TESIS DOCTORAL

AÑO 2022

AMOSE^2: UNA METODOLOGÍA DE CARACTERIZACIÓN DEL ERROR EN SEGMENTACIÓN DE OBJETOS AMORFOS.

LA SEGMENTACIÓN DE HIPERINTENSIDADES CEREBRALES COMO CASO DE ESTUDIO.

ESTELA DÍAZ LÓPEZ

PROGRAMA DE DOCTORADO EN SISTEMAS INTELIGENTES

DIRECTORES:

Dr. MARIANO RINCÓN ZAMORANO

Dra. MARGARITA BACHILLER MAYORAL

«No hay magia en la magia, todo está en los detalles», Walt Disney.

A mi familia,

de sangre y de corazón,

por hacer que esta investigación haya sido posible.

RESUMEN

El proceso de diseño de un sistema de segmentación de objetos suele seguir un patrón iterativo e incremental de desarrollo, donde los expertos analizan, en la etapa de evaluación, los errores cometidos por el sistema con el objetivo de conocerlos y proponer soluciones para el refinamiento del sistema. Aunque existe un conjunto de métricas bien definidas para evaluar la calidad de la segmentación de manera automática, la evaluación del error con el objetivo de encontrar su causa ha sido siempre una tarea muy artesanal. Por ello, son necesarios nuevos métodos y herramientas para facilitar la evaluación de un sistema durante su diseño que sean explicables, flexibles, interactivos, analíticos y contextuales.

En este trabajo, con el objetivo de descubrir nuevo conocimiento del error de una forma automatizada, se propone una metodología para la caracterización del error en segmentación de objetos amorfos bajo la hipótesis de que existe un número reducido de patrones que explican una buena parte de los errores. Esta metodología, denominada AMOSE^2, propone modelar el error por objetos individuales y realiza una descripción detallada de ellos mediante un vector de características para cada objeto de error, lo que permite realizar un análisis más profundo de los errores de segmentación mediante técnicas de clustering y de detección de outliers propias de la IA. Además, se introducen modelos de conocimiento mediante ontologías para describir las características visuo-espaciales (reutilizables entre dominios) y de contexto (propias del dominio) de los objetos segmentados y de su error. Se distinguen dos tipos de características según el origen de la información, las generadas a partir de información externa al sistema de segmentación y las generadas internamente por el sistema de segmentación (que solo son accesibles en los sistemas de caja gris). La descripción ontológica del conocimiento permite mejorar la manipulación de las variables, facilitar la interacción con el experto o realizar una selección de características basada tanto en medidas estadísticas como semánticas. Los patrones de error detectados son analizados por los diseñadores del sistema con el objetivo de seleccionar los más relevantes, los cuales permitirán orientar los esfuerzos en el siguiente ciclo de refinamiento en la etapa de diseño.

Se ha implementado un prototipo siguiendo las definiciones de la metodología, el cual se compone de una herramienta para la descripción y análisis de los errores en segmentación (AMOSE^2 analysis) y una herramienta de exploración interactiva de resultados (AMOSE^2 web report). Para su validación y ejemplo de uso, se ha utilizado un problema real, la segmentación de manchas de sustancia blanca cerebral en imágenes de resonancia magnética. Se han evaluado tres sistemas sobre cinco datasets distintos para mostrar los dos formas de uso de la metodología AMOSE^2: el análisis individual, donde se compara la propuesta de un sistema de visión por computador (SVC) respecto a una referencia dada, y el análisis comparado, donde se compara, además, respecto a otro sistema de segmentación. Los hallazgos de patrones de error relevantes han sido de utilidad para el refinamiento del sistema de segmentación AMOS-2D.

PALABRAS CLAVE: Diseño de sistemas cognitivos en visión artificial, caracterización del error en segmentación, reconocimiento de patrones, IA explicable, detección de agrupaciones de error, detección de anomalías de error, visualización de datos, objetos amorfos, lesiones de sustancia blanca cerebral

ABSTRACT

The design process of an object segmentation system usually follows an iterative and incremental development pattern where experts analyse, in the evaluation stage, the errors made by the system, in order to know them and propose solutions for the refinement of the system. Although there is a set of well-defined metrics to evaluate the quality of the segmentation in an automatic way, the evaluation of the error with the objective of finding its cause has always been a very handcrafted task. Therefore, new methods and tools are needed to facilitate the evaluation of a system during its design, which must be explainable, flexible, interactive, analytical and contextual.

In this work, with the aim of discovering new knowledge of the error in an automated way, a methodology is proposed for the characterization of the error in segmentation of amorphous objects under the hypothesis that there are a reduced number of patterns that explain a good part of the errors. This methodology, named AMOSE^2, proposes to model the error by individual objects and performs a detailed description of them by means of a vector of characteristics for each error object, which allows a deeper analysis of the segmentation errors using artificial intelligence clustering and outlier detection techniques. In addition, knowledge models are introduced using ontologies to describe the visuo-spatial (reusable across domains) and contextual (domain specific) features of the segmented objects and their errors. Two types of features are distinguished according to the origin of the information, those generated from external information to the segmentation system and those generated internally by the segmentation system (only accessible in grey box systems). Their ontological description allows to improve the manipulation of variables, to facilitate the interaction with the expert or to make a feature selection based on both statistical and semantic measures. The detected error patterns are analysed by the system in order to select the most relevant ones, which will allow to guide the efforts in the next refinement cycle in the design stage.

A prototype has been implemented following the definitions of the methodology, consisting of an error object segmentation description and analysis tool (AMOSE^2 analysis) and a tool for interactive exploration of results (AMOSE^2 web report). For its validation and example of use, a real problem has been used, the segmentation of

cerebral white matter hyperintensities in magnetic resonance images. Three systems have been evaluated on five different datasets to show the two ways of using the AMOSE^2 methodology: the individual analysis, where a proposed segmentation system is compared with respect to a given reference segmentation, and the comparative analysis, where it is as well compared with respect to another segmentation system. The findings of relevant error patterns have been useful for refining the AMOS-2D segmentation system.

KEY WORDS: Cognitive System (COGS) design in computer vision, segmentation error characterization, pattern recognition, explainable AI (XAI), error clustering detection, error outlier detection, data visualization, amorphous objects, White Matter Hyperintensities (WMH)

AGRADECIMIENTOS

Esta tesis doctoral ha sido un largo y complejo camino, un poco solitario y a veces incomprendido (incluso por mí), pero que me ha hecho crecer como persona y como investigadora. Ha habido etapas oscuras, de las que aún hoy estoy aprendiendo a gestionar (con inteligencia emocional), y etapas arcoíris, que me acompañaran siempre, y de las que estoy muy orgullosa, y por ello, quiero deciros a todas las personas que me habéis acompañado, sufrido y ayudado: "¡Gracias!".

En primer lugar quiero agradecer a mi director, Mariano, la confianza que ha depositado en mí durante todos estos años. Me has enseñado mucho, hemos trabajado codo con codo y tus comentarios y críticas, han sido siempre constructivas y bien recibidas. Gracias por ofrecerme este proyecto, por tu paciencia infinita, tus conversaciones sinceras y tu gran esfuerzo por hacer que esta investigación tenga un final exitoso, del que espero que sea un punto y seguido y no un punto y final. En segundo lugar quiero agradecer a mi co-directora, Margarita, su apoyo y motivación en la difícil tarea de la escritura de la tesis, de su visión organizada y jerárquica de todos los elementos que se tratan. Gracias por ayudarme a desenmarañar mi amalgama de ideas y hacer un documento legible. A los dos, gracias por vuestro tiempo y dedicación a horas intempestivas.

De forma especial quiero agradecer a mi familia todo el apoyo y soporte que he recibido para poder hacer realidad este proyecto. A Dani, por estar ahí siempre, a las duras y a las maduras, formando equipo y creando una familia maravillosa. A mis pequeñas, Hanna y Noa, que aunque han retrasado este trabajo me han dado la fuerza para finalizarlo. Y por supuesto, a mis padres y a mi hermana, por su gran apoyo tanto económico como emocional. Y a mis suegros, por su gran ayuda y disposición para lo que haga falta. Y no me quiero olvidar de todas las personas que me han ayudado, animado y criticado todos estos años, las que viven lejos pero están muy cerca en mi corazón, las que estaban cerca y por las vueltas de la vida, están algo lejos pero que tenemos una unión más fuerte, y a las que estuvieron en el pasado y han dejado una huella imborrable.

Por último, quiero agradecer a las diferentes instituciones el apoyo financiero para poder llevar a cabo esta investigación. A la UNED por concederme una beca FPI (2012-2016) y una beca para una estancia investigadora de dos meses en 2014 en el grupo de

investigación de visión por computador del departamento de Interfaces Avanzadas de la escuela de informática de la universidad de Mánchester bajo la supervisión del profesor Dr. Tim Morris. A la beca ofrecida por Islandia, Liechtenstein y Noruega a través del mecanismo financiero EEA, soportado y coordinado por la Universidad Complutense de Madrid (ABEL-CM-01-2013) para realizar una estancia de seis meses en 2015 en el Centro de Intervención del Hospital Univesitario Rikshospitalet de Oslo, Noruega, bajo la supervisión el Dr. A. Bjørnerud.

A todos, gracias.

Índice general

Glosario de abreviaturas y acrónimos

CNN	Redes Neuronales Artificiales / Convolutional Neural Network
CSF	Fluido cerebro-espinal
CSVD	Enfermedad de los vasos sanguíneos pequeños/ Cerebral Small Vessel Disease
DSC	Coeficiente de similitud Sørensen-Dice
IA	Inteligencia Artificial
MDS	Metodología de Desarrollo Software
O_P	Símbolo utilizado para describir a un objeto de la segmentación propuesta
$O_{R.P}$	Símbolo utilizado para describir a un objeto de segmentación obtenido por la unión de los objetos en contacto de la segmentación propuesta y la segmentación de referencia
O_R	Símbolo utilizado para describir a un objeto de la segmentación de referencia
RM	Resonancia Magnética
RX	Rayos X
SDLC	Ciclo de Vida de Desarrollo Software / Software Development Life Cicle
SVC	Sistema de visión por computador
TAC	Tomografía Axial Computerizada
WMH	Sustancia Blanca Cerebral / White Matter Hiperintensity

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto y motivación

Desde sus inicios, allá por los años 60 del siglo XX, se han desarrollado multitud de sistemas de visión por computador (SVC) en todos los ámbitos de la vida (medicina, seguridad, ocio, agricultura, industria, etc.), lo que ha permitido automatizar muchos procesos que anteriormente se realizaban de forma manual, mediante inspección visual [Yuille and Oliva, 2010, de Souza Alves et al., 2018, Voulodimos et al., 2018, O'Mahony et al., 2020]. Su evolución se muestra en la figura 1.1, donde se observa el número de publicaciones científicas relativas a visión por computador desde sus inicios [Price, 2022] hasta hoy día.

En la última década, el número de SVC ha crecido de manera exponencial debido fundamentalmente a tres factores: 1) a la mejora de la tecnología, que ha aumentado la capacidad de cómputo de los sistemas; 2) a la digitalización de la información, que ha permitido disponer de grandes bases de imágenes para usarlas en aprendizaje a partir de datos; y, sobre todo, 3) a los sistemas y entornos de programación de aprendizaje profundo [Voulodimos et al., 2018], que han proporcionado marcos de programación sencillos y accesibles junto a la posibilidad de entrenar sistemas de visión extremo a extremo. Esto último ha permitido obtener soluciones directamente a partir de imágenes de ejemplo, esto es, sin tener que codificar previamente los operadores para extraer las características relevantes para cada problema concreto.

Los SVC tienen como objetivo global imitar la labor del experto humano a partir de la adquisición, el procesado y el análisis de imágenes digitales del mundo real. Una de las tareas intermedias clave para conseguir este objetivo es la segmentación semántica de la escena, que consiste en el etiquetado de los píxeles de la imagen con las etiquetas de los objetos presentes. Esta segmentación semántica es clave porque es el punto de partida para la segmentación de objetos, o sea, para toda tarea cuyo objetivo sea localizar, caracterizar

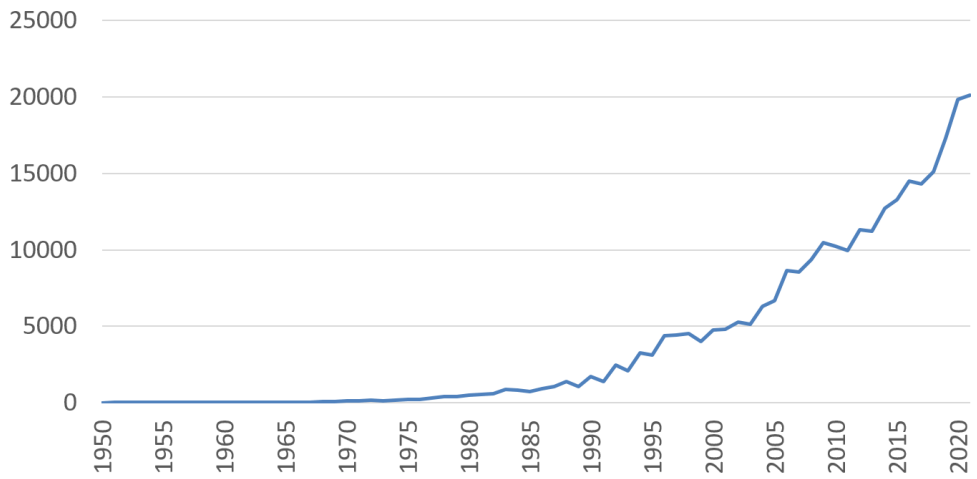


Figura 1.1: Evolución de las publicaciones científicas en el área de la visión por computador Fuente: http://www.visionbib.com/bibliography/stats.html

mediante métricas y monitorizar la evolución de los objetos de interés presentes en la escena [Hariharan et al., 2014].

La experiencia en el diseño e implementación de diferentes sistemas computacionales para la segmentación semántica de objetos que se han desarrollado en el grupo de investigación SIMDA, en el que he participado, ha puesto de manifiesto la dificultad que entraña la creación de un sistema computacional automático que imite la labor del experto de un dominio. La segmentación de objetos es una tarea compleja, pues conlleva la detección, localización y delineación precisa del contorno de los objetos presentes en la escena[1].

El primer problema que nos encontramos es que muchos de los procesos implicados en la visión pertenecen al subconsciente, lo que hace que el experto tenga un conocimiento incompleto y que el diseño del sistema suela seguir un patrón iterativo e incremental, en el que se parte de una primera aproximación para construir el sistema y después se va refinando a partir de la evaluación de los errores que éste comete. Este proceso de diseño sigue, por tanto, un modelo en espiral que suele ser muy artesanal, pues una buena parte del conocimiento utilizado para simplificar el problema pertenece al dominio de la aplicación. No cabe duda que la automatización de este proceso de diseño iterativo sería un gran avance en la definición de SVCs.

Una solución es utilizar una metodología de desarrollo software (MDS) iterativa e incremental [Larman and Basili, 2003, Basil and Turner, 1975], ya sea clásica, como la del ciclo de vida o la metodología en espiral/evolutiva [Boehm, 1986], orientada a objetos,

[1]En nuestro caso, simplificamos el paso de segmentación semántica a segmentación de objetos porque solo trabajamos con una clase de objetos y suponemos que cada región aislada e independiente es un objeto.

como RUP ("Rational Unified Process"), o una más actual basada en la metodología ágil, como XP o Scrum [Mishra and Dubey, 2013, González, 2013]. También es importante seguir los siete pasos del ciclo de vida del desarrollo software: (1) validación y concepto, (2) planificación, (3) análisis, (4) diseño, (5) implementación, (6) testeo e integración y por último, (7) mantenimiento. Además, es recomendable aplicar, en todos ellos, algún proceso de evaluación para «garantizar que cualquier sistema inteligente produzca resultados que sean válidos, verificados, basados en datos, confiables y explicables para cualquier persona, ético en el contexto de su implementación, imparcial en su aprendizaje y justo para sus usuarios» [Batarseh et al., 2021, p. 2].

En la figura 1.2 se muestra un modelo iterativo e incremental simplificado basado en la metodología en espiral, donde se representan las cuatro fases del ciclo de diseño: planificación, diseño, implementación y evaluación. En primer lugar, se realiza un análisis del problema a solucionar y se definen los requisitos y restricciones que dan lugar al plan inicial. A partir de éste, se realiza el diseño inicial, que se implementa en algún lenguaje computacional, y se obtiene el prototipo inicial. A este prototipo se le realizan diferentes pruebas para su evaluación, donde se verifican y validan los requisitos iniciales, y se detectan discrepancias (críticas). Mediante un proceso cíclico de refinamiento, estas críticas dan lugar a nuevos requisitos y restricciones que permiten mejorar el sistema, y así hasta alcanzar el objetivo con el nivel de exigencia deseado. Éste es un proceso complejo, con ciclos de grandes avances, cuando aparecen nuevos métodos que mejoran los resultados, y ciclos de mejoras leves, en los que se refinan los métodos actuales.

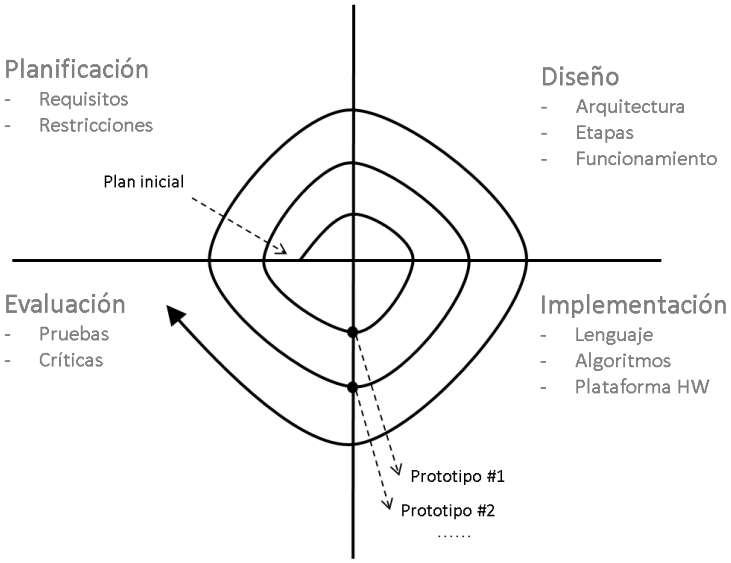


Figura 1.2: Estructura básica de una metodología iterativa e incremental

Para acotar el problema, esta tesis doctoral se centrará en la segmentación de objetos amorfos en imagen médica, aunque muchos de los resultados obtenidos serán extrapolables a otros contextos.

¿Por qué imagen médica? En la actualidad, existen múltiples tipos de imagen médica (RX, TAC, RM, ultrasonido,etc.) que proporcionan gran cantidad de información sobre el comportamiento y la estructura del cuerpo humano de una manera precisa, rápida y, en general, poco invasiva. Su análisis es un objetivo de primer nivel, pues permite extraer información útil tanto para investigación como para la práctica clínica. De hecho, su uso es fundamental en diferentes campos: investigación biomédica, diagnostico de enfermedades, seguimiento de terapias, intervenciones quirúrgicas, etc. [Ferrante, 2021].

A día de hoy, se generan una gran cantidad de imágenes médicas, y analizarlas para detectar lesiones o enfermedades forma parte del trabajo diario. La segmentación manual es costosa y subjetiva, por lo que disponer de herramientas de segmentación automática que faciliten dicha tarea es de gran ayuda. Estas herramientas también pueden cometer errores, y es importante descubrirlos para acotar su alcance y poder así confiar en su uso.

¿Por qué objetos amorfos? Porque muchos de los objetos asociados a patologías son así, sin una forma definida (tumores, hemorragias, quistes, verrugas, lesiones cerebrales, manchas, etc.). Una región amorfa es una zona que no tiene unas propiedades físicas bien definidas, su forma es irregular y sus bordes son imprecisos. Al no ser posible distinguir elementos constituyentes (subregiones), no es posible descomponer el análisis con un mayor nivel de detalle. En la figura 1.3 se muestra algunos ejemplos de objetos amorfos frente a objetos estructurados.

1.2. Objetivos

Para facilitar la segmentación precisa de los objetos presentes en una imagen (o, en general, para construir un sistema de visión artificial) es necesario automatizar el proceso de diseño, que, como ya hemos dicho, es un proceso cíclico e iterativo. La automatización completa es compleja, pues requiere sacar a los expertos (del dominio y del diseño de sistemas) del lazo de realimentación del proceso de diseño. En el estado de la cuestión, consideramos que la tecnología no puede reemplazar a los expertos en el ciclo de diseño, pero sí puede ayudarles quitándoles carga de trabajo rutinaria en tareas que sí pueden realizar los SVC. Por ello, nuestro objetivo va a ser un poco menos ambicioso y se limitará a desarrollar herramientas para facilitar la etapa de evaluación del sistema, lo que ayudará a estos expertos en el refinamiento incremental del mismo.

Dentro de la evaluación del sistema, una de las etapas que todavía no han sido automatizadas, porque es muy dependiente de conocimiento propio del dominio, es

Figura 1.3: Ejemplos de objetos amorfos y objetos estructurados

la caracterización del error. En esta tesis doctoral se propone desarrollar una nueva metodología para caracterizar los errores de segmentación, o sea, para descubrir los distintos patrones de error que describen los comportamientos detectados. Nuestra hipótesis es que cada patrón de error detectado tiene un origen y causa distinto, por lo que identificarlos ayuda a detectar los problemas del sistema de visión (*críticas* dentro de la tarea de diseño), y esto permitirá a los expertos implicados en el diseño del sistema proponer modificaciones (*mejoras* dentro de la tarea de diseño) siguiendo la estrategia "proponer-criticar-modificar" [Chandrasekaran, 1990].

Esta metodología de caracterización del error en segmentación permitirá sistematizar el proceso de evaluación del sistema. En ella, se evaluará el error en segmentación de manera supervisada, es decir, respecto a una referencia, y se distinguirán distintos tipos de error. A continuación se analizará cada tipo de error por separado para detectar patrones de error y se facilitará una descripción sencilla de sus características. Además, dado que son los expertos los que tienen que proponer las mejoras en el sistema a partir de estos patrones de error detectados, se propone crear una herramienta interactiva y visual que les facilite el trabajo, esto es, que les permita identificar qué tipo de errores existen, cómo son, dónde se localizan, a cuántos casos representa un determinado patrón de error, etc. Esta herramienta facilitará la integración de los expertos en el ciclo de diseño.

Para desarrollar esta metodología (objetivo principal) se cubrirán los siguientes objetivos parciales:

1. Detectar las discrepancias en la segmentación: Se deberá proponer una forma de identificar las discrepancias existentes entre la segmentación propuesta y la de referencia. Esto se realizará por objetos independientes, de manera que se pueda realizar un análisis detallado del error para cada objeto. Al elemento que describe el error lo llamaremos "objeto de error".

2. Describir los objetos de error: una vez detectado, se deberá describir el error en cada objeto para buscar posteriormente errores similares en el espacio de características definido.

3. Clasificar los objetos de error: Los objetos erróneos se clasificarán en distintos tipos de error en función de su composición y de los requisitos de calidad exigidos.

4. Descubrir patrones de error: Se analizarán las descripciones de los objetos de error para detectar y describir tanto patrones de error como errores aislados relevantes. De manera que se pueda actuar sobre su origen y solucionarlos.

5. Presentar resultados de manera visual e interactiva: Se implementará una herramienta de exploración con facilidades visuales e interactivas para mostrar los resultados del análisis del error y facilitar la interacción con los expertos implicados en el diseño.

1.3. Metodología

La metodología de caracterización del error en segmentación de objetos amorfos que se propone en este trabajo se enmarca dentro de la etapa de evaluación del desarrollo de sistemas inteligentes y, dado que también va a ser un sistema computacional, para su definición se van a aplicar los fundamentos de una metodología iterativa en su desarrollo.

Se pretende diseñar una metodología para caracterizar el error en segmentación que sea configurable, modular y extensible, y que permita realizar (semi)automáticamente un análisis exhaustivo del error en segmentación para la mejora de SVCs. Esta metodología deberá permitir identificar los errores, reunir información de múltiples fuentes y realizar un análisis profundo del error, de manera que se puedan identificar y describir los comportamientos de error relevantes.

Además, ha de diseñarse bajo los principios de reutilización de herramientas ya existentes, de explicabilidad tanto para los métodos de análisis como para las representaciones de los datos, de inteligencia para aprender de los datos, razonar por si mismos y entender los problemas reales, y por último, de interacción con el humano,

pues es a quien van dirigidas las herramientas informáticas que se implementen, las cuales deben facilitar la manipulación de los datos y sus hallazgos.

Para ello, se utilizarán técnicas de inteligencia artificial (IA) teniendo en cuenta que el resultado final debe ser interpretado por un experto humano especializado en el dominio, por lo que se tratará de generar descripciones con un reducido número de variables para facilitar su comprensión. Para acotar el problema, los objetos a analizar se tratarán como regiones amorfas, es decir, sin forma definida ni subregiones, y se analizarán todos los objetos sin importar su tamaño mínimo. Además, se promoverá el uso de sistemas de caja gris, pues el uso de información del dominio y de la aplicación facilitará la descripción de los errores y la interacción con los expertos.

Siguiendo los subobjetivos planteados en la sección anterior, se identifican una serie de subtareas que es necesario realizar:

1. **Tarea 1: Revisión del estado de la cuestión.** En primer lugar, se va a realizar un análisis preliminar del modelado del error en segmentación de objetos, los métodos existentes para su evaluación y se estudiaran los diferentes tipos de error que se pueden encontrar al comparar una segmentación propuesta respecto a una referencia.

2. **Tarea 2: Detección de discrepancias.** A partir de la comparación espacial entre la segmentación propuesta y la segmentación de referencia, se identificarán las discrepancias entre los objetos de ambas segmentaciones (objetos de error). Se deberá definir una representación que permita identificar estos objetos de error.

3. **Tarea 3: Descripción de los objetos de error.** Una vez detectados los objetos de error, éstos se describirán mediante un vector de características que podrá contener tanto información visual (basada en descriptores espaciales y visuales de los objetos presentes en la imagen) como información propia del dominio de aplicación si ésta está disponible (sistema de caja gris).

4. **Tarea 4: Clasificación de los objetos de error.** Dado que el error resulta de comparar una segmentación propuesta y una referencia, se pueden dar distintas situaciones. Por ejemplo, que se detecte o no un objeto, que se segmente un objeto inexistente, que la segmentación sea perfecta, que solo haya cierto solape, etc. Será necesario definir una categorización que permita organizar esta información.

5. **Tarea 5: Caracterización del error en segmentación.** Se analizarán, mediante técnicas de exploración y agrupación (clustering) propias de la IA, las descripciones de las distintas categorías de objetos de error para detectar y describir los patrones más interesantes. También se aplicarán técnicas de detección de anomalías (outlier) y comparación de múltiples soluciones para detectar comportamientos singulares.

6. **Tarea 6: Presentación de resultados.** Se implementará una herramienta visual e interactiva para permitir a los expertos implicados en el diseño, la exploración de los resultados. Por ejemplo, se podrá filtrar por las diferentes variables para ver los datos tanto en formato tabla como en una figura, se podrá seleccionar un objeto de error o un grupo de ellos y observarlos en su contexto original (sobre la imagen de entrada), etc. Además, dado que se utilizará una identificación local a cada objeto de error, será posible relacionar diferentes soluciones a partir de su identificación y localización, lo que permitirá comparar distintos sistemas de segmentación y extraer conclusiones.

Una vez diseñado el sistema, se implementará la metodología de caracterización del error siguiendo los principios de reutilización de código, y se procurará utilizar algoritmos explicables y de fácil interpretación. En cuanto a la reutilización de código, existe una gran comunidad de desarrolladores que en los últimos años ha creado una serie de repositorios públicos y de acceso abierto donde se publican la implementación de algunos algoritmos, funciones, métodos, etc. que han sido descritos en artículos científicos.

Finalmente, para ejemplificar el uso de la metodología de caracterización del error, ésta se aplicará a un problema real, la segmentación de hiperintensidades en la sustancia blanca cerebral que se aprecian en imágenes de resonancia magnética (RM). Estas lesiones hiperintensas son un ejemplo de objetos amorfos en imágenes tridimensionales de resonancia magnética, y su segmentación precisa es un proceso complejo en el que se investiga activamente [Wang et al., 2012, Ithapu et al., 2014, Caligiuri et al., 2015, Griffanti et al., 2016, Atlason et al., 2019, Kuijf et al., 2019, Brugulat-Serrat et al., 2020, Heinen et al., 2019, Hotz et al., 2022, Tran et al., 2022]. Se mostrarán distintos casos de uso de la metodología, bien para caracterizar el error en un sistema de visión, bien para comparar varios sistemas.

1.4. Organización del documento

Esta memoria está formada por varios capítulos que se describen a continuación.

En el capítulo 1, se describe el problema de la dificultad de diseñar un sistema de visión y la motivación que ha llevado a proponer una metodología de caracterización del error para facilitar la tarea de diseño. Se describen los objetivos propuestos y la manera de llevarlos a cabo.

En el capítulo 2, se realiza una revisión bibliográfica sobre los actuales sistemas de visión y su evolución hacia sistemas cognitivos, donde se combina la explicación y el razonamiento basado tanto en conocimiento como en experiencia. También se analiza la evaluación del error en segmentación de objetos, la problemática de obtener una buena

segmentación de referencia y las métricas y herramientas que se utilizan en la actualidad para su estudio.

En el capítulo 3, se propone una metodología de caracterización del error en segmentación basada en el modelado local de los objetos segmentados, el uso de ontologías para la descripción de características visuo-espaciales y la utilización de técnicas de clustering y de detección de outlier propias de la IA para detectar comportamientos de error. Sus principios son la transparencia, explicabilidad e interacción para facilitar a los expertos un mejor entendimiento de los errores con el objetivo de proponer mejoras y refinar el SVC bajo evaluación.

En el capítulo 4, se valida la metodología propuesta con un ejemplo real, la segmentación de hiperintensidades en la sustancia blanca cerebral. Se describe el problema, se presentan los materiales y algoritmos utilizados, y se muestran los resultados obtenidos, mediante el conjunto de herramientas informáticas de la metodología, de varios estudios realizados que describen las dos formas de uso.

Por último, en el capítulo 5 se presentan las conclusiones, las aportaciones y el trabajo futuro de esta tesis doctoral.

Como material adicional, en el anexo A, se describe el funcionamiento y la arquitectura de la herramienta informática implementada (un prototipo) de la metodología propuesta y se proporciona el manual de uso. En el anexo B, se detallan las características utilizadas para el análisis de caracterización del error en el sistema AMOS-2D, el cual ha servido de base para desarrollar la metodología. Por último, se incluyen en el anexo C otros resultados de los diferentes análisis y configuraciones estudiadas con la metodología propuesta.

Capítulo 2

Estado de la cuestión

En los últimos años, la visión por computador ha experimentado un gran avance, sobre todo con los sistemas basados en aprendizaje máquina y, más concretamente, con los sistemas de aprendizaje profundo, pero a pesar de dichos avances, la visión artificial todavía no puede igualar el desempeño de la visión humana en muchas tareas [Ji, 2019]. Como comentan de Souza Alves et al. [2018] «el rendimiento de los sistemas actuales depende en gran medida de los datos, carece de generalización más allá de los conjuntos de datos de entrenamiento y requiere adaptaciones importantes para incluir nuevas condiciones. Por lo tanto, el desafío actual en los sistemas de visión es crear sistemas robustos y flexibles que puedan reconocer clases de objetos complejos en entornos mal controlados y que no requieran cantidades prohibitivas de datos de entrenamiento para resolver problemas de visión por computador sin restricciones». Por ello, la tendencia de los nuevos sistemas de visión es dirigirse hacia sistemas de visión cognitivos, donde se combine conocimiento explícito de la aplicación con aprendizaje máquina y que, además, esté centrado en la interacción humano-máquina. Esta tendencia actual ya la propuso hace unos años Ashok K. Goel [Goel et al., 2012] cuando dijo que «los futuros sistemas de visón se basarán en explicaciones cognitivas del diseño y permitirán tanto el diseño colaborativo, como el conceptual y el creativo».

En el problema de la segmentación precisa de objetos amorfos, la tarea de diseño de un SVC es un proceso complejo, ya que se desarrolla en un escenario de contexto incompleto y variable, y con datos imprecisos. Por un lado, a la hora de extraer el conocimiento del dominio y de la tarea, hay mucho conocimiento tácito (no consciente) y, por tanto, su modelado es incompleto. Por otro lado, los objetos amorfos no tienen una caracterización bien definida, lo que implica que las referencias son imperfectas y variables, tanto intra- como inter-experto, por lo que los datos de ejemplo hay que tratarlos con confianza relativa y precaución.

En un SVC influye de gran manera tanto el conocimiento y experiencia de los expertos como la cantidad y calidad de los datos. También influyen otros factores, como la opacidad del sistema (caja negra, caja gris o caja blanca), la adaptación a contextos dinámicos (imágenes reales con escenarios variables) o el grado de interacción hombre-máquina (explicaciones declarativas, de una dirección o de doble dirección).

Ante esta situación de conocimiento incompleto y datos imprecisos, los SVC comenten fallos, y una forma de refinar su diseño es analizar su comportamiento. Una solución para facilitar la tarea de diseño en SVC es mejorar los procesos de evaluación, con el objetivo de detectar y describir los errores (críticas) para facilitar el refinamiento del sistema (mejoras).

Actualmente, los métodos de evaluación con el objetivo de caracterizar el error no están muy desarrollados y, por tanto, los procesos de evaluación suelen llevarse a cabo de forma artesanal, con procedimientos ad-hoc para cada aplicación concreta. En otras etapas del ciclo de desarrollo software (SDLC de sus siglas en inglés) como en la etapa de implementación o en la etapa de testeo e integración, los procesos de evaluación están más desarrollados. En la primera, los procesos de evaluación permiten el ajuste del parámetros y/o la comparativa entre algoritmos para una o varias métricas de evaluación utilizando un proceso de mejoras iterativo e incremental, mientras que en la segunda, su evaluación permite validar y demostrar la efectividad del sistema para resolver la tarea concreta para la que ha sido diseñado.

La evaluación es un proceso complejo y menos tratado en visión artificial (respecto a otras etapas, como segmentación o el reconocimiento), ya que en la mayoría de los estudios sólo se utiliza para mostrar el rendimiento del sistema mediante un porcentaje de éxito (o de error), su eficiencia en cuanto a tiempo de computo, etc. Gracias a la evolución tecnológica de los últimos años, los procesos de evaluación han virado su enfoque y han cobrado más relevancia, sobre todo, desde la aparición de la inteligencia artificial explicable, ya que debido a la mejor calidad de los resultados en los sistemas actuales, en algunos campos se ha empezado a enfocar el interés en conocer más información de los procesos, de su arquitectura, de los datos utilizados, de sus razonamientos o de sus fallos. Es decir, se ha comenzado a diseñar los sistemas desde un enfoque cognitivo, lo que permitirá, por un lado, conocer cómo se relacionan los actores, los datos y el conocimiento disponible, y por otro lado, manejar los diferentes conceptos, criterios y restricciones del problema a resolver y aprender a razonar para evaluar, con verdadera inteligencia, nuestro entorno.

El objetivo de este capítulo es, por tanto, realizar una revisión de las características que deben tener los sistemas de visión actuales para que los nuevos procesos de evaluación puedan ayudar a facilitar su diseño y refinamiento (sección 2.1). En ella, se comentan

características como el modelado de conocimiento (2.1.1), la IA explicable (2.1.2) y las propiedades de los sistemas cognitivos (2.1.3). También, se realiza una revisión de los métodos y métricas de evaluación del error en segmentación de objetos, sobre todo de aquellas utilizadas en la evaluación de hiperintensidades cerebrales (sección 2.2). Por último, se analizan las herramientas de evaluación que existen para describir y analizar la segmentación de objetos desde la perspectiva de caracterizar el error (sección 2.3).

2.1. Los sistemas de visión actuales

En los últimos años, se han propuesto enfoques para imitar las capacidades de la inteligencia humana mediante la combinación de conocimientos previos e información visual en sistemas de visión basados en conocimiento (KBS), lo que supone un paso hacia la visión cognitiva (CVS). Los sistemas de visión basados en conocimiento integran los enfoques ascendentes (basados en datos), que realizan tareas de visión por computador a través de algoritmos de aprendizaje automático, con los enfoques descendentes (basados en experiencias), que ofrece el conocimiento previo en un proceso de inferencia inspirado en la visión biológica [de Souza Alves et al., 2018].

Para desarrollar sistemas de visión cognitivos es necesario implementar mejoras para aumentar su funcionalidad, su explicabilidad y su interacción con los humanos. Para ello, en primer lugar, es necesario modelar el conocimiento mediante representaciones formales como las ontologías, utilizar algoritmos explicables que permitan conocer su proceso de razonamiento y tener herramientas de evaluación que realicen análisis más complejos ara ofrecer nuevo conocimiento a los expertos en el diseño de SVC.

2.1.1. El modelado del conocimiento

Los procesos de evaluación para sistemas cognitivos necesitan de un mayor conocimiento de los datos, de los procesos y de la tarea que trata de resolver el SVC bajo estudio. Para ofrecer mejores resultados, es fundamental disponer de conocimiento estructurado mediante algún tipo de representación formal que aumente la caracterización de los objetos segmentados, que permita realizar nuevas medidas basadas en información semántica para descubrir nuevo conocimiento sobre los errores en segmentación y que facilite la selección de una buena representación visual para entender los datos.

2.1.1.1. Representación del conocimiento

La representación del conocimiento en inteligencia artificial estudia la estructuración de la información y su utilización en herramientas software. La información es esencial

a la hora de prestar servicios o desarrollar aplicaciones, pero ésta debe encontrarse bien definida y ser correcta, aunque no esté completa. Un exceso de información no implica mayor conocimiento y crea dificultades a la hora de incorporar el nuevo conocimiento para que sea utilizado de forma compartida.

Los sistemas necesitan de información estructurada para resolver problemas en un dominio del que se posee un conocimiento específico. Este conocimiento de dominio es necesario formalizarlo utilizando para ello sistemas de representación de la información como las tripletas Objeto-Atributo-Valor, las redes semánticas o los marcos, que han posibilitado el desarrollo de diferentes formalismos y lenguajes de representación, de los que se destacan las ontologías.

Las ontologías han alcanzado una gran importancia en las últimas décadas como herramientas conceptuales que definen un vocabulario común en un determinado dominio, legible por computadores, consensuado, reutilizable y que ayuda a la gestión del conocimiento [Brewster et al., 2004, Gómez-Pérez et al., 2007, Clouard et al., 2010]. Dentro del campo de la IA, las ontologías se han definido de múltiples formas dependiendo del enfoque del autor. Gruber [1993] y Borst [1997] definen de forma similar el concepto de ontología como una especificación formal de una conceptualización compartida. Otra definición dada por Uschold and Gruninger [1996] define ontología como un vocabulario de términos y alguna especificación de su significado.

Para su desarrollo, son necesarias varias acciones: una especificación del objetivo de la ontología, una conceptualización donde se defina la estructura del conocimiento del dominio, una formalización que transforme el modelo conceptual a un modelo formal y, por último, una implementación que transforme el modelo formal a un modelo computacional.

Los principales componentes de una ontología son los conceptos, las relaciones y las instancias, y en función de su arquitectura, su uso o su aplicación, se pueden diferenciar diferentes tipos. En la figura 2.1 se distinguen algunos tipos en función de su conceptualización:

- Ontologías de alto nivel o genéricas: Describen conceptos muy abstractos y generales como espacio, tiempo, acción,... que pueden ser compartidos en múltiples dominios y aplicaciones ya que son independientes de un problema o dominio particular. Estas ontologías describen conceptos básicos en los sistemas de información. Ejemplos de este tipo de ontologías de alto nivel son DOLCE Gangemi et al. [2002] y SUMO Niles and Pease [2001].

- Ontologías de dominio: Describen el conocimiento de un dominio concreto, como medicina o geografía, especializando conceptos introducidos en la ontología de nivel superior e independiente a las posibles tareas asociadas a un dominio.

- Ontologías de tareas: Describen una tarea o actividad especializada, como diagnosis o monitorización, especializando las ontologías de alto nivel.
- Ontologías de aplicación: Describen conceptos específicos de las ontologías de dominio y de tarea. Los conceptos en estas ontologías a menudo se corresponden con los roles propios de las entidades del dominio mientras que realizan una cierta actividad.

Las ontologías inferiores son más específicas y por tanto de un uso más restringido en su ámbito de aplicación, mientras que las ontologías superiores permiten una mayor reutilización.

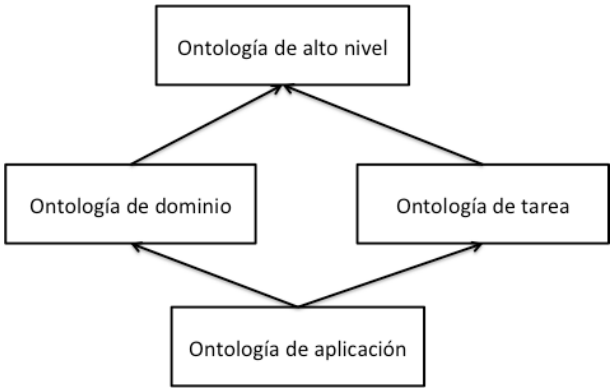


Figura 2.1: Tipos de ontologías según su conceptualización

Desde el punto de vista de la representación, las ontologías se pueden representar en una amplia variedad de formalismos, dependiendo de la expresividad requerida y de cómo se va a utilizar dicho conocimiento. Una de las características fundamentales de las ontologías es la representación de un conocimiento consensuado para que pueda ser reutilizado por diferentes sistemas. Además, una ontología debe ser clara y coherente permitiendo la realización de inferencias, y también debe ser precisa y permitir ser extensible a nuevo conocimiento.

Uno de los proyectos más importantes donde se utilizan hoy día ontologías es la Web Semántica, cuyo objetivo es la creación de tecnologías para publicar datos legibles por aplicaciones informáticas. En el ámbito médico, el uso de ontologías para el desarrollo de aplicaciones ha ido en aumento en los últimos años [Hoehndorf et al., 2013] con proyectos como BrainVisa [Rivière et al., 2009], OntoFIS [Romá Ferri, 2009] o NOA [Wang et al., 2011]. Gracias a instituciones como el National Center for Biomedical Ontology (NCBO) o la Open Biological and Biomedical Ontology Foundry (OBO Foundry) se están desarrollando aplicaciones para compartir nuevas ontologías y posibilitar su reutilización así como aplicaciones de búsqueda, sistemas de anotación o sistemas de análisis como se recoge en `https://ncbo.bioontology.org/technology`. En el área de la visión por

computador se han utilizado las ontologías para definir procesos generales [Niles and Pease, 2001, Clouard et al., 2010] o específicos [Maillot and Thonnat, 2008] y para describir soluciones [Camous, 2007, Sánchez et al., 2012, Fatimaezzahra et al., 2016].

En el diseño y desarrollo de SVC centrados en la tarea de segmentación, es de gran ayuda disponer de una ontología que modele el objeto de interés, sus propiedades locales, relativas y globales junto a información del proceso utilizado para su detección. Por ejemplo, en la etapa de evaluación, una mayor y mejor caracterización de los datos y del sistema ayuda en los procesos de búsqueda y extracción de nuevo conocimiento e incluso facilita la visualización de resultados adaptando las representaciones visuales al tipo de variable a mostrar.

2.1.1.2. Medidas de similitud semántica

A la hora de gestionar las características que describen el conjunto de datos se pueden encontrar tres situaciones. La más básica se da cuando se utilizan descriptores genéricos, que se identifican por una numeración y sólo permiten realizar medidas estadísticas de su valor ya que no aportan información sobre qué mide. En la segunda, se utiliza un descriptor especifico, como por ejemplo "MeanIntensityObjectPeriphery", que mediante medidas léxicas basadas en la composición de la palabra se podría calcular medidas de similitud semántica. En este ejemplo, el nombre de la variable informa de que se trata de una medida de la periferia del objeto donde se ha calculado la media de sus valores de intensidad. Y por último, la solución óptima es utilizar un nombre específico junto a una ontología de dominio que permita calcular de una forma objetiva medidas de similitud semántica.

Existen distintas aproximaciones para medir distancias entre conceptos. Se tienen medidas basadas en la jerarquía entre nodos [Rada et al., 1989, Wu and Palmer, 1994, Pekar and Staab, 2002, Riku et al., 2011], basadas en la información del contenido [Resnik, 1995, Lin, 1998, Blanchard et al., 2008] o basadas en las características de los conceptos [Dice, 1945, Mirkin and Koonin, 2001, Sánchez et al., 2012]. Para profundizar en las medidas de similitud semántica, se puede consultar el trabajo de Lastra-Díaz and García-Serrano [2015].

En esta tesis, se ha utilizado la definición de Wu-Palmer [Wu and Palmer, 1994] para calcular la medida de similitud semántica entre descriptores, una medida basada en el parentesco de conceptos y su estructura de información que se muestra en la ecuación 2.1.

$$Sim_{Wu-Palmer}(ci, cj) = \sum_{i} \frac{2 * depth(LCS(ci, cj))}{depth(ci) + depth(cj)} \; (LCS : Least\,Common\,Subsummer)) \tag{2.1}$$

La profundidad de un concepto ci (depth(ci)) es el camino más corto desde éste al concepto raíz en la jerarquía de conceptos de la ontología. El LCS (Least Common Subsumer) se define como el concepto más específico, esto es, es el ancestro común de los conceptos [García et al., 2018]. Un ejemplo de jerarquía y valor de distancia entre conceptos se muestra en la figura 2.2. Mediante la ecuación de Wu-Palmeer la similitud semántica entre el concepto ci="T1Image" y cj="FlairNsigmaImage" es de $Sim_{Wu-Palmer}(T1Image, lairNsigmaImage) = \frac{2*4}{5+6} = 0{,}7273$.

La información de similitud semántica será utilizada en el sistema AMOSE[2] para jerarquizar y organizar las variables descriptivas de los objetos de error y así, facilitar el análisis de datos y la detección de patrones.

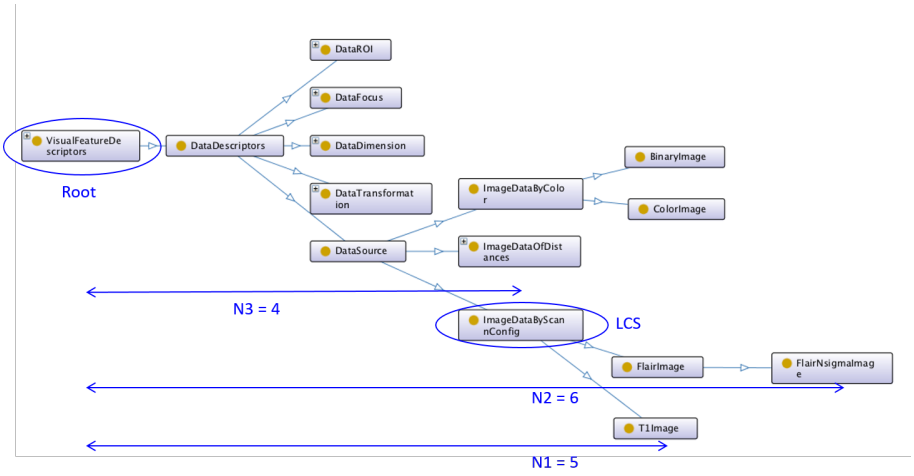


Figura 2.2: Ejemplo de similitud semántica Wu-Palmer en un fragmento de la ontología "AMOS-2D_VOF"

2.1.2. La inteligencia artificial explicable

A día de hoy, los sistemas basados en IA influyen de gran manera en las decisiones que afectan a la vida de las personas, ya que están presentes en decisiones tan importantes como los diagnósticos médicos, la conducción autónoma, los préstamos bancarios o las recomendaciones en sentencias judiciales. Por ello, en los últimos años, ha comenzado a crecer una preocupación sobre los riesgos y los sesgos de la IA.

La instituciones europeas han publicado, recientemente, un creciente conjunto de normativas, declaraciones, directrices y estudios entre los que se destacan:

- El Reglamento 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, de Protección de Datos.
- La Resolución del Parlamento Europeo, de 16 de febrero de 2017, con recomendaciones destinadas a la Comisión sobre normas de Derecho civil sobre robótica.

- La Declaración sobre inteligencia artificial, robótica y "sistemas autónomos", de 9 de marzo de 2018.
- El estudio "Algorithms and Human Rights", publicado por el Consejo de Europa en marzo de 2018.
- La Comunicación de la Comisión Europea al Parlamento, al Consejo, al Comité Económico y Social y al Comité de las regiones titulada Generar confianza en la inteligencia artificial centrada en el ser humano, de 8 de abril de 2019.
- Las Directrices éticas para una IA fiable, publicadas por la Comisión Europea en abril de 2019.
- El Libro blanco sobre la inteligencia artificial de la Comisión Europea, de 19 de febrero de 2020.

La Comisión Europea, consciente de esta situación, presentó el 21 de abril de 2021 la primera legislación sobre IA en aras de "garantizar la seguridad y los derechos fundamentales de los usuarios, a fin de que haya confianza en el desarrollo y la adopción de la IA".

Debido, en gran manera, a que las nuevas leyes y reglamentos han establecido que las decisiones de un sistema de IA deben tener capacidad de explicación (ya sea mediante la explicación del razonamiento del caso concreto o mediante la confirmación de que el sistema no introduce sesgos en su respuesta) junto al deseo de crear una IA confiable, ha estimulado la creación de algoritmos, métodos y técnicas para acompañar los resultados de los sistemas de IA con explicaciones. Se trata de la llamada inteligencia artificial explicable (Explainable AI o xAI, de sus siglas en inglés). La explicabilidad es una de las propiedades que mejora la confianza en los sistemas de IA junto a otras propiedades como la resiliencia, el sesgo y la responsabilidad. Por lo general, estos términos no se definen de forma aislada, sino como parte o conjunto de principios o pilares.

Los xAI pertenecen a un campo nuevo, de carácter indisciplinar, donde aún no se han definido completamente los conceptos clave. Según Phillips et al. [2021], estos sistemas deben respetar los siguientes principios fundamentales:

- Explicación: este principio establece que un sistema de IA debe proporcionar evidencia, soporte o razonamiento para cada decisión tomada por el sistema, tanto en su salida como en sus procesos. Además, esa caracterización de buena explicación, debe estar centrada en el humano pues él es quien la consume.
- Significativo: este principio establece que la explicación proporcionada por el sistema de IA debe ser comprensible y significativa para sus usuarios. Dado que diferentes

grupos de usuarios pueden tener diferentes necesidades y experiencias, la explicación proporcionada por el sistema de IA debe ajustarse para satisfacer las diversas características y necesidades de cada grupo.

- Precisión de la explicación: este principio establece que la explicación proporcionada por el sistema de IA debe reflejar correctamente el motivo por el que se genera el resultado y/o reflejar con precisión los procesos del sistema.

- Límites de conocimiento: este principio establece que un sistema sólo opera bajo las condiciones para las que fue diseñado y cuando alcanza la suficiente confianza en su salida. Si el sistema está en desarrollo, por un lado, se debe identificar, por si los casos analizados no cumplen los criterios para los que fue diseñado para operar, y por otro lado, se debe conocer su nivel de confianza, por si sus respuestas aún no son confiables.

Dada la amplia gama de necesidades y aplicaciones de los sistemas de IA explicables, un sistema puede considerarse más explicable o más capaz de cumplir los principios si puede generar más de un tipo de explicación. Las explicaciones se pueden describir con dos propiedades según Phillips et al. [2021]: propósito y estilo. El propósito es la razón por la cual una persona solicita una explicación o es la pregunta que pretende responder la explicación. El estilo describe cómo se entrega una explicación y se pueden diferenciar tres elementos: el nivel de detalle, el grado de interacción humano-máquina y el formato, como se muestran en la figura 2.3.

El nivel de detalle de la explicación puede ser escaso o detallado. El grado de interacción puede dividirse en tres modos: 1) explicaciones declarativas, donde la IA propone una explicación y no hay interacción; 2) explicaciones de una dirección, donde la IA propone una explicación en función de una consulta o pregunta al sistema o 3) explicaciones bidireccionales, donde la IA interviene en una conversación entre personas para mejorar los hallazgos, hacer preguntas aleatorias o proporcionar nuevas vías de exploración. Por último, el formato define la forma en que se realiza la explicación y ésta puede ser visual, verbal, auditiva o una combinación de ellas. Además, la audiencia influirá fuertemente en el propósito de la explicación y la información que proporciona. Esta información variará según los diferentes grupos de personas y su función en el sistema.

En el desarrollo de sistemas inteligentes existe un gran objetivo que es el diseño y desarrollo de algoritmos para que funcionen de forma autónoma (idealmente sin un ser humano en el circuito) y que mejoren su comportamiento de aprendizaje con el tiempo. Pero hasta llegar ese momento, es necesario tener al experto en el proceso y facilitarle la explicación de los procesos y sus resultados. Por ello, desde el punto de vista de los desarrolladores de sistemas, según Gerlings et al. [2021] los xAI son herramientas que

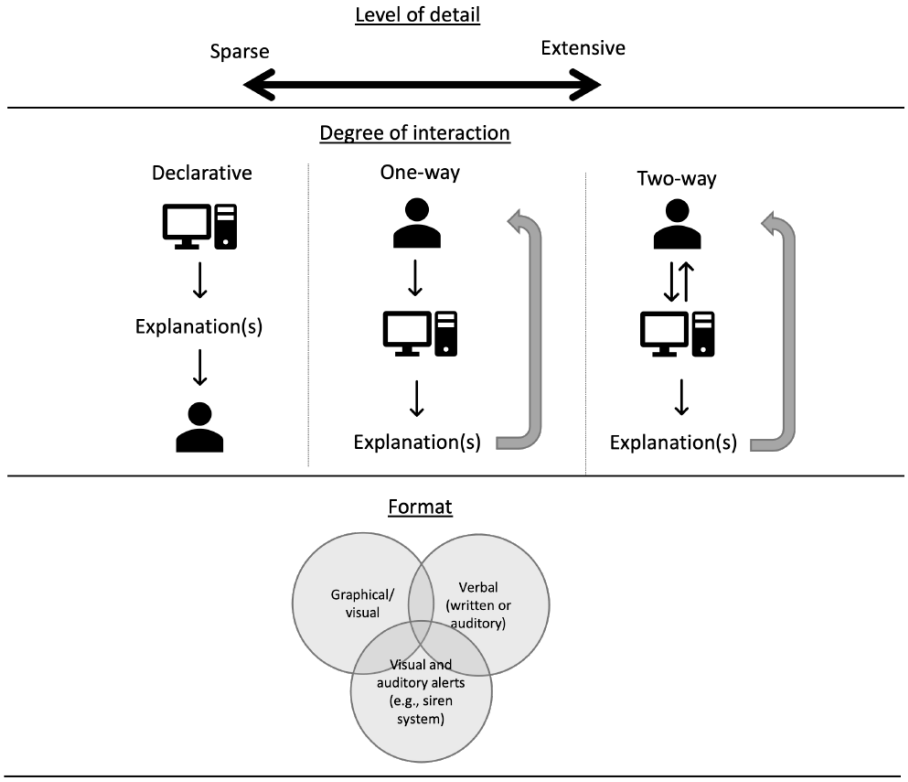


Figura 2.3: Elementos del estilo de una explicación
Fuente: Imagen extraída de [Phillips et al., 2021]

permiten crear marcos para compartir información de forma interdisciplinar como medio para minimizar los sesgos, garantizar la responsabilidad social y la equidad tanto en la preparación de los datos como en el diseño del modelo, garantizando que se base en causalidades y no en correlaciones. La xAI permite explorar y comprender la decisión tomada por los sistemas de IA y son un medio para validar modelos y el rendimiento de los sistemas. Sin embargo, validar las explicaciones y los resultados de un xAI es un proceso mucho más complejo, ya que incluye la percepción de la parte interesada que podría no poseer el nivel de alfabetización técnica necesario para seguir la cadena de lógica del modelo y la explicación combinados.

Aplicado a los sistemas de evaluación, los conceptos de explicabilidad, interpretación y transparencia son elementos opuestos a los modelos de caja negra y propician una serie de cambios [Angelov et al., 2021]. La explicabilidad implica que han de seleccionarse algoritmos que descubran nueva información que ofrezcan datos útiles para los expertos, preferiblemente con facilidades visuales que permitan conocer las explicaciones de los hallazgos detectados e incluso observarlos en su contexto. La interpretabilidad conlleva que se pueda entender el significado de una explicación, de sus variables, y que para no desvirtuar su explicación han de utilizarse transformaciones lineales en los datos. Por

último, la transparencia significa que se conozcan las razones y los procesos por los que se llegan a la conclusiones. Además, en medicina se puede también introducir el concepto de causalidad [Holzinger et al., 2019] como una forma de medir la calidad de las explicaciones, su efectividad, eficiencia y satisfacción de los usuarios.

En segmentación de objetos, un sistema de IA que facilite la explicación de los objetos erróneos en un proceso de evaluación aumentará el conocimiento del problema, ayudará a conocer casos concretos y tratará de descubrir los sesgos de los errores con el objetivo de minimizarlos.

2.1.3. Los sistemas cognitivos

Un sistema cognitivo como explica Elorriaga [2018] será aquel que sea capaz de tomar decisiones o resolver problemas a partir de la información que obtiene de su contexto, combinada con la información histórica que almacena (conocimiento) y las reglas que se han programado (razonamiento). Además, estos sistemas cuentan con la capacidad de aprender y adaptarse al contexto en el que se encuentran. El contexto de un sistema lo forman sus usuarios, sus estados internos y el resto de sistemas con los que interactúa, de los que obtiene información a través de mecanismos de monitorización de la actividad y de diferentes interfaces de entrada de datos: gráficos, textuales o de voz. Un sistema cognitivo será capaz de detectar cambios en su contexto y evaluar el impacto que puedan tener éstos en sus procesos de razonamiento y de resolución de problemas para dar la respuesta adecuada en cada momento.

Los sistemas cognitivos se basan en tecnologías de Inteligencia Artificial para simular el comportamiento humano, analizar grandes volúmenes de datos y automatizar tareas. Las principales características y beneficios de la computación cognitiva son [UNI, 2021]:

- **Evolución de la IA**, ya no es cuestión de tratar datos no estructurados y automatizar las respuestas ante un suceso A que produzca una respuesta B, hablamos de complementar a la inteligencia humana para mejorar la toma de decisiones.

- El aprendizaje significa **adquirir conocimiento** de cualquier tipo de información y del lenguaje natural.

- Estos sistemas tienen la **capacidad de interactuar con las personas** y proponer acciones a sus necesidades.

- **Automatizar** la respuesta o determinar la mejor solución a un problema determinado.

Para que un sistema sea considerado cognitivo, debe tener las siguientes características:

- **Flexible**: el sistema debe ser capaz de adaptarse a cualquier situación, recopilar los datos y entender las necesidades.
- **Interactivo**: los humanos deben poder interactuar de manera sencilla y natural con el sistema, al igual que lo harían con otra persona.
- **Analítico**: la información que recibe el sistema debe ser suficiente, fiable y de calidad para que sean capaces de determinar los problemas o necesidades y, si no es así, que sean capaces de plantear las preguntas adecuadas para ello.
- **Contextual**: el sistema debe entender y determinar el contexto de la comunicación, como, por ejemplo, significado, sintaxis, tiempo, ubicación...

2.2. La evaluación en segmentación de objetos

Uno de los objetivos de la evaluación es buscar defectos en los productos desarrollados, ya sea por una mala operación, por una respuesta incorrecta o por una respuesta indeseada Natalia Juristo [2006]. Existen múltiples técnicas para realizar la evaluación de software. En una primera división, se diferencia entre técnicas estáticas y dinámicas. Las estáticas buscan faltas sobre el sistema en reposo, esto es, estudian los distintos modelos que componen el sistema software buscando posibles faltas en los mismos. Estas técnicas se pueden aplicar, tanto a requisitos como a modelos de análisis, diseño y código. Por otro lado, las técnicas de evaluación dinámicas generan entradas al sistema con el objetivo de detectar fallos cuando el sistema ejecuta dichas entradas. Los fallos se observan cuando se detectan incongruencias entre la salida esperada y la salida real. La aplicación de técnicas dinámicas es también conocida como pruebas de software o testing y se aplican generalmente sobre código, puesto que es, hoy por hoy, el único producto ejecutable del desarrollo.

La evaluación de un SVC es una etapa necesaria dentro del SDLC que permite depurar errores, validar su uso y verificar su comportamiento. Todos los sistemas inteligentes necesitan ser evaluados de forma que se pueda conocer su comportamiento y poder, así, tomar decisiones. Los conceptos que se manejan en la literatura científica para llevar a cabo diferentes procesos de evaluación son verificación ("verification"), validación ("validation"), testeo ("testing") y garantía ("assurance"), cada uno de los cuales está asociado a una etapa del SDLC [Batarseh et al., 2021].

Existe una amplia cantidad de factores que influyen en un SVC, por lo que existen diferentes metodologías, protocolos, métricas y herramientas para evaluar sus diferentes

partes y garantizar su ejecución exitosa [Cockburn, 2002, Mishra and Dubey, 2013, Bourque and Fairley, 2014]. Un factor influyente en la evaluación es el tipo de problema, es decir, el dominio de la aplicación, el contexto de la escena, las características de la tarea, etc. Otro factor influyente en la evaluación es la forma de resolver el problema de forma computacional, esto es, el tipo de SVC, los datos disponibles o los agentes que interactúan en él. Además, cuando se desarrolla un SVC para un ecosistema variable, como por ejemplo, los sistemas que utilizan imágenes de RM, la generalización de algoritmos de segmentación es problemática ya que el ajuste de parámetros que se hace en la etapa de implementación no ofrece buenos resultados cuando cambian las propiedades de las imágenes de entrada, por lo que una buena solución es aprender de los errores [Wang et al., 2013].

Los sistemas tradicionales de evaluación en segmentación de objetos han tratado al sistema, de forma general, como una caja negra y han analizado su rendimiento con una métrica o un conjunto de ellas en función de los datos disponibles y del objetivo de evaluación. Existen multitud de métodos y métricas en la literatura científica, que varían según Wang et al. [2020] en función del tipo de aplicación de las imágenes. En el caso de aplicaciones médicas, los requisitos que se utilizan para la evaluación son más estrictos, por lo que la mayoría de los métodos son supervisados, es decir, utilizan una imagen de referencia para comparar los resultados, permitiendo analizar de forma objetiva y veraz el error en segmentación.

En una reciente revisión sobre la evaluación en inteligencia artificial, Batarseh et al. [2021] analizaron múltiples trabajos, entre los años 1985 y 2021, y concluyeron que para garantizar el éxito de los nuevos métodos de evaluación éstos deben ser automáticos, permitir la participación de los usuarios y ser específicos a un objetivo y a una subárea de la IA.

2.2.1. La segmentación de referencia

Disponer de una referencia precisa, completa y confiable, es un gran reto en proyectos de gran envergadura y con contextos cambiantes. El patrón de oro (también llamado Gold Standard (GS) o referencia) es, según Segen's Medical Dictionary, «una evaluación clínica estandarizada, método, procedimiento, intervención o medida, que se utiliza para conocer la validez y la fiabilidad de una prueba y que es, generalmente, tomada por ser la mejor disponible, contra el cual se comparan los nuevos ensayos, resultados o protocolos».

En imagen médica, la segmentación de referencia se crea, generalmente, mediante delineación y etiquetado manual. Es un proceso complejo, que depende de los diferentes expertos, costoso en tiempo y puede contener imprecisiones. La imprecisión se puede deber a diferentes causas: 1) a la forma de mostrar los datos a los expertos (iluminación

del monitor y de su entorno, calidad de la pantalla, etc.), 2) a la interpretación humana de los datos (diferentes criterios inter-experto) y/o 3) a la naturaleza del problema (oclusiones, ruido, contornos difusos, etc). En cualquier caso, aunque imprecisa y costosa, la segmentación de referencia permite una evaluación correcta y objetiva.

Cuando se cuenta con resultados de diferentes expertos de dominio, para mejorar la robustez y eficiencia de la referencia, se suele generar una única solución en función de las existentes, que será utilizada para la comparación con los resultados propuestos por un sistema computacional. Para ello, hay que seleccionar el método por el cual se genera la solución a partir de las múltiples soluciones proporcionadas por los diferentes expertos. Existen varios métodos de fusión de etiquetas como el voto por mayoría [Artaechevarria et al., 2009], el voto ponderado global [Sabuncu et al., 2010] o local [Isgum et al., 2009], el método STAPLE [Warfield et al., 2004] o el promedio basado en la forma [Rohlfing and Maurer, 2007]. En la figura 2.4 se muestra un ejemplo gráfico de fusión de etiquetas por el criterio del voto por mayoría, donde, dadas tres segmentaciones con igual peso, la segmentación de referencia es aquella en la que están presentes al menos dos de ellas.

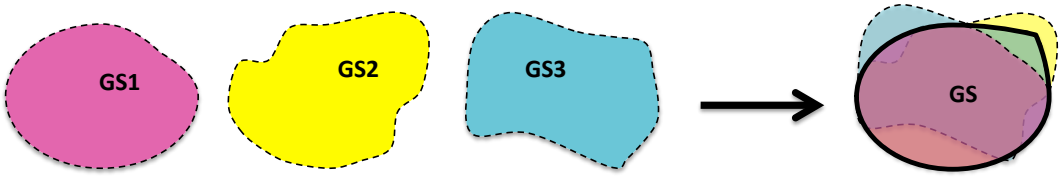


Figura 2.4: Ejemplo de generación de segmentación de referencia mediante voto por mayoría

2.2.2. El error en segmentación

El error en segmentación se analiza, generalmente, mediante análisis cuantitativos que estudian la correcta localización y delineación de los objetos. Los objetos de la segmentación propuesta por el SVC se comparan con los objetos de la segmentación de referencia mediante sus máscaras binarias, esto es, se trata, básicamente, de un problema de clasificación binaria. En su evaluación se utiliza la matriz de confusión a partir de la cual se calculan multitud de métricas del redimiendo del sistema. Se puede obtener dos tipos de medidas de evaluación en función del elemento a comparar: (1) los objetos, para evaluar su rendimiento en cuanto a detección y (2) los píxeles/vóxeles, para evaluar su rendimiento en cuanto a delineación.

De forma general, en un problema de clasificación binaria, se pueden dar dos tipos de error, como se muestra en figura 2.1: los falsos positivos (el error de tipo I) y los falsos negativos (el error de tipo II). En la comparación de dos segmentaciones a nivel

de píxel/vóxel, el error de tipo I o falso positivo (FP) se da cuando un píxel/vóxel de la segmentación propuesta no se detecta (su valor es 0) y en la segmentación de referencia sí se detecta (su valor es 1) mientras que el error de tipo II o falso negativo (FN) se da cuando el píxel/vóxel de la segmentación propuesta sí se detecta (su valor es 1) y en la segmentación de la referencia no (su valor es 0). La operación lógica que se utiliza para su detección es la comparación espacial entre las imágenes binarias de la propuesta y de la referencia.

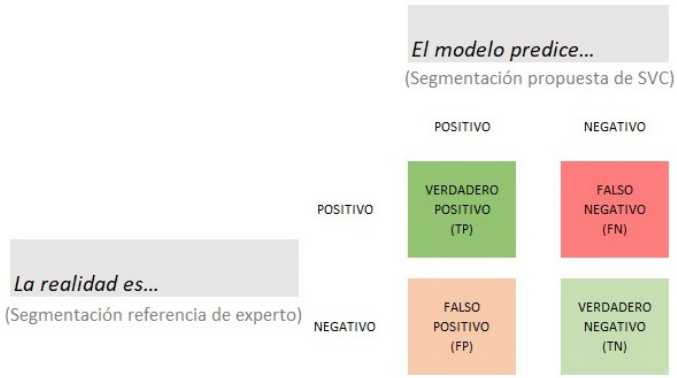


Tabla 2.1: Matriz de confusión
Fuente: Imagen adaptada de [Gil, 2022]

A nivel de objetos, esto es, analizando todos los píxeles contiguos que están activados (valor a 1) en una de las segmentaciones, se pueden diferenciar tres situaciones, una de éxito y dos de error (ver figura 2.5). Los objetos de éxito, es decir, detectados ("Detected") se dan cuando hay contacto entre las segmentaciones y puede haber zonas correctas, es decir, con píxeles que son verdaderos positivos (TP, de su abreviatura en inglés) y zonas de error (FP1/FN1). Los objetos de error, es decir, los objetos no detectados se dan cuando no hay contacto entre las segmentaciones y se diferencian en dos tipos en función de la procedencia del objeto. Los objetos no detectados ("Miss") cuando todos los píxel son de la referencia, es decir, son todos de error tipo I (FN2) y los objetos extra detectados ("Extra") cuando todos los píxeles son de la propuesta, es decir, son todos de error tipo II (FP2).

Con la información de la comparación entre los valores reales (los de la referencia) y los valores propuestos (los ofrecidos por el SVC) se crea una matriz de confusión que permite calcular diferentes medidas estadísticas para evaluar el rendimiento del sistema. En la tabla 2.2 se muestra la definición matemática de algunas de ellas, como la sensibilidad, la especificidad, el valor predictivo positivo, el valor predictivo negativo o la exactitud, los cuales son conceptos fundamentales para valorar un problema de clasificación.

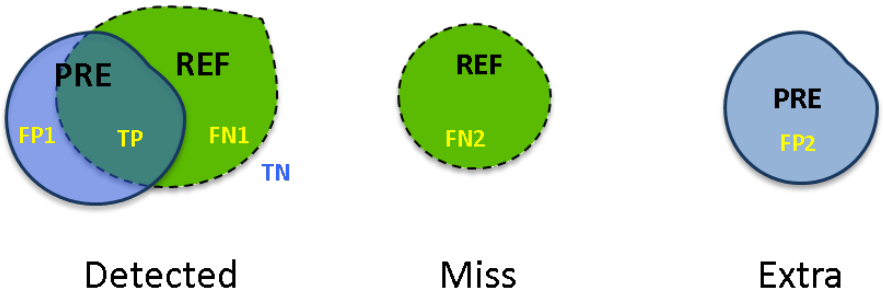


Figura 2.5: Tipos de error de detección en segmentación de objetos

Sensibilidad (Recall)	$\frac{TP}{TP+FN}$
Exactitud (Accuracy)	$\frac{TP+TN}{TP+FP+TN+FN}$
Especificidad (Especificity)	$\frac{TN}{TN+FP}$
Precisión o Valor Predictivo Positivo (Precision)	$\frac{TP}{TP+FP}$
Índice F1 (F1-Score)	$\frac{2(precision*recall)}{precision+recall}$

Tabla 2.2: Medidas estadísticas básicas de precisión

Los métodos de evaluación supervisada, también denominados métodos de discrepancia empírica [Zhang, 1995], son los usados habitualmente para estudiar el error en segmentación. Según Taha and Hanbury [2015], se diferencian seis categorías: 1) medidas basadas en solape espacial, 2) medidas basadas en volumen, 3) medidas basadas en conteo de pares, 4) medidas basadas en información teórica, 5) medidas probabilistas y 6) medidas basadas en distancia espacial. En función de los elementos que utilizan para su cálculo se diferencian tres métodos, en los que de forma general, se pueden inscribir la mayoría de las métricas: los métodos basados en píxeles, los métodos basados en objetos y los métodos basados en distancias.

El método supervisado es un método empírico, directo y objetivo Yang et al. [1995] que se basa en la utilización de una segmentación de referencia con conocimiento a priori y avalada por, al menos, un experto en el dominio. Algunas medidas son la sensibilidad o la especificidad, el coeficiente de similitud Sørensen-Dice (DSC) [Dice, 1945], la consistencia de error global (GCE) Martin et al. [2001], el índice NPR Unnikrishnan et al. [2007], la consistencia de error a nivel objeto (OCE) Polak et al. [2009], la medida F de objetos y partes Pont-Tuset and Marques [2016], etc.

Cuando no se dispone de imagen de referencia, otra forma objetiva de analizar las imágenes es mediante el método no supervisado [Zhang et al., 2008]. Este método se basa en el análisis de características propias de la imagen como medidas de tendencia

central (media, mediana y moda), de dispersión (varianza y desviación típica) o de posición (cuartiles y percentiles) para describir las distribuciones de la solución. Algunas de estas medidas son la variación de uniformidad inter- e intra-región Otsu [1979], medidas basadas en entropía Zhang et al. [2003], medidas de regularidad de forma Correia and Pereira [2003], etc. que permiten realizar un análisis preciso y objetivo, aunque no pueden asegurar que sea correcto. Este método permite filtrar datos y/o detectar valores anómalos. También permite extraer patrones de error a partir de la descripción multiespectal de los objetos, reuniendo información de las distintas fuentes disponibles, ya sea información visuoespacial de los objetos o información del procesado, pre-procesado o la adquisición.

Otro aspecto importante en la evaluación de segmentación de imágenes es la elección de la forma para describir los objetos, su dimensionalidad, el tipo de segmentación de los objetos y el método de comparación entre ellos. En la descripción de objetos mediante métodos de grano grueso se utilizan máscaras rectangulares que contienen la región segmentada ("Bounding box") y permiten conocer de forma aproximada la similitud entre segmentaciones (posición, tamaño, área ocupada, etc.), mientras que los métodos de grano fino describen los objetos con máscaras precisas a nivel de píxel, mediante la identificación de los bordes del objeto o mediante la identificación de toda la región del objeto, y permiten conocer información más objetiva y precisa del error (diferencias en los contornos, sobre-segmentación y infra-segmentación, etc.). La dimensionalidad de la imagen influye en los diferentes análisis de evaluación que se pueden realizar, ya que pueden ser análisis 1D, 2D, 3D, etc.

También hay que conocer el contexto de las imágenes ya que se pueden realizar análisis a diferentes niveles: a nivel de objetos, a nivel de imagen, a nivel de caso o a nivel de estudio. En cuanto al tipo de segmentación, en la segmentación semántica se detectan los objetos de la misma clase como máscaras binarias mientras que en la segmentación por instancias se utilizan máscaras multinivel para diferenciar entre objetos de la misma clase. Por último, existen diferentes métodos para comparar segmentaciones. Algunos análisis realizan comparaciones del número de objetos o de su volumen, otros realizan estudios estadísticos como test Wilcoxon o el test Friedman para comprobar la similitud entre poblaciones, pero para obtener información exacta y precisa de evaluación es necesario utilizar más información de los objetos como su localización espacial y sus valores de intensidad.

La comparación espacial es el método más objetivo, a pesar de ser más costoso, para evaluar la segmentación de objetos. Al comparar los objetos segmentados producidos por un SVC respecto a una segmentación de referencia, se pueden dar múltiples situaciones, como se muestran en la figura 2.6. En ella, se muestran seis situaciones distintas [Nascimento and Marques, 2006], desde la situación ideal con el solape 1-1, hasta la

situación más compleja con el solape M-N, siendo el primer término el número de objetos de la referencia y el segundo término el número de objetos de la segmentación propuesta por el SVC.

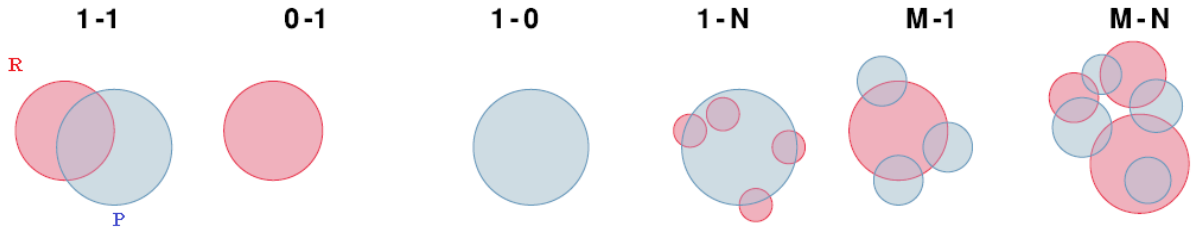


Figura 2.6: Tipos de solape entre segmentaciones según nomenclatura de Nascimento
Fuente: Imagen adaptada de [Carass et al., 2020]

Además, como se muestra en la figura 2.7, se pueden dar ocho relaciones topológicas entre dos regiones binarias según el modelo RCC-8 [Randell et al., 1992]. Por ello, hay que prestar especial atención a las diferentes métricas que existen, como se verá a continuación, y conocer sus fortalezas y debilidades para que ofrezcan una información veraz sobre la situación evaluada.

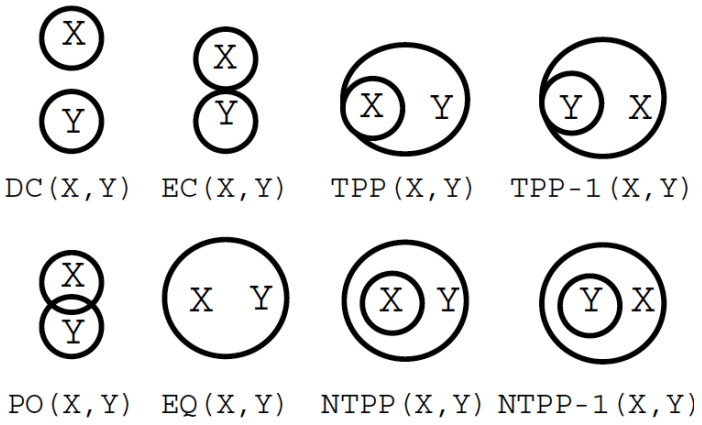


Figura 2.7: Relación topológica entre dos regiones
Fuente: Imagen procedente de Maillot and Thonnat [2008]

2.2.3. Métricas de evaluación en segmentación de objetos

En evaluación de segmentación de objetos existen métricas simples y métricas complejas, y cada una tiene sus características, fortalezas y debilidades. En un estudio realizado por Garcia et al. [2015] se analizaron la invariancia y el grado se similitud de 17 métricas de discrepancia frente a cambios en la matriz de confusión. De sus

conclusiones, cabe destacar que el coeficiente Dice detecta cinco de siete cambios en la matriz de confusión, aunque que no diferencia entre escalados uniformes. Everingham et al. [2002] definen un método general de comparación con múltiples medidas en un espacio multidimensional. Crum et al. [2006] presentan una generalización de medidas de solape que utiliza etiquetado fraccional para reflejar la proporción de tejido en cada vóxel y un análisis de los elementos sin solape utilizando tolerancia espacial.

En imagen médica, existen múltiples métricas de evaluación de la calidad de la segmentación en cuanto a detección y delineación [Polak et al., 2009, Taha et al., 2014, Garcia et al., 2015, Nai et al., 2021, Goumeidane et al., 2003, Cheng et al., 2021, Hao et al., 2009]. Particularizando al caso de lesiones hiperintensas en la sustancia blanca cerebral, las más utilizadas se muestran en la tabla 2.3, cuya descripción se de detalla a continuación [Commowick et al., 2018, Atlason et al., 2019, Kuijf et al., 2019].

DSC	$\frac{2(P \cap R)}{(P \cap R + P \cup R)} = \frac{2TP}{2TP+FN+FP}$
IoU	$IoU = \frac{\lvert P \cap R \rvert}{\lvert P \cup R \rvert} = \frac{TP}{TP+FN+FP}$
AVD	$\frac{\lvert V_R - V_P \rvert}{V_R}$
L-TPR	$\frac{N_P^{TP}}{N_R}$
L-PPV	$\frac{N_P^{TP}}{N_P}$
L-F1	$\frac{2(precision * recall)}{precision + recall}$
$d_H(x,y)$	$max\{sup_{x \epsilon X} inf_{y \epsilon Y} dist(x,y), sup_{y \epsilon Y} inf_{x \epsilon X} dist(x,y)\}$

Tabla 2.3: Métricas de evaluación en segmentación de hiperintensidades de la sustancia blanca cerebral (WMH)

- Basadas en área/volumen:

  - El coeficiente Sørensen-Dice (DSC/SI) [Dice, 1945]: Es un estimador cuantitativo que describe el grado de similitud entre los elementos comparados, expresado en valor numérico, entre 0 y 1. Siendo P la segmentación propuesta por un SVC y R la segmentación de la referencia. Un DSC de 1 indica un solape perfecto.
  - La intersección sobre la unión (IoU): También conocido como indice Jaccard, es otro estimador cuantitativo que describe el grado de similitud entre los elementos comparados. Es siempre mayor que DSC excepto en los extremos [0,1], además las dos métricas se pueden relacionar $DSC = \frac{2IoU}{1+IoU}$.

- La diferencia volumétrica absoluta (AVD, de sus siglas en inglés): Es una medida de la diferencia de volúmenes entre la referencia y la propuesta en valor absoluto en función del volumen de la referencia, donde V_R y V_P indican el volumen de la referencia y la propuesta del SVC respectivamente. Bajos valores de AVD indican una buena propuesta de segmentación.

- Basadas en objetos/lesiones:

  - La tasa de verdaderos positivos (L-TPR) o también llamada sensibilidad ("recall"): Mide el número de lesiones correctamente localizadas en la propuesta (N_P^{TP}) respecto al total de lesiones de la referencia (N_R). Valores altos de L-TPR/recall indican un mejor rendimiento.
  - El valor predictivo positivo (L-PPV) o también llamado precisión ("precision"): Mide el número de lesiones correctamente localizadas en la propuesta (N_P^{TP}) respecto al total de las lesiones predichas (N_P).Valores altos de L-PPV/precisión indican un mejor rendimiento.
  - El valor F1 (L-F1): Relaciona los valores de la precisión y de la sensibilidad mediante una media armónica. Valores altos de L-F1/precisión indican mejor rendimiento.

- Basadas en distancia entre contornos:

  - La distancia modificada de Hausdorff (H95): La distancia de Hausdorff mide la distancia más larga que se debe recorrer desde un punto de un conjunto hasta un punto del otro conjunto, donde dist(x,y) denota la distancia entre x e y, sup denota el supremo e inf el ínfimo. En esta medida se utiliza el percentil 95 en lugar de la distancia máxima, ya que la distancia de Hausdorff es sensible a los valores atípicos. Las puntuaciones más bajas de H95 indican un mejor rendimiento.

En imagen médica, el coeficiente de similitud Sørensen-Dice (DSC) o el índice Jaccard son medidas basadas en píxel muy utilizadas [Guindon and Zhang, 2017, Carass et al., 2020, Eelbode et al., 2020] pero hay que tener en cuenta que son sensibles a la carga de lesiones y a su tamaño, por lo que es importante conocer su comportamiento para los distintos escenarios.

En el caso concreto del estudio de manchas de sustancia blanca cerebral, las medidas de evaluación pueden estar influenciadas por la cantidad de lesiones del paciente, el uso de diferentes instrumentos de captura, el uso de distintos protocolos de adquisición (como variaciones en el nivel del campo magnético), como destacan Heinen et al. [2019].

Por ejemplo, Griffanti et al. [2016] comparan su método respecto a diferentes algoritmos de segmentación de lesiones hiperintensas, muestran el valor medio total y el valor para diferentes cargas de lesiones del coeficiente de similitud Sørensen-Dice (ellos lo denominan SI, "similarity index") en sus resultados. Commowick et al. [2018] también analizan la influencia del número de lesiones y de su carga total, cuyas gráficas de resultados se muestran en la figura 2.8. Tanto en uno como en otro artículo, se puede observar una gran variabilidad en los valores, por lo que es importante tener herramientas que permitan acceder y manipular los resultados de evaluación para poder entender los diferentes comportamientos.

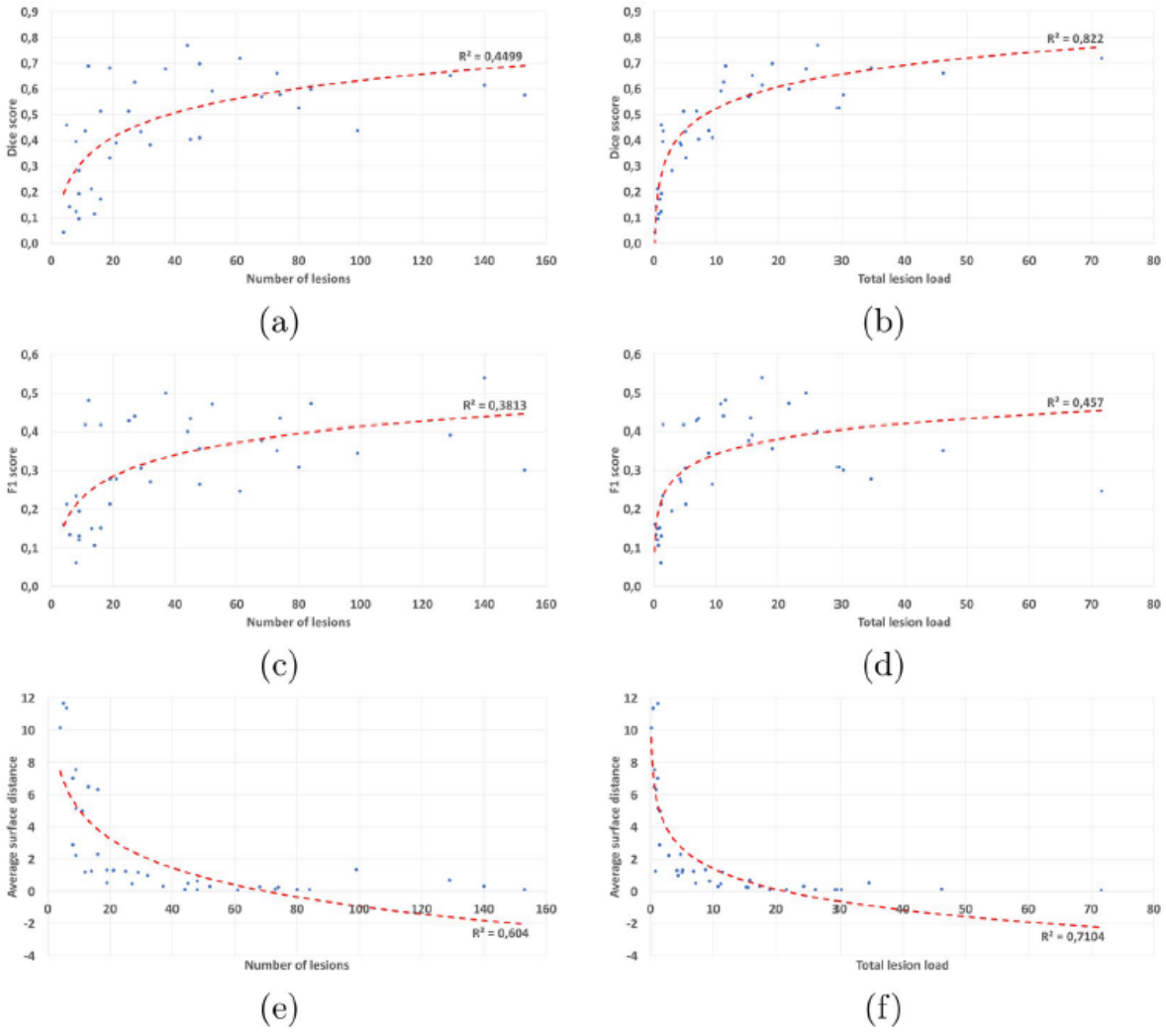


Figura 2.8: Ejemplo de dependencia de distintas métricas con el número de lesiones y con su volumen total
Fuente: Imagen procedente de Commowick et al. [2018]

Todas estas medidas de evaluación permiten conocer la calidad de la segmentación de forma global y ayudan en el proceso de ajuste de parámetros y en la comparación de

algoritmos, pero no permiten conocer características concretas de dónde se producen los errores, cómo son o si existen zonas con los mismos comportamientos erróneos, por lo que es necesario desarrollar nuevos modelos de evaluación en segmentación de objetos en los que el análisis del error sea el protagonista.

2.3. Las herramientas de caracterización del error en segmentación

En la literatura científica, hemos encontrado pocos estudios que se centren en el análisis del error en segmentación de objetos con el objetivo de obtener su caracterización. La mayoría de herramientas se centran en su detección e identificación o en analizar de forma extensa la influencia de alguna propiedad de los objetos segmentados, como por ejemplo, estudiar los tipos de error, analizar la influencia del tamaño de los objetos o cualquier otra propiedad.

De los primeros trabajos en analizar la influencia de las características de los objetos en el rendimiento del sistema de detección, se destaca el trabajo realizado por Hoiem et al. [2012]. En él, se examinan en particular los efectos de la oclusión, el tamaño de los objetos, su relación de aspecto y la visibilidad de partes, y permite medir su frecuencia e impacto en diferentes tipos de error, principalmente en los falsos positivos. Proponen una línea de investigación para conocer más información sobre los errores, identificando las debilidades significativas y sugiriendo direcciones de mejora.

En una investigación reciente, donde se analiza de forma más exhaustiva el error en segmentación de objetos la realizan, Caicedo et al. [2019] presentan un marco de evaluación enfocado en la precisión a nivel de objetos que captura las interpretaciones biológicas de forma más natural que las realizadas a nivel de píxel. Para comprender las diferencias de rendimiento entre los métodos evaluados, clasifican los objetos perdidos (“missed”) y los objetos inventados (“extra”) por tamaño y los errores de segmentación por tipo (“splits”/ “merged”).

Otra propuesta reciente es la herramienta TIDE [Bolya et al., 2020] para identificar errores de detección y segmentación, que mejora la propuesta de Hoiem et al. [2012]. Permite resumir, de forma compacta, los tipos de error para facilitar comparaciones, aislar la contribución de cada tipo de error para obtener conclusiones específicas, ser usado para cualquier conjunto de datos, estudiar todos los tipos de error y ajustar el nivel de profundidad del estudio del error.

El problema de estos trabajos es que asumen que los modelos están entrenados con conjuntos de datos adecuados y sus objetos tienen características bien definidas, por lo que sería interesante comprobar su funcionamiento para el caso de conjunto de datos

imprecisos como los objetos amorfos. También sería interesante conocer si existen otras características que permitan mejorar el conocimiento de los errores.

2.4. Conclusiones

Disponer de un buen sistema de evaluación general, que gestione la gran cantidad de elementos relacionados en un SVC, que organice todos los datos y sus características, es un gran reto. Los procesos de evaluación actuales analizan tanto el sistema como sus resultados, y ofrecen medidas en función del objetivo del análisis (rendimiento, fiabilidad, eficiencia, usabilidad, etc.) y la profundidad del análisis.

Los métodos tradicionales de evaluación en segmentación de objetos se centran, generalmente, en analizar el acierto de los resultados de forma global, pero no consideran dónde fallan los sistemas computacionales. No ofrecen información para conocer en profundidad cómo son los errores, dónde se producen o porqué se producen. Tampoco permiten saber si existen patrones de comportamiento erróneo, cuáles son las características de los casos atípicos, o relacionarlos con su contexto, entre otras acciones. Además, para realizar una evaluación objetiva y veraz de la calidad de una segmentación, especialmente en aplicaciones sensibles como la medicina o la seguridad, es necesario disponer de una segmentación de referencia completa y confiable, aunque crearla sea un proceso costoso.

El análisis de los resultados de evaluación es todo un arte, donde la influencia de los métodos empleados, la distribución de los datos y los objetivos de análisis hacen que se puedan obtener distintas medidas, condicionando la interpretación de los resultados. Un valor que en un primer momento puede parecer bueno, tras analizar cómo ha sido calculado puede ocultar problemas o zonas de menor éxito.

La mejora del proceso de refinamiento de un SVC pasa por mejorar la etapa de evaluación con nuevas metodologías de análisis de los resultados que permitan automatizar la extracción de nuevo conocimiento y ayude a enfocar los esfuerzos de los expertos siguiendo los principios de los sistemas cognitivos. Para ello, es necesario realizar mejoras en los sistemas para que compartan información, y no solo la intercambien [Romá Ferri, 2009], mediante representaciones formales como las ontologías, que permiten modelar el conocimiento, tanto de la tarea de evaluación como del dominio de la aplicación. También es recomendable que los SVC se diseñen con arquitecturas abiertas, de caja gris o de caja blanca, donde las diferentes etapas sean descritas de forma completa por modelos ontológicos [Maillot and Thonnat, 2008, Clouard et al., 2010, Sánchez et al., 2012, Hoehndorf et al., 2013, Fatimaezzahra et al., 2016], se utilicen algoritmos explicables [Došilović et al., 2018, Markus et al., 2021, Gerlings et al., 2021] y se facilite

la interacción con los expertos mediante herramientas con facilidades visuales [Mazza, 2009] que permitan analizar los resultados con transparencia, reducción de sesgos e interpretabilidad.

Capítulo 3

La metodología AMOSE^2

La automatización del proceso de diseño de un SVC no puede ser completa en el momento actual de la tecnología porque no todo el conocimiento está codificado y se depende mucho de los expertos. Actualmente, durante el proceso de diseño, para mejorar el comportamiento de un SVC es necesario que los expertos involucrados en su definición detecten dónde se cometen los fallos y propongan mejoras que los solucionen. La detección de los fallos se realiza en la etapa de evaluación del sistema, normalmente mediante un procedimiento ad hoc, creado expresamente para cada SVC.

Por tanto, sería de gran ayuda el desarrollo de herramientas avanzadas de evaluación, interactivas y explicables, que faciliten la integración de los expertos en el ciclo de diseño. Estas herramientas les permitirán realizar estudios más sistemáticos y detallados de los errores y, en consecuencia, ayudarán a establecer el origen de los fallos y buscar soluciones. Además, estas herramientas pueden acabar, en un futuro, formando parte de nuevos sistemas cognitivos centrados en la automatización de la tarea de diseño en SVCs.

Para avanzar en este camino, se describe en este capítulo una metodología para la caracterización automática del error en segmentación centrada en objetos amorfos. Se ha denominado AMOSE^2 de las siglas en inglés "AMorphous Object Segmentation Error Evaluation".La metodología propuesta se lleva a cabo dentro de un proceso de evaluación dinámica ya que se realiza una validación de la funcionalidad de los algoritmos de segmentación de objetos con el objetivo de extraer nuevo conocimiento a partir del estudio del error. Esta metodología es un paso hacia la automatización del análisis del error con técnicas de IA y, para ello, propone introducir nuevos módulos en el ciclo de diseño para automatizar el análisis del error y facilitar la descripción de los errores cometidos.

En la sección 3.1 se dará una visión general de la metodología AMOSE^2, la cual pone el foco en el proceso de evaluación dentro del ciclo de diseño de sistemas de visión artificial. La metodología consta de cuatro módulos que se describen en las secciones 3.2, 3.3, 3.4 y 3.5. A partir de sus definiciones, se han implementado varías herramientas software

que componen el sistema AMOSE². Por último, en la sección 3.6 se detallan los tipos de análisis que se pueden realizar siguiendo esta metodología.

3.1. Descripción de la metodología

La metodología AMOSE² es una metodología de caracterización del error en segmentación de objetos amorfos cuyo objetivo es la identificación de patrones de error y anomalías mediante técnicas de inteligencia artificial. Se enmarca dentro de la etapa de evaluación del ciclo de desarrollo iterativo e incremental de un SVC, cuyos fundamentos se describieron en el capítulo 1 (figura 1.2).

Siguiendo la figura 3.1, en un desarrollo de software iterativo e incremental se parte, al inicio, de un conocimiento preliminar, "K(t_0)", compuesto por unos requisitos y preferencias, y de un conjunto de datos de referencia. A partir de ellos, se diseña e implementa un primer SVC que propone una solución, la cual se evalúa de manera supervisada utilizando los datos de referencia (una serie de imágenes de segmentación previamente etiquetadas).

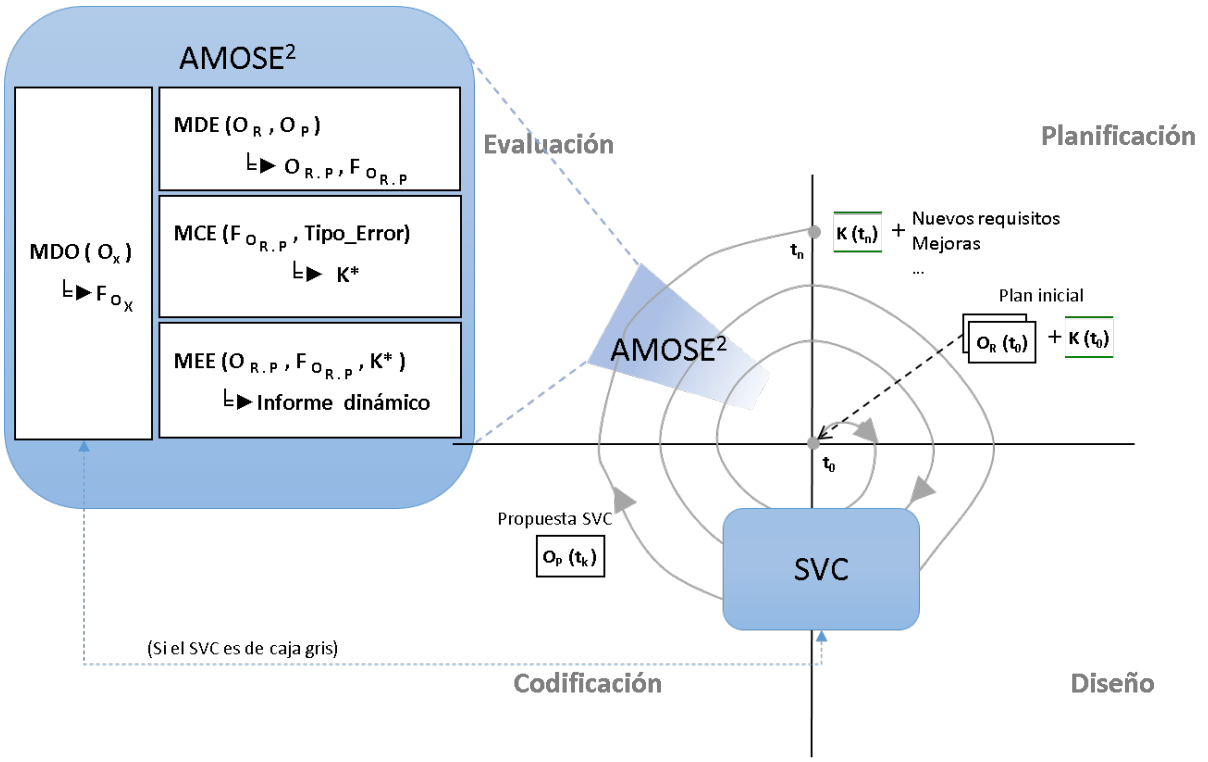


Figura 3.1: La metodología AMOSE² dentro del ciclo de diseño iterativo e incremental

Particularizando al problema de la segmentación de objetos amorfos, el SVC ofrece una imagen de segmentación que contiene los objetos de la segmentación propuesta "$O_P(t_0)$", y es evaluado utilizando una imagen de segmentación de referencia que contiene los

objetos de referencia "$O_R(t_0)$" con el objetivo de detectar errores en la segmentación. El análisis ponderado de estos errores generará críticas en el proceso de diseño del SVC que darán lugar a nuevos requisitos y restricciones (y posiblemente también a una mejora de los datos de referencia). Con cada análisis de evaluación del error se generará nuevo conocimiento, "$K(t_k)$" y nuevos datos de referencia, '$O_R(t_k)$", lo que se utilizará para, en una nueva iteración del ciclo de diseño, definir un nuevo sistema mejorado que proporcionará una nueva segmentación, "$O_P(t_k)$". Este proceso cíclico de refinamiento continuará hasta alcanzar el objetivo, esto es, el nivel de exigencia deseado.

Una forma de aumentar la información de los objetos segmentados es utilizar SVCs que aporten información, aunque sea limitada, de su comportamiento y estructura interna. En esta línea, según Mohd Ehmer Khan [2012], los sistemas software se pueden dividir en tres categorías en función del conocimiento interno de su estructura y su funcionamiento: (1) Caja-negra, cuando se conocen aspectos generales pero no se conoce el funcionamiento interno; (2) Caja-gris, cuando se dispone de un conocimiento limitado del funcionamiento y estructura interna del sistema; y (3) Caja-blanca, cuando se conocen en detalle la estructura y el funcionamiento de todo el sistema.

Para profundizar en el estudio del error en la segmentación de una imagen, es deseable disponer de una mayor información sobre el SVC, sus procesos internos y sus fuentes de datos. El sistema ideal sería el de caja blanca, donde se tiene acceso total a la información, pero esto sólo es posible si se ha desarrollado el sistema o si se dispone del código fuente. Por tanto, un sistema de caja-gris es un primer paso para mejorar la evaluación, ya que permite añadir nuevo conocimiento al proceso de análisis del error.

Una forma sencilla de pasar de un sistema de caja negra a un sistema de caja gris es mediante el acceso a información sobre características calculadas en etapas internas del proceso de segmentación, información de configuración, etc., como se muestra en la figura 3.2. Un sistema de caja gris ofrece a la salida, junto a la segmentación propuesta, una serie de información asociada a dicha imagen, como se muestra con la línea discontinua azul "*1". Los datos accesibles de la "Información Interna de la Propuesta" contendrán información sobre la configuración utilizada en el algoritmo e información de características locales, globales o del entorno de etapas internas y finales de la segmentación propuesta por el SVC. Para poder comparar estos resultados internos con los que se obtendrían con la segmentación de referencia, es necesario poder acceder al sistema para alimentarlo con datos de la referencia para obtener la "Información Interna de la Referencia".

Para descubrir nuevo conocimiento sobre los errores de segmentación de objetos es necesario mejorar los procesos de evaluación del error. La metodología AMOSE^2 presenta dos novedades principales, 1) el modelado individual de los objetos de error y 2) una

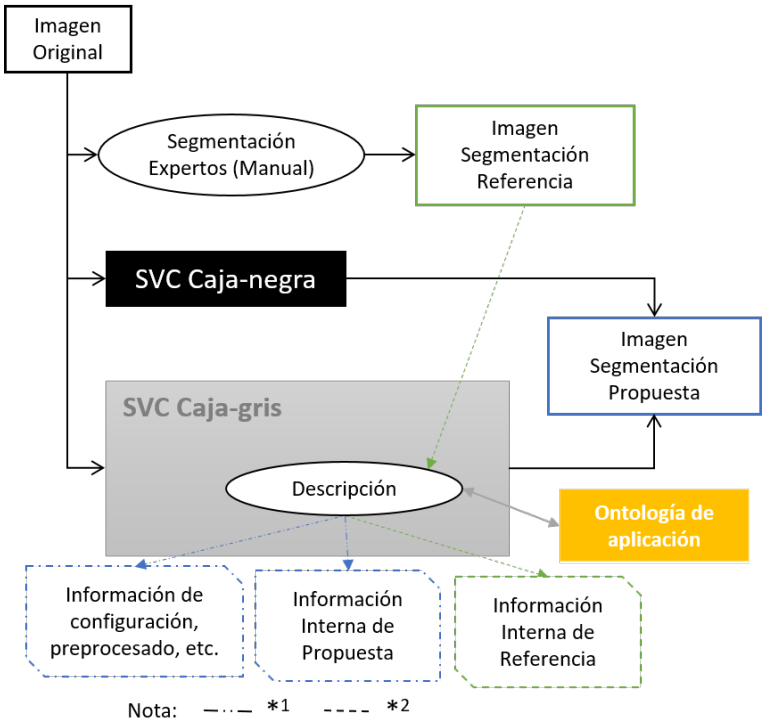


Figura 3.2: Diagrama de funcionamiento del módulo del módulo de descripción ontológico (MDO)

descripción más detallada de los objetos segmentados, lo que le permite realizar un análisis más profundo de los errores de segmentación.

La metodología AMOSE[2] parte de las siguientes premisas:

1. Al segmentar una imagen se pueden encontrar múltiples objetos, también llamados "blobs" en el campo del procesado de imagen y la visión artificial. Si en una segmentación se separa el primer plano del fondo, los blobs son los conjuntos de píxeles que comparten alguna característica común, están contiguos y pertenecen al primer plano. Estos blobs pueden describirse de manera independiente mediante características propias de los objetos y de su contexto.

2. La evaluación del error sigue una estrategia supervisada, esto es, se compara la segmentación propuesta por el sistema con una segmentación de referencia.

3. Si dos segmentaciones (blobs) coinciden en algún punto de la imagen, entonces corresponden (hacen referencia) al mismo objeto.

4. Para comparar la segmentación propuesta con una de referencia, se combinan ambas segmentaciones, lo que genera nuevos blobs. Estos nuevos blobs, que llamaremos blobs de agrupación, darán lugar a los "objetos de agrupación de error", que tendrán

características propias de la agrupación más otras derivadas de las propiedades de los blobs participantes en la agrupación. Estas propiedades servirán para detectar y describir los errores.

5. La metodología AMOSE^2 se particularizará para el análisis de imágenes biomédicas volumétricas (3D), cuya unidad mínima es el vóxel. Sin embargo, en muchas ocasiones, por reducir el coste computacional, se trabaja con imágenes 2D (procesado por lonchas o, en inglés, "slices"). Por este motivo, tanto en el capítulo 3 como en el 4, se utilizarán casi indistintamente los términos vóxeles y píxeles.

En AMOSE^2 se definen los elementos, los procesos y los métodos para facilitar que los expertos involucrados en el diseño del sistema descubran nuevo conocimiento sobre las características del error. La metodología consta de cuatro módulos, los cuales están asociados a las tareas que fueron identificadas en la sección 1.3. Avanzamos aquí una breve descripción de estos módulos que completaremos en las secciones siguientes:

MDE: El "Módulo de Descripción del Error" (MDE) describe cómo comparar dos segmentaciones, la segmentación propuesta y la de referencia, para detectar las discrepancias entre ellas. Este módulo realiza una descripción independiente para cada objeto, tanto individual ("objeto ancesto") como de error ("objeto de agrupación"), genera un vector de características multidimensional que describe cada objeto de error y lo clasifica por tipo de error. El módulo MDE cubre las tareas 2, 3 y 4.

MCE: El módulo MDE proporciona información independiente de cada error que se produce sobre el conjunto de datos utilizado para evaluar el sistema de segmentación. Esta información está, por tanto, desagregada. El "Módulo de Caracterización del Error" (MCE) realiza el análisis para agregar esta información y descubrir los comportamientos de error relevantes (agrupaciones de error, outliers y error comparado). El módulo MCE cubre la Tarea 5.

MEE: El "Módulo de Exploración del error" (MEE) proporciona las herramientas visuales e interactivas necesarias para explorar el nuevo conocimiento sobre los patrones de error detectados y facilitar así su comprensión y posterior uso. El módulo MEE cubre la Tarea 6.

MDO: El "Módulo de Descripción Ontológica" (MDO) es un módulo transversal, basado en ontologías, creado para dar asistencia al resto de módulos. En él se define el modelo ontológico utilizado para describir cada objeto de error (objeto de agrupación). Este modelo define la estructura de relaciones semánticas entre

las características utilizadas para describir el error y la implementación de los operadores que van a cuantificar dichas características. El módulo MDO sirve, por tanto, de apoyo a las tareas 3 y 5 .

3.2. El módulo de descripción del error

El Módulo de Descripción del Error (MDE) tiene como objetivo identificar, describir y clasificar los objetos de agrupación obtenidos al comparar los objetos de la segmentación propuesta por un SVC "O_P" con los objetos de la segmentación de referencia "O_R". Se compone de tres etapas, como se muestra en el diagrama de la figura 3.3. Estas etapas coinciden, respectivamente, con las tareas 2, 3 y 4 propuestas en la metodología (sección 1.3). A continuación, se describe en detalle cada una de estas etapas.

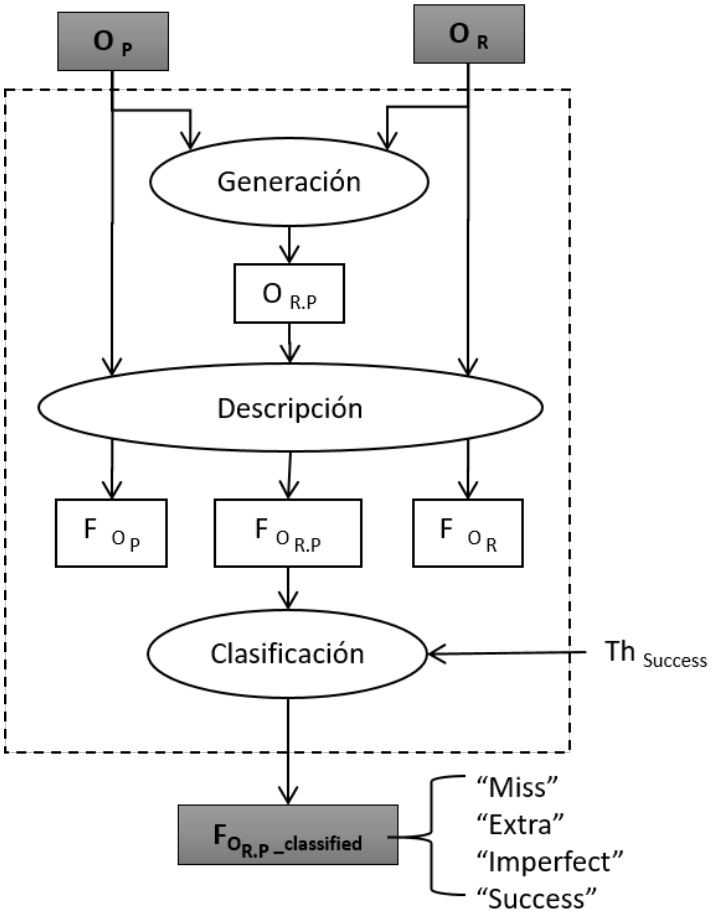


Figura 3.3: Diagrama del módulo de descripción de error (MDE)

3.2.1. Generación de agrupaciones

Al comparar los objetos de la segmentación propuesta por un SVC "O_P" con los objetos de la segmentación de referencia "O_R" se define un nuevo objeto, el objeto de agrupación "$O_{R.P}$", el cual se utiliza para analizar la posibilidad de error en la segmentación.

Para generar los objetos de agrupación $O_{R.P}$ y preservar la información de sus ancestros (sus padres), se utiliza una representación que permite identificar el solape espacial. Para ello, se genera una nueva imagen de fusión en la que se asigna, a cada píxel, una etiqueta en formato decimal en la que la parte entera hace referencia al objeto de la referencia y la parte decimal hace referencia al objeto de la segmentación propuesta. En caso de que no haya ningún objeto de la referencia o de la segmentación en un píxel, su valor será 0 en la parte entera o la parte decimal respectivamente. La ecuación 3.1 describe esta representación. Posteriormente, se distinguen los objetos de agrupación separando los grupos de píxeles en contacto en esta imagen de fusión (se utiliza vecindad 8 para el análisis bidimensional y vecindad 26 en el tridimensional).

$$Etiqueta(O_{R.P}) \quad = \quad Etiqueta(O_R) + \frac{Etiqueta(O_P)}{10^{lenght(char(max(Etiqueta(O_P))))}} \tag{3.1}$$

La figura 3.4 muestra un ejemplo de generación de un objeto de agrupación bidimensional a partir de un objeto de referencia y de tres objetos de propuesta. En el nuevo objeto aparecen diferentes sub-regiones (grupos de vóxeles con la misma etiqueta) que definen las distintas zonas de error. Cada píxel, y por extensión la zona, puede tener una de estas etiquetas:

- "Sin error", éxito o verdadero positivo (TP), que se da cuando las partes entera y decimal son mayores que cero (en color rojo).
- "Con error de infra-segmentación" o falso negativo (FN), que se da cuando el valor de la parte decimal es cero (representada en color verde).
- "Con error de sobre-segmentación" o falso positivo (FP), que se produce cuando el valor de la parte entera es cero (representada en color azul).

En función de los objetos de la segmentación propuesta y de la segmentación de referencia que participan en un objeto de agrupación se pueden dar diferentes situaciones, tal como se muestra en la figura 3.5. A la izquierda se muestran 9 objetos segmentados de la referencia O_R en distintas posiciones en la imagen, en el centro se muestran también 9 objetos segmentados de la propuesta O_P, y a la derecha se muestran los objetos de agrupación resultantes $O_{R.P}$, siguiendo la mismo código de color que el utilizado en la

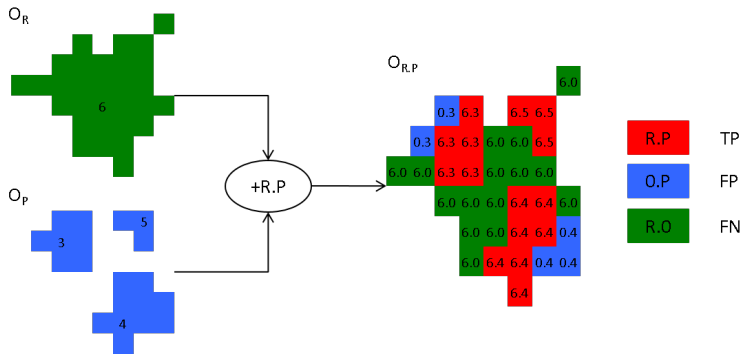


Figura 3.4: Ejemplo de representación decimal de los píxeles de la imagen fusión de la segmentación propuesta y la de referencia para obtener un objeto de agrupación

figura anterior. El resultado son 8 objetos de agrupación independientes de forma y composición diversa: dos objetos sin contacto (A y H) y seis objetos con contacto (B, C, D, E, F y G). El objeto de agrupación A es un objeto no detectado: todos los vóxeles pertenecen a la referencia, es decir, todos los vóxeles son falsos negativos. El objeto de agrupación H es un objeto extra: todos los vóxeles pertenecen a la propuesta, o sea, son falsos positivos. Otro caso reseñable es el objeto D, con solo contacto: no existen vóxeles verdaderos positivos, sólo tiene un contacto por dos lados. El resto de objetos de agrupación (B, C, E, F y G) presentan solape en distinta proporción. Como se puede apreciar, de esta manera se pueden calcular de forma exacta las zonas de error o el número de objetos implicados en la agrupación, y se pueden gestionar los casos más complejos asociados a solapes múltiples, como el objeto de agrupación F. En definitiva, hemos aumentado la información sobre los objetos segmentados, lo que permitirá mejorar la descripción de los errores y, así, proporcionar más información útil para la etapa de evaluación del sistema.

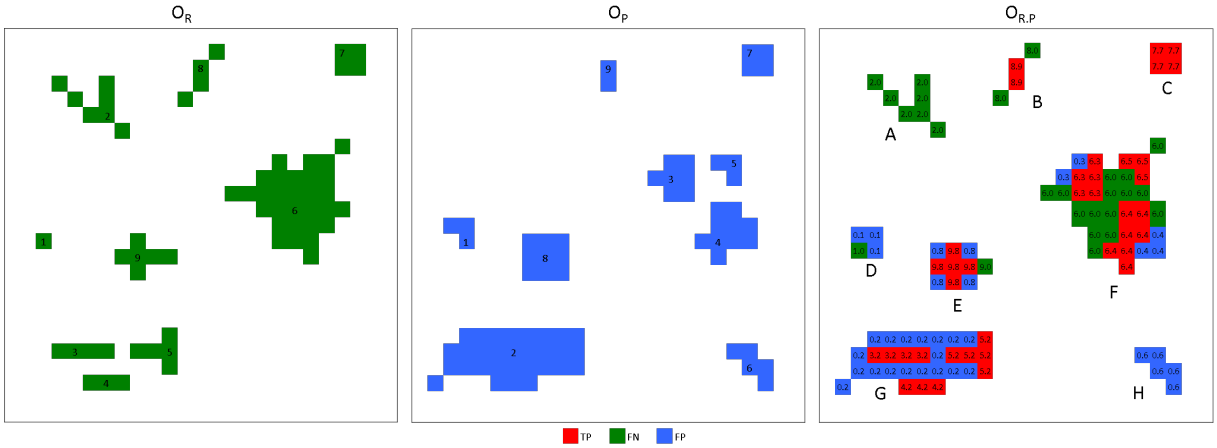


Figura 3.5: Ejemplos de composición de distintos tipos de objetos de agrupación

3.2.2. Descripción de agrupaciones

En esta segunda etapa, se describen los objetos de agrupación $O_{R.P}$. Para ello, por cada objeto de agrupación, se crea un vector de características multidimensional con toda la información descriptiva que pueda ser relevante para la caracterización del error de segmentación. Este vector de características contiene tanto propiedades globales del propio objeto de agrupación $O_{R.P}$, como propiedades de los objetos que lo componen, O_P y O_R.

Por un lado, analizando el objeto de agrupación en su conjunto, se añaden propiedades de las sub-regiones de error como su número, la localización del error (interior o periferia del objeto), la existencia de agujeros o surcos, la disposición (si el error está disperso o agrupado), etc. Por otro lado, se añaden propiedades de los objetos de la segmentación propuesta y de la referencia que participan en el objeto de agrupación (objetos padres o ancestros del objeto de agrupación). Además, si el SVC es de caja gris, se puede añadir al vector de características información sobre estados internos del sistema, configuración de subprocesos, resultados intermedios, etc., lo que enriquece enormemente la descripción de los objetos de agrupación. Su diagrama general se muestra en la figura 3.6.

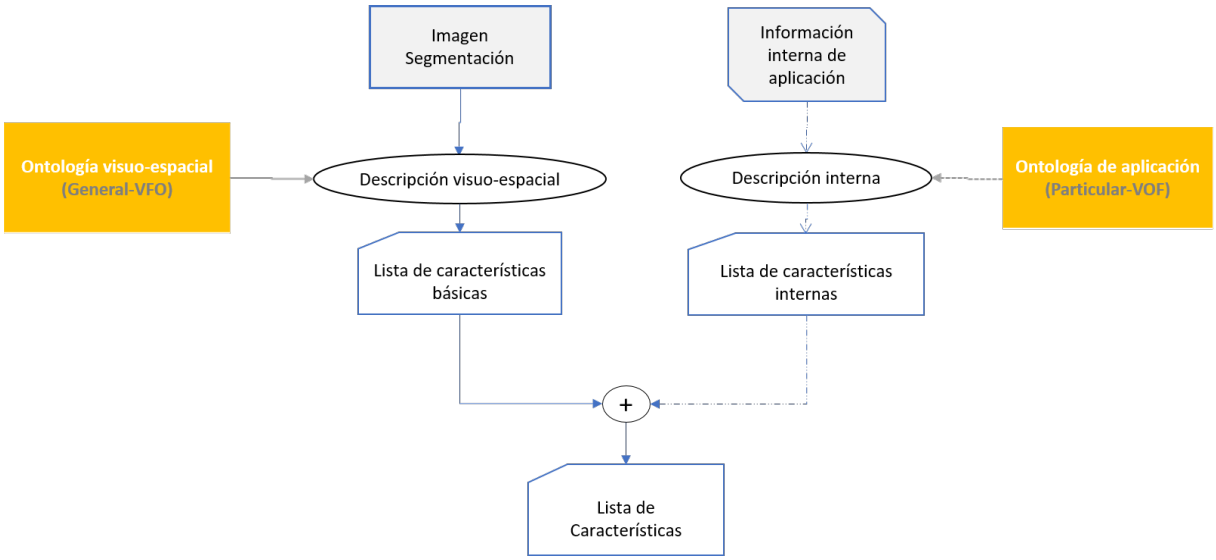


Figura 3.6: Diagrama de generación de características multidimensionales de objetos de agrupación

Un objeto de agrupación de puede caracterizar de forma externa a partir de la imagen de segmentación y del resto de fuentes de datos que estén disponibles, como la imagen de entrada al sistema de segmentación u otras imágenes disponibles en el dominio. A modo de ejemplo, en la tabla 3.1 se describen las características básicas de los $O_{R.P}$ para el caso de estudio de la segmentación de objetos amorfos que se detallará en el capítulo 4. Solo con máscaras binarias de la segmentación, podemos determinar el

tamaño, la posición o la forma de los objetos, mientras que con éstas y las imágenes de entrada se puede calcular características de color, textura, etc. Además, con información adicional a la proporcionada por el SVC, por ejemplo, con otros algoritmos que identifique otras estructuras en la imagen aparte de las de interés para la tarea, podemos generar características adicionales relacionadas con el contexto del objeto, como distancias a distintas estructuras, características relativas, etc. Finalmente, también se pueden añadir otras características relacionadas con el contexto de la imagen, como el tipo de escáner en el que se tomó la imagen o datos del estudio al que pertenece.

Una vez calculadas todas las características de interés de los objetos de agrupación $O_{R.P}$, se acumula toda la información en un vector de características de múltiples dimensiones denominado "$F_{O_{R.P}}$" para su uso en las siguientes etapas.

3.2.3. Clasificación del error

Teniendo en cuenta la hipótesis de partida planteada en esta tesis de que distintos tipos de error de segmentación tienen distinto origen, en esta tercera etapa se realiza una primera clasificación de los objetos de agrupación atendiendo al tipo de error encontrado, lo que permitirá, posteriormente, analizarlos por separado.

Los objetos de agrupación $O_{R.P}$ se clasifican en función de los objetos que componen la agrupación (sus ancestros) y su disposición. Según Nascimento and Marques [2006] se pueden diferenciar seis situaciones (ver figura 3.7): detección correcta o correspondencia "1 - 1"[1], falsa alarma o correspondencia "0 - 1", fallo de detección o correspondencia "1 - 0", segmentación unida o correspondencia "M - 1", segmentación partida o correspondencia "1 - N", segmentación mixta o correspondencia"M - N".

Los objetos sin contacto forman objetos de agrupación de un único objeto y el origen de su ancestro marca su tipología. El tipo "1 - 0", también denominado "Fallo en la detección" u objeto "Miss", corresponde a un objeto de la segmentación de referencia no detectado. El tipo "0 - 1", "Falsa alarma" u objeto "Extra", corresponde a una sobresegmentación, es decir, un objeto de la segmentación propuesta que no tiene correspondencia con ningún objeto de la referencia. Los objetos con contacto forman objetos de agrupación de, al menos, dos objetos y también se denominan "Detección correcta" u objeto "Detected". El caso ideal es el objeto de agrupación de tipo "1 - 1" con una gran proporción de área solapada entre la segmentación propuesta y la referencia, pero no siempre sucede. En el caso extremo, los objetos tienen contacto pero no solape (en la figura 3.5, etiquetado como caso D, se muestra un ejemplo de esta situación). Existen otras clasificaciones de los objetos de agrupación en función del número de sus objetos ancestros: los de tipo "1 - N",

[1]Esta notación, "#(O_R) - #(O_P)" indica el número de objetos de la referencia y de la propuesta del SVC que componen el objeto de agrupación.

Tipo	**Nombre**	**Descripción**
Identificación	Id_scaner	Identificación del instrumento de adquisición
	Id_estudio	Identificación de estudio que describe las segmentaciones comparadas
	Id_caso	Identificación anonimizada del paciente
	Id_RuP_2D	Identificación del objeto de agrupación bidimensional
	Id_RuP_3D	Identificación del objeto de agrupación tridimensional
Configuración	Th_P	Umbral utilizado cuando la segmentación de la propuesta no es binaria
	Id_Config_P	Identificación del fichero de la segmentación de la propuesta
	Id_Config_R	Identificación del fichero de la segmentación de la referencia
Composición	Num_obj_R	Número de objetos 2D segmentados en la referencia
	Num_obj_P	Número de objetos 2D segmentados en la propuesta
Posición	XYZ_centroide	Posición del centroide del objeto de agrupación en coordenadas mundo
	XYZ_1er_voxel	Posición del primer vóxel del objeto de agrupación en coordenadas mundo
	IJK_1er_voxel	Posición del primer vóxel del objeto de agrupación en coordenadas imagen
Tamaño	Volumen	Volumen del objeto de agrupación 2D tanto en vóxeles como en mm3
	TP	Volumen solapado entre la propuesta y la referencia
	FP	Volumen de la propuesta no solapado con la referencia
	FN	Volumen de la referencia no solapado con la propuesta
Forma	Eje mayor	Longitud (en píxeles) del eje mayor de la elipse que tiene los mismos segundos momentos centrales normalizados que la región, devuelto como un escalar
	Eje menor	Igual que el anterior pero del eje menor de la elipse
	Orientación	Ángulo entre el eje x y el eje mayor de la elipse que tiene los mismos segundos momentos que la región, devuelto como un escalar [-90º , 90º]
Color	Intensidad máxima	Valor del píxel con mayor intensidad en la región, devuelto como escalar
	Intensidad media	Media de todos los valores de intensidad en la región, devuelta como un escalar
	Intensidad mínima	Valor del píxel con mayor intensidad en la región, devuelto como escalar
Distancia	Distancia	Mide cuan lejos están uno de otro dos subconjuntos compactos de un espacio métrico ($d_H$modificado, percentil 95).
	Distancia euclidea a otra estructura	Mide la distancia de un objeto a otro elemento de interes (distancia entre centroides en píxeles).
Similitud espacial	DSC	Coeficiente de similitud Dice: 2*TP/(2*TP+FN+FP)
	TPR	Tasa de Verdaderos Positivo («True Positive Rate»): TP/R
	OSR	Tasa de SobreSegmentación («Over Segmentation Rate»): FP/R
	USR	Tasa de Infrasegmentación («Under Segmentation Rate»): FN/R
Diferencias	AVD	Diferencia ponderada de volumen entre la propuesta y la referencia: abs (P-R)/R
Clasificación	AgrupType	Descripción del tipo de agrupación {«Extra», «Miss», «Detectedt»}
	SubAgrupType	Descripción avanzada de agrupación compuestas {«Single», «Multiple», «Split», «Merge»}

Tabla 3.1: Descripción de características básicas de $O_{R.P}$ en AMOSE^2

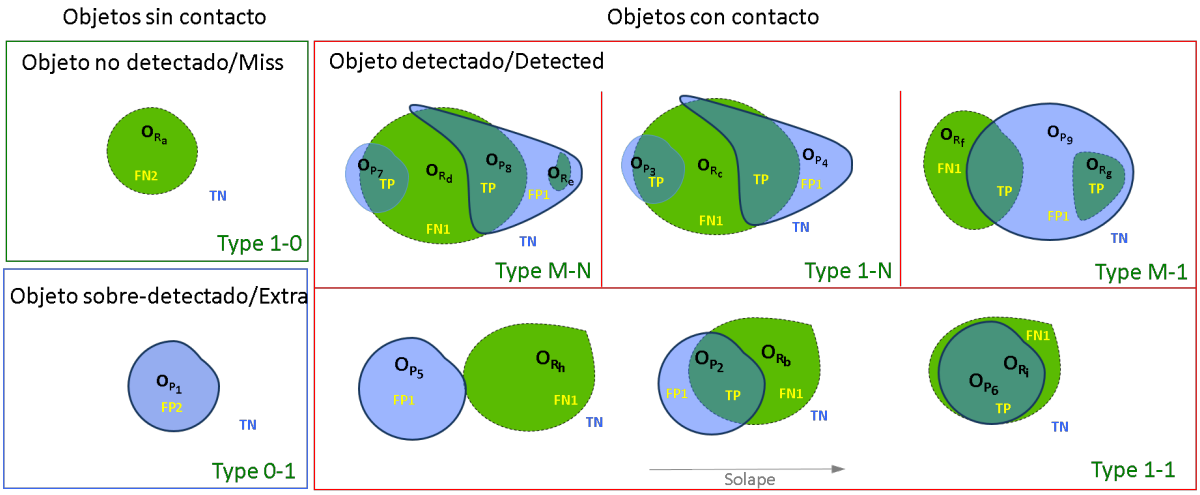


Figura 3.7: Tipos de objetos de agrupación $O_{R.P}$

cuando un objeto de la referencia conecta varios objetos de la propuesta (la segmentación está partida); los de tipo "M - 1", cuando varios objetos de la referencia se conectan con uno de la propuesta (la segmentación une varios objetos); y los de tipo "M - N", cuando varios objetos de la referencia se conectan con varios objetos de la propuesta (la segmentación es poco precisa).

Nosotros proponemos una clasificación algo diferente, donde se utiliza, además de la disposición de los objetos, la proporción de solape entre ellos. Esto nos proporciona más flexibilidad a la hora de considerar qué segmentaciones son correctas en función de la fase del diseño en la que nos encontremos. Debemos tener en cuenta que los recursos son limitados y tienen un coste (número de ingenieros implicados en el diseño, número de desarrolladores, coste de anotación del dataset de referencia, etc.), por lo que, en cada iteración del ciclo de diseño, se deben focalizar los recursos en solucionar los problemas más graves. Por ejemplo, en las primeras iteraciones del diseño, no tiene sentido resolver problemas de delineación de los contornos (problema relacionado con el solape entre los blobs de la segmentación propuesta y la de referencia) cuando no se detectan gran cantidad de objetos o se segmentan demasiados objetos erróneos (problema relacionado con los objetos miss y/o extra). Sin embargo, en las últimas iteraciones, buscaremos más precisión a la hora de segmentar los objetos de interés, fijándonos más en el solape entre segmentaciones.

Por ello, proponemos flexibilizar la definición de "segmentación correcta" introduciendo un umbral, "$Th_{success}$", en el grado de solape entre los objetos de la segmentación propuesta y de la referencia que participan en un objeto de agrupación. Por ejemplo, se pueden considerar diferentes opciones con distinto grado de exigencia en el solape para considerar correcta una segmentación: 1) que la segmentación propuesta sea correcta sólo con tener

contacto con un objeto de la referencia, 2) exigir al menos un píxel/vóxel de solape o 3) exigir un solape mayor (utilizando un criterio más complejo como por ejemplo el DSC). También se pueden utilizar criterios todavía más específicos al analizar el solape, por ejemplo, la granularidad de los objetos erróneos (su composición en número de objetos), la distancia entre subregiones de error, o la profundidad de la zona de error. Por ejemplo, se pueden dar casos con un porcentaje alto de similitud espacial, es decir, un DSC alto, pero con una granularidad elevada, lo que indicaría problemas en la delineación de los contornos.

Un caso típico, en el que los criterios para clasificar las distintas agrupaciones son más exigentes conforme se avance en el diseño, podría ser el siguiente: Un error que inicialmente es pequeño, se desprecia en una fase inicial del desarrollo, pero se convierte en objetivo en una fase posterior. Según la literatura científica Zijdenbos et al. [1994], Zou et al. [2004], en imagen médica y objetos con contornos con transiciones suaves (poco abruptos), se puede considerar una segmentación correcta cuando el coeficiente DSC es mayor de 0.7. Por tanto, un ejemplo de criterio poco estricto sería clasificar como objeto de agrupación correcto a aquel que simplemente presenta solape entre el objeto de segmentación propuesto y el de la referencia. Mientras que un criterio más estricto sería considerar objeto de agrupación de éxito a aquel que supere un cierto umbral de solape, por ejemplo, con un DSC mayor del 90 % y cuya composición sea de tipo 1-1. De esta forma, se reducen los objetos de error (o se subdividen) en los primeros ciclos de desarrollo para focalizar en los más graves y facilitar la explicación de los patrones de error.

En la figura 3.8 se muestra la descomposición jerárquica de los tipos de error que presentan los objetos de agrupación para diferentes criterios. En la primera división, se utiliza una función "f_1" para relacionar las sub-regiones de error (TP, FN , FP y $\#O_R$-$\#O_P$) y separar las agrupaciones en detectadas "Detected" y erroneas "Error" (las no detectadas). Una vez que la agrupación se clasifica como errónea, en la segunda división, por un lado se subdivide los objetos de error en dos sub-tipos, "Miss" y "Extra", en función del origen de los objetos agrupados "$f_2(\#O_R, \#O_P)$". Por otro lado, los objetos detectados se clasifican como exitosos "Success" o imperfectos "Imperfect" en función de si su grado de solape supera cierto umbral $Th_{Success}$. En una tercera division, los objetos imperfectos pueden subdividirse empleando otros criterios, ya sea una relación de tamaño, de forma, segun el porcentaje de solape espacial, granularidad, etc., o una combinación de éstos ("f_3"). Todas estas particiones diferencian distintos tipos de objetos de agrupación, lo que simplifica el reconocimiento de patrones en etapas posteriores.

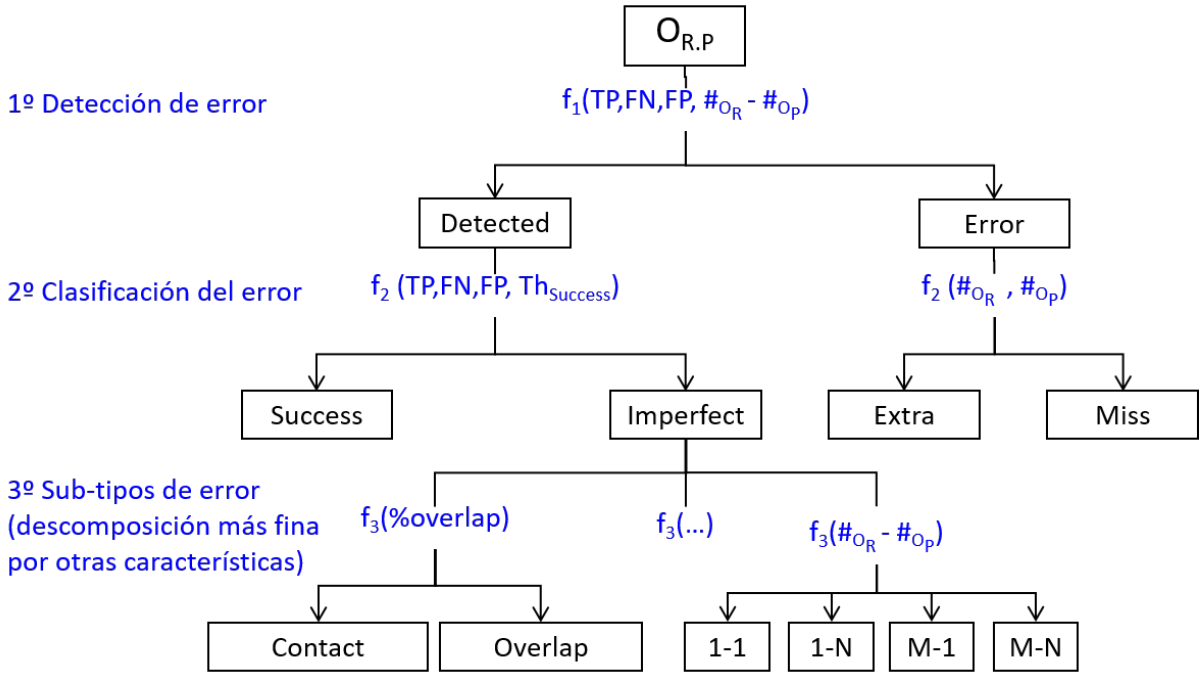


Figura 3.8: Descomposición jerárquica de tipos de error en objetos de agrupación

3.3. El módulo de caracterización del error

El Módulo de Caracterización de Error (MCE) tiene como objetivo el análisis de los vectores de características multidimensionales que describen los objetos de agrupación de error para extraer nuevo conocimiento que ayude a los expertos en el refinamiento del SVC. Se buscarán descripciones sencillas y de fácil interpretación, de forma que se facilite su exploración e interpretación por parte de los expertos. El análisis se realiza de forma independiente por cada tipo de error, pues se supone que su origen y causa son distintos.

El MCE permite realizar tres tipos de análisis a la hora de caracterizar comportamientos erróneos, como se muestra en la figura 3.9:

- Análisis de errores agrupados: el objetivo del análisis es detectar y caracterizar bolsas de error (K_B*), esto es, descubrir patrones que describan gran cantidad de instancias de objetos de agrupación de error.

- Análisis de errores aislados: el objetivo es identificar casos anómalos, errores puntuales con características que se salen de lo normal. (K_A*).

- Análisis de error comparado: el objetivo es comparar dos predicciones respecto a la referencia para detectar similitudes y diferencias entre los errores cometidos por el modelo propuesto respecto a otro alternativo (K_S*).

En las siguientes subsecciones se describen en detalle los distintos tipos de análisis realizados y sus análisis, aunque previamente, para algunos de ellos, será necesario realizar un preprocesado de los conjuntos de datos para reducir su dimensionalidad o transformar el espacio de datos para que los algoritmos de IA utilizados convergan a una solución en un tiempo razonable, como se comenta a continuación.

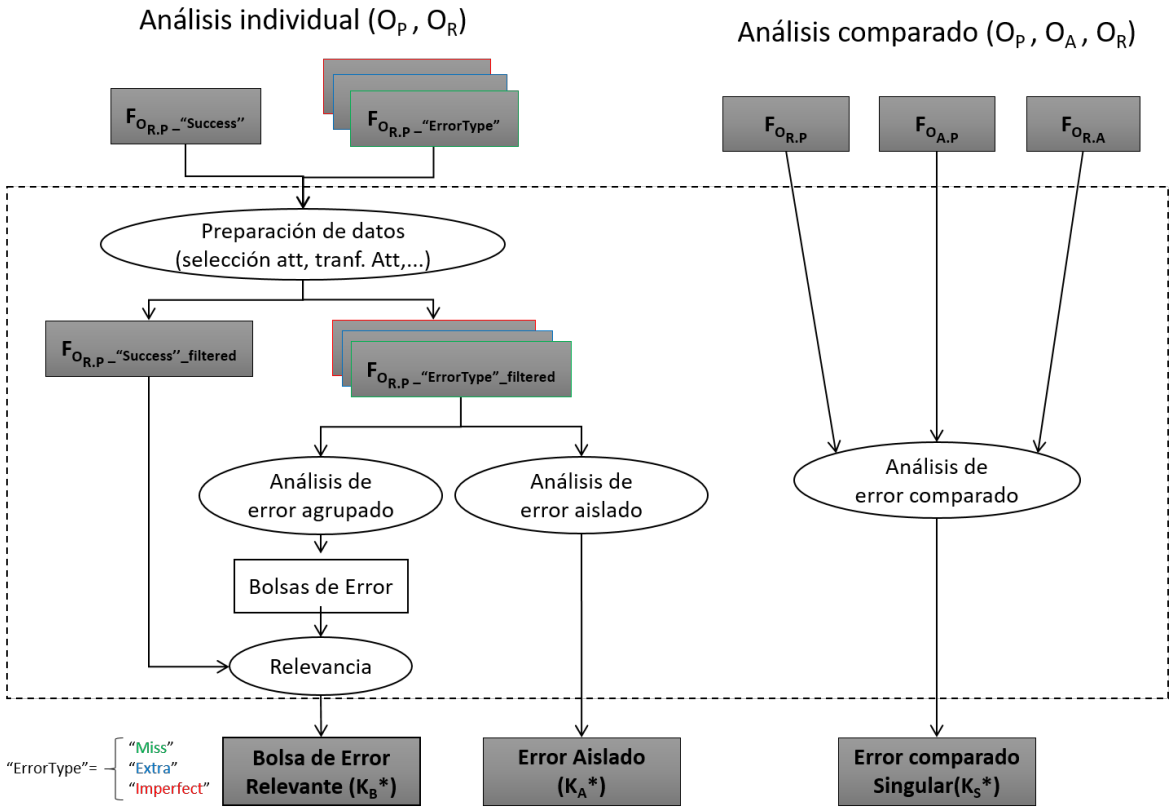


Figura 3.9: Diagrama del módulo de caracterización de error (MCE)

3.3.1. Preprocesado de las características

Antes de aplicar algoritmos de IA para detectar patrones de error (mediante técnicas de clustering) y casos anómalos (mediante técnicas de detección de outliers), es necesario realizar un preprocesado de los datos que facilite su tratamiento posterior y permita obtener resultados óptimos.

Un primer problema a tener en cuenta es la diversidad de tipos de datos que manejará el sistema AMOSE^2 para adecuarlos a los formatos que necesitan los algoritmos. En este trabajo, se han utilizado únicamente transformaciones de normalización y escalado en la preparación de los datos para no mezclar variables, garantizar que las variables sean comparables entre sí y evitar que las diferencias de escala sesguen los resultados. De este modo, nos aseguramos de que no se pierde el significado de las variables.

Por otro lado, asociado al análisis de datos de alta dimensionalidad (cuando hay más de 16 atributos según Berkhin [2006]) tenemos el llamado "problema de la dimensionalidad" [Bellman, 1966], que se refiere a los potenciales efectos negativos derivados del aumento del número de variables respecto a las observaciones. Esto complica el procesado de la información y llega a hacer inviable la aplicación de algunos algoritmos por el coste computacional que supone.

La causa común de estos problemas es que, cuando aumenta la dimensionalidad, el volumen del espacio aumenta exponencialmente, lo que hace que los datos disponibles se dispersen. Este problema afecta a los métodos que se basan en el análisis de distancias entre los datos, como son el clustering o la detección de outliers, que son los problemas que se tratan en el MCE. Para reducir su efecto, se puede aumentar el volumen de datos o aplicar técnicas de reducción de la dimensionalidad. El primer caso resulta difícil en proyectos de medicina y, además, implica un aumento de los costes, por lo que normalmente se opta por la reducción de la dimensionalidad, que es más sencillo, aunque puede conllevar pérdida de información.

Por tanto, en la preparación de los datos, interesa reducir la dimensionalidad para identificar y eliminar variables irrelevantes o que estén fuertemente correlacionadas, de manera que se mejore el rendimiento computacional (menor uso de memoria y de CPU) y se reduzca la complejidad de los modelos, lo que también facilitará su comprensión y mejorará la confianza en los resultados.

Existen múltiples técnicas de reducción de la dimensionalidad, para simplificar o "comprimir" los descriptores de los datos originales. Por un lado, están las técnicas basadas en la transformación de atributos, como el análisis de componentes principales (PCA), DFT, wavelets, etc., cuyo problema principal es que resulta difícil entender el significado de las nuevas variables. Por otro, están las técnicas basadas en la selección de atributos, que permiten eliminar atributos redundantes o irrelevantes [Saeys et al., 2007, Lima, 2019]. Algunas de las técnicas más populares son los métodos de filtrado, los métodos de envoltura o los métodos integrados. En los primeros, algunas técnicas utilizadas son la ganancia de información, la prueba de chi-cuadrado o el coeficiente de correlación de Pearson, etc. En los segundos encontramos la selección hacia delante (Forward Selection), la eliminación hacia atrás (Backward Elimination) o la eliminación recursiva (Recursive Elimination) y, en los terceros, las técnicas de regularización o los árboles de decisión.

Dado que uno de los requisitos de este trabajo es que la explicación del error resulte sencilla, se descartará la trasformación de atributos y sólo se utilizarán técnicas de selección de variables. En la implementación del sistema AMOSE[2] se han utilizado diferentes estrategias para reducir el número de variables. Desde la más simple, que es la selección manual, donde se filtran las variables según las preferencias de los expertos;

pasando por la selección computacional, donde el filtrado se realiza mediante métricas que miden la relación entre las variables, como la similitud semántica o la ganancia de información; o una mezcla de ambas. Por el momento, en el sistema AMOSE^2, estas métricas han servido para proporcionar distintos criterios de selección, pero han sido los expertos los que han establecido el orden de importancia de las características y han tomado la decisión final de qué variables utilizar en cada momento.

3.3.2. Análisis de errores agrupados

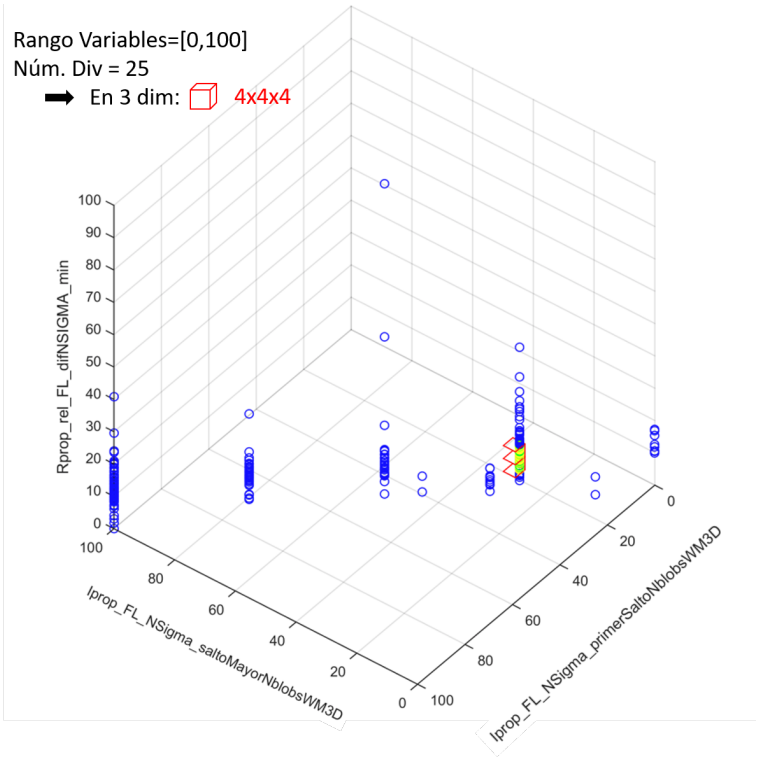
El objetivo del análisis de error agrupado es descubrir bolsas de error, esto es, descubrir patrones de error que representen a gran cantidad de objetos de agrupación erróneos. Para su detección, utilizaremos técnicas de clustering sobre las características de los diferentes subconjuntos de tipos de error "$F_{O_{R.P\_'ErrorType'}}$".

Existen múltiples algoritmos de clustering Guojun et al. [2007], Aggarwal and Reddy [2013], basados en densidad, rejillas, adyacencias, redes neuronales, etc. La elección de uno u de otro es una tarea compleja, ya que depende del problema y de la aplicación final. Para el caso de conjuntos de datos de múltiples dimensiones, una buena solución son los algoritmos de clustering subespacial, como CLIQUE Agrawal et al. [1998], SUBCLU Kailing et al. [2004], CLTREE Liu et al. [2000] o MAFIA Nagesh et al. [2001], y los de clustering por proyección, como P3C Moise et al. [2006] o PreDeCon Bohm et al. [2004]. Todos estos algoritmos analizan los datos utilizando un subconjunto de múltiples dimensiones y se basan en el análisis de densidad para detectar grupos de comportamiento similar. De los citados, se ha seleccionado para su implementación en esta metodología de caracterización el algoritmo CLIQUE Agrawal et al. [1998], pues la representación de los clusters en forma de hipercubos de las variables relevantes resulta muy sencilla de interpretar.

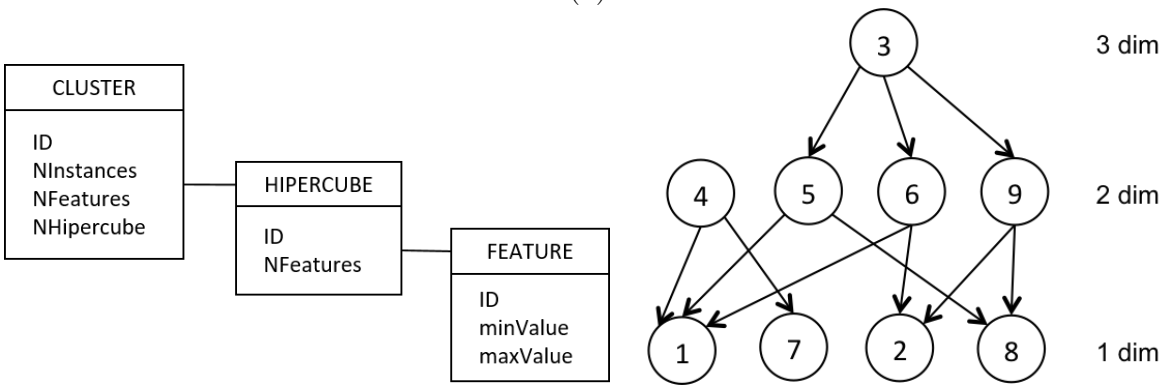
3.3.2.1. El algoritmo CLIQUE

CLIQUE es un algoritmo de clustering que permite identificar zonas densas en subespacios de máxima dimensionalidad. La figura 3.10 muestra la idea del algoritmo en un espacio tridimensional para facilitar la explicación. Inicialmente, el espacio de datos se divide en una rejilla fija y equiespaciada paralela a los ejes, resultando un número ξ de intervalos en cada dimensión (figura 3.10.a). Sólo aquellas unidades que contienen al menos τ puntos se consideran densas. Un cluster se define como el máximo conjunto de unidades densas adyacentes, las cuales se describen mediante los valores máximo y mínimo de la división en cada característica (figura 3.10.b).

CLIQUE sigue una estrategia "bottom-up" y encuentra múltiples descripciones de los mismos datos en múltiples subespacios de distinto número de dimensiones. Estas descripciones se pueden relacionar mediante una estructura jerárquica (figura 3.10.c). Los cluster de mayor dimensionalidad (cluster padre), serán los cluster de interés, los cuales se utilizarán para describir los patrones de comportamiento erróneo. En la imagen de ejemplo serían los cluster identificados con el valor 3, de dos dimensiones, y con el valor 4, de 3 dimensiones.



(a)



(b) (c)

Figura 3.10: Ejemplo de cluster de hipercubos de 3 dimensiones mediante algoritmo CLIQUE

Una vez detectados los cluster que describen las bolsas de error, es necesario valorar su relevancia para presentarlos a los expertos en orden de importancia. Los patrones de

error se ordenan utilizando medidas globales, como el número de atributos que participan en el patrón, el número de agrupaciones de error que describe (cobertura), etc., o medidas relativas, como la importancia o la densidad del cluster. En la tabla 3.2 se muestra la definición de algunas funciones que se pueden utilizar para ordenar los patrones de error.

Descripción			Función
Global	Número de instancias		count()
	Número de atributos		count()
Relativa	Importancia	Por característica	$I_A = \frac{Natt_{cluster}}{Natt_{total}}$
		Por región	$I_R = \frac{1}{Nregion}$
		Por valor	$I_V = sum(\frac{Range_i}{Range_{total}})$
		Por tamaño	$I_S = \frac{Ninstances_{cluster}}{Ninstances_{total}}$
	Densidad	Clásica	$D_C = \frac{Ninstances_{cluster}}{Size_{cluster}}$
		Difusa (1er nivel)	$D_{F1} = \frac{Ninstances_{cluster\ 1erChild}}{Size_{cluster\ 1erChild}}$
	Calidad	Relativa	$C_R = \frac{Ninstances_{cluster}}{Ninstances_{success}}$
		Diferencial	$C_D = Ninstances_{cluster} - Ninstances_{success}$

Tabla 3.2: Métricas de relevancia de cluster

Cuando veamos el caso de uso en el capítulo 4, utilizaremos la tasa error-acierto $C_R = \frac{Ninstances_{cluster}}{Ninstances_{success}}$, que es una medida relativa de calidad que indica si el cluster de error es diferenciable respecto al conjunto de éxito. Para ello, se calcula el número de objetos que contiene el cluster detectado, tanto para el conjunto de objetos de agrupación de error correspondiente ("Miss", "Extra" o "Imperfect") como para el conjunto de objetos de agrupación de éxito ("Success"), y se calcula su proporción. Aquellos cluster que superen un cierto umbral (valor configurable en la herramienta de AMOSE^2 analysis) se considerarán relevantes y su información descriptiva se mostrará a los expertos para informar sobre el patrón de error.

A la hora de mostrar los cluster de error relevantes, con el fin de simplificar la descripción, es bueno analizar si existen conjuntos de datos muy similares con diferentes descripciones, de forma que se pueda seleccionar aquella descripción más sencilla o aquella que aporte información más útil. También, si se dispone de la descripción semántica de las características de los objetos, se puede utilizar la similitud semántica entre los atributos que describen el cluster para eliminar aquellos atributos redundantes. De esta forma, se reduce la dimensionalidad del cluster, lo que facilitará la explicación o la visualización en futuras fases.

3.3.3. Análisis de errores aislados

Un error aislado o atípico, denominado en inglés outlier, se define como un error que es numéricamente distante del resto de errores. Puede ocasionarse por errores en la adquisición, en el proceso, por acontecimientos extraordinarios, o por otras causas no conocidas. Detectar y describir los errores aislados en segmentación de objetos es una tarea importante por varias razones. Por un lado, para analizar si estos valores se deben a la naturaleza del problema o si se deben a fallos en alguna parte del proceso, ya sea en el procesado del SVC, en la definición de la referencia, o en el proceso de caracterización y exploración del error. Por otro lado, para estudiar su importancia con el fin de saber si son errores tolerables o, por el contrario, son errores inasumibles en los que sería necesario un nuevo ciclo de refinamiento.

Dada la dificultad que existe a la hora de definir qué es un outlier y definir un método concreto que lo detecte Hawkins [1980], en esta tesis se exploran diversos métodos como solución de compromiso, y se seleccionan aquellos que facilitan una explicación sencilla de los errores.

Para el análisis unidimensional existen diferentes métodos. Uno de ellos es el método de desviación estándar, que considera outlier a todos los datos que están fuera del intervalo dado por la definición 3.2

$$(\mu - k \cdot \sigma, \mu + k \cdot \sigma)\ k \geqq 3 \tag{3.2}$$

Otro es el método del rango intercuartil cuyo acrónimo es IQR (Inter Quartile Range), utiliza los valores del primer y tercer cuartil para definir el rango a partir del cual cualquier valor que esté fuera se considera un valor anómalo. Este método también se le conoce como Test de Tukey Tukey [1977]. Los valores de este rango vienen dados por la definición 3.3

$$[Q_1 - k \cdot IQR, Q_3 + k \cdot IQR],\ k \geqq 1{,}5\ IQR = Q_3 - Q_1 \tag{3.3}$$

En este trabajo se ha implementado el método IQR, ya que es un método sencillo que, mediante diagramas de cajas y bigotes, permite identificar rápida y claramente los casos anómalos.

Para el análisis multidimensional de anomalías existen diferentes métodos basados en aprendizaje no supervisado que abordan el problema Schubert et al. [2014], desde modelos basados en la desviación respecto a los vecinos hasta modelos más complejos basados en la matriz de covarianza, como el método robusto MCD ("Minimum Covariance Determinant")[Hubert et al., 2018]. También existen métodos basados en análisis robustos de componentes principales (RPCA) [Candès et al., 2011] y métodos basados en grafos Akoglu et al. [2015]. Dado que no existe ningún método óptimo, se opta por la exploración

de algunos de ellos, los cuales están disponibles en el paquete "OutlierO3" del software R desarrollado por Antony Unwin (Unwin [2019]). En la tabla 3.3 se muestran los métodos del paquete junto a su descripción. De todos ellos, se seleccionan tres métodos: HDoutliers, adjOutlyingness y covMcd y se configura, para cada uno, un valor de tolerancia individual de valores atípicos elegido de forma experimental siguiendo las recomendaciones de su descripción. Estos algoritmos son opacos y generan un listado con las instancias outlier detectadas. Para obtener un listado robusto de instancias outlier, se ha implementado en este trabajo un último paso de selección de los casos configurable según uno de estos dos criterios, el de voto por mayoría o el de intersección de los tres métodos.

Método	Paquete	Autor(es)	Fecha	Descripción
HDoutliers()	HDoutliers	C. Fraley, L. Wilkinson	09/02/2018	Detección de outliers basados en modelos probabilísticos (Wilkinson, 2018)
DectectDeviatingCells()	CellWise	J. Raymaekers, P. Rousseeuw, W. Van den Bossche	01/02/2018	Detección de outliers basado en celdas (Rousseeuw and Van den Bossche, 2016)
adjOutlyingness()	robustbase	Martin Maechler et al.	30/10/2017	Cálculo multivariante del «Outlyingness» con sesgo ajustado
covMcd()	robustbase	Martin Maechler et al.	30/10/2017	Cálculo del estimador MCD («the Minimum Covariance Determinant»), un estimador robusto y multivariante de localización y escala con punto de ruptura alto
mvBACON()	robustX	W. Stahel, M. Maechler et al.	02/02/2017	Versión multivariante que detecta outlier basados en una estimación iterativa a partir de un subconjunto sin outlier (Billor, Hadi and Velleman, 2000)
FastPCS()	FastPCS	K. Vakili	13/08/2014	Cálculo multivariante del «Outlyingness index» de forma rápida y robusta (Vakili and Schmitt, 2014)

Tabla 3.3: Métodos del paquete "OutlierO3"

3.3.4. Análisis de error comparado

El análisis de error comparado tiene como objetivo confrontar, a nivel de objetos de agrupación, dos propuestas de segmentación, que llamaremos segmentación propuesta (P) y segmentación alternativa (A), respecto a una referencia (R) para conocer sus semejanzas y diferencias en cuanto a los errores que comenten. Este análisis es útil para conocer el comportamiento de distintas configuraciones del mismo sistema o de éste respecto a otro sistema previo o alternativo, posiblemente más avanzado, que se basa en otras premisas. Desde el punto de vista del sistema propuesto, de la comparación pueden resultar tanto líneas de mejora fiables, en las que el sistema alternativo consigue mejores resultados y, por tanto, ya se sabe que ahí se puede mejorar, como líneas en las que va a ser complicado mejorar, pues el sistema alternativo también comente ese mismo tipo de errores.

Para comparar tres segmentaciones, se puede definir un método basado en visión por computador que combine la información de las tres segmentaciones en un único objeto de agrupación, pero el proceso de análisis es lento y tedioso. Otra opción es generar y describir las agrupaciones por composición de tres parejas de segmentaciones (P-R, A-R y P-A), y utilizar la etapa de generación de agrupaciones del módulo MDE.

Analicemos este segundo caso. En primer lugar, se identifican y caracterizan por pares los objetos de agrupación producidos por las tres soluciones (segmentación P respecto a la segmentación R, segmentación A respecto a la segmentación R y segmentación P respecto a segmentación A). Dado que estos conjuntos de datos son independientes y los objetos de agrupación se describen como tablas, utilizaremos operaciones de álgebra relacional, como la operación de "reunión", para enlazar los objetos.

La relación entre objetos de agrupación se realiza a partir de los identificadores de sus objetos ancestros, lo que permite crear una nueva tabla de información denominada "LOJ" (Left Outer Join). Su definición matemática se presenta en la ecuación 3.4 y consiste en dos reuniones externas por la izquierda (operación típica en bases de datos relacionales y que se denota por el símbolo $\rtimes$). Nótese que, dependiendo del orden de la reunión en la LOJ, se detectarán diferentes comportamientos enlazados de error. De forma genérica, en la ecuación se indica que, en primer lugar, se realiza una reunión externa por la izquierda utilizando como unión el identificador del objeto ancestro O_{S_2} entre $O_{S_1.S_2}$ y $O_{S_3.S_2}$ y, a continuación, otra reunión externa por la izquierda utilizando el identificador del objeto ancestro O_{S_1} entre la salida anterior y $O_{S_3.S_1}$.

$$LOJ_{S_1.S_2->S_3} = ((O_{S_1.S_2} \rtimes_{Id_{S_2}} O_{S_3.S_2}) \rtimes_{Id_{S1}} O_{S_3.S_1}) \tag{3.4}$$

Para el caso de las tres segmentaciones (propuesta, alternativa y referencia), se muestran en la figura 3.11 dos comparaciones enlazadas que nos aportan información del comportamiento de los objetos segmentados. En $LOJ_{R.P\ ->A}$, primero se combina la información de la agrupación R.P con A.P a través del identificador de los objetos de P (id_P), y después con R.A a través del identificador de los objetos de R (id_R). En $LOJ_{R.A\ ->P}$ se relacionan las tres agrupaciones en un orden diferente, comenzando desde la agrupación R.A con el identificador de A (id_A) y, a continuación, con el identificador de R (id_R).

La información múltiple almacenada mediante una LOJ permite descubrir la configuración de las tres segmentaciones comparadas y, en base a ello, distintos comportamientos o tipos de error. Por ejemplo, es posible estudiar distintos tipos de agrupación, distinguir combinaciones por tamaño de los objetos de agrupación, de su composición, de su porcentaje de solape, etc.

A continuación, vamos a realizar el análisis de las configuraciones posibles para asociarlas con los distintos tipos de error que se pueden presentar. En primer lugar, las reuniones LOJ trabajan con agrupaciones de objetos pertenecientes a pares de segmentaciones. Como se vio en el apartado 3.2.3, se pueden dar cuatro situaciones en función de su composición y su contacto espacial: 1) que se conecten los objetos (Detected), 2) que no se conecten y el objeto proceda de la segmentación S1 (Miss), 3) que no se

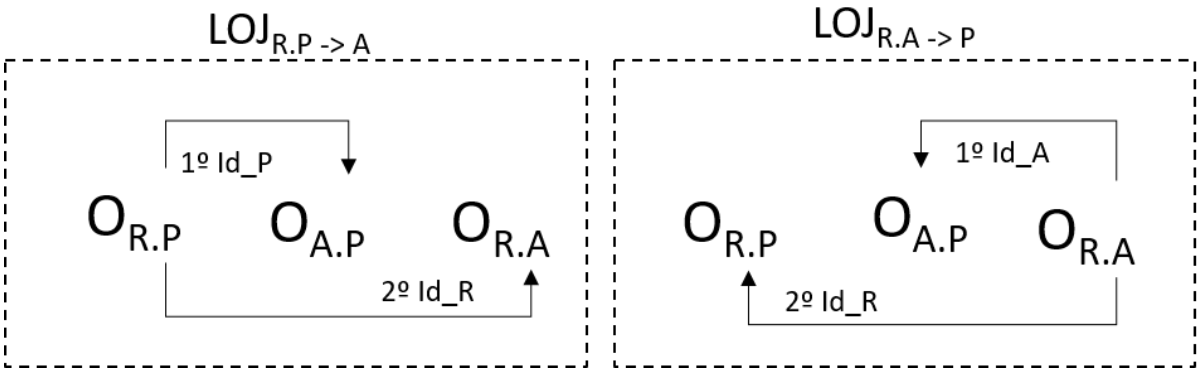


Figura 3.11: Relación de identificadores de clave para realizar las operaciones de reunión por la izquierda de las comparaciones enlazadas (LOJ$*$)

conecten y el objeto provenga de la segmentación S2 (Extra), y 4) que no existan objetos agrupados (NaN). En la tabla 3.4 se muestra dicha explicación, donde se indica el tipo de agrupación según la identificación de sus objetos ancestros. Por ejemplo, si una agrupación es de tipo "Miss", existe un objeto de la segmentación S1 (identificador "s1") que no se conecta con nada ("-").

Tipo de agrupación	Id_S1.Id_S2
Detected	s1.s2
Miss	s1.-
Extra	-.s2
NaN	-.-

Tabla 3.4: Tipos de objetos de agrupación según la identificación de los objetos ancestros

La tabla 3.5 detalla las diez situaciones posibles que se pueden presentar a la hora de analizar los objetos de agrupación resultantes de combinar los objetos de las tres segmentaciones que se desea comparar. Estas situaciones son función de la presencia o ausencia de los objetos de cada segmentación y de si tienen contacto o no entre dichos objetos. La columna "Tipo" contiene la etiqueta con la que denominaremos a cada configuración (Cx). La columna "PAR" indica la presencia (1) o ausencia (0) de objetos de la segmentación P, A y R, respectivamente. Siguiendo la notación descrita en la tabla 3.4, las columnas "$O_{R.P}$, $O_{A.P}$ y $O_{R.A}$ "contienen la descripción de la agrupación resultante, la cual depende de los objetos existentes (de la columna PAR). La columna "Descripción" describe cada combinación y en la columna "Tipos de error de detección" se clasifican los comportamientos.

Por otro lado, en la tabla 3.6, se indica cómo reconocer cada configuración a partir de la situación descrita por las agrupaciones $O_{R.P}$, $O_{A.P}$ y $O_{R.A}$, la cual se obtiene mediante las tablas LOJ. En estas agrupaciones por pares se realiza la comparación respecto a la

referencia. Así, por ejemplo, se producirá un error de tipo Miss cuando no haya objeto que contacte con el objeto R, o error de tipo Extra cuando un objeto de P o A no se conecte con un objeto R. Las descripciones proporcionadas por $LOJ_{R.P->A}$ y $LOJ_{R.A->P}$ son equivalentes, por lo que normalmente se utilizará la descripción $LOJ_{R.P->A}$ ya que está más orientada a la segmentación propuesta, P. El uso de la reunión $LOJ_{R.A->P}$ solo es necesario para detectar la configuracion C2, pues esta configuración no genera ninguna tupla en $LOJ_{R.P->A}$.

Tipo	**PAR**	$\mathbf{O_{R.P}}$	$\mathbf{O_{A.P}}$	$\mathbf{O_{R.A}}$	**Descripción**	**Tipo de error en detección**
C1	001	r.-	-.-	r.-	Solo hay un objeto segmentado de la referencia.	Miss P - Miss A
C2	010	-.-	a.-	-.a	Solo hay un objeto segmentado de la alternativa.	Extra A
C3	100	-.p	-.p	-.-	Solo hay un objeto segmentado de la propuesta.	Extra P
C4	011	r.-	a.-	r.a	Hay dos objetos, uno de A y otro de R que se conectan.	Miss P - Detected A
C5	101	r.p	-.p	r.-	Hay dos objetos, uno de P y otro de R que se conectan.	Miss A
C6	110	r.-	a.p	-.a	Hay dos objetos, uno de P y otro de A que se conectan.	Extra P - Extra A
C7	111	r.p	a.p	r.a	Hay tres objetos, uno de cada segmentación y se conectan entre ellos tres.	Detected
C8	111	r.-/-.p	a.p	r.a	Hay tres objetos y conecta el objeto de A con P y con R, pero P no conecta con R.	Extra P - Detected A
C9	111	r.p	a.-/-.p	r.a	Hay tres objetos y conecta el objeto R con A y con P, pero A no conecta con P.	Detected*
C10	111	r.p	a.p	r.-/-.a	Hay tres objetos y conecta el objeto P con R y con A, pero A no conecta con R.	Detected P - Extra A

Tabla 3.5: Situaciones de error posibles al comapar tres segmentaciones (P, A y R) en función de las agrupaciones por pares resultantes (R.P, A.P y R.A)

Desde el punto de vista del refinamiento de la segmentación propuesta, unas configuraciones de error son más interesantes que otras. Así, por ejemplo, cuando los objetos de la referencia no son detectados ni por la propuesta ni por la alternativa (configuración C1), estamos ante un tipo de error que va a ser difícil de evitar, pues tanto la segmentación propuesta como la alternativa, lo comenten. En cambio, cuando

Tipo	PAR	$O_{R.P}$	$O_{A.P}$	$O_{R.A}$	$LOJ_{R.P->A}$	$LOJ_{R.A->P}$
C1	001	r.-	-.-	r.-	Miss-NaN-Miss	Miss-NaN-Miss
C2	010	-.-	a.-	-.a	-	NaN-Miss-Extra
C3	100	-.p	-.p	-.-	Extra-Extra-NaN	-
C4	011	r.-	a.-	r.a	Miss-NaN-Detected	Miss-Miss-Detected
C5	101	r.p	-.p	r.-	Detected-Extra-Miss	Detected-NaN-Miss
C6	110	r.-	a.p	-.a	Extra-Detected-NaN	NaN-Detected-Extra
C7	111	r.p	a.p	r.a	Detected-Detected-Detected	Detected-Detected-Detected
C8	111	r.-/-.p	a.p	r.a	Miss-NaN-Detected / Extra-Detected-NaN	Miss-Detected-Detected
C9	111	r.p	a.-/-.p	r.a	Detected-Extra-Detected	Detected-Miss-Detected
C10	111	r.p	a.p	r.-/-.a	Detected-Detected-Miss	Detected-NaN-Miss / NaN-Detected-Extra

Tabla 3.6: Detección con las tablas LOJ de las distintas configuraciones de error posibles en una comparación de tres segmentaciones

los objetos de la referencia no son detectados por la propuesta pero si por la alternativa (configuración C4), estamos ante un tipo de error que debemos intentar solucionar (pues la alternativa ya lo consigue).

Otras configuraciones de interés para mejorar el sistema son C3 (errores de objetos Extra que solo comete la segmentación propuesta P), C6 (error de tipo Extra que comenten tanto P como A, que puede tener dos explicaciones: que los dos algoritmos se hayan equivocado o que no se haya anotado bien la referencia). C8, C9 y C10 son casos a valorar individualmente, pues son casos en los que las segmentaciones P y A aciertan pero no coinciden en ningún vóxel. Por último, las configuraciones C2, C5 y C7 son simplemente informativas, no ayudan a mejorar el sistema.

La implementación actual del sistema AMOSE^2 permite realizar un análisis cuantitativo de la cantidad de casos de cada situación y también un análisis manual de los comportamientos enlazados, filtrar por combinaciones y visualizar los objetos de error sobre las imágenes de entrada.

Ejemplo de análisis comparado. Como complemento a la explicación anterior, se presenta un ejemplo sencillo en la figura 3.12. En la fila superior, se muestran tres objetos segmentados en cada una de las tres segmentaciones que pertenecen a una misma agrupación: Objeto P6 en la segmentación propuesta P (en color verde oscuro), objeto A7 en segmentación alternativa A (en verde claro) y objeto R8 en la segmentación de referencia R (en azul). En el interior del objeto se muestra el valor de su etiqueta y en

el exterior el nombre de su identificador. En la fila inferior de imágenes se muestran los objetos de agrupación generados por pares de segmentaciones: objeto RP1 en la agrupación R.P, dos objetos AP3 y AP5 en la agrupación A.P y un objeto RA8 en la agrupación R.A. Se mantiene el color de los objetos ancestros de cada vóxel en las zonas sin solape y se modifica a color rojo cuando hay solape. Cada objeto de agrupación tiene un identificador de objeto que se indica fuera del elemento y dentro se indica el valor de su etiqueta según sus objetos ancestros $O_{S1.S2}$.

Con los objetos de agrupación se crean las agrupaciones enlazadas. En la figura 3.13, se describe en detalle $LOJ_{R.P\ ->A}$. En primer lugar, se realiza una primera reunión externa por la izquierda entre R.P y A.P con el identificador Id_P. A continuación, con su resultado y R.A, se realiza otra reunión externa por la izquierda con el identificador Id_R. Este ejemplo muestra la situación RP=8.6, AP=0.6/7.0, RA= 8.7, por lo que es de tipo Detected-Extra-Detected (CE9) en $LOJ_{R.P->A}$. No hay solape entre P y A pero ambos tienen contacto con R. Sería un caso a estudiar individualmente porque no es normal.

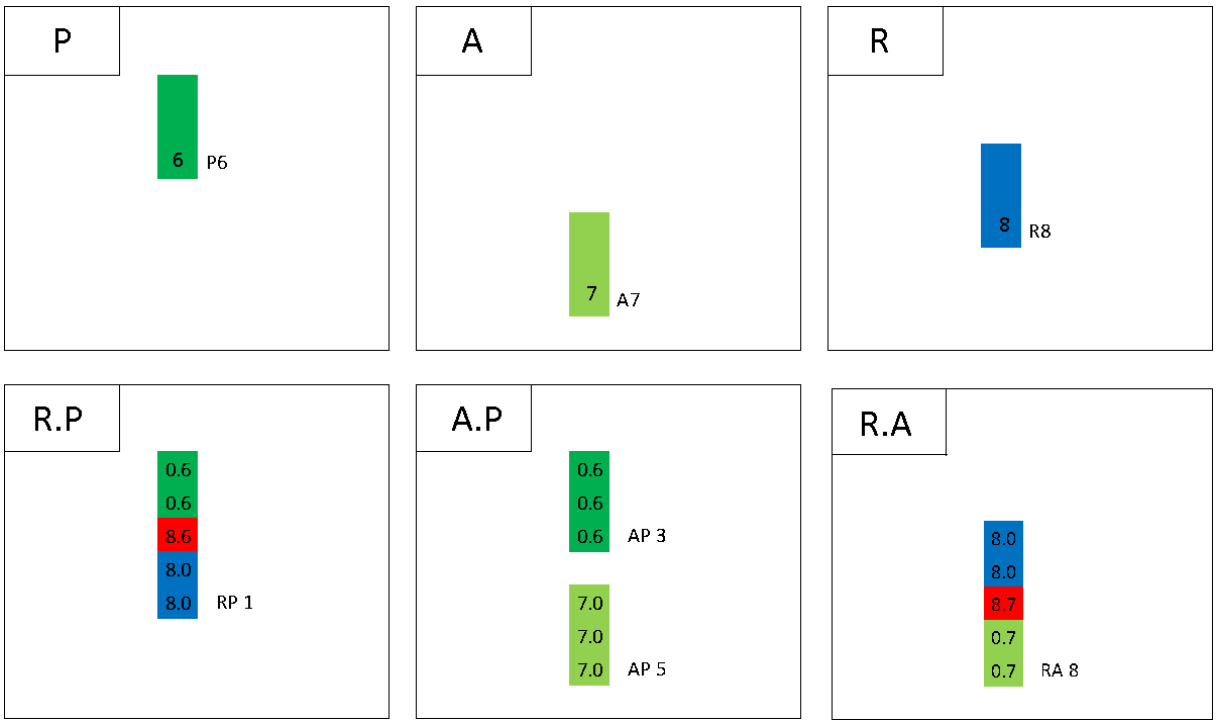


Figura 3.12: Ejemplo sencillo de análisis comparado (tres segmentaciones P, A y R) mediante enlazado de agrupaciones por pares. Objetos ancestros en las segmentaciones P, A y R (arriba) y objetos de agrupación por pares (abajo)

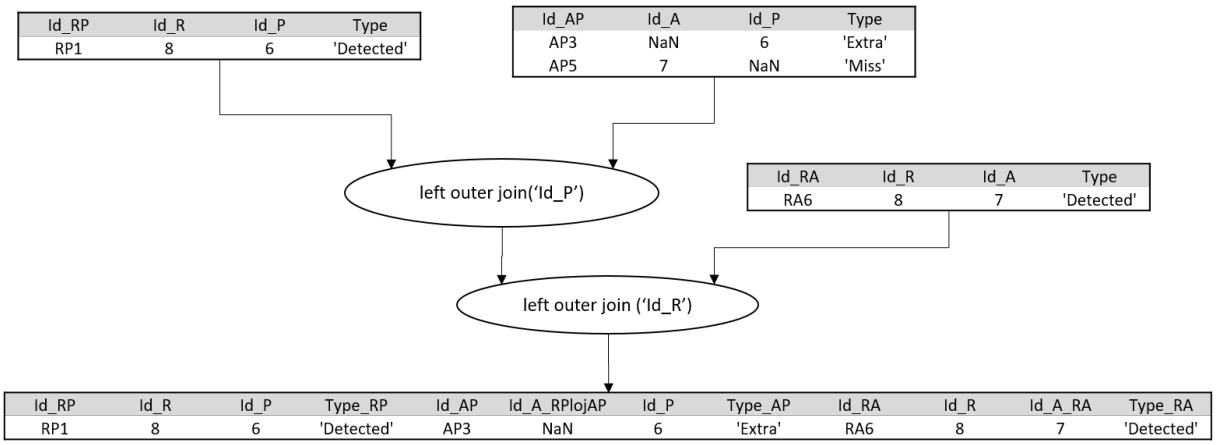


Figura 3.13: Ejemplo de comparación enlazada de tres objetos de segmentación: tablas descriptivas y reunión externa por la izquierda de $LOJ_{R.P\ ->A}$

3.4. El módulo de exploración del error

El "Módulo de Exploración del Error" (MEE) permite explorar diferentes soluciones mediante el uso de una aplicación visual e interactiva. Esta aplicación es necesaria porque las herramientas estáticas analizan los datos y proporcionan informes cuyo formato es demasiado rígido. Estos informes estáticos suelen dar pistas sobre posibles inconsistencias, pero es necesario realizar un análisis posterior, focalizado, para confirmar o desmentir los hallazgos y llegar a conclusiones. Por tanto, estos informes estáticos no son del todo prácticos y generan una barrera a la hora de conocer los hechos relevantes. Dado que el MCE proporciona diferentes comportamientos de error relevantes y pueden existir varias configuraciones para describir un conjunto de objetos de error, esta herramienta dinámica permitirá interpretar y comprender mejor estos nuevos hallazgos.

El MEE gestiona todas las información ofrecida por los módulos anteriores y permite destacar los hallazgos relevantes, proporciona una descripción de los errores y facilita la visualización en su contexto (en este trabajo, en imágenes de resonancia magnética). Como se muestra en la figura 3.14, MEE está dividido en tres vistas, las cuales permiten realizar diferentes tipos de exploración: vista resumen, vista extendida y vista de patrones de error. El MEE se ha implementado como una aplicación web, denominada "AMOSE web report", que está desarrollada en Python™[2], un lenguaje de programación interpretado, de fácil aprendizaje y de código abierto, junto a la librería Streamlit[3], que permite, de una forma rápida y sencilla, visualizar e interaccionar con los resultados del análisis en un navegador web. También se han utilizado otras librerías para la adquisición, procesado y

[2] https://www.python.org/
[3] https://streamlit.io/

representación de los datos, como pandas, numpy, plotly, awesome_streamlit o PIL, entre otras.

El manual de usuario se encuentra en el anexo A.3.4. El tipo de exploración se selecciona en el menú lateral izquierdo: "Overview" para la vista resumen, "Explore" y "Explore 2 sol" para la vista extendida y "Cluster/Isolated Error" para la vista de patrones. En las subsecciones siguientes se detallan las distintas vistas, aunque habrá que esperar al capítulo 4 para mostrar la funcionalidad de este módulo en el contexto de un caso de estudio real.

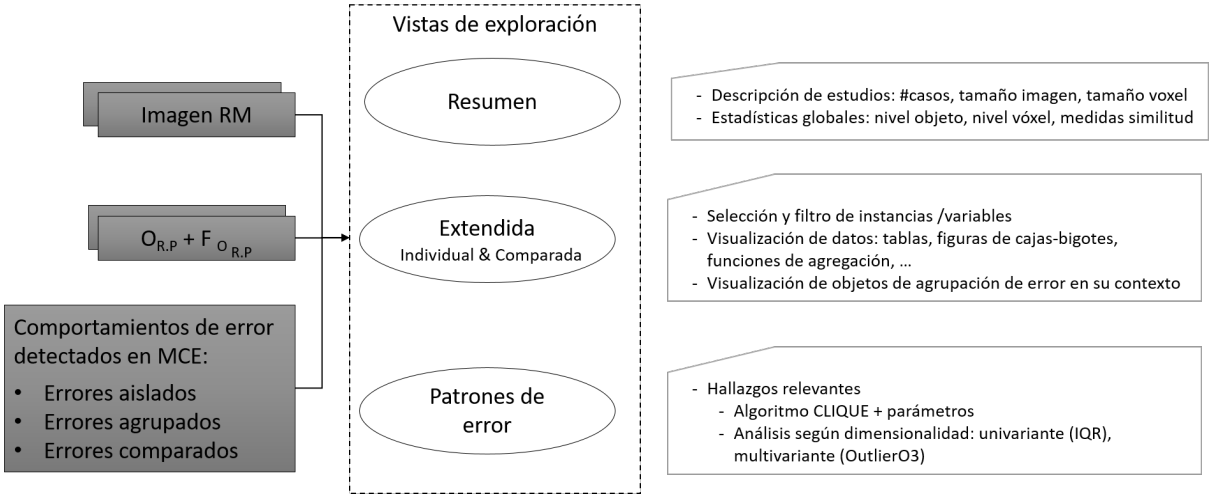


Figura 3.14: Arquitectura del módulo de exploración del error

3.4.1. Vista resumen

En la vista resumen, como se puede ver en la figura 3.15, se selecciona un estudio analizado por el MCE. La vista muestra información general de uno o varios conjuntos de objetos de agrupación $F_{O_{R.P}}$. En primer lugar, se describe el número de instancias y de variables que contiene el estudio y se muestra, de forma tabular, información de las características de los conjuntos de datos analizados, como el número de casos en el estudio, el tamaño de la imagen y el tamaño del vóxel. A continuación, se muestran distintas métricas de evaluación de los errores tanto a nivel de objeto como a nivel de vóxel. La aplicación también tiene opciones avanzadas para visualizar los resultados mediante gráficos, visualizar la información con un nivel de detalle más fino (a nivel de los casos), o permitir ver los datos originales sin procesar ("raw data"), para facilitar a los expertos una visión general del estudio realizado.

Figura 3.15: Vista resumen del MEE

3.4.2. Vista extendida

En la vista extendida se facilita la visualización de las distintas características que describen los objetos de agrupación de error en varios formatos: tablas simples de datos, gráficos avanzados con facilidades de selección y filtrado de datos, o incluso la visualización de objetos en su contexto. Esta vista tiene dos modos de uso: la exploración individual (pestaña "Explore") y la exploración comparada (pestaña "Explore 2 Sol"), que son útiles según el tipo de estudio realizado. Una imagen de su selección en la aplicación "AMOSE web report" se muestra en la figura 3.16.

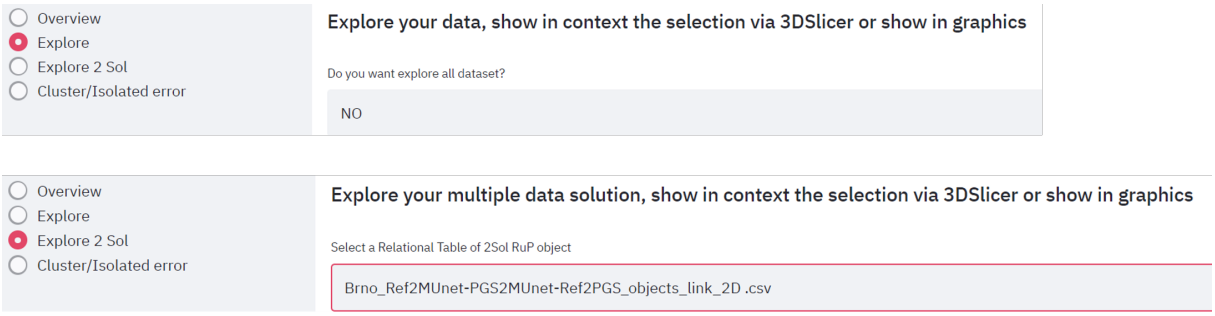


Figura 3.16: Selección de vista personalizable en "AMOSE web report": Individual o Comparada

En la exploración individual, se analiza la propuesta del sistema respecto a la referencia, y la interfaz de la aplicación permite realizar una selección de las variables y las instancias de la caracterización multidimensional de los objetos de agrupación $F_{O_{R.P}}$. De esta forma se puede analizar su comportamiento y extraer patrones de error manualmente, mediante inspección visual y las diferentes facilidades disponibles. Para visualizar los objetos en su contexto, la app web permite seleccionar un listado de objetos y, mediante el programa de visualización de imagen biomédica 3D Slicer[4], un software gratuito y de código abierto, visualizarlos sobre las imágenes originales. Su estructura se puede ver en la figura 3.17.

En el caso de exploración comparada, se analiza el error en la segmentación propuesta también respecto a la referencia, pero además, respecto a otra segmentación, tal como se comentó en la sección 3.3.4. En la figura 3.18 se muestra su interfaz. Un selector permite elegir la comparativa que se desea explorar, es decir, la información de una tabla LOJ$_*$. En esta vista se pueden conocer los diferentes comortamientos de error, su frecuencia en el conjunto de datos comparado y filtrar por los diferentes tipos de error de las agrupaciones comparadas, bien para observar los datos crudos o bien en su contexto.

[4] https://www.slicer.org/

Figura 3.17: Vista extendida: estudio individual

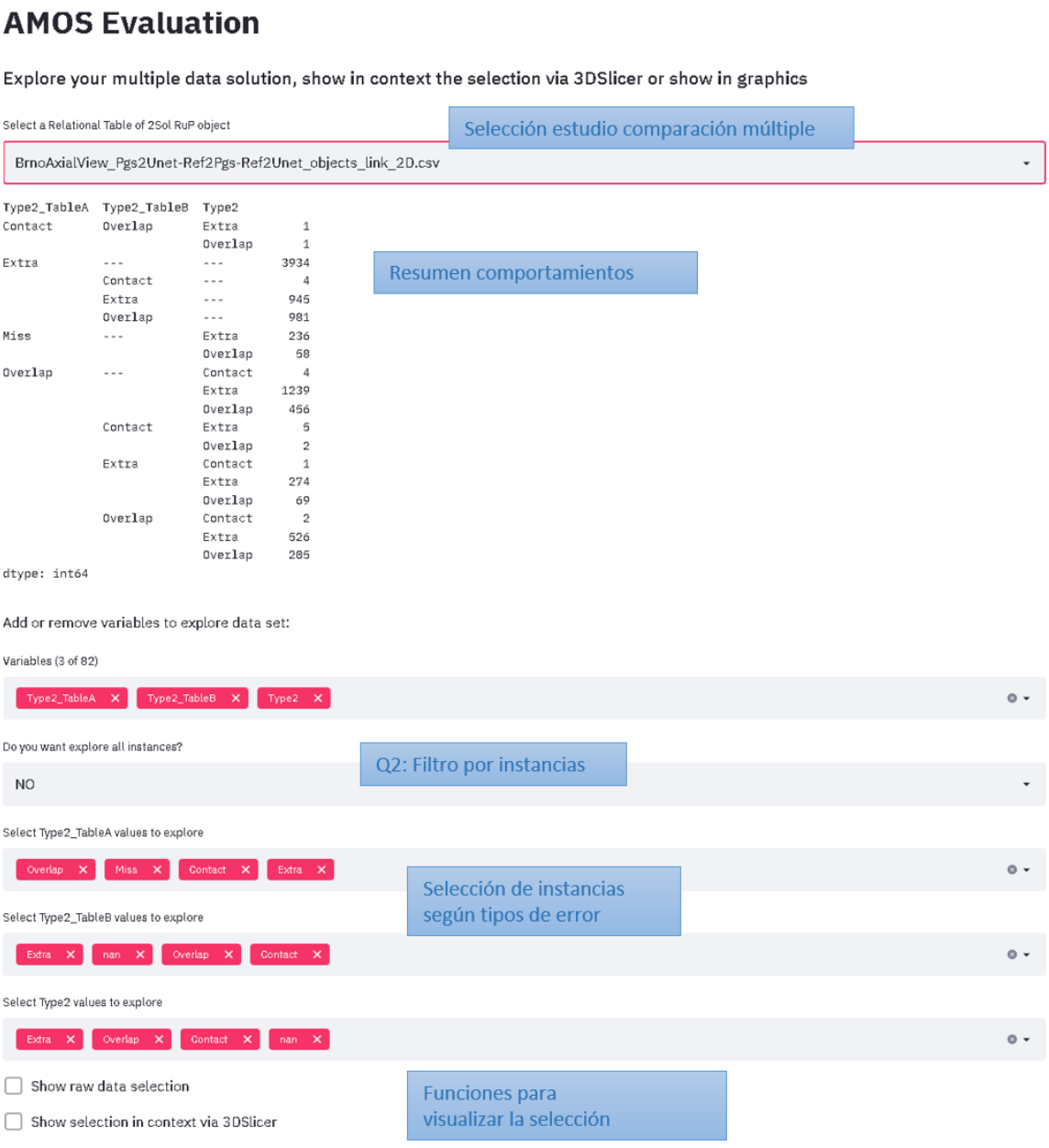


Figura 3.18: Vista extendida: estudio de error comparado

3.4.3. Vista de patrones de error

Por último, la vista de patrones de error permite examinar las características de los hallazgos detectados tras su análisis por el MCE, esto es, visualizar e interactuar con la nueva información descubierta para detectar errores agrupados (bolsas de error / clusters) y errores aislados (anomalías / outliers).

3.4.3.1. Bolsas de error

Para explorar las bolsas de error obtenidas en el MCE se ha implementado en la aplicación "AMOSE web report" la pestaña "Cluster/Isolated Error", opción "Custer" en el tipo de análisis. Esta vista permite seleccionar un directorio de análisis para un cierto dataset y algoritmo, y seleccionar los valores de la configuración que se desea explorar: lista de atributos reducida, umbral de éxito y los parámetros del algoritmo de clustering utilizado. En la figura 3.19 se muestra su apariencia y opciones.

La herramienta actual solo tiene implementado el algoritmo CLIQUE por lo que en la lista desplegable "Algorithm" solo aparece una opción. Este algoritmo tiene dos parámetros, el número de divisiones del espacio de entrada de datos y la densidad del cluster que han de seleccionarse en el desplegable de configuración "No. division- Threshold cluster".

Esta vista tiene dos funcionalidades: (1) observar de forma general todos los cluster detectados, los atributos que los describen y si superan el umbral de relevancia, en "See Maximal Cluster Overview") y (2) observar un cluster relevante concreto para conocer su tamaño, su cobertura o los rangos de las variables que describen el cluster.

La representación visual de las descripciones de los errores relevantes, esto es, de aquellos que superan el umbral ESR (definido en el módulo MCE), se puede realizar en diferentes formatos en función de su dimensionalidad. Las tablas permiten ver en detalle los datos concretos, mientras que las gráficas permiten integrarlos, lo que facilita su interpretación. Por ejemplo, la distribución de los datos se puede visualizar mediante histogramas o utilizando diagramas de cajas y bigotes ("box-plot"). Cuando los patrones de error se describen con pocas variables, se pueden utilizar gráficos simples, como un "scater-plot", para visualizar la relación entre variables. Sin embargo, en el caso de descripciones multidimensionales, se necesitan otro tipo de gráficos más complejos, como los mapas de color o los diagramas de araña, para mostrar múltiples fuentes de información de manera combinada, como el número de atributos relevantes, la cantidad y calidad de las regiones que forman el cluster o su tamaño.

En el anexo A.3.2 hay una explicación de las representaciones gráficas complejas: los mapas de color, los diagramas de araña o los diagramas de cajas y bigotes (box-plot).

También se explican en detalle las distintas funcionalidades y representaciones visuales complejas de las bolsas de error en el capítulo siguiente, con un ejemplo concreto.

3.4.3.2. Errores aislados

Para explorar los errores aislados obtenidos en el MCE se ha implementado en la aplicación "AMOSE web report" la pestaña "Cluster/Isolated Error", donde hay que seleccionar la opción "Outlier" en el tipo de análisis.

Para el análisis unidimensional se ha implementado una vista ("Univariate (IQR)") donde se pueden explorar los objetos de agrupación con comportamiento anómalo (outlier1D) para diferentes valores de K analizados, diferenciando por tipo de objetos de error. En esta vista, en el selector, se muestran las variables donde se ha detectado uno o varios comportamientos "outlier1D" y se puede elegrir entre diferentes representaciones visuales de los subconjuntos de datos (blox-plot, line plot o frecuency plot). También se presenta un listado de objetos de agrupacion "outlier1D" con su identificación, localización y valor outlier para la variable bajo exploración. Un ejemplo de su apariencia se muestra en la figura 3.20.

Para el análisis multidimensional se ha configurado otra vista ("Multivariate (OutlierO3)") donde se exploran los objetos de agrupación con comportamiento anómalo descrito con más de una variable (outlierMD). Similar al método unidimensional, hay que elegir entre las configuración de los métodos utilizados en el MCE y seleccionar el subconjunto de datos por tipo de error. En esta vista, en el selector , se muestran los objetos con comportamiento outlierMD junto a su representación gráfica y descripción textual. La representación gráfica muestra las variables que describen la anomalía y su valor concreto marcado en rojo, mientras que la descripción textual indica información sobre su identificación, localización y combinación de valores que lo definen. Un ejemplo de su apariencia se muestra en la 3.21.

3.5. El módulo de descripción ontológico

Como último módulo, tenemos el módulo de descripción ontológico. Este módulo se ha concebido como un módulo de descripción formal de los conceptos manejados dentro de la metodología AMOSE^2 y para ello, se define una ontología denominada "Visual Object Feature" VOF) para definir la arquitectura y los conceptos necesarios para la descripción semántica de variables que caracterizan un objeto de segmentación y/o un objeto de agrupación.

En las subsecciones siguientes se describen los componentes de la ontología que formalizan la información relacionada con a) los tipos de objetos con los que trabaja

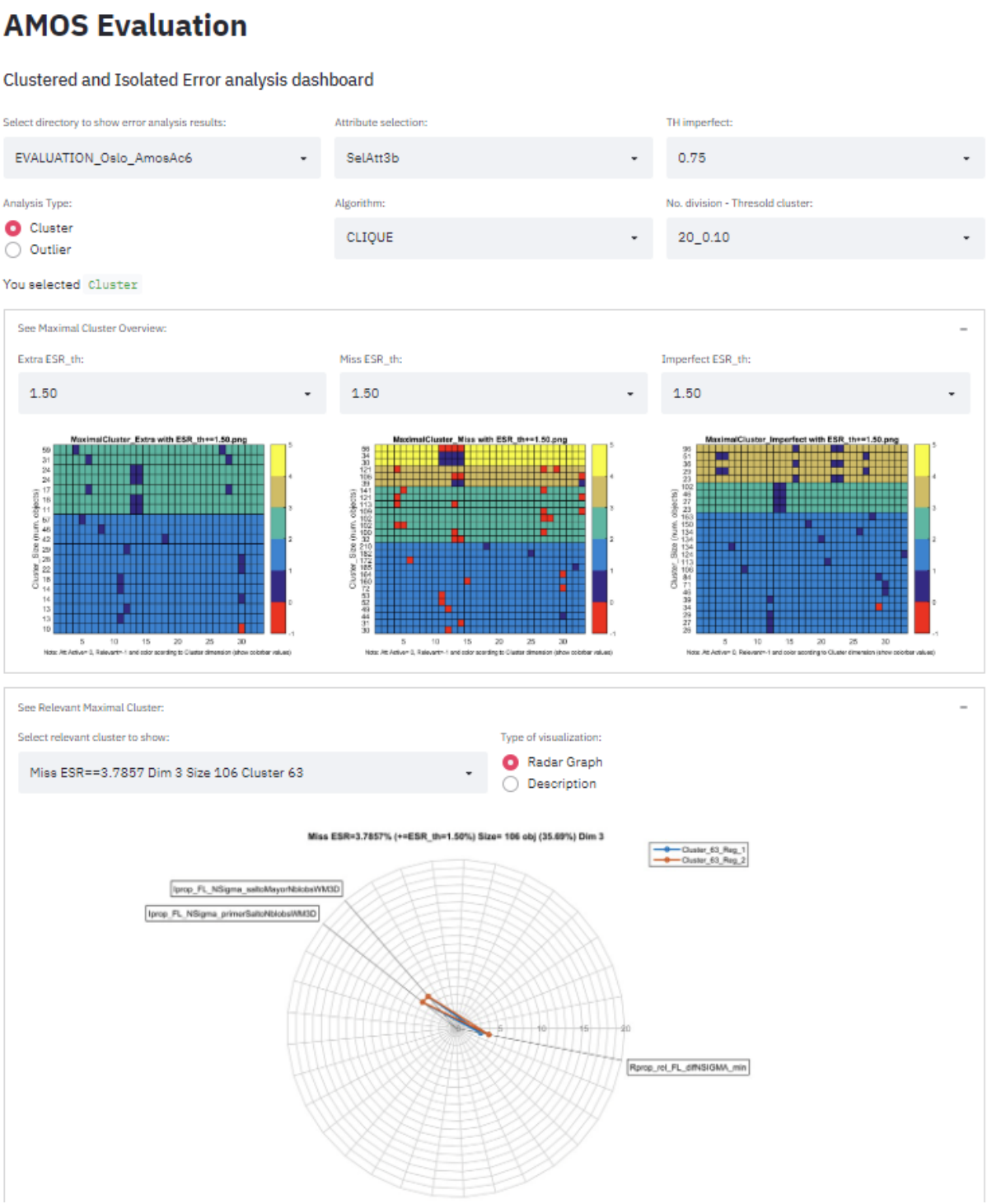


Figura 3.19: Vista de patrones de error para explorar bolsas de error

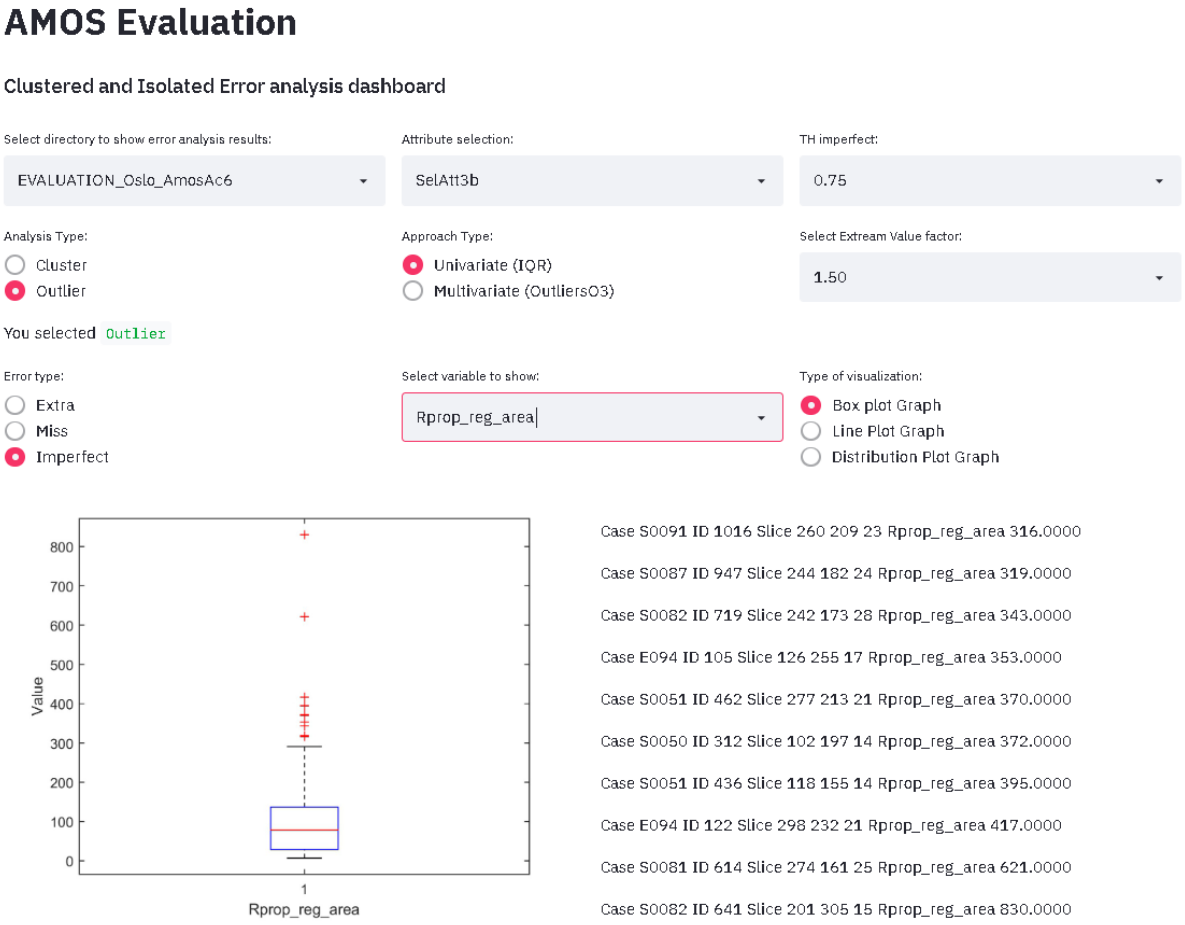


Figura 3.20: Vista de patrones de error para análisis unidimensional de errores aislados

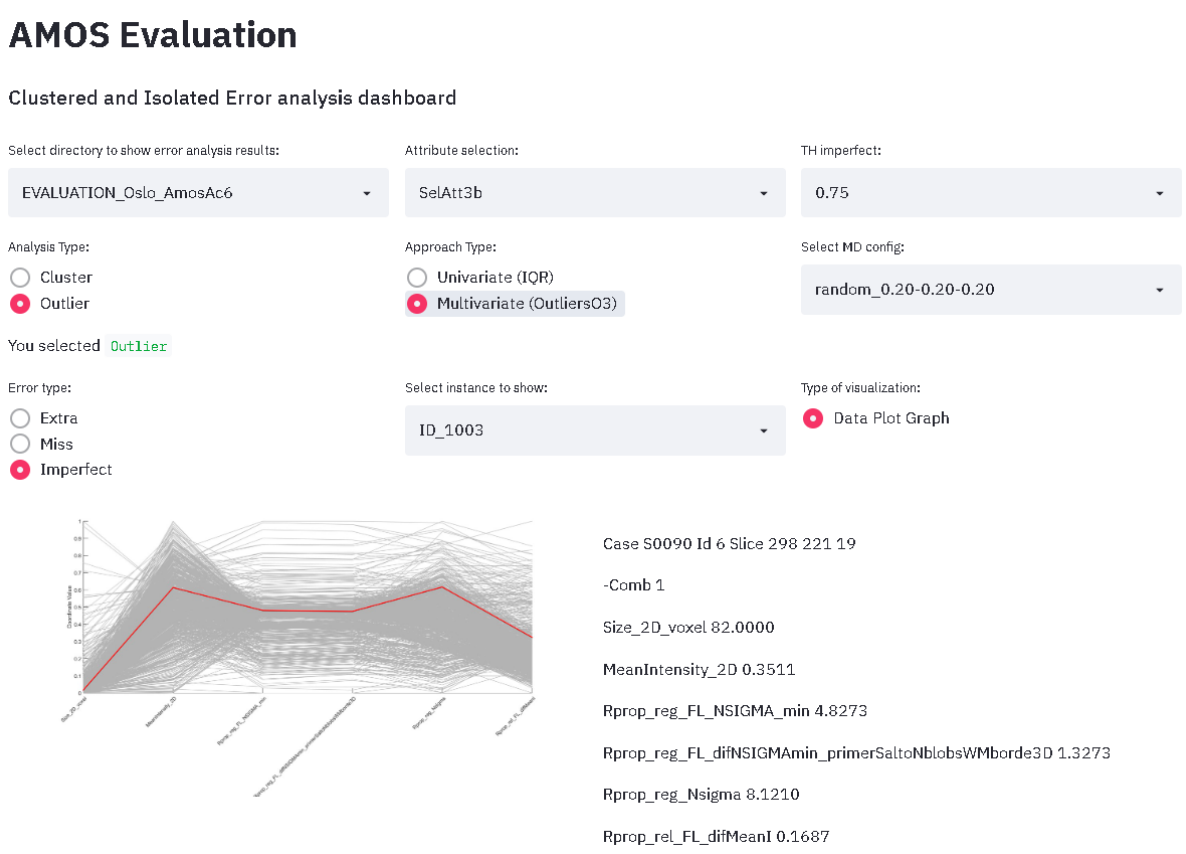


Figura 3.21: Vista de patrones de error para análisis multidimensional de errores aislados

la metodología; b) las características utilizadas para describir el error y sus tipos y c) el concepto de error y sus tipos.

En esta tesis, solo se ha hecho un uso parcial de la información recogida de manera ontológica, aunque se prevé su uso para una futura automatización completa del proceso de análisis del error. En el estado actual de los sistemas de visión, no es posible, ya que "TODO" el conocimiento debería estar formalizado, y no es el caso.

3.5.1. Las entidades visuales

Una entidad visual es un elemento que se puede diferenciar en una imagen. Se distinguen tres tipos de entidades: imagen, región y objeto visual (figura 3.22). Por imagen entendemos la entidad que contiene toda la información adquirida mediante un sensor visual.

Como se representa en la figura 3.23, una imagen tiene un contexto, es decir, procede de un conjunto de datos ("Dataset") obtenido en un proyecto ("Project"). La imagen se genera utilizado un protocolo ("Protocol") que se almacena en los meta-datos de la imagen junto a otra información como la matriz de transformación de las coordenadas imagen a

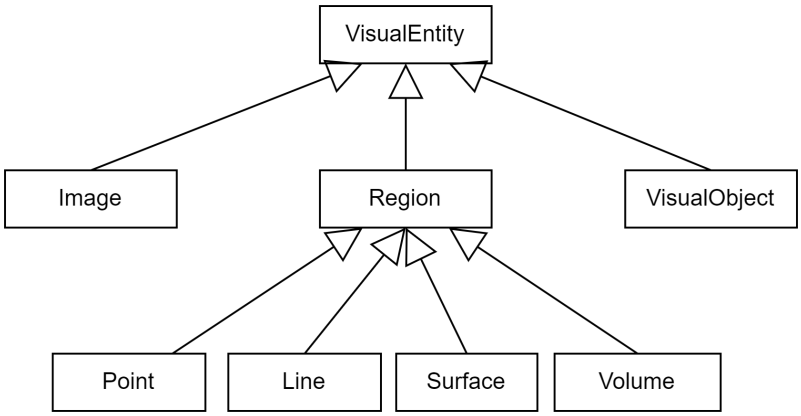


Figura 3.22: Diagrama de la clase "VisualEntity"

las coordenadas mundo ("'Matrix_xyz2ijk"), información sobre el elemento de adquisición ("AdquisitionEntity"), etc.

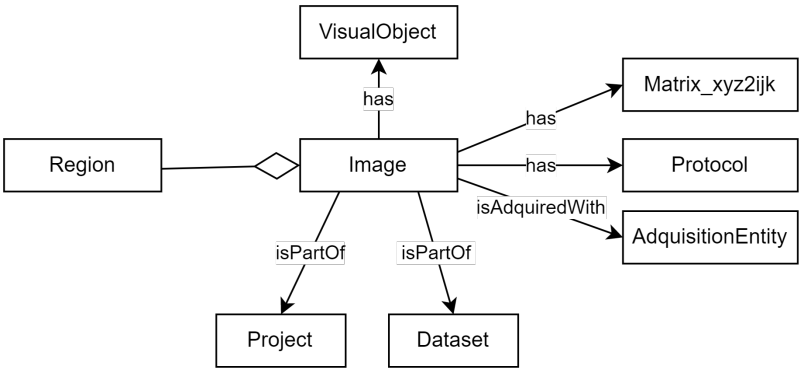


Figura 3.23: Diagrama de la clase "Image"

Además, la imagen se puede descomponer en entidades más pequeñas, como regiones u objetos visuales, siendo la entidad más pequeña el píxel (vóxel en imágenes 3D). En función de su dimensionalidad, los datos de la imagen se pueden tratar de forma unidimensional, como un listado de puntos, de forma bidimensional, como imagen 2D, o de forma tridimensional, como un volumen.

Una "región" es cualquier porción de la imagen que presenta alguna característica homogénea que la distingue del resto. Puede jugar el papel de una imagen en muchos procedimientos, pero cuando las regiones tienen entidad propia, nos referimos a ellas como "objeto visual". Como objeto visual, se le puede asociar una serie de características (figura 3.24). En el dominio de la segmentación de objetos, se distinguen los píxeles/vóxeles del primer plano (foreground) de los del fondo (background). Cada región de píxeles contiguos

del primer plano forman un blob u objeto de segmentación, el cual es un objeto visual que puede ser descrito por una serie de características, "VisualObjectFeature".

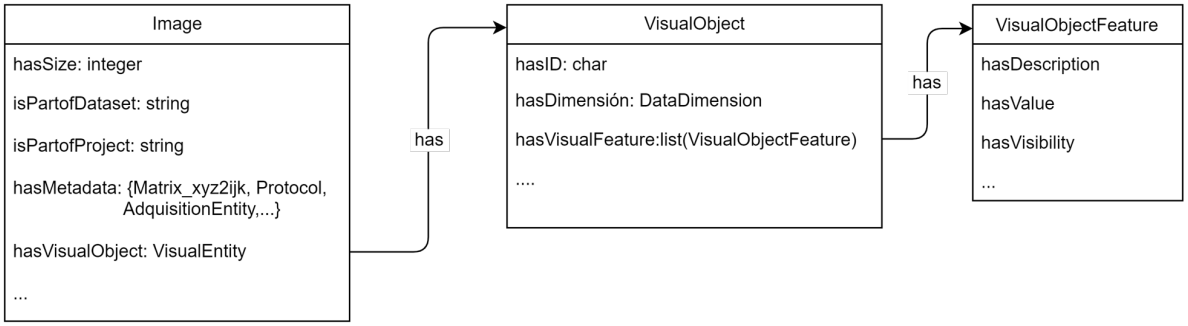


Figura 3.24: Diagrama de la clase "VisualObject"

A su vez, las regiones correspondientes a distintas segmentaciones se pueden combinar (por superposición) y producir una nueva región, la cual da lugar a un nuevo objeto visual, el objeto de agrupación. Los objetos de agrupación se describirán por las características de los objetos que componen la agrupación, más características propias del nuevo objeto. En la figura 3.25 se muestran los tipos de objetos manejados en la metodología. De estos objetos de agrupación se hará uso en la subsección 3.5.3, cuando se describan los tipos de error.

3.5.2. Las características para la descripción del error

Para poder describir distintos tipos de error, es necesario caracterizar los objetos de agrupación de manera que se puedan encontrar similitudes y diferencias entre ellos. La manera más simple es mediante un espacio de características n-dimensional en el que todas las características tengan la misma importancia. Estas características pueden ser muy variadas, dependiendo del punto de vista desde el que analizamos el sistema.

Por ejemplo, en función de su procedencia podemos encontrar características externas e internas al sistema mientras que en función de su origen, podemos diferenciar entre características visuo-espaciales y características de contexto. En el primer caso, las características externas se obtienen de analizar la información disponible desde fuera del sistema, mientras que las características internas se obtienen de los procesos internos del sistema de segmentación. En el segundo caso, las características visuo-espaciales proceden del análisis de las imágenes mientras que las características de contexto son características propias del dominio, la tarea y el sistema. Éstas últimas no están directamente relacionadas con la imagen pero ayudan a separar distintas situaciones, lo que facilita el análisis del error.

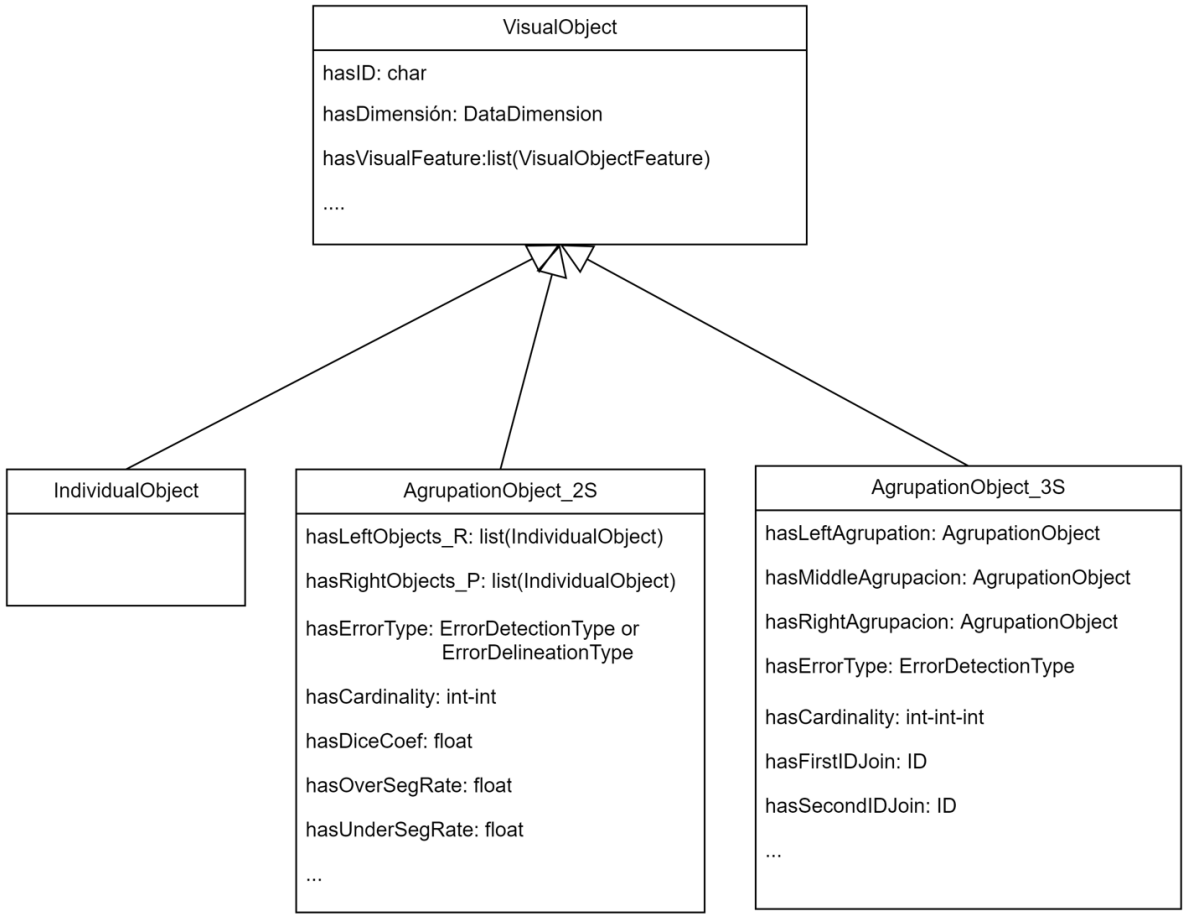


Figura 3.25: Tipos de objetos de la metodología AMOSE[2]

El número de características disponibles para describir los objetos de agrupación dependerá de la opacidad del SVC. En un sistema de caja negra solo se dispondrá de características externas, mientras que en un sistema de caja gris podremos mejorar la descripción con información sobre su funcionamiento interno, la configuración utilizada o el resultado que se obtiene en puntos intermedios del proceso. Cuanta más información esté accesible, más extensa y precisa puede ser la caracterización de los objetos de agrupación y, por tanto, más probable será encontrar patrones de error con mayor soporte en el conjunto de datos utilizado para construir el SVC.

3.5.2.1. Ontología "Visual Object Feature"

Disponer de múltiples características es importante para mejorar los procesos de evaluación, pero para que esta mejora sea real y efectiva es importante poder reutilizarlas. Para ello, es necesario formalizarlas mediante herramientas que permitan su descripción y gestión. La clase "VisualObjectFeature" (VOF), cuyo diagrama se muestra en la figura 3.26, ofrece un marco para la descripción de las características de un objeto visual.

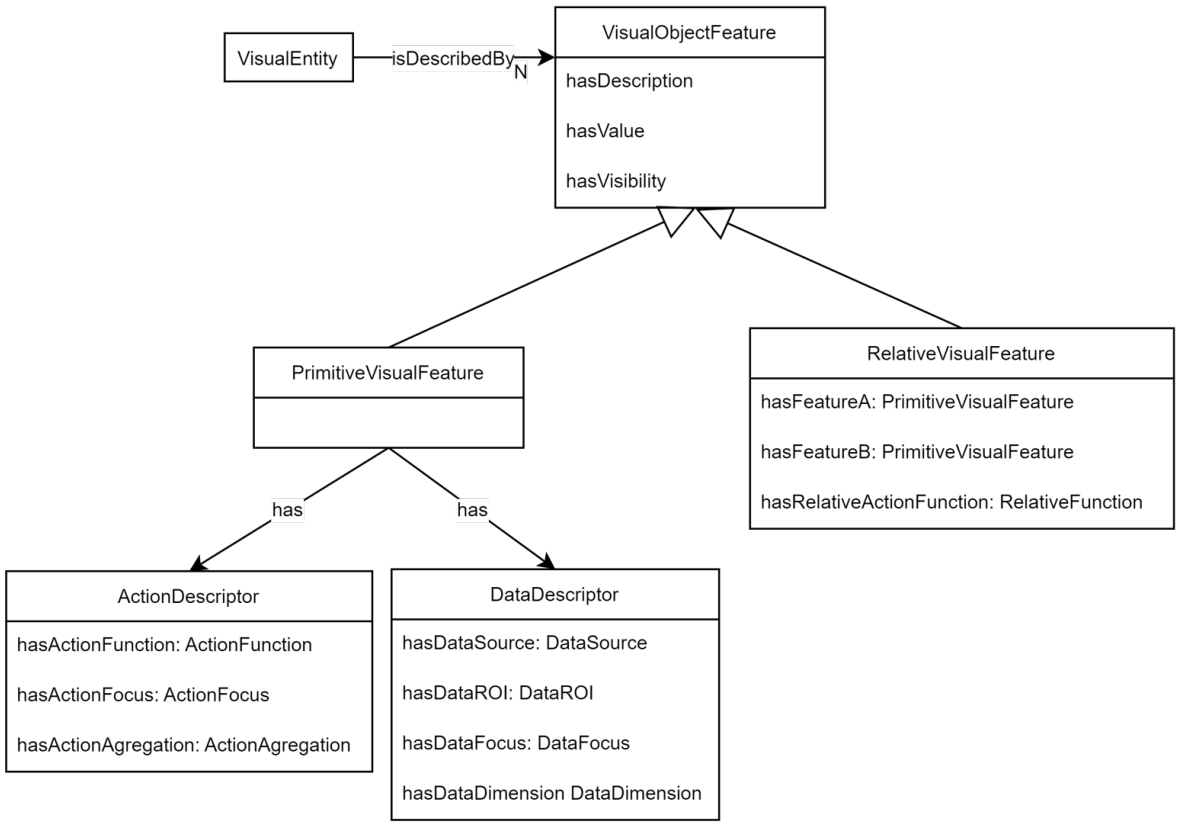


Figura 3.26: Diagrama de la clase "VisualObjectFeature"

Una característica visual se obtiene a partir de los datos asociados a la entidad visual ("VisualEntity"). Toda "VisualObjectFeature" tiene una serie de atributos comunes, como hasDescription (Descripción textual de la característica), hasValue (valor asignado), hasVisibility (si se utiliza o no en un momento dado), etc.

En una descripción de un objeto visual ("VisualObjectFeature") se distinguen dos tipos de características: (1) las características primitivas ("PrimitiveVisualObjectFeature"), cuando se trata de una característica visual directa, de la propia entidad, y (2) las características relativas ("RelativeVisualObjectFeature"), cuando se trata de una comparación (diferencia, proporción, ...) entre la característica visual de la propia entidad descrita respecto a otra que se utiliza como referencia.

Siguiendo la definición propuesta en esta ontología, una "PrimitiveVisualFeature" se compone de dos descriptores semánticos, el descriptor de los datos ("DataDescriptor"), que aporta información sobre el origen de los datos, y el descriptor de la acción ("ActionDescriptor"), que describe la operación utilizada para obtener dicha característica visual. Cada uno de ellos se describe mediante la serie de sub-descriptores que se detalla a continuación.

"DataDescriptor" se describe mediante los siguientes atributos:

- DataDimension: describe el número de dimensiones de la entidad utilizada (1D, 2D, 3D,..., nD).
- DataSource: indica la fuente usada para el cálculo, es decir, la imagen concreta de la que se obtienen los datos de la entidad utilizada.
- DataFocus: indica el foco utilizado para realizar el cálculo de la característica (toda la imagen, el objeto de agrupación u otro objeto).
- DataROI: indica si se restringe el foco a una subregión de interés dentro del elemento seleccionado en DataFocus (interior, periferia, zona concreta, etc.).

"ActionDescriptor" se describe mediante los siguientes atributos:

- ActionFocus: indica el objetivo de la característica. Describe si la característica porta información de contexto (experimento, caso,etc.), información de control del sistema, o información visuo-espacial (distancias, vecindades, propiedades locales, etc).
- ActionFunction: especifica el descriptor concreto utilizado para obtener el valor de la característica. Entendemos que existen múltiples descriptores de una misma cualidad, y aquí se selecciona uno de ellos y, de manera implícita, toda la información que lleva asociada. Por ejemplo, si queremos introducir en el sistema características de forma, podemos seleccionar el descriptor "Excentricidad", o el descriptor "Redondez" o el descriptor "DiametroEquivalente". La selección del operador, lleva implícito que estamos utilizando características de descripción de la forma del objeto.
- ActionAgregation: indica la manera de agregar los datos. Dado que las características visuales configuradas en ActionFunction suelen ser características locales a una entidad visual, en función de la naturaleza de la característica será necesario agregar un conjunto de valores. Por ejemplo, cada píxel tiene un valor de intensidad, pero para dar una característica visual asociada a un objeto, es necesario agregar las intensidades de todos los píxeles del objeto. Esta agregación puede ser tanto una función estadística (media, mediana, cuartil 95, etc.), como un valor puntual representativo (máximo, mínimo, etc.).

A partir de la estructura jerárquica y de conceptos que describe esta ontología se puede instanciar las descripciones semánticas de otras aplicaciones para un dominio y tarea concreta. Un ejemplo de descripción semántica para la característica de forma "ShapeFeature" se muestra en la figura 3.27, que proviene de la ontología de aplicación

"AMOS-2D_VOF". En la figura se presentan los diferentes conceptos, en este caso medidas y descripciones de la forma de un objeto segmentado como la excentricidad o el eje mayor, que se utilizan para describir el rol de la característica de acción ("ActionFunction").

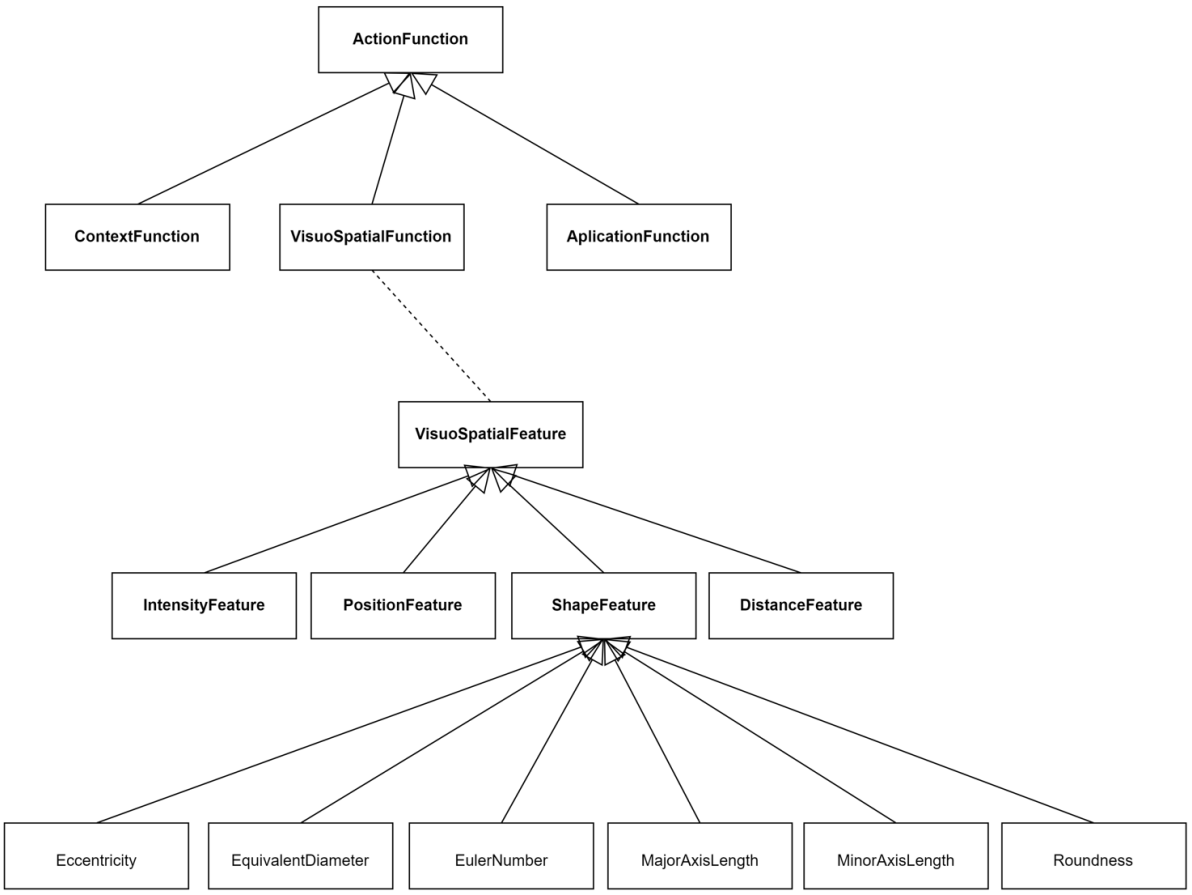


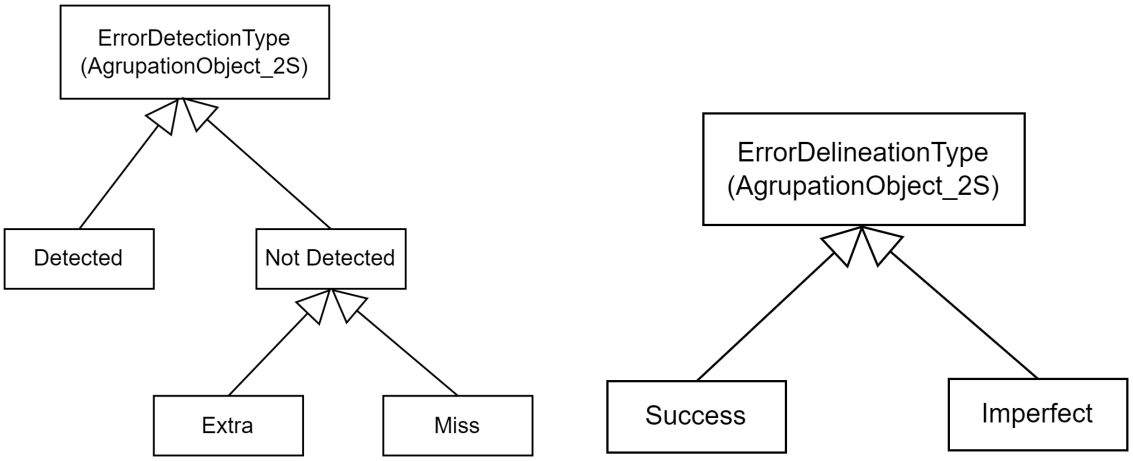
Figura 3.27: Ejemplo de jerarquía semántica de un descriptor visual

Además, el modelado de la información semántica de las características con esta ontología permitirá calcular su especificidad en función del nivel de profundidad o la similitud semántica entre características, facilitando al usuario decisiones como la selección de características o la presentación ordenada de resultados.

3.5.3. Los tipos de error

Para evaluar el error en una segmentación propuesta, P, respecto a una segmentación de referencia, R, se comparan las dos segmentaciones. Para ello, se combinan los objetos de la propuesta, O_P, con los de la referencia, O_R, y se generan unos nuevos objetos de agrupación $O_{R.P}$ de tipo AgrupationObject_2S. Del análisis de estos objetos, se deduce si la segmentación propuesta ha sido exitosa o si contiene algún tipo de error.

En la figura 3.28 se diferencian ambos tipos de error. El error de detección separa los objetos de agrupación en tres tipos: Miss, Extra y Detected, mientras que el error en delineación se da en los objetos detectados y los separa en dos tipos: Imperfect y Success.



(a) (b)

Figura 3.28: Tipos de error de un objeto de agrupación: (a) error de detección y (b) error de delineación.

En el caso de análisis del error comparado, se evalúa una segmentación propuesta, P, respecto a la referencia, R, y a otra segmentación alternativa, A. De la combinación de las segmentaciones, también se genera un nuevo objeto de agrupación, ahora comparado, que hemos denominado AgrupationObject_3S. Este análisis comparado permite detectar hasta 10 comportamientos enlazados de error desde el punto de vista del error de detección, como se muestran en la figura 3.29.

El error de delineación no se ha definido explícitamente porque no se ha considerado necesario realizar un estudio tan exhaustivo de los comportamientos enlazados de error comparados, pero no cabe duda de que podría ser una línea de trabajo futuro.

3.6. Formas de uso de la metodología

La metodología propuesta permite realizar una evaluación profunda del error a partir del análisis de los objetos de una o varias agrupaciones con el objetivo de descubrir nuevo conocimiento sobre el comportamiento del error en segmentación de objetos. En la tabla 3.7 se recogen sus dos formas de uso: el análisis individual y el análisis comparado.

Por un lado, en la forma de uso "Análisis individual" se compara la segmentación propuesta por un SVC, O_P, respecto a la segmentación de la referencia proporcionada por los expertos del dominio, O_R, con el objetivo de obtener nueva información sobre los errores para determinar sus causas. Esta forma de uso se puede generalizar si utilizamos

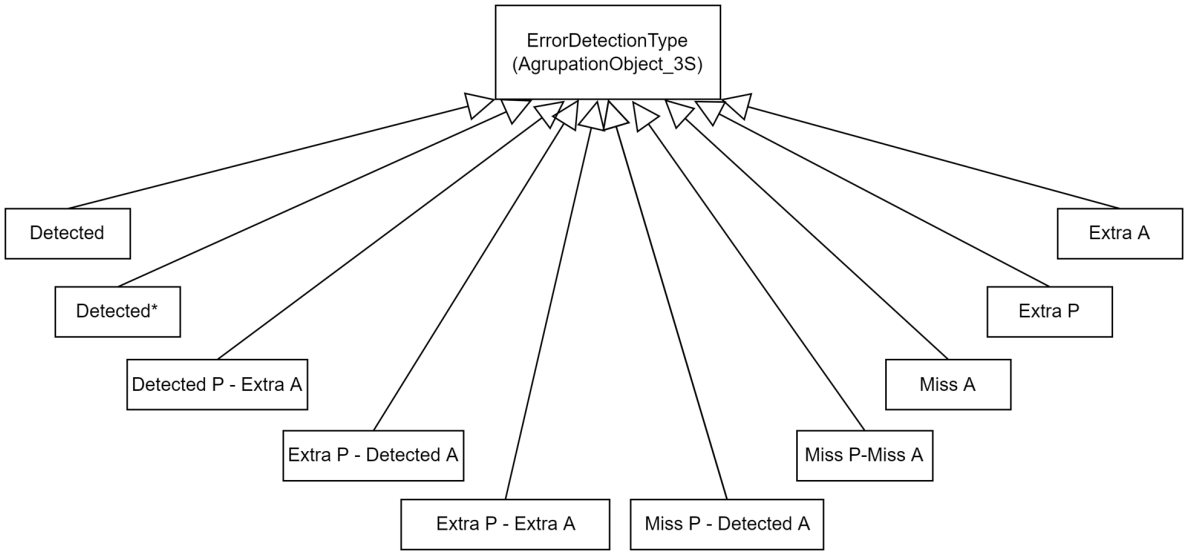


Figura 3.29: Tipos de error de detección en análisis de error comparado

Nombre	Objetos	Descripción
Individual	O_R O_P	Se compara una segmentación propuesta P respecto a una segmentación de referencia R. Se puede generalizar y comparar dos soluciones entre sí. Etapas: Se crean los objetos de agrupación, se caracterizan y clasifican por tipo de error (Miss, Extra, Imperfect y Success). Se analizan los errores con métodos clásicos (DSC, AVD, F1-score) y se buscan patrones con métodos de IA (detección de patrones de error y de errores aislados).
Comparada	O_P O_A O_R	Se compara una propuesta P respecto a otras dos, una de referencia R y otra segmentación de propuesta alternativa, A. Etapas: Se crean los tres objetos de agrupación y se enlazan mediante sus objetos componentes creando tablas LOJ que se analizar para detectar comportamientos enlazados de error, sus similitudes y diferencias entre si.

Tabla 3.7: Formas de uso de la metodología

como referencia otra segmentación. En este caso, se pueden comparar dos soluciones de segmentación, ya sean dos algoritmos de segmentación distintos, O_{P_A} respecto a O_{P_B}, o un mismo algoritmo con diferente configuración, $O_{P_{A1}}$ respecto a $O_{P_{A2}}$.

Por otro lado, en la forma de uso "Análisis comparado", se comparan tres segmentaciones entre sí: la segmentación propuesta O_P respecto a la segmentación de referencia O_R, la segmentación propuesta respecto a la segmentación alternativa O_A y la segmentación alternativa respecto a la segmentación propuesta. Su objetivo es conocer comportamientos enlazados de error para mejorar el conocimiento de estas situaciones, como por ejemplo, conocer dónde la segmentación alternativa sí detecta el objeto y no lo detecta la propuesta, o cuando la propuesta detecta objeto y no es correcto, o cuando no hay objeto de referencia, pero si lo detectan las segmentaciones de propuesta y alternativa.

Capítulo 4

Caso de uso: La caracterización del error en la segmentación de hiperintensidades de la sustancia blanca cerebral

En este capítulo, para ejemplificar el uso de la metodología propuesta, se analiza el problema de la segmentación de hiperintensidades cerebrales a partir de imágenes de resonancia magnética. Desde el punto de vista de los objetivos de esta tesis, su detección es un ejemplo de segmentación de objetos amorfos, donde las hiperintensidades son regiones generalmente pequeñas, no homogéneas y cuyos bordes son difusos. Además, su tono y textura pueden no ser lo suficientemente significativos para diferenciarlos de tejidos vecinos, por lo que la correcta detección y delineación de manchas hiperintensas es un proceso complejo.

Gracias a la evolución de la tecnología en los últimos años, la segmentación semántica y la segmentación de instancias han experimentado un gran avance. A pesar de ello, la realidad es que aún, hoy día, los SVC necesitan ser mejorados para obtener resultados más precisos, por lo que disponer de nuevas herramientas y métodos que faciliten la detección de errores y su descripción sencilla es de gran utilidad.

A continuación, se procede a describir y contextualizar el problema, se revisa el estado del arte en segmentación de hiperintensidades de sustancia blanca y se analizan varias soluciones con el objetivo de ejemplificar el uso de la metodología AMOSE^2.

4.1. Descripción del problema

4.1.1. La leucariosis

La leucoariosis es un término introducido por Hachinski et al. [1987] para designar a las alteraciones de la sustancia blanca cerebral subcortical y/o periventricular. Se trata de un hallazgo radiológico que se detecta en las imágenes obtenidas por técnicas como la resonancia magnética nuclear y la tomografía axial computarizada. En la literatura científica también se las denomina hiperintensidades o lesiones de la sustancia blanca, en inglés White Matter Hyperintensities (WMH) o White Matter Lesions (WML). Se suelen dividir en función de su origen en dos tipos: las de origen vascular y las de origen no vascular. Las primeras están asociadas a pérdida funcional, discapacidad y deterioro cognitivo. Se trata de una de las patologías de mayor incidencia, en adultos mayores se asocia principalmente a la enfermedad de los pequeños vasos sanguíneos (CSVD) [Wardlaw et al., 2015]. Las segundas se relacionan con problemas de movilidad y se asocian a enfermedades inflamatorias y neurodegenerativas como la esclerosis múltiple [Carass et al., 2017, Frey et al., 2019]. Éstas últimas tienen generalmente unos límites más nítidos y distinta localización, aunque en algunos casos su diferenciación puede ser todo un reto [Geraldes et al., 2021].

La leucoariosis es uno de los problemas más estudiados en neuroimagen por estar presente en un gran número de patologías y estar asociado a un aumento del riesgo de accidente cerebrovascular. Hasta hace poco tiempo, se descartaba su análisis ya que se pensaba que formaban parte del proceso natural de envejecimiento, pero numerosos estudios indican que tiene importantes asociaciones clínicas y son factores de riesgo [Yoshita et al., 2006, Debette and Markus, 2010, Carass et al., 2017, Salvadó et al., 2019, Brugulat-Serrat et al., 2020]. Estas alteraciones se caracterizan por áreas difusas, bilaterales, que aparecen como regiones hiperintensas en resonancia magnética craneal o hipodensas en la tomografía computerizada. Un ejemplo de imagen de resonancia magnética con leucoariosis se muestra en la figura 4.1, en la que se ha marcado en azul las alteraciones. Se presentan frecuentemente en personas de edad avanzada, particularmente aquellas que son hipertensas, o que han sufrido un ictus, aunque su prevalencia es muy variable. Los hallazgos histopatológicos, es decir, el estudio de las células y el tejido enfermo bajo un microscopio, comprenden edema local, desmielinización y pérdida axonal debido probablemente a la lesión isquémica [Mauriño Donato and Álvarez-Sabin, 2004]. Determinar su número, tamaño y localización es fundamental para averiguar la etimología y la progresión de la enfermedad, permitiendo conocer cuáles son las funciones vasculares más importantes de la patología, conocer si hay anomalías reversibles o la relación entre

lesiones y síntomas [Zijdenbos et al., 1994, Yoshita et al., 2006, Debette and Markus, 2010, Manjón et al., 2018, Gwo et al., 2019].

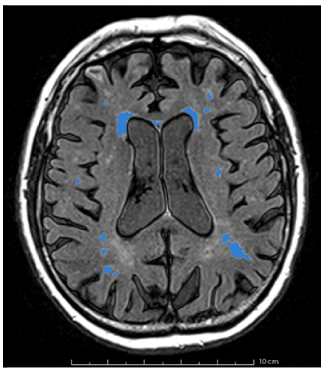


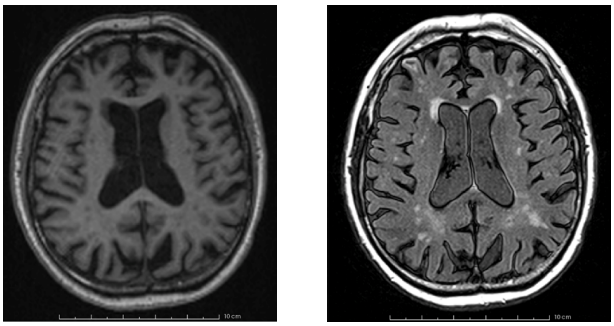
Figura 4.1: Imagen de RM de segmentación de leucariosis (en azul)

4.1.2. Detección de WMH en imágenes de resonancia magnética

La resonancia magnética (RM) es una técnica no invasiva y no ionizante que permite obtener información, en cualquier plano, de la estructura, la composición o el funcionamiento de partes corporales utilizando el fenómeno de la resonancia. La generación de imágenes mediante RM proviene de la recogida de ondas de radiofrecuencia (RF) procedentes de la estimulación de la materia (principalmente de los núcleos de hidrógeno), la cual se ha magnetizado previamente mediante la acción de un campo magnético [Morales and Torres, 2008]. Los núcleos se orientan mayoritariamente en la dirección del campo externo, con un ángulo de precesión dependiente de la fuerza del campo magnético. La estimulación de la materia mediante un pulso de RF provoca que el vector de magnetización neta gire del plano longitudinal al plano transverso. Dependiendo del giro de este vector, se diferencian dos tipos: T1 es el tiempo que tarda en recuperarse el 63 % de su magnetización longitudinal y T2 es el tiempo en perder el 63 % de su magnetización trasversal. T1 muestra de forma óptima la anatomía normal del tejido blando y la grasa y T2 muestra el líquido y alteraciones como tumores, inflamación y traumatismos. Según la sustancia que se quiera resaltar, se utilizan diferentes secuencias de pulsos de RF, lo que da lugar a diferentes modalidades de imagen.

Para el caso concreto de la segmentación de hiperintensidades en sustancia blanca, la modalidad de imagen que proporciona más información se conoce como FLAIR ("Fluid Attenuated Inversion Recovery"). FLAIR es una secuencia de imágenes por resonancia magnética que revela la prolongación T2 de tejido con supresión de fluido cerebro-espinal (CSF). Esta secuencia elimina la señal del líquido cefalorraquídeo, pero no la señal proveniente de lesiones patológicas que suelen presentar un aumento del contenido de agua, o edema, por lo que es útil en su identificación [Rivera et al., 2011]. En la figura

4.2, se muestra un ejemplo imágenes de RM de tipo T1 y FLAIR. En la primera, se distinguen los tres tipos de tejido cerebral: la sustancia blanca (WM), la sustancia gris (GM) y el líquido cerebro-espinal (CSF). En cambio, en la segunda, se distinguen las hiperintensidades en sustancia blanca (WMH) del tejido neuronal normal.



(a) T1 (b) FLAIR

Figura 4.2: Ejemplo de imágenes de resonancia magnética T1 y FLAIR

4.1.3. Almacenamiento de imágenes médicas

Existen diferentes modalidades de imagen médica: imagen plana, multicorte bidimensional, imagen tridimensional, series temporales, etc. Estas imágenes, así como la información clínica del paciente, se almacena en los sistemas de comunicación y archivado de imágenes PACS ("Picture Archiving and Communication System") de los hospitales, utilizando generalmente el formato DICOM ("Digital Imaging and Communications in Medicine"). DICOM es un formato muy poderoso y flexible, pero complejo, y proporciona interoperabilidad entre diferentes hardware y software. Sin embargo, DICOM no es eficiente para el procesamiento y análisis de imágenes ya que cada volumen se almacena como una secuencia de cortes 2D, lo que puede ser engorroso de manejar. Existen otros formatos de archivos médicos adecuados para esta tarea como NIFTI, NRRD, MNCI o el novedoso BIDS. El formato NIFTI, que es una versión mejorada del formato Analyze, fue diseñado para ser más simple que DICOM, al tiempo que conserva todos los metadatos esenciales. El formato NIFTI ha sido el elegido para este trabajo, pues permite almacenar un volumen en un solo archivo, con un encabezado simple seguido de datos sin procesar. Esto hace que sea rápido de cargar y procesar.

En el caso RM cerebral hay que prestar atención tanto a las diferentes modalidades de representación de la imagen como a la orientación del paciente en la adquisición de ésta. Es muy importante conocer estas cuestiones para no realizar manipulaciones o interpretaciones erróneas. Generalmente, los sistemas de visualización trabajan en coordenadas mundo (x,y,z) y, en función del convenio utilizado, pueden mostrar los datos

en vista radiológica o vista neurológica (figura 4.3). La diferencia está en el punto de vista desde donde se mira a la imagen, si de frente o desde atrás. Por contra, los sistemas computacionales de procesado trabajan en coordenadas imagen (i,j,k), por lo que es necesario saber establecer la correspondencia entre ambos sistemas de coordenadas. La transformación se realiza con una matriz de transformación afín que define el punto de vista del volumen 3D de datos. Para facilitar el procesamiento, será necesario llevar todas las imágenes a una representación volumétrica 3D (una matriz 3D) que tengan la misma orientación y cuyos vóxeles tengan las mismas dimensiones.

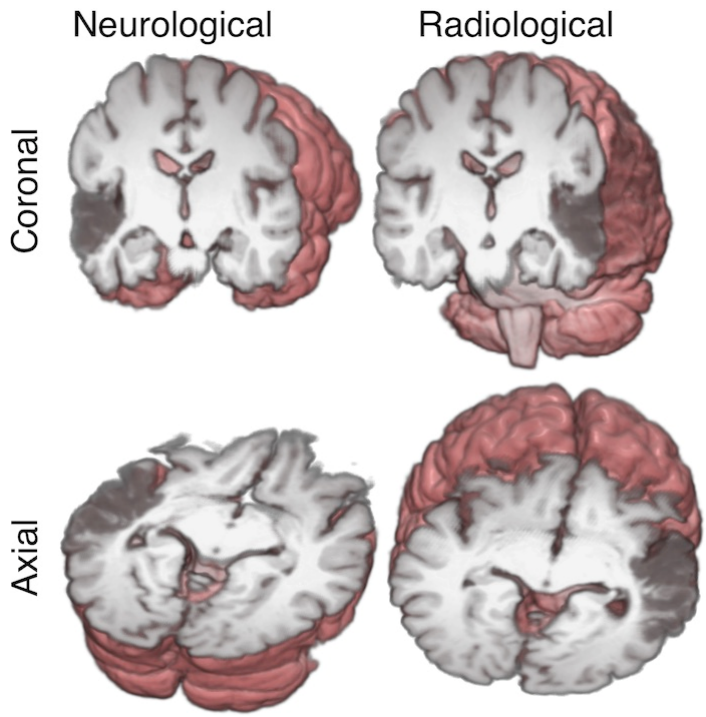


Figura 4.3: Diferentes vistas (Neurológica/Radiológica) y cortes (axial/coronal) de una imagen cerebral humana
Fuente: https://nipy.org/nibabel/neuro_radio_conventions.html

4.1.4. La segmentación de hiperintensidades cerebrales

El progreso de los métodos de segmentación ha ido de la mano de la mejora de la tecnología, con máquinas de mayor calidad y capacidad tanto para la adquisición de imágenes como para su procesado. El preprocesado previo a la segmentación es similar en la mayoría de los métodos e incluye operaciones de coregistro entre imágenes de distinta modalidad, extracción de la región del cerebro, corrección de inhomogeneidades de la intensidad debidas al sistema de adquisición (corrección del sesgo, en inglés "bias correction"), reducción del ruido y normalización de la intensidad. Además, en la

mayoría de publicaciones se combinan distintas modalidades de imagen (T1, T2 y FLAIR principalmente) para obtener segmentaciones más robustas.

Existen múltiples algoritmos de segmentación, como se recoge en revisiones recientes Caligiuri et al. [2015]Balakrishnan et al. [2021]Frey et al. [2019], donde se analizan algunos de los algoritmos de segmentación automáticos y semiautomáticos desarrollados desde los años 80 hasta hoy día.

En Caligiuri et al. [2015], se estudian 34 artículos sobre segmentación de lesiones hiperintensas publicados entre 1980 y 2014. Se comparan diferentes algoritmos supervisados y no supervisados tanto automáticos como semiautomáticos. Los algoritmos no supervisados se basan principalmente en técnicas de clustering, mediante umbralizaciones fijas o difusas de los niveles de intensidad, y se mejoran mediante la utilización de información geoestadística y/o multiespectral junto modelos con varios niveles de segmentación. Los algoritmos supervisados necesitan de la segmentación del experto en el dominio y utilizan distintas técnicas de aprendizaje supervisado para clasificar los píxeles de la imagen: clasificadores basados en los k vecinos más cercanos ("K Nearest Neighbours", K-NN), máquinas vectores soporte SVM ("Suport Vector Machine"), métodos bayasianos multiespectrales o redes neuronales, entre otros. Algunos algoritmos son completamente automáticos, otros necesitan de la interacción de un experto para ajustar parámetros, iniciar puntos semilla, etc. En dicha revisión se critica que la mayoría de los métodos analizados no están disponibles libremente, son específicos del estudio y/o del protocolo y han sido validados principalmente con muestras pequeñas.

En la segunda revisión, Balakrishnan et al. [2021] analizan 37 artículos de segmentación de lesiones hiperintensas de la sustancia blanca cerebral de origen vascular entre 2015 y 2020. Como se comenta en el artículo, en los últimos años ha habido un gran aumento del tamaño de las imágenes junto a una mejora de la calidad de las mismas, un incremento de la potencia computacional y una mejora de los métodos de aprendizaje máquina basados en aprendizaje profundo que ha fomentado el desarrollo de nuevos algoritmos. En ella se analizan artículos con conjuntos de datos tanto grandes como pequeños, donde 27 métodos son supervisados, de los cuales 10 están basados en aprendizaje profundo, y 10 son basados en métodos no supervisados, de los cuales uno se basa en redes neuronales convolucionales (CNN). Concluyen que a pesar de la creciente popularidad y la alta precisión de los esquemas basados en CNN aplicados a la segmentación de WMH, no existe evidencia de que favorezca su aplicación en la investigación clínica sobre los métodos de aprendizaje tradicionales. También destacan que hay que evitar los sesgos y mejorar la transparencia de los informes, por lo que sería recomendable que los estudios futuros analizarán el efecto combinado de varias métricas en la evaluación de los resultados de sus algoritmos.

Una revisión de estudios sobre segmentación de hiperintensidades con conjuntos de datos grandes se presenta en Frey et al. [2019]. En ella analizan los estudios de lesiones hiperintensas de origen vascular, con muestras de más de 500 casos y con pacientes de edad avanzada ($67 \pm 0.8\,\%$) realizados entre los años 2005 y 2018 (ver figura 4.4). Según su análisis, destacan dos algoritmos basados en métodos no supervisados [DeCarli et al., 2005, Brickman et al., 2011] y cuatro basados en métodos supervisados [Zijdenbos et al., 2002, Anbeek et al., 2005, de Boer et al., 2009, Maillard et al., 2008], y concluyen que existe una multitud de métodos de segmentación de hiperintensidades, aunque solo una minoría proporciona definiciones claras de WMH, ya sea con la recomendación STRIVE[1] o con otra definición, lo que limita la comparabilidad y reproducibilidad de los resultados. Por otro lado, también concluyen que los nuevos métodos de segmentación basados en aprendizaje profundo pueden mejorar aún más la segmentación automatizada en un futuro, por lo que es necesario crear y adherirse a pautas de informes que cubran tanto la definición de WMH como la descripción del enfoque de la segmentación.

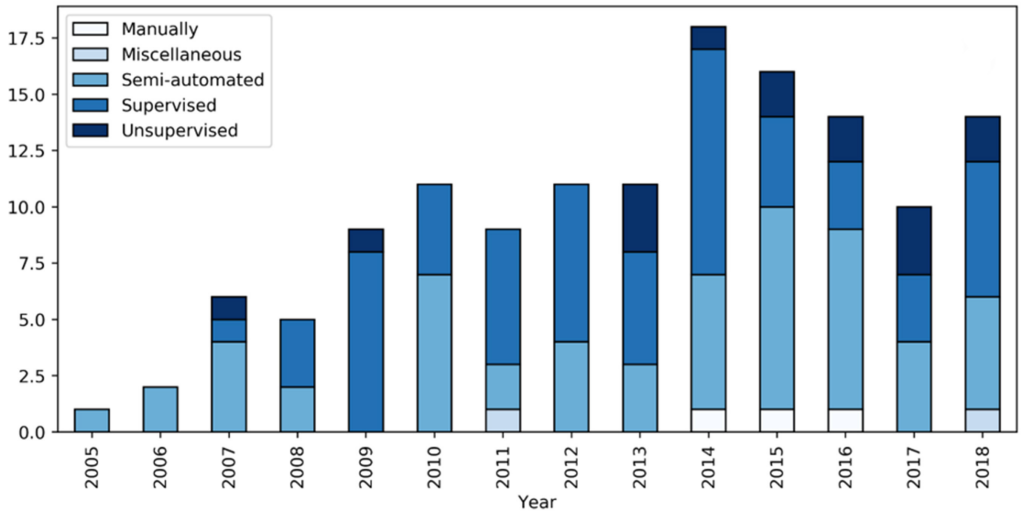


Figura 4.4: Selección de estudios y su evolución en función del número de estudios de gran escala por año separados por tipo de algoritmo
. Fuente: Frey et al. [2019]

En resumen, dada la complejidad de este problema, a día de hoy no existe un método óptimo, pero la mejora de la tecnología ha propiciado el desarrollo de nuevos algoritmos que ofrecen cada día mejores soluciones. Son sistemas (semi-)automáticos y flexibles a las

[1]El protocolo STRIVE (Standards for Reporting Vascular changes on Neuroimaging) es un consenso internacional de neuroimagen para CSVD que determina la clasificación, terminología, definiciones e incluye: infartos subcorticales pequeños recientes, lagunas de presunto origen vascular, hiperintensidad de sustancia blanca, microhemorragias cerebrales, espacios perivasculares y atrofia cerebral (Barrera and Cabezas [2021]).

diferentes características de las imágenes de resonancia magnética, que generan buenos resultados para múltiples escenarios en cuanto a carga de lesiones hiperintensas.

Algunas de las herramientas más utilizadas y disponibles de forma libre son LST-LGA [Schmidt et al., 2012] y LST-LPA [Schmidt, 2017] para SPM, BIANCA [Griffanti et al., 2016] para FLS, CASCADE [Damangir et al., 2012] o WHAT Li et al. [2022]. Sus resultados están permitiendo ayudar a los expertos en la investigación de la influencia de las lesiones hiperintensas tanto en el envejecimiento natural como en las enfermedades. Aunque hay que tratar con cautela los resultados obtenidos y las conclusiones derivadas de sus análisis, ya que debido a la estructura opaca de los nuevos algoritmos y el gran volumen de datos que manejan los nuevos proyectos, se pueden enmascarar comportamientos no deseables, que en el ámbito de la salud pueden ser cruciales. Por ello, es recomendable que, en un futuro, se utilicen algoritmos de segmentación explicables, que aporten más información para mejorar los métodos de evaluación, de forma que se facilite la toma de decisiones.

4.2. Materiales y métodos

4.2.1. Datasets

Se utilizan varios conjuntos de datos (datasets) de imágenes de RM procedentes de diferentes proyectos: AMOS-2D[2] [Rincón et al., 2017], Brno y WMH segmentation Challenge [3] [Kuijf et al., 2019], que se etiquetan con el nombre de la ciudad donde fueron adquiridos: Oslo, Brno, Amsterdam, Singapore y Utrecht. En todos ellos, se dispone de dos modalidades de imagen, FLAIR y T1, de al menos una segmentación de referencia. Para el dataset de Oslo, del que se disponía de tres segmentaciones de referencia creadas por tres expertos del dominio, se ha utilizado una segmentación agregada que se ha creado utilizando la técnica de voto por mayoría a nivel de vóxel.

En total, se dispone de 106 casos de estudio de 106 pacientes de edad media o avanzada, que proceden de cinco máquinas diferentes, cuyas características más relevantes se presentan en la tabla 4.1. Nótese que la distancia entre slices en varios de los dataset es mayor o igual a 3 mm en las secuencias FLAIR. Esta peculiaridad obliga tanto a los expertos como a los sistemas automáticos a analizar los volúmenes slice a slice, de forma bidimensional. Nuestro análisis es semejante al del experto y trataremos a los objetos como objetos 2D. Dado que la imagen procede de un volumen tridimensional, utilizaremos el término vóxel como unidad de análisis, aunque en imágenes 2D sean equivalentes a píxel.

[2] http://www.simda.uned.es/AMOS2D_SharedData/index.html
[3] https://wmh.isi.uu.nl/

			Características de las imágenes			
Dataset	**Modelo de escáner**	**Casos**	**Secuencia**	**Orientación**	**Slices**	**Tamaño Vóxel (mm^3)**
Oslo	1.5T Siemens Espree	28	2D axial FLAIR	Transverse	36	0.45 x 0.45 x 3.90
			T1-weighted MPRAGE 3D	Sagittal	160	1.35 x 1.35 x 1.20
Brno	3T Siemens Prisma	18	3D FLAIR	Sagittal	256	1.00 x 1.00 x 1.00
			T1-weighted MPRAGE 3D	Sagittal	256	1.00 x 1.00 x 1.00
Amsterdam	3T GE Signa HD x t	20	3D FLAIR	Sagittal	132	0.98 x 0.98 x 1.20
			3D T1-weighted	Sagittal	176	0.94 x 0.94 x 1.00
Utrecht	3T Philips Achieva	20	2D FLAIR	Transverse	48	0.96 x 0.95 x 3.00
			3D T1-weighted	Transverse	192	1.00 x 1.00 x 1.00
Singapore	3T Siemens TrioTim	20	2D FLAIR	Transverse	48	1.00 x 1.00 x 3.00
			3D T1-weighted	Transverse	192	1.00 x 1.00 x 1.00

Tabla 4.1: Descripción del conjunto de imágenes de resonancia magnética FLAIR y T1

4.2.2. SVC para segmentación de WMH

En este trabajo se van a utilizar tres algoritmos de segmentación de WMH para ejemplificar el uso de la metodología. Los dos primeros, AMOS-2D y M-UNET, han sido desarrollados en mi grupo de investigación, mientras que el tercero, el algoritmo PGS, se selecciona por ser el actual ganador del reto abierto "WMH Segmentation Challenge". A continuación se describen estos algoritmos.

4.2.2.1. AMOS-2D

AMOS-2D, de las siglas en inglés "AMorphous Object Segmentation in 2D", es un algoritmo basado en técnicas híbridas de extracción de características y aprendizaje máquina [Rincón et al., 2017]. Su diagrama de funcionamiento se muestra en la figura 4.5. En él se observa que las entradas son las imágenes FLAIR y T1, que son preprocesadas para realizar un corregistro de las mismas a un mismo espacio y una corrección del sesgo en la imagen FLAIR. La imagen T1 se utiliza para la extracción de estructuras cerebrales, las cuales se utilizarán para limitar la región de análisis (solo sustancia blanca) y posteriormente para extraer características del contexto de la segmentación (por ejemplo, para medir distancias de los objetos segmentados a distintas estructuras cerebrales). La imagen FLAIR se utiliza para proponer una segmentación inicial de las lesiones hiperintensas mediante un análisis de la distribución de intensidad de la materia blanca con un modelo gaussiano.

AMOS-2D utiliza una umbralización local multinivel a nivel de vóxel para generar una primera segmentación, en la que se identifican los objetos de segmentación como regiones

de vóxeles segmentados contiguos (blobs). Dado que en esta primera segmentación se generan gran cantidad de objetos falsos positivos, se utiliza un filtro clasificador basado en máquinas de vectores soporte (SVM) para distinguir los objetos de segmentación correctos e incorrectos. Para construir el clasificador, se genera un vector de múltiples características asociado a cada objeto, el cual contiene información de características visuo-espaciales de los objetos segmentados (características externas al sistema) y características de etapas internas del algoritmo. El modelo SVM se entrena mediante aprendizaje supervisado con el dataset etiquetado por los expertos.

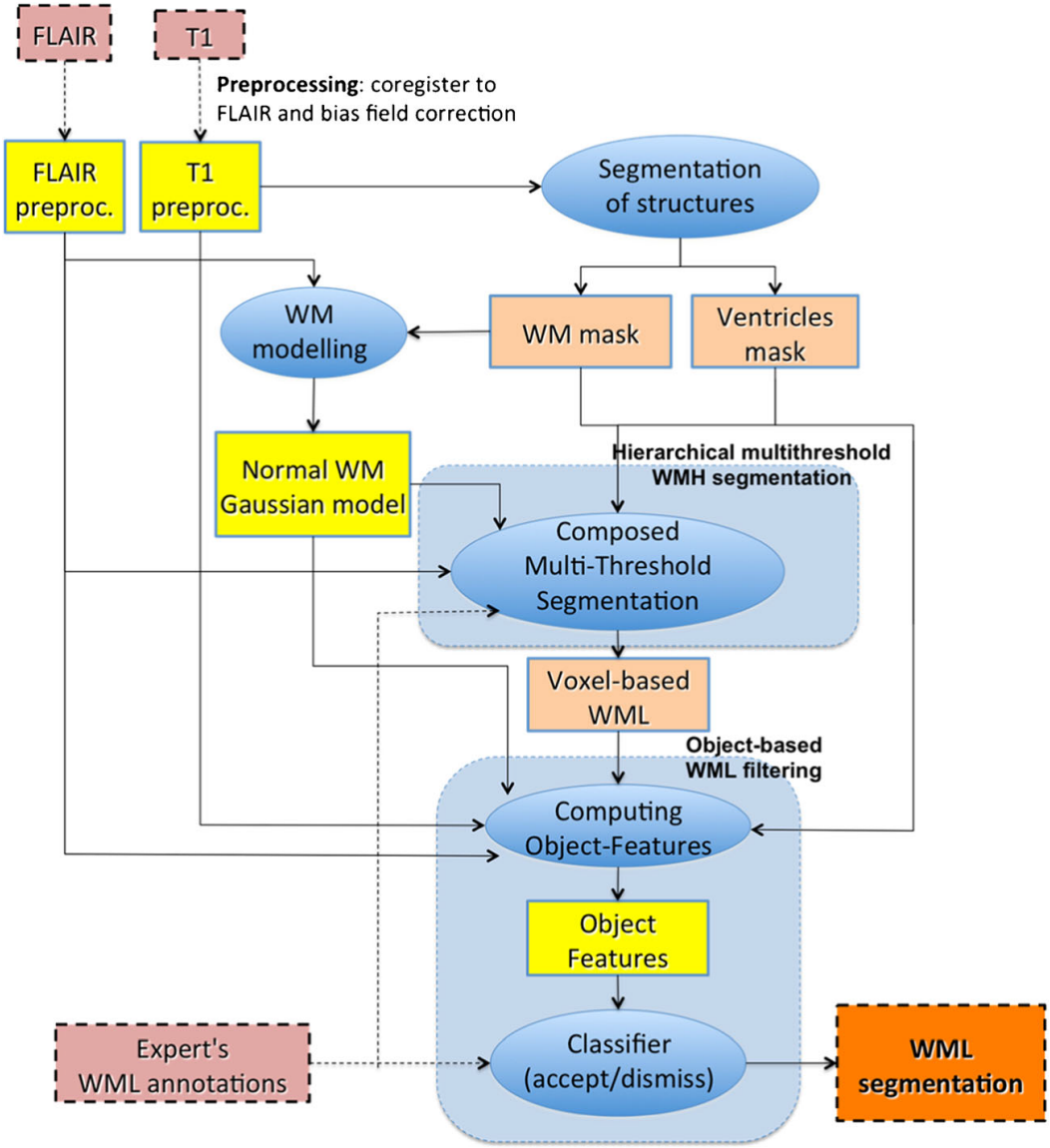


Figura 4.5: Diagrama de funcionamiento del algoritmo de segmentación de hiperintensidades AMOS-2D
Fuente: Rincón et al. [2017]

4.2.2.1.1. La ontología "AMOS-2D_VOF" La ontología "AMOS-VOF", desarrollada para describir las características internas del algoritmo AMOS-2D, se ha modelado en el lenguaje de etiquetado semántico OWL ("Web Ontology Language") con Protégé [Musen, 2015]. Esta ontología es una instanciación de la ontología de descripción de características visuales "Visual Object Feature" descrita en la sección 3.5.2.1. Está formada por 153 clases, 29 propiedades de los objetos, 12 propiedades de datos y 166 individuos.

La jerarquía de clases se compone, por un lado, de los términos asociados a los conceptos que se utilizan para describir las características visuales "VisualFeatureConcepts" y, por otro lado, de los términos asociados a los descriptores de dichas características "VisualFeatureDescriptors", particularizadas para el dominio del problema. Cada característica es un tipo de "VisualObjectFeature" que se calcula sobre una entidad visual ("VisualEntity") y que se describe a partir de descriptores de datos ("DataDescriptiors"), de acción ("ActionDescriptiors") y diferentes propiedades como su visibilidad, su descripción y su valor, es decir, con las propiedades de la clase "VisualObjectFeature" (ver figura 3.26)

Las descripciones de algunos de ellos se muestran en la figura 4.6, donde se observan sus componentes y sus descripciones. Se diferencian dos tipos de características: las características primitivas "PrimitiveVisualFeature", aquellas que se calculan directamente de la imagen y, las características derivadas "RelativeVisualFeature", aquellas que se calculan a partir de variables primitivas.

En la fila (a) se puede observar que la variable denominada "Iprop_numBlobs" es de tipo primitivo (PrimitiveVisualFeature), se describe como una característica de contexto (∃ hasFeatureFocus ContextBased), de cuantificación (∃ hasActionFunction QuantificationFunction) y que se obtiene de una imagen binaria (∃ hasDataSource BinaryImage) de dos dimensiones (∃ hasDataDimension 2D). Además, indica que la característica pertenece a la imagen (∃ hasDataFocus ImageFocus), en concreto, de las zonas hiperintensas del cerebro (∃ hasDataROI Brain_Hiperintenisty).

En la fila (b) se muestra la variable "Rprop_reg_DistToVent_cmI", que también es una variable primitiva, que describe una característica de distancia. En ella se indica que se obtiene a nivel de objeto a partir de una imagen de distancias a los ventrículos, en concreto, del centro de masas.

Por último, es la fila (c) se muestra una variable relativa denominada "Rprop_rel_Skel2area" donde se indica que es una variable de proporción entre la variable "Rprop_reg_areaSkel" y "Rprop_reg_area", que se calcula a nivel de objeto bidimensional.

4.2.2.2. M-Unet

M-Unet es un algoritmo basado en redes neuronales convolucionales (CNN), una clase de redes neuronales especialmente diseñadas para el reconocimiento y la clasificación

(a)

(b)

(c)

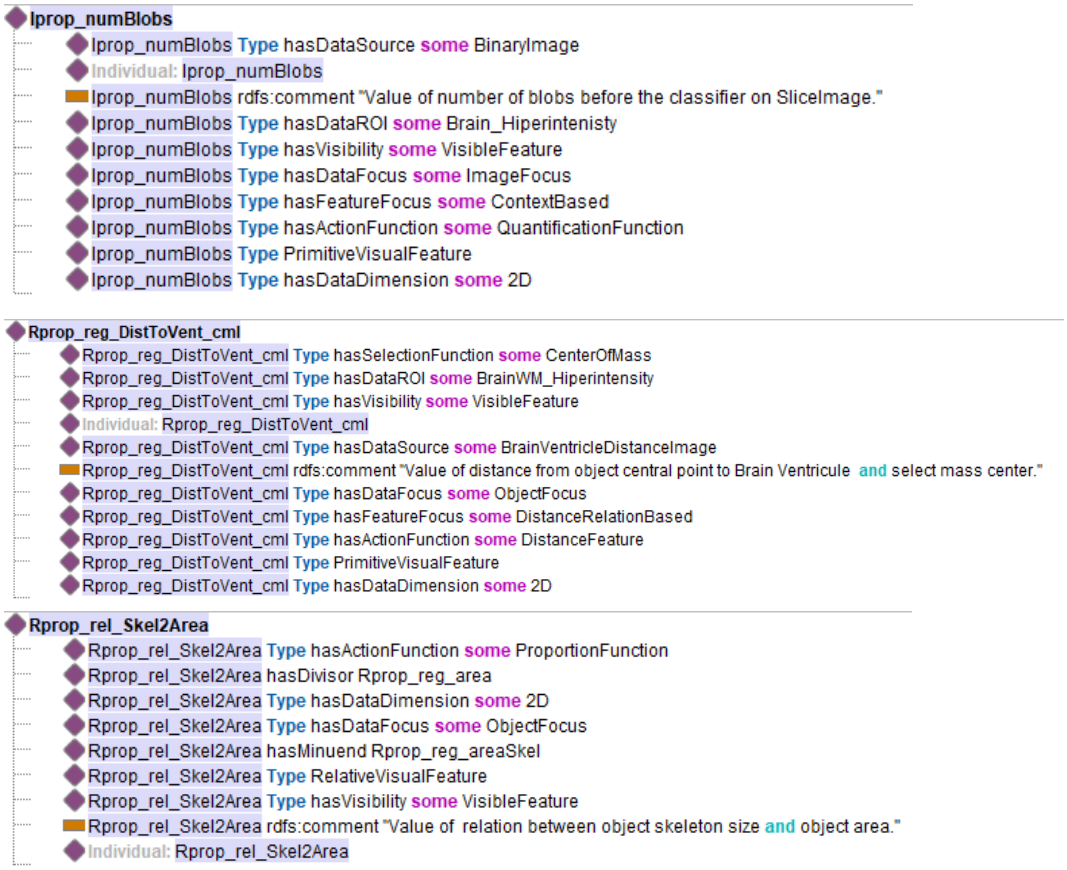


Figura 4.6: Descripción ontológica de algunas características calculadas por AMOS-2D

de imágenes. Utiliza la arquitectura U-Net propuesta por Ronneberger et al. [2015], cuya estructura se muestra en la figura 4.7. Long et al. [2015] publicó un trabajo similar donde combinaba la información semántica con detalles de la imagen. La red se alimenta de una imagen a clasificar y a la salida se obtiene un mapa de segmentación (su segmentación semántica). Consta de una parte descendente, formada por capas de convolución combinadas con otras de reducción ("max pooling"), donde se extraen rasgos de la imagen inicial; y una parte ascendente, compuesta de capas de deconvolución e incremento de resolución ("upsampling"), que se encarga de construir el mapa de segmentación semántica.

En los últimos años, han aparecido variantes de U-Net entre las que se encuentra el algoritmo M-Unet, desarrollado por Duque et al. [2020] para competir en el reto abierto "WMH Segmentation Challenge". M-Unet es un algoritmo de caja negra que ofrece a su salida imágenes de las máscaras probabilísticas de los objetos segmentados. Para obtener una salida binaria se aplica una umbralización (threshold). Cuando se envió al reto abierto, ocupó la posición 17 de 53. Actualmente se encuentra en la posición 20 de 57 participantes, en la última revisión de Diciembre de 2022.

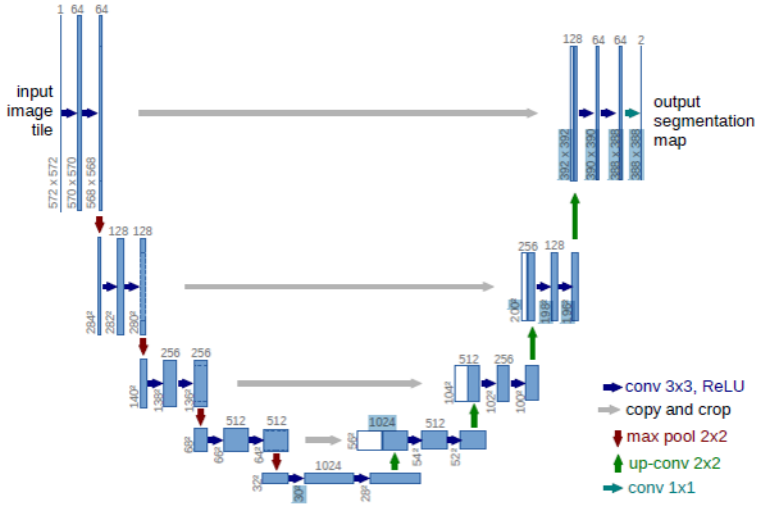


Figura 4.7: Arquitectura U-Net

La estructura de M-Unet se describe en Duque et al. [2019], consta de dos canales de entrada que alimentan a una red neuronal convolucional, uno para la imagen FLAIR y otro para la imagen T1, con dos capas de convolución seguidas de una capa de reducción que se repite 4 veces por cada camino. Utiliza un procesado bidimensional que analiza los datos del volumen slice a slice en la vista axial. Utiliza técnicas de aumento de datos, aplicando transformaciones afines, como rotaciones, cambios en los ejes x e y, zoom y volteo horizontal y vertical. Como función de pérdidas utiliza el coeficiente de similitud Dice a nivel de vóxel, ya que el conjunto de datos está muy desbalanceado porque pocos vóxeles de la imagen pertenecen a lesiones hiperintensas.

4.2.2.3. PGS

El algoritmo PGS es el actual ganador del reto abierto "WMH Segmentation Challenge", ya que se encuentra en la primera posición del ranking[4]. Según sus autores, el algoritmo se basa en la arquitectura U-Net con resaltado multiescalar del foreground (de la segmentación de referencia, que contiene los objetos que se desean resaltar). Este método está diseñado para mejorar la detección de los vóxeles de WMH con efectos de volumen parcial. El efecto de volumen parcial se produce cuando en el volumen asociado a un vóxel hay distintos tejidos, pues el valor del vóxel es proporcional al valor y cantidad de cada tejido, lo que produce valores promediados, alejados de los normales para un determinado tipo de tejido. Este efecto tiene una gran repercusión en la detección de los objetos pequeños y en la delineación de los contornos de los objetos cuyo tamaño es superior a varios vóxeles. La salida PGS se basa en 5 modelos U-Net 2D con la misma

[4] https://wmh.isi.uu.nl/results/

arquitectura pero inicializados aleatoriamente y entrenados con datos aumentados. La segmentación final se obtiene aplicando un threshold de 0.5 a la salida agregada de los 5 modelos.

El algoritmo está disponible en la página del WMH Segmentation Challenge[5], lo cual permite aplicar el algoritmo a nuestras propias imágenes y comparar los resultados. Teniendo en cuenta que la delineación de las hiperintensidades es compleja y subjetiva, compararnos con otro método que ofrece mejores resultados puede ser una manera de encontrar potenciales líneas de mejora. La implementación solo ofrece a la salida una segmentación binaria, por lo que se trata de un SVC de caja negra. Para conocer más información sobre este algoritmo se puede consultar la publicación derivada [Park et al., 2021].

4.2.3. Análisis exploratorio de los objetos segmentados

Antes de aplicar la metodología de evaluación del error en la segmentación, es interesante analizar la distribución de los datos de entrada, es decir, del total de objetos segmentados tanto en la referencia como en las propuestas de los SVC, con el fin de tener una primera visión de los datos y su comportamiento general.

4.2.3.1. Objetos segmentados en las referencias

Un primer análisis de todos los datos de referencia se muestra en la tabla 4.2, donde se describen características de cantidad (#) en número de objetos, volumen (V) en número de vóxeles e intensidad (I) en un rango normalizado [0,1] para cada uno de los dataset. En las columnas se diferencian los cinco dataset analizados: Oslo, Brno, Amsterdam, Singapore y Utrecht, y en filas se muestran sus características. Cada objeto segmentado tiene una descripción individual y pertenece a un caso y a una slice, por lo que se pueden realizar varios estudios agrupados.

En el primer análisis, se estudian todos los objetos por dataset. En "# (O_R)", se presenta el número total de objetos , en "V (O_R) Min-Max" se muestra el volumen del objeto más pequeño y del objeto más grande, respectivamente. En "V (O_R) Mean(Std)" se indica el valor medio del volumen de los objetos y su desviación estándar, y en "V (O_R) Total" el volumen total de todos los objetos en número de vóxeles. En "I (O_R) Min-Max" se muestran los valores de intensidad mínimo y máximo, respectivamente, y en 'I (O_R) Mean(Std)" el valor medio y su desviación estándar.

En el segundo análisis, se estudian los objetos agrupados por caso. En primer lugar, se muestra el número de objetos "$\#_c$ (O_R)" del menor caso y del mayor ("Min-Max") y

[5] https://wmh.isi.uu.nl/results/pgs/

el valor medio y su desviación estándar ("Mean(Std)"). En segundo lugar, se muestra el volumen del caso menor y del mayor en "V_c (O_R) Min-Max" y el volumen medio y su desviación estándar en "V_c (O_R) Mean(Std)". Finalmente, en tercer lugar, se muestra la variación de intensidad, el valor mínimo y máximo en "I_c (O_R) Min-Max" y el valor medio y su desviación estándar ("Mean(Std)").

Por último, en el tercer análisis, se estudian los objetos por slice. En primer lugar, se muestra el número de objetos "$\#_s$ (O_R)" menor y mayor ("Min-Max") en un slice y el valor medio y su desviación estándar ("Mean(Std)"). En segundo lugar, se muestra el menor y el mayor volumen de los objetos en las slices en "V_s (O_R) Min-Max" y el volumen medio y su desviación estándar en "V_s (O_R) Mean(Std)". Finalmente, se muestra la variación de intensidad en los objetos en las diferentes slices. Se muestra el valor mínimo y máximo en "I_s (O_R) Min-Max" y el valor medio y su desviación estándar ("Mean(Std)").

Se observa que, en general, se tiene un número elevado de objetos, un total de 13118, con características de volumen y niveles de intensidad variados, lo que dificulta el desarrollo de una solución automática. Además, dado el gran volumen de los datos, es inabordable la exploración manual de todos los casos, por lo que se necesitan herramientas computacionales que faciliten su evaluación.

Dataset			**Oslo**	**Brno**	**Amsterdam**	**Singapore**	**Utrecht**
Por dataset	# (O_R)	Total	2964	2129	3025	2875	4125
	V (O_R)	Min-Max	1-4764	1- 360	1- 728	1-1173	1-1395
		Mean(Std)	156 (402)	16 (32)	20 (48)	44 (104)	40 (101)
	I (O_R)	Total	150721	33154	61223	127283	66550
		Min-Max	0.18 -0.49	0 -0.59	0.13-0.76	0.08 -0.56	0 -0.48
		Mean(Std)	0.33 (0.06)	0.34 (0.09)	0.39 (0.1)	0.27 (0.07)	0.18 (0.07)
Por caso	$\#_c$(O_R)	Min-Max	0-129	10-384	43-300	33-250	33-526
		Mean(std)	36 (35)	118 (114)	151 (77)	144 (59)	206 (136)
	V_c(O_R)	Min-Max	0-65694	101-10720	433-12381	247-20337	307-27198
		Mean(Std)	5582 (13272)	1842 (2934)	3061 (3213)	6364 (4984)	8328 (8139)
	I_c(O_R)	Min-Max	0.21 -0.4	0.21 -0.44	0.27 -0.46	0.12 -0.36	0.09 -0.29
		Mean(Std)	0.33 (0.04)	0.35 (0.07)	0.39 (0.05)	0.26 (0.06)	0.18 (0.06)
Por slice	$\#_s$(O_R)	Min-Max	1-19	1-23	1-27	1-22	1-35
		Mean(std)	4 (3)	3 (3)	7 (5)	7 (5)	8 (7)
	V_s(O_R)	Min-Max	4-8571	1- 613	1- 1456	1- 2041	1- 2602
		Mean(Std)	602 (1266)	49 (80)	135 (208)	310 (383)	329 (482)
	I_s(O_R)	Min-Max	0.18 -0.44	0.17 -0.53	0.13 -0.66	0.1 -0.45	0.05 -0.43
		Mean(Std)	0.33 (0.05)	0.35 (0.08)	0.39 (0.08)	0.26 (0.07)	0.18 (0.06)

Tabla 4.2: Descripción del conjunto de objetos segmentados de referencia

Para profundizar en la exploración del dataset de referencia, en la figura 4.8 se muestra de forma gráfica la distribución por casos y tamaño de los objetos del dataset de Oslo. En concreto, en la fila de arriba se muestra, a la izquierda, el número de objetos por

caso y, a la derecha, el volumen en vóxeles de los objetos por caso. En la fila de abajo, a la izquierda, se muestra la distribución del número de objetos según su tamaño junto a una ventana ampliada del rango inferior (desde 1 a 100 vóxeles) en el que hay un mayor número de objetos. Por último, a la derecha se muestra la distribución de la intensidad media de los objetos por caso.

Se observa que hay múltiples situaciones, desde casos con ningún objeto en la referencia, por ejemplo, el caso S0058 del dataset de Oslo, hasta casos con más de 500 objetos, por ejemplo, el caso S0050 del dataset de Utrecht. Respecto al tamaño, aunque la mayoría son objetos pequeños, se aprecia que existe una gran variabilidad, con objetos de hasta 4764 vóxeles. Se ha de tener en cuenta que estos objetos grandes dominarán las métricas basadas en píxeles, como el DSC.

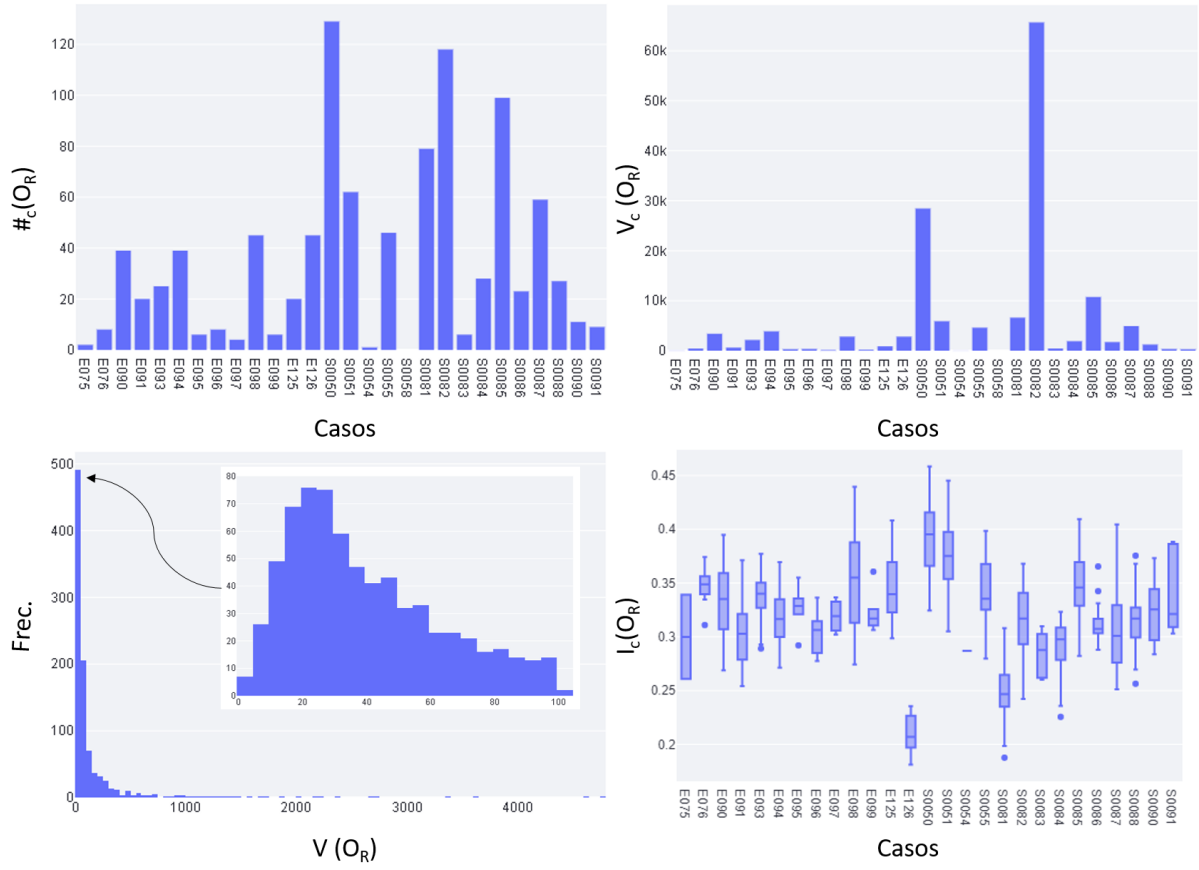


Figura 4.8: Distribución de los objetos de la referencia en el dataset OSLO

4.2.3.2. Objetos segmentados en las propuestas

De forma análoga a la descripción llevada a cabo de los objetos de referencia, se realiza la descripción de los objetos de segmentación propuestos por los algoritmos analizados. El algoritmo AMOS-2D se ha aplicado a dos de los dataset, Oslo y Brno, el algoritmo PGS

se ha aplicado al dataset Brno y el algoritmo M-Unet se ha aplicado a los dataset Brno, Amsterdam, Singapore y Utrecht. Las características generales de las distintas propuestas de segmentación se recogen en la tabla 4.3. Se realizan también tres estudios, agrupando los datos por dataset, por caso y por slices. Los datos que se muestran son los valores mínimo y máximo ("Min-Max"), su valor medio junto a su desviación estándar ("Mean(Std)") y valores totales ("Total") a nivel de objetos con información sobre su cantidad en número de objetos ("# (O_P)"), sobre su volumen en número de vóxeles ("V (O_P)") y sobre su intensidad ("I (O_P)") en un rango normalizado [0,1].

			SVC	**AMOS-2D**	**PGS**	**M-UNET**			
		Dataset	**Oslo**	**Brno**	**Brno**	**Brno**	**Amsterdam**	**Singapore**	**Utrecht**
Por dataset	#(O_P)	Total	737	1147	2873	9375	2836	2953	3915
	V (O_P)	Min-Max	6-2957	1-1267	1-341	1-315	1-736	1-1171	1-1375
		Mean(Std)	192 (345)	34 (61)	16 (28)	8 (18)	21 (49)	41 (99)	41 (102)
		Sum	141689	38494	45705	73544	58747	121752	158822
	I (O_P)	Min-Max	0.19-0.46	0.18-0.55	0.08-0.68	0.08-0.89	0.18-0.84	0.08-0.7	0.05-0.45
		Mean(Std)	0.33 (0.05)	0.36 (0.08)	0.25 (0.08)	0.29 (0.1)	0.4 (0.1)	0.28 (0.07)	0.19 (0.07)
Por caso	$\#_c$(O_P)	Min-Max	1-93	11-220	59-463	150-1051	42-271	25-345	38-497
		Mean(std)	26 (24)	64 (55)	180 (117)	521 (258)	142 (71)	148 (76)	196 (130)
	V_c(O_P)	Min-Max	42-49152	185-6846	392-12732	703-11701	407-12488	202-20115	276-26865
		Mean(Std)	5060 (10165)	2139 (2158)	2857 (3339)	4086 (3094)	2937 (3177)	6088 (4848)	7941 (8023)
	I_c(O_P)	Min-Max	0.21-0.41	0.22-0.46	0.14-0.35	0.16-0.41	0.28-0.48	0.13-0.37	0.11-0.3
		Mean(Std)	0.32 (0.04)	0.37 (0.08)	0.25 (0.06)	0.27 (0.06)	0.4 (0.05)	0.26 (0.06)	0.19 (0.05)
Por slice	$\#_s$(O_P)	Min-Max	1-13	1-11	1-19	1-28	1-27	1-32	1-42
		Mean(std)	3 (2)	2 (2)	3 (3)	6 (5)	6 (5)	7 (5)	8 (7)
	V_s(O_P)	Min-Max	6-5925	2-1267	1-513	1-519	1-1445	1-2042	1-2554
		Mean(Std)	529 (954)	77 (107)	54 (79)	49 (70)	127 (204)	284 (365)	317 (473)
	I_s(O_P)	Min-Max	0.19-0.44	0.18-0.55	0.09-0.68	0.09-0.81	0.19-0.66	0.1-0.7	0.07-0.4
		Mean(Std)	0.32 (0.05)	0.37 (0.08)	0.26 (0.08)	0.27 (0.09)	0.4 (0.08)	0.27 (0.07)	0.19 (0.06)

Tabla 4.3: Descripción del conjunto de objetos segmentados de propuesta

Se observa una gran variedad de comportamientos, desde casos con un sólo objeto segmentado hasta más de 1000, y objetos desde 1 vóxel a casi 2997 vóxeles. En la propuesta del algoritmo AMOS-2D sobre el dataset de Oslo, se observa una situación destacable: el tamaño mínimo de los objetos es de 6 vóxeles. Esto puede ser debido a que en el algoritmo hayan aplicado un filtro de tamaño 6 para mejorar los resultados globales de evaluación, por lo que habrá que tenerlo en cuenta a la hora de interpretar los resultados. Otra situación interesante se da en el dataset de "Brno_PGS", ya que sólo se tienen datos

de segmentación de 16 casos de un total de 18, por lo que faltan dos casos, y también habrá que tenerlo en cuenta en los análisis de evaluación.

Si comparamos visualmente los valores de las tablas 4.2 y 4.3 se aprecia, por ejemplo, el mejor comportamiento del algoritmo PGS frente al algoritmo M-Unet sobre el dataset Brno, ya que el número de objetos segmentados es más parecido a los de la referencia, aunque no tenemos información de si los objetos están ubicados en la misma posición de la imagen 3D. Por otro lado, el volumen máximo por objeto es, en general, bastante diferente entre la referencia y la propuesta, especialmente en el dataset Oslo (2997 en la propuesta frente a 4764 en la referencia). Se aprecia incluso esa diferencia entre las propuestas obtenidas con los diferentes algoritmos sobre el mismo dataset. Así, por ejemplo, el volumen máximo por objeto aplicando el algoritmo AMOS-2D al dataset Brno es bastante diferente del obtenido con el algoritmo PGS, 1267 y 341, respectivamente.

En resumen, se aprecian diferencias significativas entre los resultados de los distintos algoritmos y, tras realizar un rápido análisis cualitativo, se puede concluir que existen errores. Ahora, es necesario realizar una evaluación cuantitativa y exhaustiva de este error para poder mejorar el sistema.

4.3. Resultados

En esta sección, se muestran algunos análisis realizados con la metodología AMOSE^2 sobre los dataset y algoritmos descritos anteriormente. Se pretende con ello ejemplificar las dos formas de uso de la metodología, el análisis individual del error y el análisis comparado. Estos análisis se han realizado de manera independiente para los distintos conjuntos de datos porque cada conjunto de imágenes tiene características diferentes, pues han sido adquiridos con máquinas y protocolos distintos. Además, siguiendo la hipótesis planteada en esta tesis de que cada tipo de error tiene un origen y naturaleza diferentes, también se han realizado análisis independientes para los distintos tipos de error de los objetos de agrupación.

4.3.1. Análisis individual del error

El uso original de la metodología AMOSE^2 es la comparación de una segmentación propuesta por un SVC respecto a una referencia, es decir, es una comparación por pares. Para ejemplificar esta forma de uso, se presenta en las siguientes subsecciones el estudio de las segmentaciones obtenidas usando el dataset de Oslo con el algoritmo AMOS-2D, que denominaremos "Oslo_AMOS-2D", por ser el más completo. En este estudio, se tiene acceso a los procesos internos del algoritmo, lo que lo convierte en un SVC de caja gris. Este análisis también se ha aplicado a los otros algoritmos anteriormente comentados

(PGS y M-Unet) y a diferentes conjuntos de datos (Brno, Amsterdam, Singapore y Utrecht), cuyos resultados se recogen en el anexo C.

4.3.1.1. Descripción del estudio

En este estudio, "Oslo_AMOS-2D" , se dispone de las imágenes originales, FLAIR y T1, de 28 pacientes. De las imágenes FLAIR, se dispone de la imagen original y la imagen con el sesgo corregido, esta última será la que se utilice en el proceso de descripción de objetos. En cuanto al algoritmo, se ha utilizado AMOS-2D mejorado con salida de características internas del proceso de segmentación y una descripción semántica de dichas variables con la ontología "AMOS-2D_VOF".

El estudio se realiza mediante el prototipo AMOSE^2 cuya descripción se recoge en el anexo A y sigue la estructura de módulos de la metodología propuesta. En primer lugar, se presentan los resultados obtenidos con el MDE, donde se comparan los objetos segmentados de la propuesta y la referencia y se crean los objetos de agrupación para su descripción, clasificación y análisis. A continuación, se muestran los experimentos realizados para detectar agrupaciones de error (clústeres) y errores aislados o atípicos (outlier) el MCE. Estas operaciones se llevan a cabo mediante la herramienta "AMOSE^2 analysis". A su vez, se muestran los hallazgos de error relevantes mediante las facilidades de la herramienta "AMOSE^2 web report" que implementa el MEE. Por último, se describen las ontologías utilizadas en el MDO y el uso que se hace de ellas.

4.3.1.2. Módulo de descripción del error (MDE)

Este módulo tiene como objetivo clasificar los tipos de error detectados tras comparar la segmentación propuesta con la de referencia. Como se describió en el capítulo anterior, el MDE consta de tres subtareas o etapas: generación, descripción y clasificación.

En la etapa de generación, se crean los objetos de agrupación a partir de los objetos segmentados de la referencia y de la propuesta. En este estudio se tienen 964 objetos de la referencia y 737 objetos de la propuesta. Con ellos se generan 1019 objetos de agrupación.

En la etapa de descripción, se describen los objetos de agrupación mediante la interacción con el MDO para crear un vector de múltiples características. Éste contiene la información para aplicar la función correcta en las situaciones de las agrupaciones con solape, ya que, en esos casos, existen dos descripciones posibles del objeto de agrupación, una procedente de las características de los objetos de la propuesta y otra procedente de los objetos de la referencia. Es en el MDO en donde se selecciona el origen de los datos. Por defecto, se utiliza la información de la propuesta, ya que permite conocer el comportamiento del sistema de segmentación.

Además, en el caso de solape múltiple, donde hay más de un objeto de la referencia y/o de la propuesta en el objeto de agrupación, se aplican funciones de agrupación en los descriptores para trabajar con un valor único preservando su significado global. Por ejemplo, en la característica de área se utiliza la función suma, en la característica de intensidad, dependiendo de la descripción, se emplea la función media, máximo o mínimo y, en el caso de características relativas al objeto, se utilizan funciones ponderadas por el área de los objetos.

Tras esta etapa de descripción, se obtiene el vector de características de cada objeto de agrupación $F_{O_{R.P}}$ con un total de 186 valores, que engloban a las características visuo-espaciales de la agrupación (44 características) y a las características internas de sus objetos componentes, ${F^I}_{O_R}$ y ${F^I}_{O_P}$ (142 características). En el anexo B.1 se muestra el listado completo de las características externas e internas utilizadas en este trabajo junto a su identificador y una descripción breve.

Por último, en este módulo se clasifican los objetos de agrupación $O_{R.P}$ con la información del vector $F_{O_{R.P}}$. Se utiliza como primera función (f1) para clasificar los objetos de error, el porcentaje de solape definido mediante el coeficiente Dice en vóxeles (DSCv), como se comentó en la sección 3.2.3. Concretamente, se clasifica como clase "Success" si DSC_V es mayor o igual a un valor umbral $Th_{Success}$ y como clase "Error" en caso contrario. Se utiliza un valor de 0.75, para ejemplificar una fase avanzada de la evaluación, ya que según Zijdenbos et al. [1994] y Zou et al. [2004] un valor umbral de 0.7 se considera una segmentación correcta. Como se observa en la la figura 4.9, la distribución de este ejemplo muestra que dicho umbral separa los objetos detectados en una proporción suficiente para tener datos de ambas clases y poder extraer patrones de error relevantes en cuanto a error en delineación, pero en una fase inicial del diseño de un SVC, se deberá comenzar por umbrales más bajos para abordar primero los errores de detección y después proceder a analizar los errores de delineación.

Con el valor umbral de 0.75, de los 1019 objetos de agrupación, se clasifican como "Success' a 412 objetos, y como "Error" a 607. A continuación, se subdividen los objetos de error en tres subclases: "Extra", "Miss" e "Imperfect" mediante una segunda función (f2). En este caso, se utiliza la composición del objeto de agrupación $O_{R.P}$ y se considera objeto de la clase "Miss" si es de tipo 1 - 0, de la clase "Extra" si es de tipo 0 - 1 y de la clase "Imperfect" en el resto de casos. Con estos criterios, se tienen 297 objetos "Extra", 84 "Miss" y 226 "Imperfect". Finalmente, se emplea una tercera función (f3) para clasificar los objetos de agrupación "Imperfect" en la clase "Solo contacto", si no existen vóxeles coincidentes, o "Con solape", cuando sí existen. En este caso, se obtienen 0 objetos "Solo contacto" y 226 "Con solape". En la figura 4.10 se muestra esquemáticamente las distintas clasificaciones realizadas en este ejemplo.

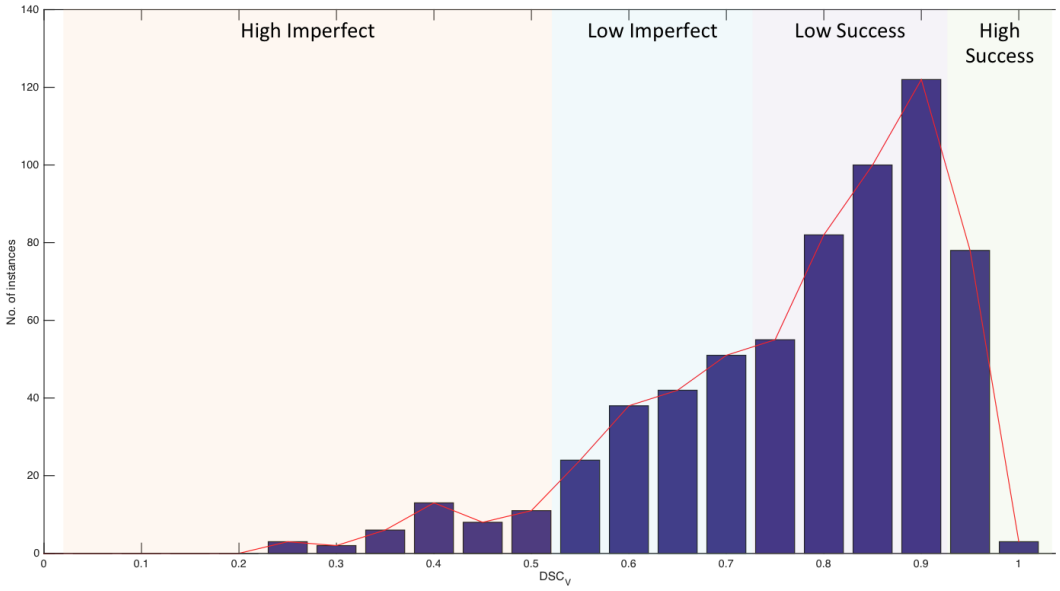


Figura 4.9: Distribución del coeficiente Dice a nivel vóxel (DSC_V) de los objetos $O_{R.P}$ con solape

Los resultados de cada caso del dataset de Oslo obtenidos tras aplicar el MDE se muestran en la tabla 4.4. La tabla se ha dividido en tres partes para presentar la información a nivel de objetos, a nivel de vóxeles y con medidas estadísticas de similitud.

En la primera parte, se presentan el número de objetos en la referencia "$\#(O_R)$" y en la propuesta "$\#(O_P)$"; el número de objetos de agrupación "$\#(O_{R.P})$" y el número de objetos de detección erronea "$\#(O_{Miss})$" y "$\#(O_{Extra})$".

En la segunda parte, se muestran los resultados a nivel de vóxeles: el número de vóxeles en la referencia "$V(O_R)$" y en la propuesta "$V(O_P)$"; el número de vóxeles de los objetos de agrupación $V(O_{R.P})$; el número de vóxeles de delineación errónea, infra-segmentados "$V(O_{UnderSeg})$" y sobre-segmentados "$V(O_{OverSeg})$" y el número de vóxeles de detección erróneos, "$V(O_{Miss})$" y "$V(O_{Extra})$".

Finalmente, en la tercera parte de la tabla, se presentan la métrica DSC a nivel de vóxeles "DSC_V" y la métrica "F1-score" a nivel de objetos. Al final de la tabla, en negrita, se muestran los valores globales del dataset realizando la media de los valores de los casos.

En una primera valoración de las métricas mostradas en la tabla 4.4, vemos que el comportamiento global es bueno, con un DSC igual a 74.3 % y F1-score igual a 71,2 %, pero mejorable. Para lograr esta mejora es necesario profundizar en el análisis del error e intentar conocer dónde, cómo y por qué se producen.

Al llegar a este punto, tenemos dos opciones: (1) explorar los objetos de agrupación clasificados por tipo de error y descritos con múltiples características para descubrir de forma manual comportamientos de error relevantes mediante las facilidades de la

ID_case	Núm. Objeto					Núm. Vóxel							DSC_V	F1-score
	$\#O_R$	$\#O_P$	$\#O_{R.P}$	$\#O_{Miss}$	$\#O_{Extra}$	$V(O_R)$	$V(O_P)$	$V(O_{R.P})$	$V(O_{UnderSeg})$	$V(O_{OverSeg})$	$V(O_{Miss})$	$V(O_{Extra})$		
S0058	0	1	1	0	1	0	42	42	0	0	0	42		
S0054	1	3	3	0	2	28	255	255	0	18	0	209	75.7	50.0
E075	2	2	3	1	1	40	52	67	0	12	15	15	80.7	50.0
E097	4	18	18	0	14	148	1260	1268	8	163	0	957	62.1	36.4
E095	6	8	10	2	4	288	834	961	7	145	120	528	67.9	57.1
E099	6	8	8	0	2	192	1028	1028	0	607	0	229	38.8	85.7
S0083	6	6	8	2	2	422	630	721	0	221	91	78	75.0	66.7
E076	8	5	8	3	0	414	419	537	53	123	65	0	77.1	76.9
E096	8	15	15	0	7	328	1043	1044	1	354	0	362	64.8	69.6
S0091	9	6	9	3	2	279	600	655	1	300	54	76	59.8	61.5
S0090	11	11	13	2	3	324	581	677	11	248	85	105	63.8	76.2
E091	20	14	23	9	4	612	888	1095	23	220	184	263	76.9	60.6
E125	20	14	24	10	4	866	1294	1659	8	553	357	240	64.1	58.8
S0086	23	22	25	3	2	1720	2878	3101	33	1164	190	217	71.4	88.9
E093	25	19	26	7	2	2145	2960	3215	34	672	221	398	84.3	79.1
S0088	27	12	25	13	0	1232	984	1649	245	417	420	0	63.1	64.9
S0084	28	20	29	9	4	1896	1887	2569	272	394	410	279	78.5	71.1
E090	39	21	41	20	4	3386	3664	4735	247	745	824	604	82.4	58.6
E094	39	33	40	8	2	3870	5189	5734	184	1669	361	195	78.2	85.7
E098	45	34	45	11	4	2808	4067	4587	218	1101	302	678	77.6	80.0
E126	45	39	48	9	3	2807	2679	3525	294	624	552	94	81.0	85.7
S0055	46	39	50	12	4	4604	5330	5953	234	1234	389	115	84.4	81.0
S0087	59	42	58	16	0	4920	4371	5650	807	730	472	0	82.6	84.0
S0051	62	46	61	16	5	5894	6357	7848	765	1849	726	105	77.1	79.2
S0081	79	60	81	21	2	6614	6788	8363	866	1732	709	17	79.5	83.5
S0085	99	79	104	30	5	10742	8265	11721	2441	826	1015	153	81.7	79.8
S0082	118	67	118	56	0	65694	49152	67060	15660	1366	2248	0	84.9	68.9
S0050	129	93	125	34	1	28448	28192	31290	1724	2829	1374	13	91.8	83.7
28	**964**	**737**	**1019**	**297**	**84**	**150721**	**141689**	**177009**	**24136**	**20316**	**11184**	**5972**	**74.3**	**71.2**

Tabla 4.4: Información detallada de la evaluación por caso del conjunto de datos “Oslo”

Nota: El coeficiente Dice (DSC) se calcula a nivel vóxel para los objetos de contacto. El valor-F1 (F1-Score) se calcula a nivel objeto.

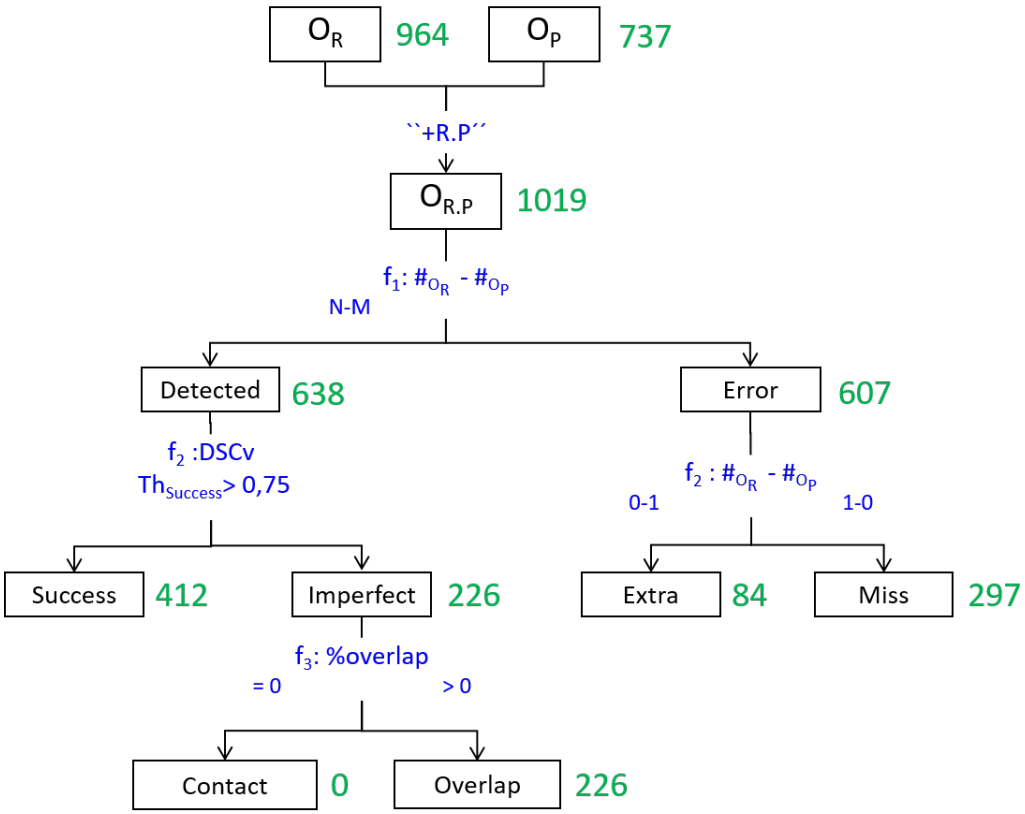


Figura 4.10: Jerarquía de los objetos de agrupación, origen y tipo de error

aplicación "AMOSE² web report" o (2) utilizar el módulo MCE para analizar los conjuntos de datos en busca de patrones de error relevantes: agrupaciones/ bolsas de error mediante técnicas de clustering o anomalías/errores aislados mediante técnicas de detección de outlier. Comenzaremos por las últimas, ya que son las que aportan novedad y valor a este trabajo de tesis doctoral y dejaremos para la sección siguiente la exploración de comportamientos de forma manual con el MEE.

4.3.1.3. Módulo de caracterización del error (MCE)

Este módulo tiene como objetivo extraer patrones de comportamiento erróneo a partir del análisis de los vectores de características de los diferentes tipos de error proporcionados por el MDE. Como se describió en el capítulo anterior, el MCE consta de dos subtareas o etapas: la preparación de los conjuntos de datos (transformación y/o reducción de dimensionalidad) y análisis del error. Para este caso, donde se ejemplifica la forma de uso individual, se tienen dos sub-análisis, el análisis de error agrupado y el análisis de error aislado.

Preparación de los dataset El algoritmo CLIQUE, utilizado para la búsqueda de patrones de agrupación de error, tiene problemas a la hora de analizar conjuntos de datos de alta dimensionalidad (más de 16 atributos) por lo que necesita una reducción de características para que sus procesos converjan a una solución. Además, dado que utiliza valores con sólo dos decimales, también necesita una transformación de los datos, que se normalizan al rango [0, 100].

Para la reducción de características se han realizado varios análisis para ayudar al experto a tomar una decisión, que se explican a continuación. Se realiza en primer lugar un análisis de la relevancia estadística de cada una de las variables respecto a la clase del objeto de agrupación. Y en segundo lugar, se calcula la profundidad semántica de las variables que describen a los objetos de agrupación. Por ahora, con toda esta información, la decisión es tomada por la persona encargada de realizar el estudio de evaluación de forma manual.

Relevancia estadística La relevancia estadística de las variables respecto a la clase de error se analiza con el software WEKA[6], un entorno para el análisis de datos de la Universidad de Waikato, escrito en Java y gratuito (GNU GPL[7]). Antes de aplicar diferentes algoritmos de exploración de los datos, se preparan mediante la eliminación de los atributos identificativos y de posición, quedando reducidos a 152 variables numéricas, salvo la última que corresponde al tipo de objeto de agrupación que es de tipo enumerado (categórico). A continuación, se aplica un filtro para detectar atributos constantes (weka.filters.unsupervised.attribute.RemoveUseless -M 99.0) con lo que se consigue eliminar 3 atributos invariantes.

Una vez preparado el vector de características se analizan diferentes algoritmos para estudiar su relevancia semántica, cuyos resultados se muestran en la tabla 4.5. Se utilizan dos algoritmos de clasificación, uno basado en árboles de decisión, C4.5 [Quinlan, 1993] y otro, basado en las reglas difusas, FURIA [Hühn and Hüllermeier, 2009], para conocer las variables más influyentes respecto a la clase de salida, que en este caso es el tipo de error del objeto de agrupación. También se han aplicado cuatro métodos de selección de atributos: (1) basado en la correlación entre atributos con "weka.attributeSelection.CorrelationAttributeEval", (2) basado en la ganancia de información (o entropía) entre atributos con "weka.attributeSelection.InfoGainAttributeEval", (3) basado en métodos de envoltura junto a un árbol de decisión como clasificador con búsqueda hacia delante y (4) el mismo con búsqueda hacia atrás, cuya configuración en ambos es "weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R

[6] https://www.cs.waikato.ac.nz/ml/weka/
[7] http://www.gnu.org/licenses/gpl-3.0.html

1 -E DEFAULT -- -C 0.25 -M 20" y ofrecen a su salida un listado ordenado de los atributos más influyentes respecto a la clase de salida.

Esquema	**Selección /ranking de atributos**
weka.classifiers.trees.J48 -C 0.25 -M 20	3 5 8 9 10 21 70 90 102 135
weka.classifiers.rules.FURIA -F 3 -N 20.0 -O 2 -S 1 -p 0 -s 0	1 2 3 4 5 7 8 9 12 20 21 23 24 25 27 28 30 32 60 69 71 73 80 81 86 88 90 98 100 101 103 113 115 122 123 131 135 138 139 141 144 145
weka.attributeSelection.CorrelationAttributeEval	8 26 24 9 73 27 28 70 74 72 137 135 123 3 122 25 143 142 32 115 101 106 93 116 60 54 1 57 51 36 42 29 48 63 66 45 39 33 138 2
weka.attributeSelection.InfoGainAttributeEval	9 8 26 24 27 28 73 70 30 50 65 68 47 74 59 53 56 44 62 35 41 38 72 137 20 22 135 3 140 31 49 64 67 46 136 1 52 58 43 37 55 61 40 34 115 123 32 122
weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -E DEFAULT -- -C 0.25 -M 20 (forward)	3 8 9 38 69 87
weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -E DEFAULT -- -C 0.25 -M 20 (backward)	1 3 4 6 7 8 9 13 21 22 25 37 42 43 70 73 75 85 91 92 93 94 95 116 120 122 123 125 130 131 132 134 135 139

Tabla 4.5: Estudio de la influencia de las variables del conjunto de datos Oslo

Por ejemplo, en el primer algoritmo, el que se basa en árboles de decisión, se obtiene una lista de las 10 variables más relevantes de las 150 variables analizadas. En la figura 4.11 se muestran los primeros niveles del árbol de decisión obtenido, en el que se observa que la variable número 3 ("Iprop_stdWM") es la más significativa.

Figura 4.11: Relevancia estadística de las variables internas de AMOS-2D mediante árbol de decisión C-4.5

Relevancia semántica Respecto a la información semántica de las características, se ha utilizado la ontología "AMOS-2D_VOF" para calcular la similitud semántica y

la profundidad semántica de los atributos. Con estas medidas el experto puede realizar una reducción de características seleccionando, dentro del conjunto de características, aquellas que cumplan un cierto criterio. Por ejemplo, si tenemos varias características con una descripción semántica similar, se puede seleccionar aquella con los atributos más específicos porque se supone que es más precisa.

Utilizando la definición de Wu and Palmer [1994], que se describe en la sección 2.1.1.2 es posible analizar la similitud semántica navegando por la ontología, esto es, a partir de su estructura jerárquica y las relaciones entre sus términos. La matriz de similitud semántica entre las características definidas en la ontología se muestra en la figura 4.12.

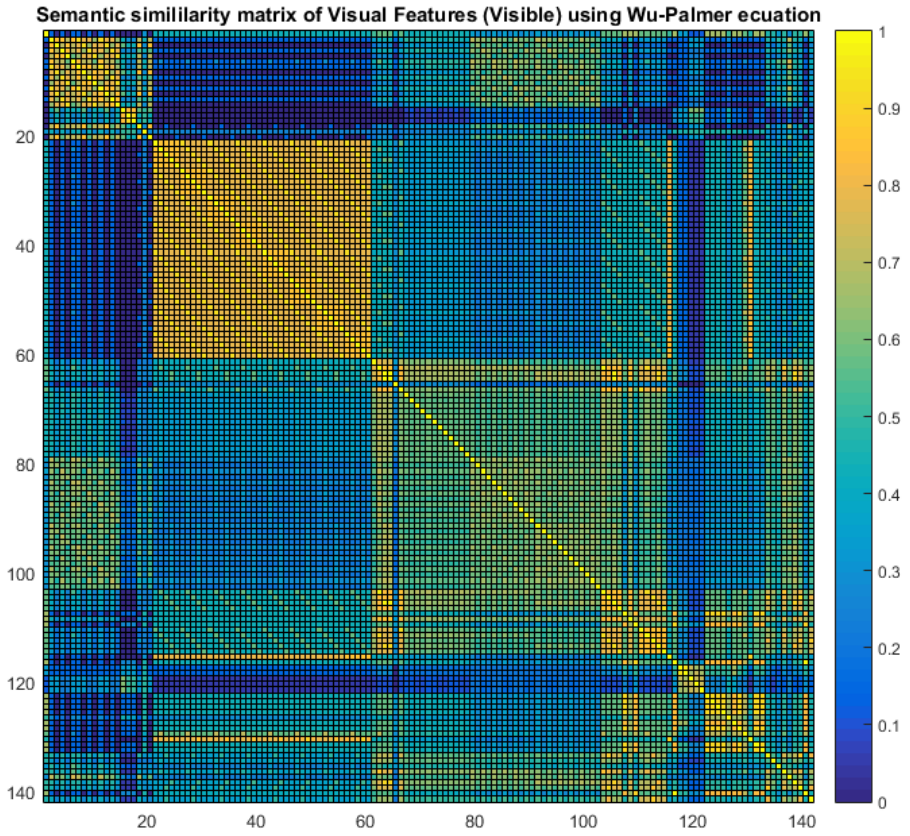


Figura 4.12: Similitud semántica de una las variables internas de AMOS-2D

Se observa que existen variables con una alta similitud semántica (valores cercanos a uno, en color amarillo) y otras con baja similitud (valores cercanos a cero, en color azul). Al analizar en detalle el listado de variables internas (en el anexo B.1 se identifican y describen todas las variables utilizadas), se observa que la alta similitud semántica entre las variables 21 a 60, se corresponde con variables de distancia inter-objetos. También se observa que hay una baja similitud semántica entre las variables 2-20 respecto a las variables 21-60. Estas variables describen propiedades de la imagen bidimensional donde se localiza el objeto hiperintenso. Por ejemplo, la mayor similitud semántica se da entre las

variables "Rprop_reg_area" y "Rprop_rel_Skel2Area" con un valor de 0.99. La primera es una variable interna a la región que indica su área y la segunda una variable de proporción, donde se relaciona el área del esqueleto con el área de la región.

Reducción de la dimensionalidad El resultado de aplicar los criterios anteriores con el objetivo de reducir la dimensionalidad del dataset no ofrece resultados concluyentes sobre la mejor elección de variables, ya que son características muy heterogéneas y reflejan múltiples propiedades, por lo que se generan varias propuestas para su reducción. En la tabla 4.6 se muestran tres listados de selección de variables, que han sido utilizadas en los experimentos realizados en este trabajo. En todos ellos se utilizan los atributos 1-10, aunque no se muestran en los listados, ya que son las características externas de los objetos de agrupación que se calculan por defecto en el MCE.

Nombre	Identificador de atributo	Núm. atributos
SelAtt1	26, 33, 36, 39, 42, 45, 72, 73, 74, 75, 79, 84, 94, 99, 117, 119, 120, 123, 124, 130, 131, 132, 134, 136, 137, 138, 140, 141, 142, 143, 144, 145, 146, 147	44
SelAtt2	17, 18, 20, 22, 24, 26, 32, 33, 36, 38, 39, 42, 45, 51, 72, 78, 79, 83, 93, 94, 98, 119, 121, 123, 127, 128, 130, 131, 134, 135, 136, 137, 138, 140, 141, 142, 143, 144, 145, 146, 147	51
SelAtt3	11, 12, 24, 26, 30, 36, 37, 38, 59, 62, 68, 71, 74, 76, 77, 80, 121, 128, 129, 141, 142, 143, 146	33

Tabla 4.6: Listado de atributos seleccionados

Seguidamente, en la etapa de análisis del error, se pueden realizar dos tipos de análisis sobre los objetos de agrupación: 1) un análisis de errores agrupados y 2) un análisis de errores aislados o anomalías. Cada uno de estos análisis se realiza separando los dataset por tipo de error.

Análisis de errores agrupados El objetivo de este proceso es encontrar patrones de error mediante la búsqueda de agrupaciones de objetos en el espacio de características con técnicas de aprendizaje no supervisado de IA. Para ello, utilizaremos el algoritmo CLIQUE, un algoritmo explicable de clustering que identifica zonas densas en espacios n-dimensionales. Bajo el principio de reutilización, se utiliza la implementación realizada en el software ELKI[8], un software de minería de datos, gratuito y escrito en Java. El algoritmo CLIQUE tiene dos parámetros, el número ξ de intervalos en cada dimensión y la densidad mínima τ del cluster. Su funcionamiento consta de tres pasos: primero, se identifican los subespacios que contienen regiones (hipercubos) densos; a continuación, se identifican los clusters; y finalmente, se obtiene la descripción mínima de las agrupaciones. Este último

[8] https://elki-project.github.io/

paso no había sido aún implementado en ELKI, por lo que en esta tesis se han desarrollado métodos que jerarquizan todos los cluster detectados y se identifican las agrupaciones de mayor dimensionalidad, denominándolos clústeres maximales ("MaximalCluster").

Por ejemplo, se explica en detalle los resultados para el dataset de tipo Miss reducido siguiendo el listado "SelAtt3" y con una configuración de CLIQUE de 20 divisiones y una densidad mínima de 0.1. El algoritmo detecta 95 clústeres, 31 de una dimensión, 42 de dos dimensiones, 19 de tres dimensiones y 4 de cuatro dimensiones, ya que como se comentó en el capítulo anterior este algoritmo encuentra múltiples descripciones de los mismos datos en múltiples subespacios. Su relación jerárquica se muestra en la figura 4.13, donde la dimensión del clúster se aprecia en el número de niveles de la jerarquía.

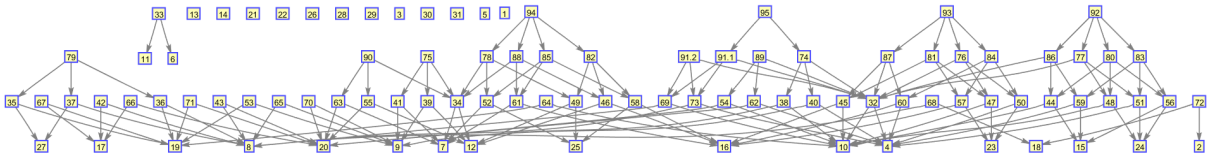


Figura 4.13: Jerarquía de clúster del dataset Miss con SelAtt3 y CLIQUE 20 0.10

De todos ellos, son clústeres maximales 33 con una distribución de 12 de una dimensión, 12 de dos dimensiones, 5 de tres dimensiones y 4 de 4 dimensiones. Dado que la representación no aporta mucha información, se ha creado un mapa de calor para representar los clústeres y marcar sus atributos descriptivos. En la figura 4.14 se muestran en cada fila los clústeres maximales y en las columnas marcado en azul oscuro las variables que describen el clúster. En el eje Y en vez de su identificador se indica el tamaño del clúster, ya que aporta más información conocer el número de objetos que se engloban en la agrupación.

Para identificar agrupaciones de tipo Miss que sean diferentes a los objetos de agrupación de tipo Success se define una función para seleccionar los casos relevantes. En este caso se ha definido la tasa de exito-error (ESR) con un valor de 1.5. Las agrupaciones que superen este umbral se consideran agrupaciones de error relevantes y son los hallazgos que se presentan a los expertos para conocer nuevos patrones de error en los datos. En la figura anterior los clústeres relevantes se marcan en rojo, como se muestra en la figura 4.15.

Los clústeres maximales relevantes por tipo de error se pueden explorar con la aplicación "AMOSE^2 web report", en concreto en la pestaña "Cluster/Isolated error". Por un lado, se puede ver con los mapas de color los clústeres relevantes y sus atributos activos y, por otro lado, seleccionar un clúster concreto y conocer su descripción, ya sea mediante una representación gráfica en forma de radar o mediante una descripción textual, un ejemplo se muestra en la figura 4.16.

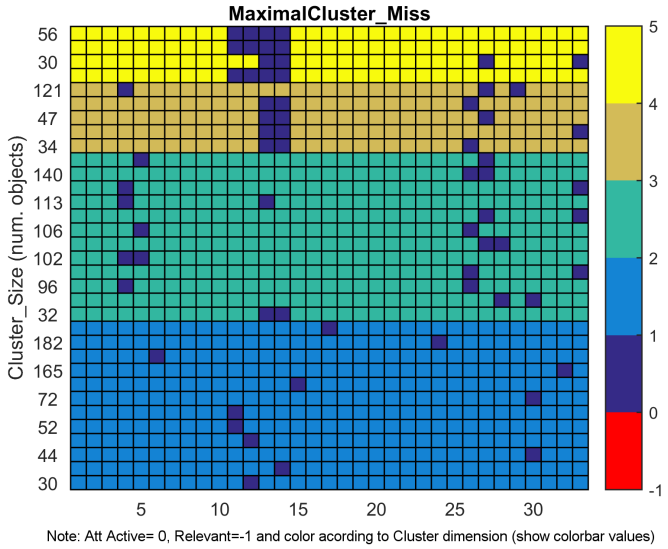


Figura 4.14: Clústeres maximales del dataset Miss con SelAtt3 y CLIQUE 20 0.10

El inconveniente de este algoritmo es que realiza un división fija del espacio de datos por lo que hay que configurar diferentes valores de división y diferentes valores de densidad para descubrir distintos patrones de comportamiento erróneos relevantes mediante su exploración en el MEE.

Análisis de errores aislados En este análisis el objetivo es descubrir comportamientos de error aislados, es decir, anomalías, en dataset multidimensionales con datos heterogéneos. Es uno de los procesos más complejos dada la dificultad que entraña definir que es un dato anómalo. En la tarea de evaluar un SVC con el objetivo de conocer comportamientos de error, utilizar este método puede ayudar a detectar errores singulares, ayudar en la descripción de dichos comportamientos y facilitar la tarea de refinamiento de sistemas.

Siguiendo la descripción de la metodología AMOSE^2 la herramienta "AMOSE^2 analysis" ha implementado dos formas de búsqueda de anomalías: el análisis unidimensional y el análisis multidimensional. Para el análisis unidimensional se ha implementado el método del rango intercuartil (IQR) y para el análisis multidimensional se ha reutilizado un método de detección de outlier n-dimensional del paquete OutlierO3[9].

Análisis unidimensional En este estudio, "Oslo_AMOS-2D", se ha realizado un análisis de detección de outlier unidimensionales con IQR y k=3, y se ha detectado múltiples situaciones, que se pueden explorar con el MME. Por ejemplo, se ha detectado que existe un valor anómalo en la variable "Rprop_reg_DistToVent_maxI', como se

[9] https://cran.r-project.org/web/packages/OutliersO3/OutliersO3.pdf

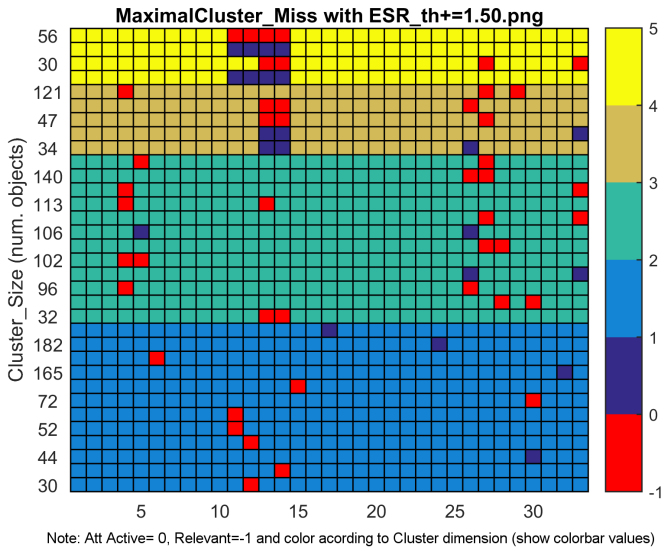


Figura 4.15: Clústeres maximales relevantes (ESR>1.5) del dataset Miss con SelAtt3 y CLIQUE 20 0.10

muestra en la figura 4.17. Este valor está muy alejado del resto, los cuales están comprendidos entre 0 y 100. Al investigar sobre esta variable, dicho valor elevado es un valor por defecto que se fija cuando no hay ventrículo. El problema de su presencia es que provoca un funcionamiento incorrecto en los algoritmos que se basan en la división del espacio de datos, por lo que para reducir su efecto una opción sería trasladar dicho valor a uno más cercano al resto, pero diferenciable, como por ejemplo al valor -20.

Análisis múltidimensional Para el análisis multidimensional se utilizan varios métodos de detección de anomalías ("outlier") implementados en el paquete OutlierO3 de R, un lenguaje de programación y un entorno computacional dedicado a la estadística.

La implementación realizada en "AMOSE^2 analysis" ejecuta tres métodos de detección de anomalías en espacios de datos multidimensionales, en concreto, HDo (HDoutliers) del paquete "HDoutlier", adjOut (adjOutlyingness) y MDC (covMcd) de "robustbase") y se seleccionan aquellos objetos de agrupación que se consideran outlier en múltiples dimensiones y que cumplen el criterio de voto por mayoría. A estos outliers los denominaremos outlierMD. Por tanto, al finalizar el análisis se tiene un conjunto de outlierMD que se describe con un conjunto de variables y sus valores. Además, los outlierMD pueden estar descritos por múltiples combinaciones de variables. La aplicación "AMOSE^2 web report" en la pestaña "Cluster/Isolated error" permite explorar el listado de outlierMD para una configuración dada y ver, mediante facilidades visuales o descripciones textuales, las características del mismo.

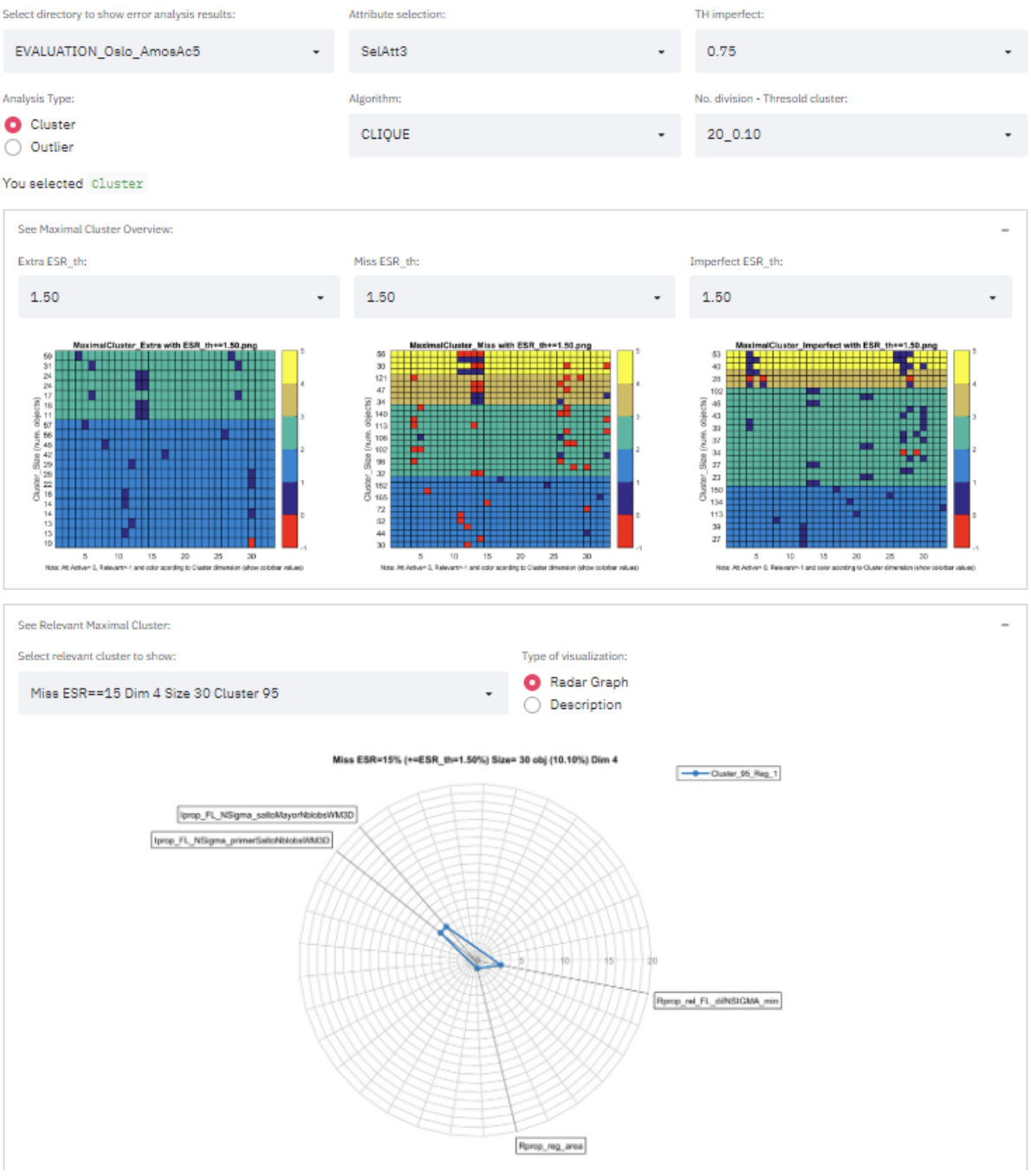


Figura 4.16: Exploración de un clúster maximal relevante (ESR>1.5) del dataset Miss con SelAtt3 y CLIQUE 20 0.10

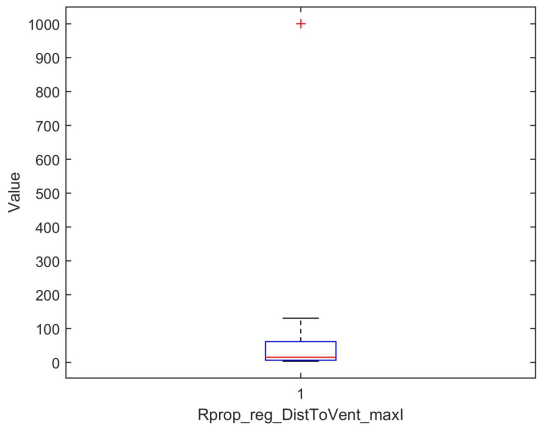


Figura 4.17: Situación con valor anómalo en el análisis unidimensional

Dado que estos métodos son opacos en su ejecución y ofrecen un listado de instancias marcadas como anómalas con información sobre los conjuntos de variables sobre los que se detectó la anomalía, para detectar un outlierMD con un mayor nivel de confianza, hemos estudiado su presencia en varias configuraciones. Para el subconjunto de datos de tipo Imperfect se muestran, en la tabla 4.7, cuatro configuraciones junto a un listado reducido de los objetos de agrupación marcados como outlierMD. En la última fila, se indica el número total de objetos de agrupación detectados como anomalías en cada configuración. Se han ordenado los outlierMD en orden decreciente al número de combinaciones de variables que detectan dicho objeto.

0.20-0.20-0.20		**0.15-0.15-0.15**		**0.12-0.12-0.12**		**0.05-0.05-0.05**	
outMD	N. Comb	outMD	N. Comb	outMD	N. Comb	outMD	N. Comb
241	4	641	5	440	6	641	9
60	3	719	5	641	6	440	6
122	3	122	4	122	5	614	6
238	3	436	4	614	5	719	6
312	3	440	4	719	5	122	5
436	3	614	4	436	4	436	4
641	3	1016	4	1016	4	875	4
992	3	233	3	60	3	940	4
1005	3	241	3	238	3	60	3
1016	3	696	3	241	3	241	3
92		**74**		**67**		**66**	

Tabla 4.7: Identificación de objetos de agrupación con características anómalas en diferentes configuraciones

Al observar la tabla se detecta, por ejemplo, que el outlierMD 122 aparece en las cuatro configuraciones. Además, se observa que tiene múltiples combinaciones de variables que detectan dicho outlier, por lo que se concluye que es un outlierMD relevante.

El siguiente paso es conocer como es este outlierMD y cuales son sus descripciones, para ofrecer al experto esta información de manera que pueda tomar una decisión de refinamiento en el SVC. El outlierMD 122 se corresponde con el objeto ID_R.P 21 del caso E094 y se presenta en su contexto en la figura 4.18.

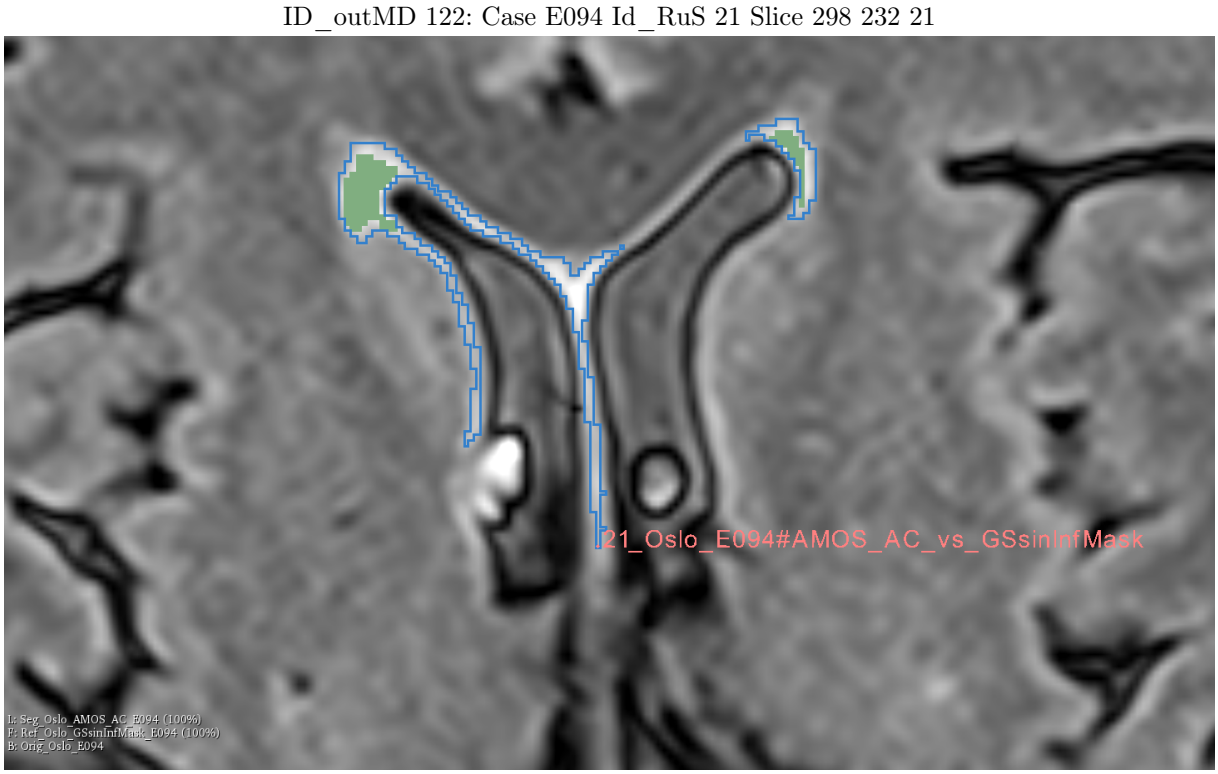
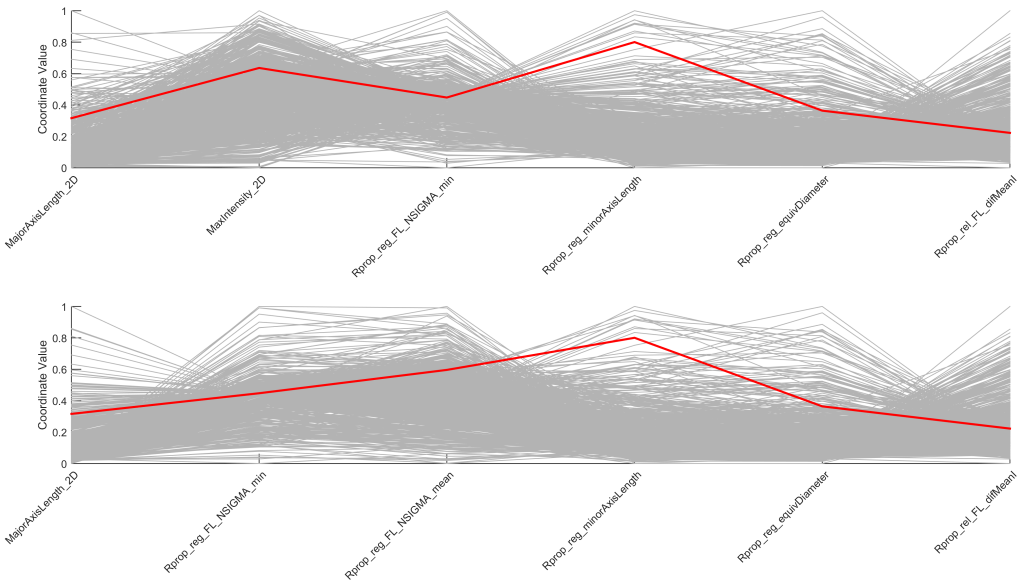


Figura 4.18: Objeto de agrupación que se detecta como error anómalo en el subconjunto de error “Imperfect”

Una detección del outlierMD 122 con dos combinaciones y 6 variables para la configuración “0.12-0.12-0.12” se describe en la figura 4.19. En ella, se presenta las gráficas de la distribución de los datos, donde se marca en color rojo el caso anómalo, y también se presenta su descripción textual. Se observa que un outlierMD es indistinguible del resto de casos en la representación gráfica, por lo que es su descripción textual la que nos permite conocer qué variables y qué valores detectan el outlierMD. Interesa conocer primero aquellas combinaciones de menor número de variables que detectan un outlierMD, ya que permite guiar al experto al análisis de unas variables frente a otras con el objetivo de refinar el SVC para eliminar dicho comportamiento anómalo.

Config: 0.12-0.12-0.12



Comb 1	MajorAxisLength_2D	82.7616	Comb 2	MajorAxisLength_2D	82.7616
	MaxIntensity_2D	0.4529		Rprop_reg_FL_NSIGMA_min	4.5246
	Rprop_reg_FL_NSIGMA_min	4.5246		Rprop_reg_FL_NSIGMA_mean	7.9194
	Rprop_reg_minorAxisLength	32.7284		Rprop_reg_minorAxisLength	32.7284
	Rprop_reg_equivDiameter	23.0422		Rprop_reg_equivDiameter	23.0422
	Rprop_rel_FL_difMeanI	0.126		Rprop_rel_FL_difMeanI	0.126

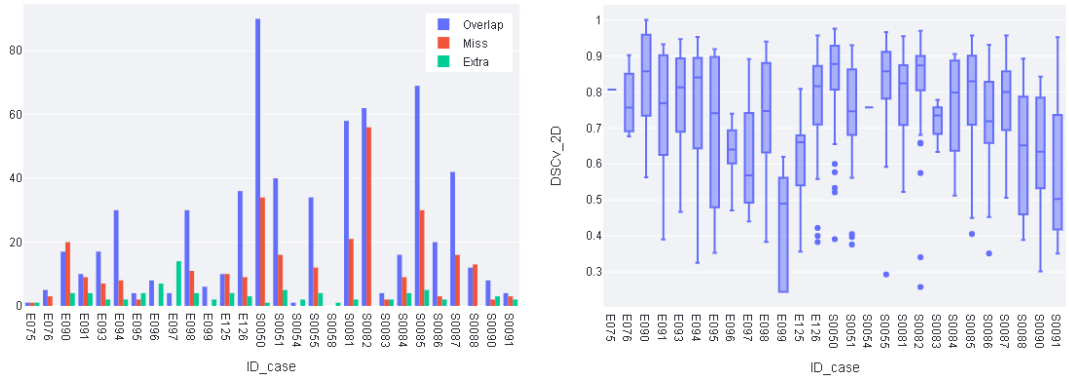
Figura 4.19: Ejemplo de una situación con valores anómalos en análisis multidimensional de dos combinaciones

4.3.1.4. Módulo de exploración del error (MEE)

El módulo de la metodología AMOSE^2 que define las vistas para explorar los hallazgos relevantes de error se ha implementado mediante una aplicación web, "AMOSE^2 web report", para mejorar la iteración con el usuario y facilitar nuevas descripciones del error.

Para este análisis, como se comentó anteriormente, la exploración se puede realizar de forma manual a partir de la descripción de múltiples variables de los objetos de agrupación (pestaña "Explore") o de forma guiada a partir de los resultados del análisis de agrupaciones o del análisis de anomalías realizado en el MCE (pestaña "Cluster/Isolated error").

Exploración manual Un primer paso en el análisis de evaluación de la segmentación es conocer de forma global la calidad de la propuesta de segmentación del SVC respecto a la referencia. Para ello, la herramienta "AMOSE^2 web report" permite realizar una exploración por caso de la distribución del número de objetos de agrupación según su tipo de error: Miss, Extra y Detected o analizar el coeficiente DSC de los objetos con solape como se muestran en la figura 4.20 (a) y (b), respectivamente.



(a) (b)

Figura 4.20: Distribución por caso del número de objetos por tipo de detección (a) y del coeficiente Dice de objetos con solape (b) del caso de estudio Oslo AMOS-2D

En el gráfico de barras (a) se observa que existen casos con comportamientos raros, con valores muy elevados de objetos Miss u objetos Extra, como son el caso E0097 o el S0082. En (b) se muestra el coeficiente Dice de objetos de solape mediante un diagrama de cajas y bigotes ("Box-plot") en el que se representan los valores máximo, mínimo, mediana, 1er cuartil y 3er cuartil.

En general, se observa que el comportamiento del coeficiente DSC es bueno, pero hay un caso con valores más bajos que el resto, el caso E099. También se observa que hay casos con valores mínimos anómalos los cuales se representan mediante puntos, como el

caso S0082. Con el sistema AMOSE[2], cada caso destacado puede ser explorado de forma individual de manera que se pueda descubrir una explicación. Algunos casos destacados se presentan a continuación.

1. **Valores elevados de objetos Extra**

   El caso E097 es un ejemplo de un número elevado de objetos de tipo Extra. Está formado por 18 objetos "$O_{R.P}$" de los cuales 14 son de tipo Extra y 4 de tipo Overlap. El tamaño de los objetos tipo Extra varía entre 8 y 176 vóxeles y se sitúan en 10 slices distintos, como se muestra en la figura 4.21. Consultando los objetos en su contexto, esto es, en la imagen de resonancia magnética, se observa que la mayoría de los objetos extra de mayor tamaño se producen alrededor de los ventrículos (ver figura 4.22).

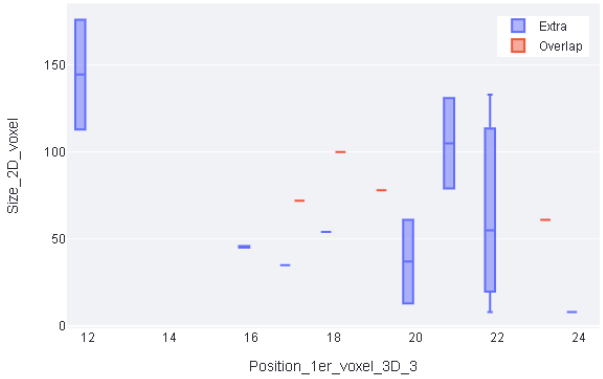


Figura 4.21: Tamaño de los objetos en las diferentes slices del caso Oslo AMOS-2D E097

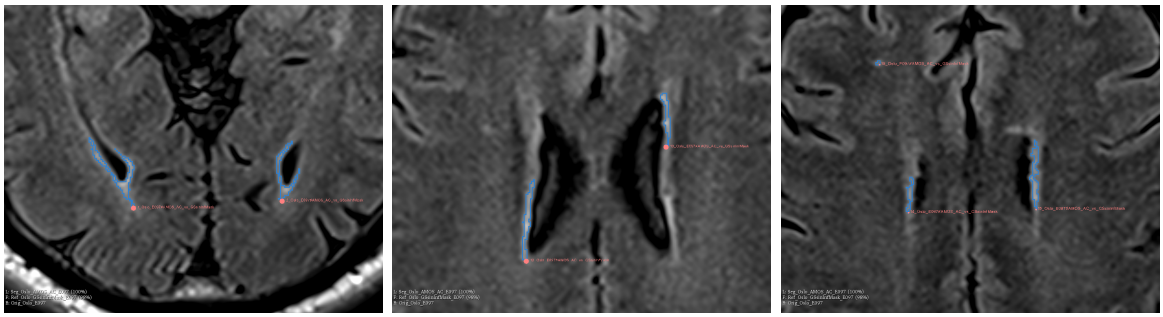


Figura 4.22: Objetos $O_{R.P}$ en su contexto, caso Oslo E097
Nota: Objecto O_R opaco en verde y objeto O_P transparente con bordes en azul

2. **Valores distintos a la media**

   Al analizar en profundidad el caso E099 se encuentra que se compone de 8 objetos "$O_{R.P}$", 2 de los cuales son de tipo Extra y el resto de tipo Overlap tipo 1-1, con tamaños que varían entre [51-251] vóxeles. Los objetos se detectan en 4 slices, entre

la 16 y la 19. Dado el número tan bajo de casos, se puede hacer una inspección visual en su contexto para conocer la causa de tan bajo porcentaje de solape. En las imágenes de la figura 4.23 se aprecia que existe una sobresegmentación extrema en los objetos 2 y 4, debida a la hiperintensidad de la zona de los ventrículos.

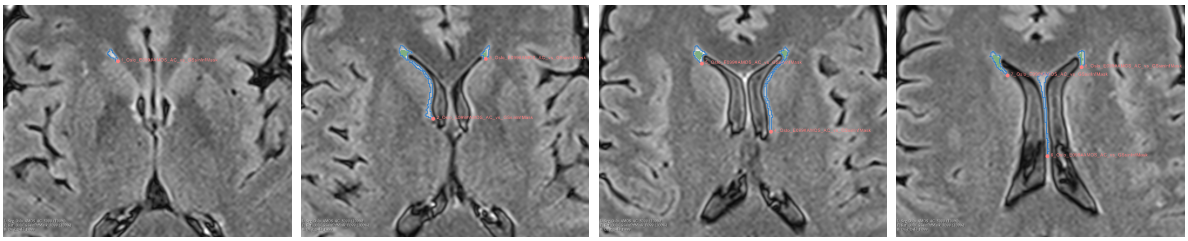


Figura 4.23: Objetos $O_{R.P}$ en su contexto, caso Oslo E099 slices 16 a 19
Nota: Objecto O_R opaco en verde y objeto O_P transparente con bordes en azul

3. **Valores elevados de objetos tipo Miss y valores anómalos de DSC_overlap**

El caso S0082 se compone de 118 objetos de agrupación, 58 de tipo Miss y 62 de tipo Overlap. El tamaño de éstos varía entre 7 y 4774 vóxeles y su distribución por slices y tipo de objeto se muestra en la figura 4.24. Si se analiza la evolución del DSC por slice, se observa que en la slice 12 existe un caso con un valor muy bajo que hay que estudiar. Si se utiliza un valor umbral de DSC < 0.6 se detectan tres objetos que cumplen dicho criterio.

(a) (b)

Figura 4.24: Caso S0082: (a) Distribución por slices del coeficiente Dice (DSC) y (b) Tamaño de los objetos de agrupación

En la figura 4.25 se muestran las lesiones hiperintensas en su contexto. Se observa que el objeto con un menor DSC se sitúa cerca de un ventrículo, una zona conflictiva de segmentación ya que es una zona hiperintensa que a veces está segmentada por los expertos y otras no. Los objetos tipo Miss tienen un tamaño que varía entre [7, 288] vóxeles y su distribución respecto a la intensidad mínima y máxima del objeto

se muestra en la figura 4.26 (a) y (b), respectivamente. Se observa que la distribución de la intensidad mínima es similar tanto para objetos Overlap como Miss mientras que en la intensidad máxima se observa un desplazamiento del valor medio de los distintos tipos. S0082 es un caso con un nivel elevado de hiperintensidades tanto en número como en tamaño de las mismas que provoca fallos de infrasegmentación y un número elevado de objetos no detectados.

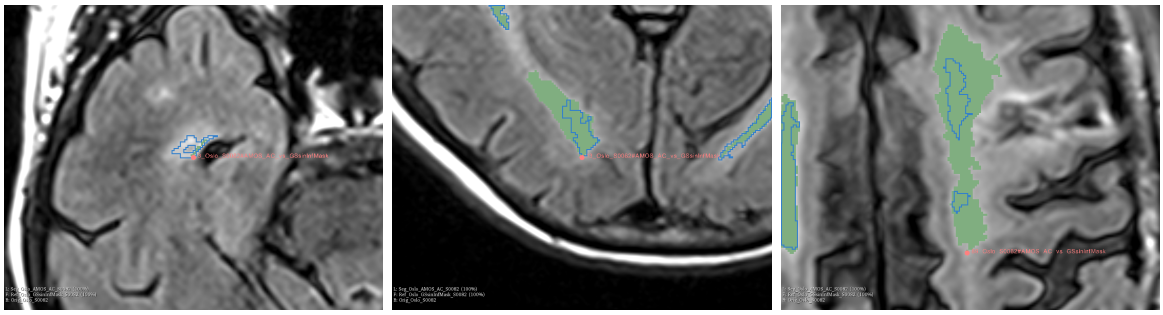
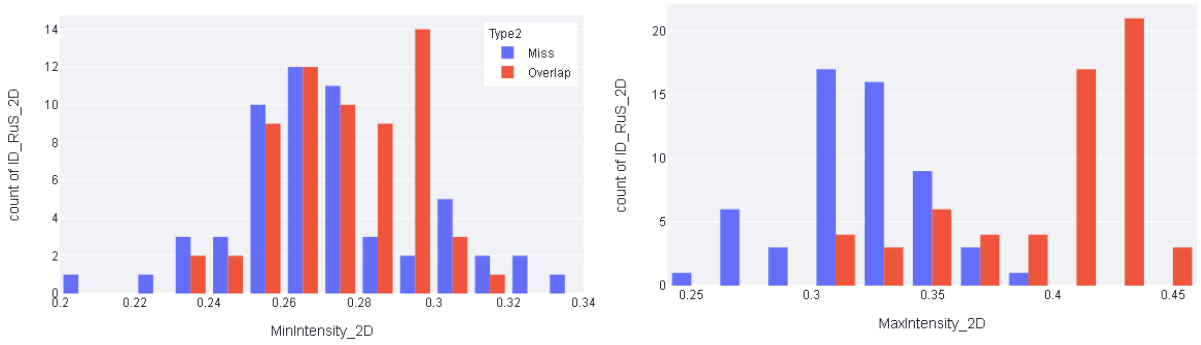


Figura 4.25: Objetos de agrupación $O_{R.P}$ en su contexto, caso Oslo S0082
Nota: Objecto O_R opaco en verde y objeto O_P transparente con bordes en azul



(a) (b)

Figura 4.26: Distribución de número de objetos según su intensidad mínima y máxima para el caso Oslo S0082

Exploración de agrupaciones de error La herramienta "AMOSE^2 web report" en su pestaña "Cluster/Isolated Error" permite conocer y explorar todos los experimentos realizados en la búsqueda de bolsas de error de objetos de agrupación mediante el algoritmo CLIQUE. Dado que el algoritmo trabaja con una rejilla fija para subdividir el espacio de entrada, se ejecutan diferentes configuraciones para explorar el conjunto de datos multidimensionales para un tipo de error.

Los mapas de color permiten ver de forma global y rápida los atributos descriptivos del clúster. Cuando existen atributos que se repiten en una gran mayoría de clústeres, puede indicar un comportamiento anómalo. En la figura 4.27 se muestra una comparativa de

diferentes selecciones de atributos para el subconjunto de datos Miss, en ella se observa que en la selección SelAtt1 de la tabla 4.6, los atributos 20 y 32 están presentes en la mayoría de los clústeres.

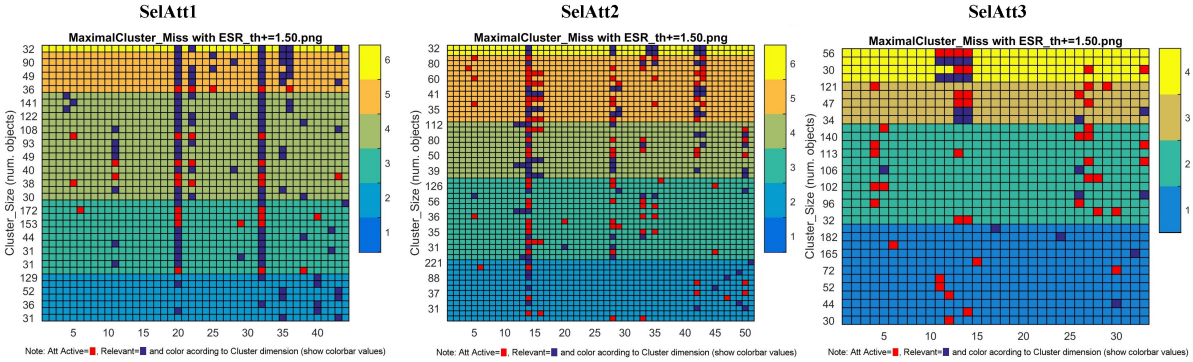


Figura 4.27: clúster detectados con diferentes selección de atributos

En el primer caso, al comprobar los valores numéricos de los rangos de esos atributos en el subconjunto de datos de tipo Miss, se vio que todas las instancias tenían el mismo valor para todo el subconjunto de error bajo análisis. Por tanto, esas variables no aportan información para la caracterización del error.

En la selección SelAtt2, en el atributo 14 se observa otro comportamiento raro. Al comprobar el subconjunto de datos de tipo Miss se observó que el atributo presentaba solo dos valores y que estaban muy desbalanceados, el 99.3 % de instancias se daba para uno de estos valores (ver figura 4.28). La variable tiene en la mayoría de los objetos de agrupación, independientemente de su tipo, el mismo valor. Habrá que consultar con el experto si tiene sentido o hay un fallo en el SVC que se deba depurar. Por último, en la selección SelAtt3 no se aprecia ninguna anomalía en esta vista resumen.

La información de los clústeres maximales relevantes, es decir, aquellos clústeres de mayor dimensionalidad que superan el umbral de relevancia ESR, se pueden representar en AMOSE^2 web report mediante un gráfico de radar o mediante una descripción textual. Se muestran dos ejemplos en la figuras 4.29 y 4.30 mediante la herramienta "AMOSE^2 web report". En la primera figura se muestra un clúster relevante (ESR>1'5) de tipo "Extra", con una cobertura de 10 objetos (un 11 % del subconjunto de tipo Extra) y de tres dimensiones. El clúster está formado por un único hipercubo, denominado R1. Tanto en el gráfico radar como en la descripción textual se muestran las variables que participan en el clúster y sus rangos. La segunda figura ejemplifica un clúster de dos dimensiones y tres hipercubos.

Exploración de anomalías de error La herramienta "AMOSE^2 web report" en su pestaña "Cluster/Isolated Error" también permite conocer y explorar todos los

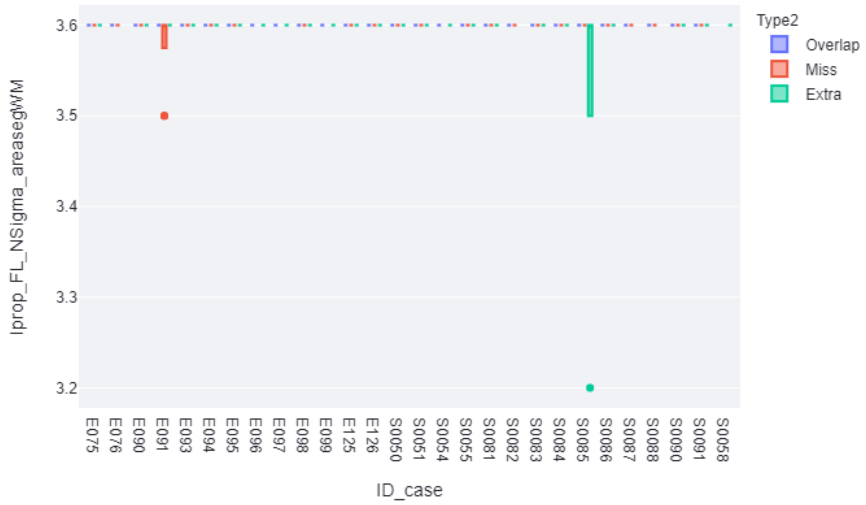


Figura 4.28: Box-plot de variable con comportamiento raro en mapa de calor: datos muy desbalanceados

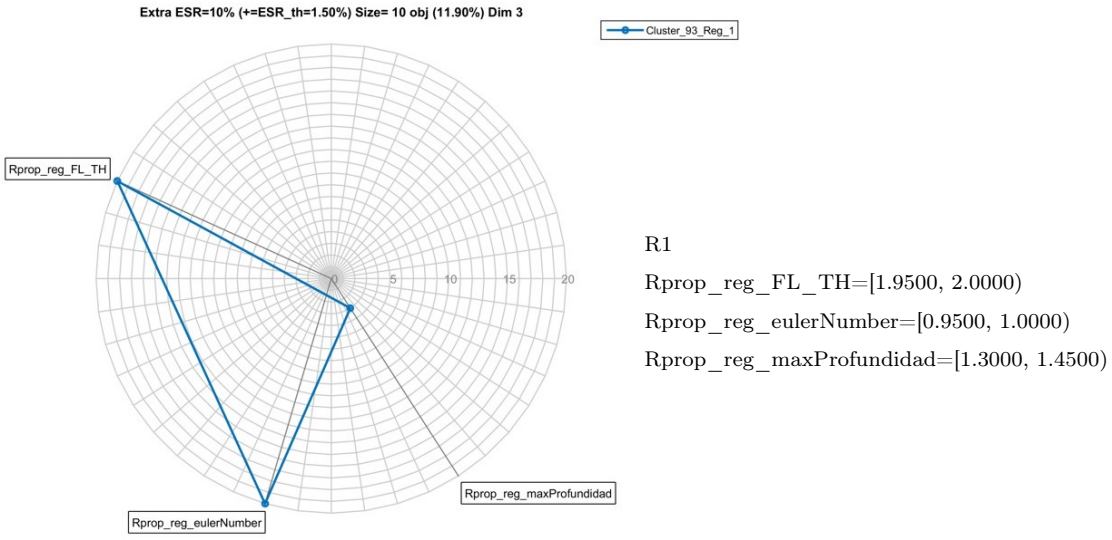


Figura 4.29: Descripción de un clúster maximal relevante de tres dimensiones y 1 región. Gráfico radar (dcha) y Texto (izda)

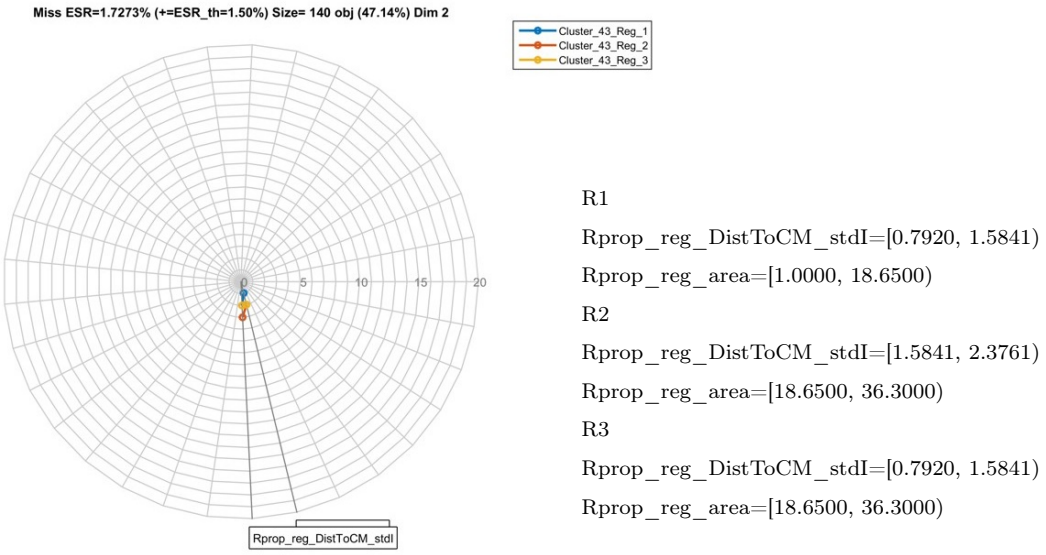


Figura 4.30: Descripción de un clúster maximal relevante de 2 dimensiones y 3 regiones. Gráfico radar (dcha) y Texto (izda)

experimentos realizados en la búsqueda de anomalías de error. Su ejemplo de uso se comentó en este capítulo anteriormente, a la par que se explicaban los dos métodos de análisis del MCE, el unidimensional y el multidimensional, por lo que no se repite aquí la misma información.

4.3.1.5. Módulo de descripción ontológico (MDO)

El módulo de descripción oncológico gestiona las diferentes ontologías que se relaciona con el SVC para mejorar la descripción y el uso de las características de los objetos de segmentación.

En este ejemplo, en el módulo ontológico, por un lado, se modelan las características ofrecidas por la versión actualizada de AMOS-2D con la ontología "AMOS-2D_VOF". Las variables internas de AMOS-2D se agregan al vector de características externas que se calculan para cualquier sistema, creando un vector de alta dimensionalidad para describir los objetos de agrupación. La instanciación de esta ontología se utiliza para mejorar la caracterización de los objetos de agrupación.

Por otro lado, se usa para calcular medidas de relevancia estadística de las variables respecto al tipo de error del objeto de agrupación y para calcular medidas de relevancia semántica a partir de la distancia semántica entre variables o su profundidad semántica, con el fin de mejorar el conocimiento de las variables analizadas y facilitar la toma de destino en procesos como la reducción de dimensionalidad o la presentación jerárquica de resultados.

4.3.2. Análisis comparado del error

Otra de las formas de uso de la metodología AMOSE^2 es el análisis que relaciona tres soluciones de segmentación de objetos con el objetivo de comparar las soluciones y conocer dónde falla y acierta el sistema evaluado frente a otros desarrollos.

4.3.2.1. Descripción del estudio

Para ejemplificar este uso se utiliza el dataset de Brno y dos algoritmos, el algoritmo M-Unet que es el sistema que se desea evaluar y el algoritmo PGS que es el sistema cuya solución tiene un mejor comportamiento general (es la ganadora actual del reto abierto "WMH segmentation Challenge"). Se dispone de las imágenes de resonancia magnética originales, FLAIR y T1, y de imágenes binarias de objetos segmentados, tanto de la referencia como de las segmentaciones propuestas por los SVCs. Estos algoritmos son de caja negra por lo que el conjunto de características es más reducido y no hay que realizar reducción de dimensionalidad.

Como en el ejemplo anterior, se ha utilizado el prototipo AMOSE^2 siguiendo sus diferentes módulos. En este ejemplo, sólo nos centraremos en ejemplificar el uso del análisis comparado con el objetivo de detectar comportamientos de error singulares. Como se comentó en el capítulo anterior, se han definido 10 comportamientos de error (ver tabla 3.5). La detección y exploración de alguno de ellos se explica a continuación.

4.3.2.2. Módulo de descripción del error (MDE)

En primer lugar, con la metodología AMOSE^2, en concreto, en el módulo MDE, se comparan las propuestas respecto a la referencia y las propuestas entre ellas, generando los objetos de agrupación "$O_{R.M\text{-}Unet}$", "$O_{R.PGS}$" y "$O_{PGS.M\text{-}Unet}$", respectivamente. En la tabla 4.8, se observa el número de estos objetos de agrupación junto al número de objetos de su componentes ("#O_R" y "#O_P"), el número de objetos de agrupación detectados ("#O_Detected"), los no detectados ("#O_Miss") y los extra detectados ("#O_Extra"), tanto en valor absoluto como en porcentaje. También se muestra su comportamiento a nivel de vóxeles: el número de vóxeles en la referencia "$V(O_R)$", en la propuesta "$V(O_P)$", el número de vóxeles de los objetos de agrupación "$V(O_{R.P})$", el número de vóxeles correctos, es decir, con solape en "$V(O_{Overlap})$" y los vóxeles con delineación errónea, infra-segmentados "$V(O_{UnderSeg})$" y sobre-segmentados "$V(O_{OverSeg})$", y por último, el número de vóxeles de detección erróneos, "$V(O_{Miss})$" y "$V(O_{Extra})$". Finalmente, se muestran medidas de evaluación a nivel de vóxel mediante el coeficiente de similitud Dice (DSCv) y a nivel de objeto mediante la exhaustividad (Recall), la precisión (Precision) y el valor-F1

(F1-Score). Estos datos se han extraído de la página principal de la aplicación "AMOSE² web report", seleccionando la vista resumen del MEE.

		Brno		
		R.M-Unet	**R.PGS**	**PGS.M-Unet**
Núm. Objeto	$\#(O_R)$	2129	2003	2873
	$\#(O_P)$	9375	2873	8438
	$\#(O_{R.P})$	9571	3171	8638
	$\#(O_{Detected})$	1719 (18 %)	1527 (48.2 %)	2536 (29.4 %)
	$\#(O_{Miss})$	297 (3.1 %)	370 (11.7 %)	291 (3.4 %)
	$\#(O_{Extra})$	7555 (78.9 %)	1274 (40.2 %)	5811 (67.3 %)
Núm. Voxel	$V(O_R)$	33154	32096	45705
	$V(O_P)$	73544	45705	68647
	$V(O_{R.P})$	80976	52682	76042
	$V(O_{Overlap})$	25722 (31.8 %)	25119 (47.7 %)	38310 (50.4 %)
	$V(O_{UnderSeg})$	5986 (7.4 %)	4630 (8.8 %)	6211 (8.2 %)
	$V(O_{OverSeg})$	11436 (14.1 %)	10818 (20.5 %)	4723 (6.2 %)
	$V(O_{Miss})$	1446 (1.8 %)	2347 (4.5 %)	1184 (1.6 %)
	$V(O_{Extra})$	36386 (44.9 %)	9768 (18.5 %)	25614 (33.7 %)
Métricas	DSCv	0.4821	0.6457	0.67
	$DSCv_{Detected}$	0.747	0.7648	0.8751
	Recall	0.8527	0.805	0.8971
	Precision	0.1854	0.5452	0.3038
	F1-score	0.3045	0.6501	0.4539

Tabla 4.8: Resultados generales de las soluciones comparadas respecto a la referencia y entre ellas

Se observa que la solución PGS obtiene mejores resultados tanto a nivel de precisión de los vóxeles de los objetos con solape (DSCv_overlap) como a nivel de los objetos (F1-score) aunque hay una alta tasa de objetos extra detectados. Al comparar la solución PGS respecto a M-Unet se observa que el valor DSCv_Overlap es de 0.87, un valor que supera el umbral 0.7 que según la bibliografía significa que la segmentación es similar, mientras que hay un peor resultado en la detección de los objetos que se muestra con el valor F1-score igual a 0.45. Se detecta que tanto PGS como M-Unet tienen una alta tasa de objetos extra detectados respecto a la referencia y eso da lugar a un nivel elevado de objetos de error.

Con las facilidades de "AMOSE² web report", en concreto, con la vista personalizada del MEE (pestaña "Explore"), podemos explorar con mayor detalle estos comportamientos. Por ejemplo, podría ser interesante conocer la distribución del error de detección, es decir, los objetos Miss y Extra, desde el punto de vista del refinamiento del sistema. En las

figuras 4.31 y 4.32 se puede observar la distribución de objetos no detectados, por caso y por tamaño en vóxeles, respectivamente. Con esta herramienta también podemos conocer la distribución del tamaño de los objetos de error, tanto Miss como Extra, para comprobar si el problema se da en objetos pequeños, medianos o grandes. En la figura 4.33 se muestra su distribución. Se comprueba que la mayoría del error se da en objetos pequeños o muy pequeños (menos de 5 vóxeles).

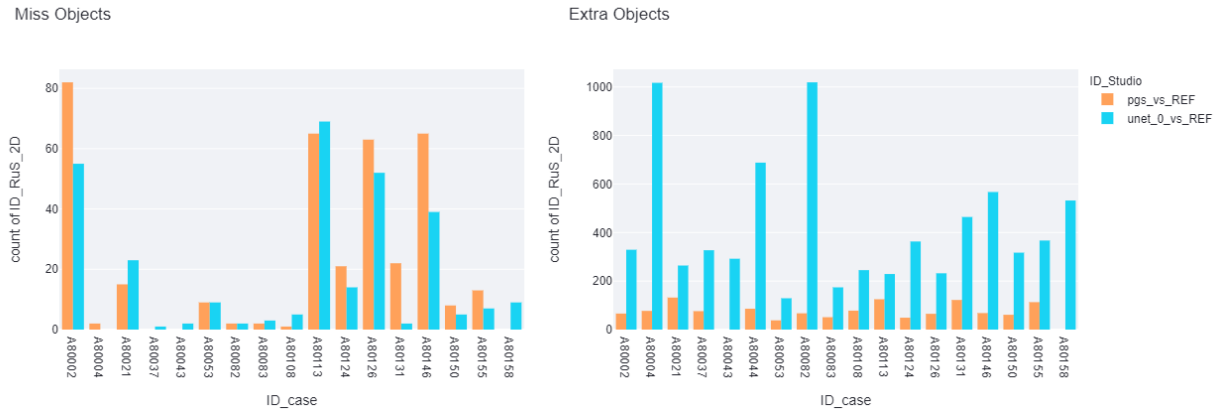


Figura 4.31: Comparativa entre diferentes soluciones del número de objetos Miss y Extra por caso

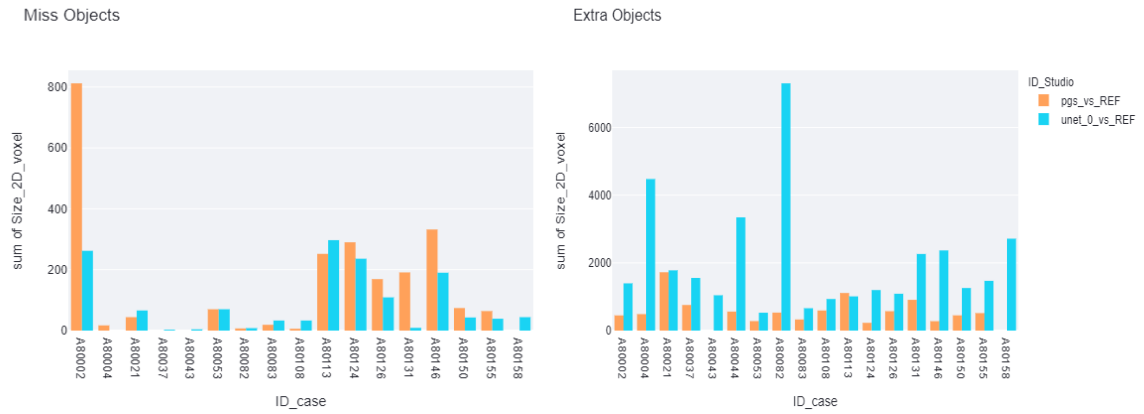


Figura 4.32: Comparativa entre diferentes soluciones del número de vóxeles Miss y Extra por caso

De manera similar podría ser interesante conocer la evolución, por caso, del coeficiente DSC en volumen para los objetos detectados, como se muestra en la figura 4.34. En ella se observa, de forma generalizada, que casi todos los casos PGS respecto a la referencia ("pgs_vs_REF") son mejores (más cercano a 1 es mejor comportamiento) que M-Unet respecto a la referencia ("unet_0_vs_REF"), como ya se vio anteriormente con la tabla de resultados. También se observa que existen comportamientos anómalos (los puntos) que será interesante explorar para conocer sus características (cuántos son, cómo son,

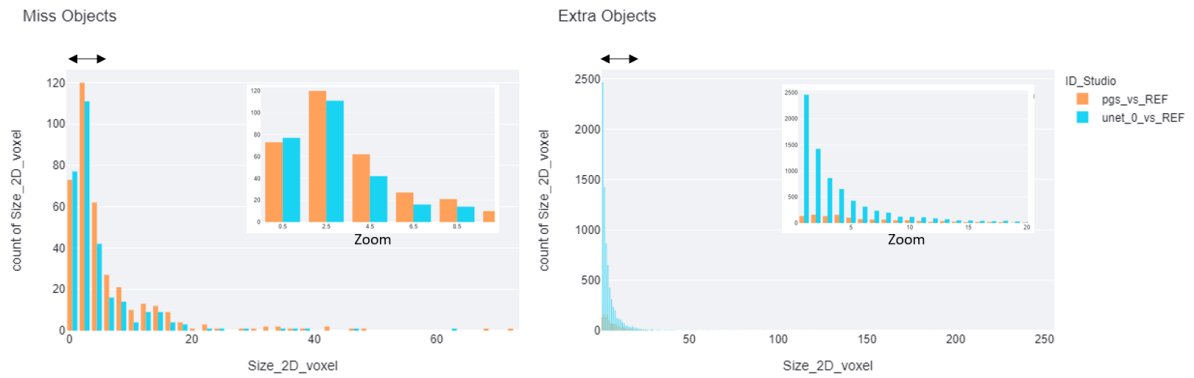


Figura 4.33: Distribución del tamaño de los objetos (núm vóxeles) error de tipo Miss y tipo Extra en la comparativa entre dos propuestas de segmentación

dónde se localizan,...) y, además, se detecta que hay dos casos, los dos últimos, donde no hay solución del algoritmo PGS. Este hallazgo es importante y hay que hacérselo saber al desarrollador del algoritmo para que repare el error que ha provocado que no se genere solución. Por último, al analizar el comportamiento de M-Unet respecto a PGS ("unet_0_vs_PGS") se observa que hay una gran coincidencia en delineación en la mayoría de los casos, aunque también hay casos anómalos que han de ser analizados con mayor profundidad.

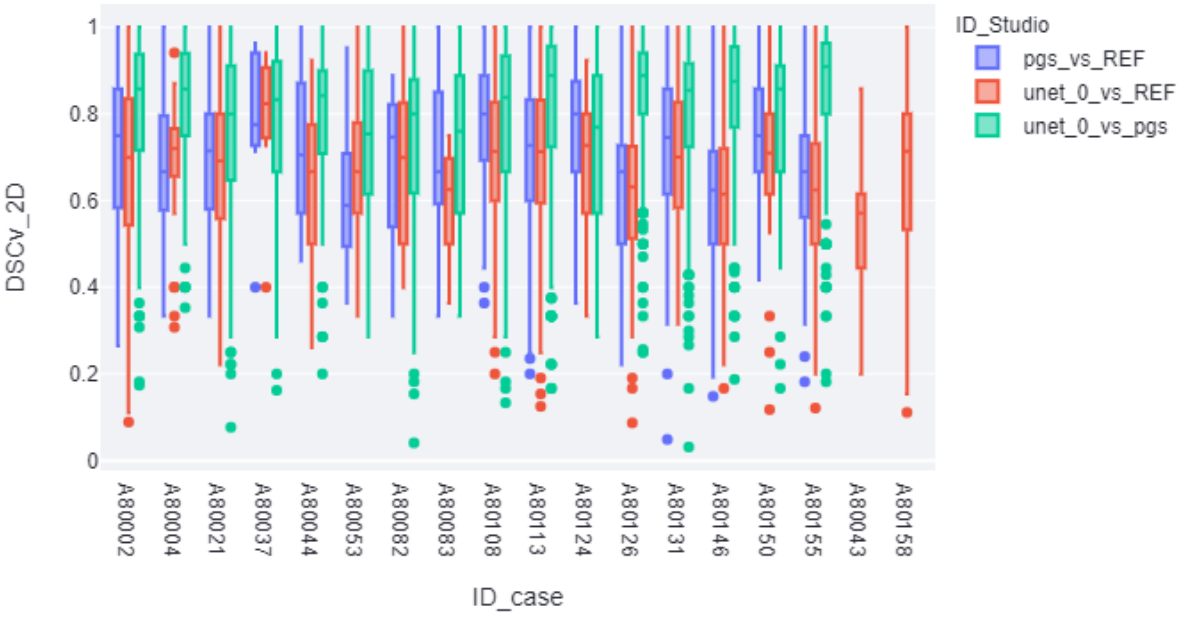


Figura 4.34: Comparativa entre diferentes soluciones de la distribución del DSC de objetos con solape por caso

El problema de este análisis es que es muy artesanal y que sólo posibilita ver el comportamiento general de los diferentes algoritmos. Por ejemplo, ayuda a averiguar en que casos se localizan mayores errores o cómo es el tamaño de los objetos, lo que permite priorizar acciones, pero para profundizar en sus diferencias y obtener información es insuficiente. Para extraer nuevo conocimiento del comportamiento a nivel de los objetos de agrupación hay que, siguiendo la metodología AMOSE^2, realizar el análisis del error comparado del MCE, que se explica a continuación.

4.3.2.3. Módulo de caracterización del error (MCE)

En el módulo de caracterización del error, en su modo de funcionamiento de análisis comparado, se crean las estructuras de información necesarias para descubrir patrones de comportamiento de singularidades. Como se comentó en el capítulo anterior, esto se realiza a partir de las tablas LOJ, en las que se relacionan los objetos de agrupación de las tres segmentaciones comparadas a partir de sus objetos componentes.

En este estudio, a partir de los objetos de agrupación "$O_{R.M\text{-}Unet}$", "$O_{R.PGS}$" y "$O_{PGS.M\text{-}Unet}$" se crean las tablas $LOJ_{R.M\text{-}Unet\text{->}PGS}$ y $LOJ_{R.PGS\text{->}M\text{-}Unet}$ a partir de la relación entre los objetos componentes "O_R" y "O_P", según corresponda. La frecuencia de sus comportamientos de error se muestra en la tabla 4.9, una información extraída de la vista personalizada de exploración comparada de la aplicación web (pestaña "Explore 2 sol"). El orden de la reunión es importante por lo que en cada tabla LOJ el significado de las columnas es diferente.

El tipo de error del objeto de agrupación se almacena en la tabla LOJ en la columna "Type2" y para la tabla $LOJ_{R.M\text{-}Unet\text{->}PGS}$ que se muestra en la tabla 4.9 (a), la primera columna de tipo de error, denominada Type2_TableA, corresponde a la agrupación $O_{R.M\text{-}Unet}$, la segunda, denominada Type2_TableB, corresponde a $O_{PGS.M\text{-}Unet}$ y la tercera, denominada Type2 a $O_{R.PGS}$. Para la tabla $LOJ_{R.PGS\text{->}M\text{-}Unet}$ (tabla 4.9 (b)) la primera corresponde a $O_{R.PGS}$, la segunda a $O_{PGS.M\text{-}Unet}$ y la tercera a $O_{R.M\text{-}Unet}$. En el módulo de exploración (MEE) se explicará el uso que se le puede dar.

4.3.2.4. Módulo de exploración del error (MEE)

La versión actual de prototipo AMOSE^2 no tiene implementado ningún procedimiento de búsqueda de agrupaciones o comportamientos aislados en el análisis comparado por lo que hay que realizar una exploración manual de las tablas LOJ para detectar y describir comportamientos relevantes de error.

Desde el punto de vista de la evaluación de los SVCs, puede ser interesante comenzar por conocer, en primer lugar, cuáles son los objetos de la referencia que no se detectan por ningún algoritmo. Este hecho se corresponde con el tipo de error comparado C1 (ver tabla

Select a Relational Table of 2Sol RuP object

Brno_Ref2MUnet-PGS2MUnet-Ref2PGS_objects_link_2D.csv

Type2_TableA	Type2_TableB	Type2	
Detected	---	---	120
		Detected	1141
		Miss	89
	Detected	Detected	720
		Miss	62
	Miss	Detected	62
		Miss	3
Extra	---	---	5386
	Detected	---	1998
	Miss	---	236
Miss	---	---	11
		Detected	64
		Miss	229

(a)

Select a Relational Table of 2Sol RuP object

Brno_Ref2PGS-PGS2MUnet-Ref2MUnet_objects_link_2D.csv

Type2_TableA	Type2_TableB	Type2	
Detected	Detected	Detected	882
		Miss	28
	Extra	Detected	1033
		Miss	36
Extra	Detected	---	334
	Extra	---	945
Miss	---	Detected	152
		Miss	229

(b)

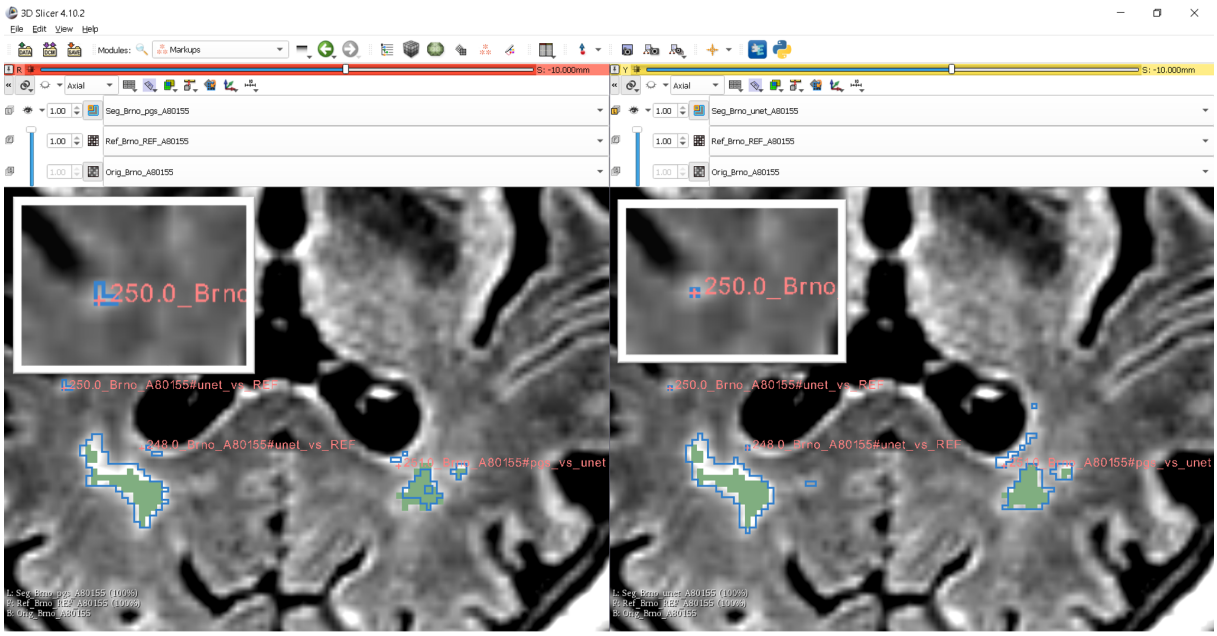
Tabla 4.9: Frecuencia de comportamientos de error comparado en tablas LOJ: (a) $LOJ_{\text{R.M-Unet -> PGS}}$ y (b) $LOJ_{\text{R.PGS -> M-Unet}}$

3.5). Para ello, hay que seleccionar en "AMOSE^2 web report" pestaña "Explore 2 sol" el fichero correspondiente a la tabla $LOJ_{\text{R.M-Unet -> PGS}}$. A continuación, seleccionar de cada agrupación comparada el tipo de error a explorar, en este caso la triada "Miss-NaN-Miss" para la variable "Type2". Se observa en la tabla 4.9 que son 229 los objetos de la referencia que no se detectan por ninguna propuesta. El experto con las facilidades proporcionadas por "AMOSE^2 web report" podrá seleccionar este grupo de objetos de error tipo Miss para analizar sus propiedades.

También puede ser interesante conocer si hay objetos segmentados por ambas soluciones propuestas, pero que no están segmentadas en la referencia, es decir el comportamiento C6. En este caso, hay que seleccionar los objetos que cumplen con la triada "Extra-Detected-NaN" en $LOJ_{\text{R.M-Unet -> PGS}}$. En la figura 4.35 se muestra el caso de objetos PGS y M-Unet detectados entre ellos y que no tienen contacto con un objeto de la referencia. Este suceso puede indicar un fallo de la referencia al no detectar esa zona hiperintensa ya que los dos sistemas de propuesta si lo hacen.

Otra situación interesante para mejorar el sistema bajo evaluación es cuando la segmentación propuesta no detecta el objeto, pero si lo detecta la segmentación alternativa, es decir, el comportamiento C4. Este comportamiento se extrae de la triada "Miss-NaN–Detected" en $LOJ_{\text{R.M-Unet -> PGS}}$. Se observa en la tabla 4.9 que hay 64 situaciones con este comportamiento y, mediante la aplicación web, podemos explorar sus características.

Utilizando una clasificación del error más detallada, separando los objetos detectados en objetos de contacto "Contact" y de solape "Overlap", como se comentó en la figura 3.8, se pueden detectar otro tipo de situaciones singulares. Por ejemplo, puede ser



(a) Propuesta PGS (b) Propuesta M-Unet

Figura 4.35: Solape entre segmentaciones propuestas y sin contacto respecto a la referencia. Los objetos de la referencia se muestran en verde mientras que en borde azul se muestra la propuesta

interesante detectar situaciones donde haya solape entre las propuestas y sólo contacto con la referencia. Para ello, hay que selecciona la triada "Contact-Overlap-Contact" en $LOJ_{R.M\text{-}Unet \text{ -> } PGS}$. En la tabla 4.10 se muestra la información de la frecuencia de las situaciones con dicha sub-clasificación, donde se puede ver que son situaciones poco frecuentes.

Un ejemplo sobre la imagen de RM se muestra en la la figura 4.36. Este suceso podría deberse a un fallo en la referencia ya que los dos segmentaciones propuestas informan de que la zona hiperintensa estaría un píxel desplazada. El experto al conocer esta información y observar el suceso podrá tomar una decisión.

En la mayoría de los casos, una misma información se puede extraer de diferentes tablas LOJ pero hay que resaltar que se encuentran variaciones en las frecuencias de las selecciones. Esto es debido a los casos con solape múltiple, ya que aparece una entrada por cada objeto que forman la agrupación $O_{R.P}$, ya sea de la referencia o de la propuesta.

El uso del análisis comparado sirve tanto para comparar dos soluciones de diferentes SVC como de un mismo SVC con diferentes soluciones. Que puede ser

Select a Relational Table of 2Sol RuP object

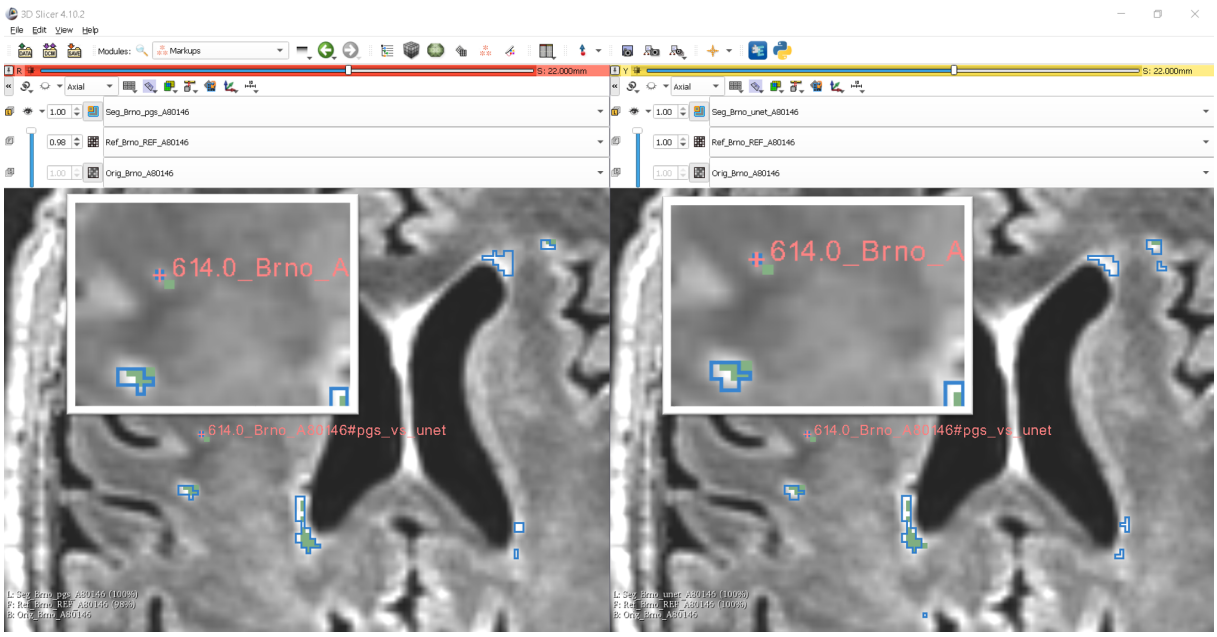
BrnoAxialView_Ref2Unet-Pgs2Unet-Ref2Pgs_objects_link_2D.csv

Type2_TableA	Type2_TableB	Type2	
Contact	---	---	3
		Contact	3
		Miss	6
		Overlap	8
	Overlap	Contact	1
		Miss	3
		Overlap	3
Extra	---	---	5386
	Contact	---	1
	Miss	---	236
	Overlap	---	1997
Miss	---	---	11
		Contact	5
		Miss	229
		Overlap	59
Overlap	---	---	117
		Contact	2
		Miss	83
		Overlap	1128
	Contact	Overlap	1
	Miss	Miss	3
		Overlap	62
	Overlap	Miss	59
		Overlap	715

Tabla 4.10: Frecuencia de comportamientos de error comparado en tabla LOJ con subclasificación en objetos "Contact" y "Overlap"

4.3.2.5. Módulo de descripción ontológico (MDO)

El módulo ontológico para el análisis de dos SVC de caja negra no tiene, por ahora, ningún uso ya que como sólo se calculan 10 variables visuo-espaciales de los objetos de agrupación, suponen un número manejable que se pueden gestionar sin una formalización de su descripción ontológica. Sin embargo, no cabe duda que para mejorar la automatización del proceso será necesario instanciar su estructura mediante la ontología de caracterización "VisualObjectFeature" definida en el capítulo anterior.



(a) Propuesta PGS (b) Propuesta M-Unet

Figura 4.36: Solape entre segmentaciones propuestas y con contacto respecto a la referencia. Los objetos de la referencia se muestran en verde mientras que en borde azul se muestra la propuesta

Capítulo 5

Conclusiones, aportaciones y trabajo futuro

5.1. Conclusiones

Diseñar un SVC es una tarea compleja, y disponer de herramientas basadas en IA que ayuden a los expertos en su construcción es una buena solución en la actualidad y una línea de progreso hacia la construcción de sistemas cognitivos para el diseño de SVC. Un módulo muy importante en estos sistemas será el encargado de realizar la evaluación del sistema con la idea, no solo de validar el diseño, sino de caracterizar el error (encontrar qué errores se comenten, qué patrones de error se repiten) para buscar sus causas y poder proponer mejoras para refinar el sistema de manera iterativa.

La metodología AMOSE^2 se desarrolló y utilizó durante la fase de diseño del sistema de segmentación AMOS-2D. En aquel tiempo, los sistemas de aprendizaje profundo no estaban tan desarrollados y era necesaria más ingeniería de características para construir sistemas basados en aprendizaje. La metodología sirvió para mejorar iterativamente el diseño a través de la detección de grandes bolsas de error mediante clustering y la mejora del etiquetado del dataset de entrenamiento mediante la detección de outliers. Los resultados obtenidos dieron lugar a una publicación de alto índice de impacto [Rincón et al., 2017].

Para mostrar el uso de la metodología se han analizado los resultados de tres algoritmos de segmentación de hiperintensidades en la sustancia blanca cerebral: AMOS-2D, un algoritmo de caja gris, M-Unet, un algoritmo de caja negra y PGS, otro algoritmo de caja negra. Estos algoritmos se han aplicado a un conjunto de imágenes de RM de pacientes de edad avanzada y enfermedades neurodegenerativas de cinco máquinas distintas. En total se han analizado 106 imágenes de las que se disponía de dos modalidades de imagen (FLAIR y T1) y, al menos, una segmentación de referencia.

Se demuestra, por tanto, que la metodología permite la evaluación y la caracterización del error en segmentación que comete un sistema respecto a una referencia dada y también, respecto a otro sistema de segmentación.

5.2. Aportaciones

La aportación principal de este trabajo es la propuesta, desarrollo y aplicación de una nueva metodología de caracterización del error en segmentación de objetos amorfos que permite detectar y describir comportamientos de error relevantes. Esta metodología propone una forma sistemática de evaluar el error en segmentación que aporta nuevo conocimiento para el refinamiento del sistema de segmentación. La metodología se ha denominado AMOSE^2, de las siglas en inglés "AMorphous Object Segmentation Error Exploration", y presenta las siguientes características principales:

(a) En cuanto a la descripción del error: la metodología AMOSE^2 presenta dos novedades principales, 1) el modelado individual de los objetos de error y 2) una descripción más detallada de los objetos segmentados, lo que le permite realizar un análisis más profundo de los errores de segmentación. Se define un nuevo objeto de segmentación denominado objeto de agrupación $O_{R.P}$ mediante la comparación espacial entre dos segmentaciones, O_R y O_P, referencia y propuesta, respectivamente. Cada objeto de agrupación $O_{R.P}$ es descrito con múltiples características visuo-espaciales procedentes del propio objeto de agrupación y de sus objetos ancestros.

Cuantas más características, mejor descripción de los comportamientos de error. Para ello, 1) se promueve el uso de sistemas de caja gris, de manera que se proporcione información de etapas internas del proceso para que pueda ser utilizada en la descripción del error en segmentación; y 2) se promueve el uso de ontologías de dominio para la definición formal de las variables descriptivas de los objetos segmentados, de manera que se puedan estructurar y jerarquizar estas variables con información semántica, y así mejorar y automatizar los procesos de análisis y descripción de resultados. En general, se recomienda formalizar toda la información disponible en función de las fuentes de información existentes.

(b) En cuanto al análisis y caracterización del error: Los esfuerzos se focalizan en realizar un análisis profundo del error en segmentación, donde se extrae nuevo conocimiento a partir del uso de técnicas de inteligencia artificial. Dada la descripción individualizada de los objetos de agrupación de error mediante conjuntos multidimensionales de características, se aplican diferentes análisis para detectar distintos tipos de error (errores aislados con algoritmos de detección de outliers y

patrones de error con algoritmos de clustering subespacial). Los análisis se realizan tanto respecto a una referencia fiable, como respecto a otro sistema alternativo del que se desea aprender. En cada caso, se han identificado los tipos de error posibles, lo que permite simplificar el análisis y obtener patrones de error relevantes.

Lo ideal sería poder asociar estos patrones de error con sus causas, sin embargo, esto requiere formalizar mucho conocimiento del dominio. Esta tesis ha dado los primeros pasos en la línea de proporcionar herramientas para facilitar la detección de estos patrones de manera sencilla, pero el control recae todavía en los expertos.

(c) En cuanto a la exploración interactiva de los resultados: se definen vistas personalizadas para gestionar la gran cantidad de resultados de evaluación que se generan y se facilita la interacción con otros sistemas de visualización para analizar los comportamientos de error relevantes en su contexto. Los informes visuales e interactivos facilitan la exploración y la interpretación de los distintos análisis realizados, lo que facilita la toma de decisiones.

También se ha implementado una herramienta informática siguiendo la metodología AMOSE^2 para la caracterización del error en segmentación en objetos amorfos. De forma genérica nos referimos a ella como sistema AMOSE^2 y está compuesta por una serie de herramientas que llevan a cabo los diferentes módulos descritos en la metodología.

(a) Para el módulo de descripción y análisis de agrupaciones de error, se ha desarrollado un prototipo denominado "AMOSE^2 analysis". Este software se ha creado mediante código propio en el lenguaje MATLAB junto a herramientas existentes en la comunidad científica (ELKI, WEKA, R). El software se encarga de describir los objetos segmentados, combinarlos para crear objetos de agrupación y lanzar la ejecución de los algoritmos de análisis avanzados para extraer nueva información que caracterice los errores en la segmentación.

(b) Para la visualización interactiva de los resultados de los análisis, se ha desarrollado "AMOSE^2 web report", una aplicación web en código Python junto a funciones de librerías existentes como pandas, StreamLit, etc. Esta aplicación permite gestionar e interactuar con los distintos estudios de evaluación que se han realizado.

(c) Para el modelado semántico de las características de los objetos segmentados se ha definido la ontología "Visual Object Feature". Esta ontología ha sido el resultado de la experiencia y del uso del sistema AMOSE^2 para definir las variables descriptivas de los objetos segmentados durante el desarrollo del algoritmo AMOS-2D. Su objetivo es contribuir a la organización de las variables explicativas del error, de manera

que se pueda utilizar tanto información estadística como semántica para la toma de decisiones y, en un futuro, automatizar todo el proceso. Por ejemplo, la ontología ha permitido utilizar información semántica para mejorar el proceso de selección de variables en los algoritmos de clustering, para realizar análisis jerarquizados y para ordenar las explicaciones.

5.3. Trabajo futuro

Esta tesis doctoral se puede interpretar como un primer paso hacia la caracterización automática del error en segmentación para ayudar en el desarrollo de sistemas cognitivos de diseño de sistemas de visión artificial. Como trabajo futuro, proponemos las siguientes líneas de mejora:

1. En cuanto a la extensión de la metodología, el prototipo implementado utiliza métricas basadas en medidas de similitud de área con el DSC, y métricas basadas en objetos con el F1-score, por lo que queda, como trabajo futuro, añadir métricas basadas en distancias entre contornos, como, por ejemplo, la distancia de Hausdorff.

   Además, dado que es deseable disponer de información de los procesos internos del algoritmo de segmentación, sería interesante aplicar técnicas de inteligencia artificial explicable a los algoritmos de aprendizaje profundo para convertir los sistemas de caja negra en sistemas de caja gris.

   La metodología es fácilmente extensible a la segmentación de otros tipos de objetos. Bastaría con utilizar las características estructurales de los objetos para definir los tipos de error que se van a caracterizar.

2. En cuanto al análisis de los patrones de error, la metodología AMOSE^2 está abierta a utilizar nuevos algoritmos de reconocimiento de patrones, o a profundizar en la manera de describir los patrones encontrados.

   Por otro lado, en el análisis de error comparado, no se ha realizado el estudio del error de delineación porque no se consideró necesario profundizar en dichos comportamientos de error para el análisis del sistema AMOS-2D durante su desarrollo, pero no cabe duda de que sería una línea de trabajo futuro.

3. En cuanto a la automatización del uso de la metodología, actualmente, muchas decisiones que se toman durante el proceso de análisis recaen en el experto porque, en muchos casos, no se dispone de la información digitalizada y accesible. En el desarrollo de la tesis se ha hecho un esfuerzo, con el módulo de descripción ontológico, para describir formalmente los conceptos manejados dentro de la

metodología AMOSE². Sin embargo, solo se ha hecho un uso parcial de la información recogida en este módulo, pues para la automatización completa del proceso de análisis del error, "TODO" el conocimiento debería estar formalizado, y no es el caso. Un futuro trabajo sería integrar los distintos módulos en una única plataforma que permitiera mayor interacción y un uso mayor de razonamiento ontológico lo que contribuiría a diseñar finalmente un sistema cognitivo.

Otra acción de mejora relacionada con la automatización consistiría en integrar esta metodología dentro de un sistema de control de versiones, de manera que se pudieran organizar y comparar distintas soluciones de segmentación de forma dinámica y automática.

4. En cuanto al uso, también sería interesarte utilizar la metodología AMOSE² para evaluar otros SVC de segmentación de imágenes de diferentes contextos, como por ejemplo, con los conjuntos de imágenes disponibles de forma pública de "Berkeley Segmentation Dataset and Benchmark" (BSDB) o ADE20K (Mittal et al. [2021]). También se podría utilizar su funcionalidad y evaluación exhaustiva en nuevos retos de segmentación de la comunidad académica y científica.

   Finalmente, también se podrían considerar otros usos de la metodología, no solo para caracterizar el error a la hora de refinar un sistema de segmentación durante su diseño, sino para desarrollar un sistema de IA explicable, en el que se caracterizara el error para aumentar la confianza en el sistema.

Apéndice A

El prototipo AMOSE^2

A.1. Descripción y estructura

El prototipo desarrollado para implementar la metodología propuesta "AMOSE^2" ha seguido los principios de reusabilidad, sencillez y explicabiliad que se comentaron en el capítulo 1. El sistema utiliza diferentes lenguajes de programación (Matlab, Python, R) y reutiliza código y herramientas de uso libre cuando es posible.

La forma de evaluar los resultados de la segmentación de objetos para presentarlos de una manera sencilla y explicable, se realiza mediante un análisis variado de los datos mediante las distintas técnicas implementas en el sistema AMOSE^2. Sus diagrama se muestra en la figura A.1. Se compone, por un lado, de una herramienta de descripción y análisis de objetos desarrollada con el software MATLAB, con código propio, denominada "AMOSE^2 analysis" y que utiliza funciones y métodos de otras herramientas (WEKA, ELKI y R) y, por otro lado, de una herramienta de visualización implementada en lenguaje Python para explorar los resultados de la caracterización y los objetos de error en su contexto, las imagenes de RM. Su objetivo es realizar una búsqueda y descripción de nuevo conocimiento sobre el error de segmentación de objetos.

El sistema AMOSE^2, a partir de las imágenes de segmentación, realiza en primer lugar una comparación entre los objetos segmentados de la propuesta del SVC y la referencia, y crea los nuevos objetos de agrupación $O_{R.P}$ para su identificación, caracterización y clasificación. Si existe información asociada a las imágenes de segmentación como las imágenes originales, imágenes de estructuras cerebrales o vectores de características de los objetos segmentados proporcionados en el procesos de segmentación, se enriquece la caracterización de los objetos de la agrupación. En segundo lugar, a partir de la caracterización múltiple de los objetos de agrupación de error, se realiza una exploración con algoritmos de IA para la búsqueda de patrones de error y errores aislados. En este trabajo se ha utilizado el algoritmo CLIQUE, utilizando el algoritmo implementado en el

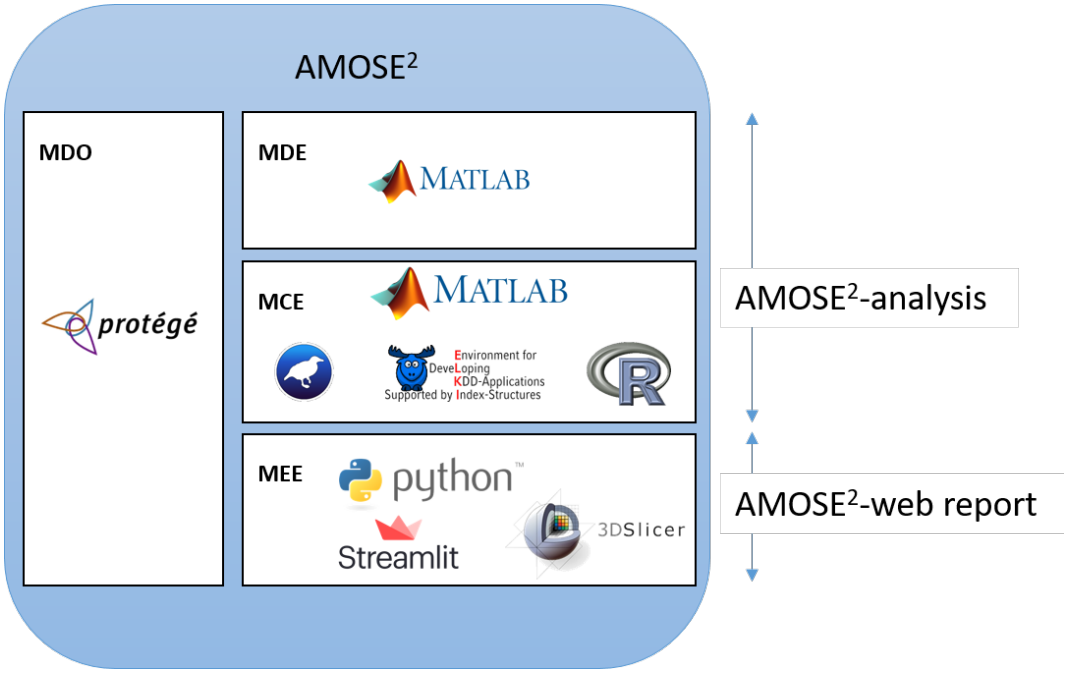


Figura A.1: Diagrama de relaciones entre las herramientas utilizadas en el prototipo AMOSE²

software ELKI e implementando su última etapa en Matlab, para la detección de bolsas de error multidimensionales. Para la detección de errores aislados n-dimensionales se han utilizado algunos de los algoritmos implementados en el paquete "OutlierO3" de R. La preparación de los conjuntos de datos de error se ha realizado mediante el software WEKA, donde se han normalizado los datos y realizado diferentes experimentos de selección de características para proponer vectores de dimensión reducida que faciliten su análisis e interpretación.

Por otro lado, para la visualización de resultados se ha desarrollado una aplicación web denominada "AMOSE² web report", implementada en lenguaje Python y con librerías como Streamlit para la interfaz visual o las librerías pandas, numpy o PIL para la adquisición, gestión, manipulación y representación de los datos, entre otras.

La visualización de las imágenes de resonancia magnética cerebral, tanto las imágenes originales como las imágenes segmentadas, se realiza con el software 3DSlicer. Y se ha utilizado el software Protégé para definir una ontología de aplicación para modelar las características descriptivas de los objetos segmentados para mejorar el algoritmo AMOS-2D.

A.2. La herramienta "AMOSE^2 analysis"

La herramienta que implementa el módulo de descripción de objetos y el módulo de caracterización del error se ha realizado principalmente con el software Matlab junto a otras herramientas de acceso libre como Weka, R o ELKI.

Esta herramienta necesita que se configure el experimento que se desea realizar para evaluar de forma supervisada la segmentación de un SVC. Para ello, en primer lugar hay que indicar el directorio de salida (dirOut) y la localización de las imágenes binarias que se desean comparar, esto es, el volumen con la segmentación propuesta por el SVC que se va a analizar (IbinaryP) y el volumen de la segmentación de referencia (IbinaryR) respecto a la que se va a evaluar. Para introducir características de intensidad se necesita localizar la imagen original, en este caso las imágenes de RM que contiene el volumen FLAIR (IorigFLAIR). Si existe caracterización interna del SVC, hay que indicar la estructura de información que contiene el vector de características de cada objeto (FeatP, FeatR) para que sean incorporadas en la descripción de los objetos de agrupación. Además, si las características se han descrito mediante una estructura semántica de sus términos (AppOntology) hay que indicar la ubicación del fichero de la ontología. Una vez caracterizados los objetos de agrupación, se utiliza el valor de umbral de éxito ($\mathrm{Th}_{\mathrm{Success}}$) para clasificar los objetos de agrupación por tipo de error en delineación.

En segundo lugar, para la caracterización del error hay que configurar los parámetros de las distintas técnicas utilizadas de aprendizaje no supervisado para detectar agrupaciones de error o anomalías. En la preparación de los datos se configura el listado de atributos seleccionados (SelAtt) para reducir el espacio de datos y mejorar el computo de los algoritmos.Para las agrupaciones se utiliza el algoritmo CLIQUE y hay que configurar el número de intervalos en los que se divide el espacio de datos (CLIQUE_ξ) y la densidad mínima de la rejilla (CLIQUE_τ). Para seleccionar aquellos clusters que son relevantes respecto a los objetos de éxito hay que configurar el valor mínimo de la tasa de éxito-error por cada tipo de error (CLIQUE_ESR"Miss","Extra", "Imperfect"). Para la detección de outlier hay que configurar el valor k del método IQR para una dimensión (IQR_k) y los parámetros de los métodos n-dimensionales del paquete OutlierO3 (O3_c1, c2,c3) junto al tipo de voto para la detección de un outlierMD robusto (TipoVoto), ya sea por mayoría o la intersección.

A.3. La herramienta "AMOSE^2 web report"

La visualización e interpretación de resultados tras el análisis de datos con técnicas de IA es una tarea fundamental en la fase de evaluación de cualquier sistema computacional.

Para facilitar dicha tarea, dentro del sistema AMOSE^2 se ha implementado una aplicación web que permite: a) visualizar el resultado global del análisis comparativo entre diferentes segmentaciones de objetos amorfos; b) realizar una exploración de los distintos objetos a través de la selección y filtrado de sus características visuo-espaciales y, si están disponibles, de características procedimentales, calculadas internamente por el sistema de segmentación (SVC de caja gris); y c) conocer los patrones de error y los errores aislados calculados por diferentes algoritmos dentro del marco de evaluación desarrollado.

A.3.1. Características del software

La aplicación ”AMOSE^2 web report” es un interfaz de visualización compuesto por varias páginas que muestran tablas, gráficas y descripciones textuales de los resultados de evaluación de la comparación de dos o más soluciones de segmentación de objetos amorfos. Está implementado en Python^{TM} []Rossum2011(un lenguaje de programación interpretado, de fácil aprendizaje y de código abierto) junto a la librería Streamlit [Str, 2021], que permite de una forma rápida y sencilla visualizar e interaccionar con los resultados del análisis en un navegador web. También se han utilizado otras librerías para la adquisición, procesado y representación de los datos, como pandas, numpy, plotly, awesome_streamlit o PIL, entre otras.

Esta aplicación permite también visualizar en su contexto los objetos segmentados por las soluciones comparadas mediante la aplicación externa 3D Slicer Fedorov et al. [2012], un software gratuito, abierto y multiplataforma para visualizar y procesar imágenes biomédicas.

A.3.2. Descripción de representaciones gráficas complejas

Gráfico radar o araña

Un gráfico radar es una representación visual de múltiples variables, una por cada radio, donde se divide el espacio de salida en un número de divisiones, los anillos concéntricos. Es útil para describir un cluster compuesto de múltiples hipercubos (la salida del algoritmo CLIQUE) ya que permite su representación con multiplexes variables y combinaciones. En la figura A.2 se muestra la descripción de un cluster de una región descrito con 3 atributos con un diagrama de araña o gráfico de radar. El espacio de datos es de 36 variables cada una dividida en 20 intervalos.

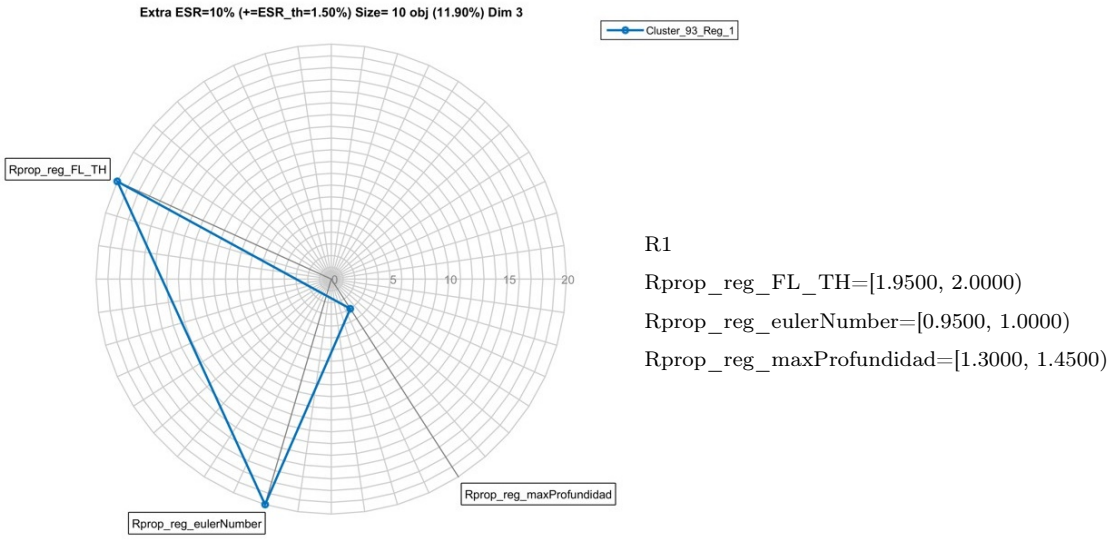


Figura A.2: Descripción de un cluster maximal relevante de tres dimensiones y 1 región. Gráfico radar (dcha) y Texto (izda)

Mapa de calor

Un mapa de calor (heatmap) es una representación gráfica bidimensional de colores para indicar el nivel de actividad. En esta tesis, se utiliza para representar los atributos que describen los clusters descubiertos por el algoritmo CLIQUE, y aunque sea una representación compleja permite ver de un rápido visitado mucha información.

En la figura A.3 se muestran un ejemplo de mapa de calor que representa a todos los cluster detectados por CLIQUE para el subconjunto de datos Miss de "Oslo_AMOS-2D' y con la selección de atributos "SelAtt3". La información que describe es la siguiente:

Cada columna de la figura (eje X) se corresponde con los atributos del subconjunto de datos, en este ejemplo son 33 variables. Cada fila (eje y) se corresponde con un cluster maximal.

- En el eje X cada columna se corresponde con los atributos del subconjunto de datos, en este caso es de 33 variables.

- En cada fila, se representa un cluster maximal encontrado. En el eje Y, se indica el número de instancias contenidas en el cluster y las variables que participan en su definición se marcan con el color azul oscuro.

- El color de fondo de cada fila (color según la leyenda: azul claro, verde, naranja y amarillo) indica la dimensión del cluster; las filas están organizadas de manera que los clusters de mayor dimensionalidad están más arriba. Esta información que se

puede obtener también contando el número de variables activas, pero el uso de un color de fondo facilita el trabajo.

- Por último, se diferencia entre clusters relevantes (variables marcadas en color rojo) y no relevantes (variables marcadas en azul).

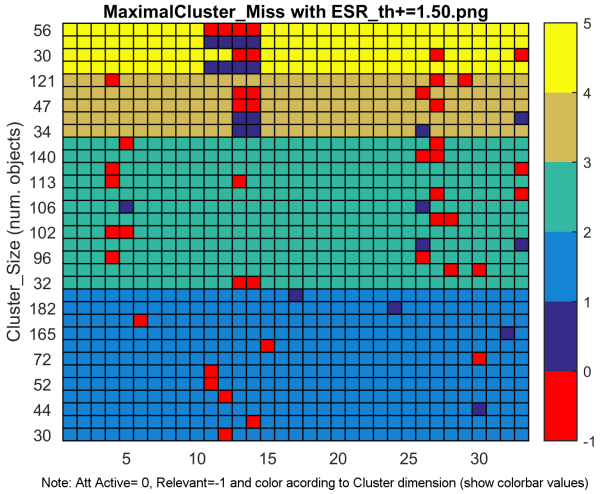


Figura A.3: Descripción de todos los cluster maximales en el subconjunto Miss (en fondo azul oscuro) e identificación de los cluster relevantes (en fondo rojo)

A.3.3. Requisitos de instalación

La instalación local de esta aplicación necesita de los siguientes componentes:

- Python 3. Se puede descargar desde https://www.python.org/downloads/ e instalar seleccionando el enlace correspondiente a la plataforma del sistema operativo o instalar Anaconda, una distribución de Python gratuita que contiene las principales librerías para ciencia de datos (https://www.anaconda.com/products/individual). En este caso se utiliza Anaconda Navigator version 1.9.12 y python versión 3.7.6

- Librerías python. Las librerías utilizadas junto a su versión en la aplicación son las siguientes:

  - awesome-streamlit 20200728.1
  - numpy 1.18.1
  - pandas 1.0.3
  - pillow 7.1.2

  - plotly 4.8.2
  - streamlit 0.82.0

- 3D Slicer. Se utiliza para visualizar los objetos en su contexto, la imagen de resonancia magnética cerebral. Se puede descargar desde https://download.slicer.org/. La versión utilizada en este trabajo es la 4.10.2.

A.3.4. Manual de usuario

La herramienta "AMOSE^2 web report" permite visualizar los resultados de la descripción y caracterización realizada de los objetos de segmentación evaluados. La caracterización utiliza un enfoque local a los objetos de agrupación, $O_{R.P}$, resultantes de establecer la correspondencia a nivel de blob entre las segmentaciones de propuesta y de referencia. Por cada objeto de agrupación resultante se obtiene un conjunto de múltiples características (visuo-espaciales, de composición, internas a los procesos, etc.) que permite explorar los datos y analizar las relaciones existentes entre distintas agrupaciones para detectar patrones de error y situaciones singulares.

La página principal de la aplicación web consta de un menú lateral y una ventana de visualización de resultados como se observa en la figura A.4. En el menú lateral se puede seleccionar entre las distintas páginas de resultados (Overview, Explore, Explore 2 Sol y Cluster/Isolated Error) y configurar el tamaño de la ventana de visualización. En la ventana de visualización se mostraran diferentes vistas en función de la página seleccionada como se describe en detalle a continuación.

Resumen de resultados

En la página "Overview" se muestran los resultados globales de la evaluación en formato tabular. Para ello hay que seleccionar entre el listado de estudios de evaluación realizados, aquel que se desea explorar y que contiene los ficheros de resultados de los objetos comparados, es decir, sus identificadores y sus múltiples características. Tras el selector desplegable se muestra una descripción general del contenido de los ficheros, número de instancias y número de variables, de forma textual. Los estudios pueden contener una o más soluciones comparadas que se identifican por dos descriptores: "ID_Scanner_Location" e "ID_Studio", que describen el conjunto de datos de las imágenes de RM y las soluciones comparadas, respectivamente. En la tabla "Dataset Overview" se muestra el número de casos que se analizan (#Cases), el tamaño de las imágenes (Image Size) y el tamaño de los vóxeles en milímetros (Voxel Size). En "Global Evaluation Results Overview" se muestran tres tablas, las dos primeras con resultados basados en objetos y basados en vóxel, respectivamente. En ellas se muestra la información

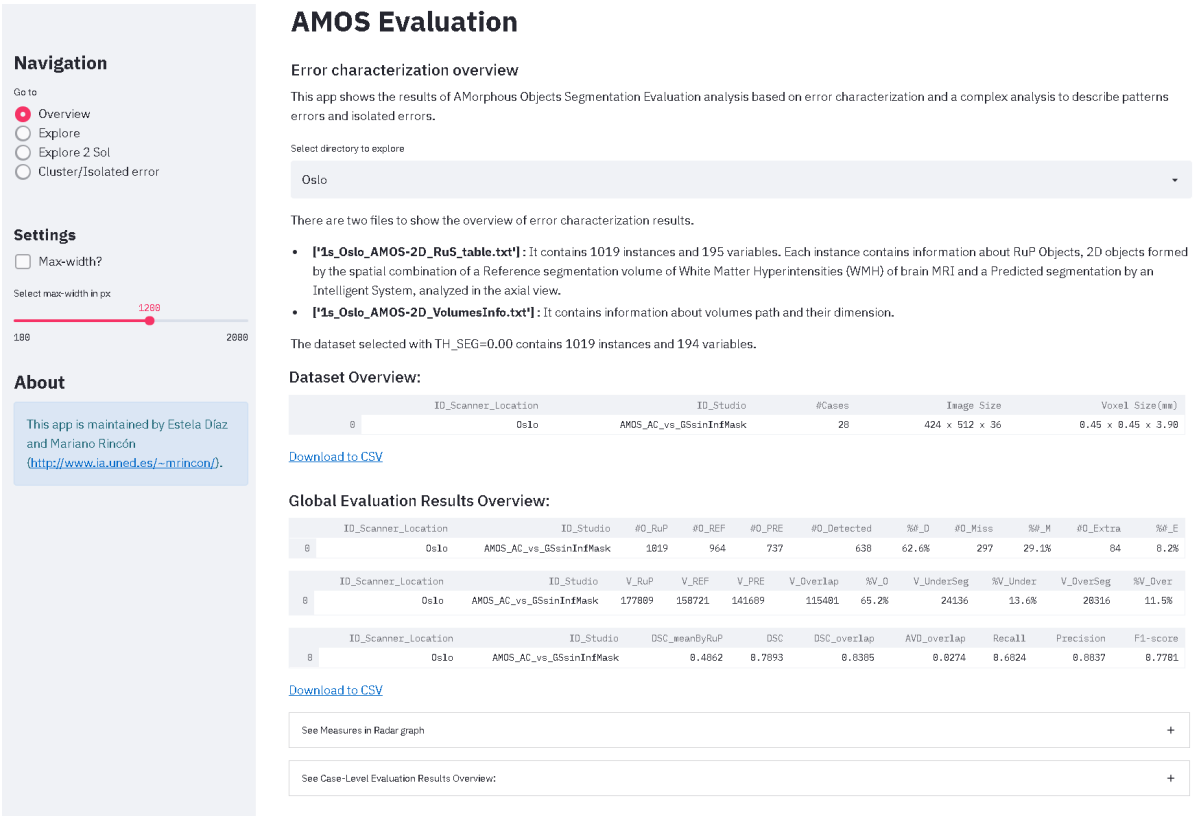
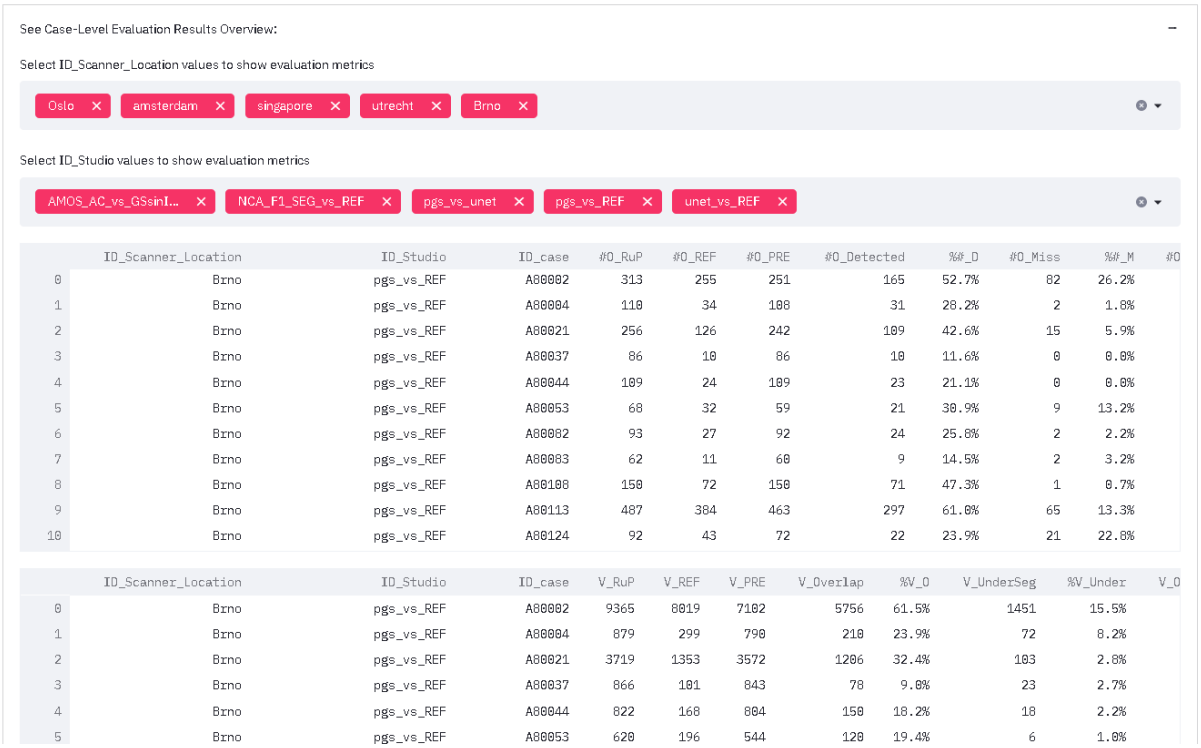


Figura A.4: Página principal: vista resumen a nivel del dataset

de los objetos de agrupación $O_{R.P}$, los objetos de referencia O_R y los objetos de propuesta O_P y de su distribución en función del tipo de error (Detected/Miss/Extra a nivel objeto o Overlap/UnderSeg/OverSeg a nivel vóxel) tanto en número como en volumen. La última tabla presenta medidas estadísticas globales por cada solución analizada. Por un lado medidas basadas en volumen como el coeficiente Soresen-Dice para todos los objetos (DSC) y solo para los objetos con solape (DSC_Overlap) y la diferencia volumétrica media (AVD, por sus siglas en ingles) y por otro lado medidas basadas en objetos como la exhaustividad (Recall), la precisión (Precision) y el valor-F (F1-Score).

Para profundizar en la exploración de resultados globales la aplicación también permite analizar los resultados estadísticos agrupando los datos por caso. Para ello hay que desplegar el panel “See Case-Level Evaluation Results Overview” y si hay más de una solución filtrar por los campos descriptivos del estudio como se observa en la figura A.5. Igual que antes, aparecen las tres tablas con una nueva columna que identifica el caso (ID_Case).

Dado que el uso del formato tabular es muy informativo pero costoso a la hora de comparar resultados se ha añadido la funcionalidad de visualización de los resultados estadísticos globales en un diagrama de araña, lo que permite comparar de forma más



	ID_Scanner_Location	ID_Studio	ID_case	#O_RuP	#O_REF	#O_PRE	#O_Detected	%#_D	#O_Miss	%#_M	#O
0	Brno	pgs_vs_REF	A80002	313	255	251	165	52.7%	82	26.2%	
1	Brno	pgs_vs_REF	A80004	110	34	108	31	28.2%	2	1.8%	
2	Brno	pgs_vs_REF	A80021	256	126	242	109	42.6%	15	5.9%	
3	Brno	pgs_vs_REF	A80037	86	10	86	10	11.6%	0	0.0%	
4	Brno	pgs_vs_REF	A80044	109	24	109	23	21.1%	0	0.0%	
5	Brno	pgs_vs_REF	A80053	68	32	59	21	30.9%	9	13.2%	
6	Brno	pgs_vs_REF	A80082	93	27	92	24	25.8%	2	2.2%	
7	Brno	pgs_vs_REF	A80083	62	11	60	9	14.5%	2	3.2%	
8	Brno	pgs_vs_REF	A80108	150	72	150	71	47.3%	1	0.7%	
9	Brno	pgs_vs_REF	A80113	487	384	463	297	61.0%	65	13.3%	
10	Brno	pgs_vs_REF	A80124	92	43	72	22	23.9%	21	22.8%	

	ID_Scanner_Location	ID_Studio	ID_case	V_RuP	V_REF	V_PRE	V_Overlap	%V_O	V_UnderSeg	%V_Under	V_O
0	Brno	pgs_vs_REF	A80002	9365	8019	7102	5756	61.5%	1451	15.5%	
1	Brno	pgs_vs_REF	A80004	879	299	790	210	23.9%	72	8.2%	
2	Brno	pgs_vs_REF	A80021	3719	1353	3572	1206	32.4%	103	2.8%	
3	Brno	pgs_vs_REF	A80037	866	101	843	78	9.0%	23	2.7%	
4	Brno	pgs_vs_REF	A80044	822	168	804	150	18.2%	18	2.2%	
5	Brno	pgs_vs_REF	A80053	620	196	544	120	19.4%	6	1.0%	

Figura A.5: Página principal: desglose a nivel de casos

sencilla diferentes resultados. Para ello hay que desplegar el panel "See Measures in Radar graph" para ver una gráfica como la que se muestra en la figura A.6.

Exploración de resultados

Las páginas "Explore" y "Explore 2 Sol" permiten realizar una exploración de los resultados a nivel objeto de agrupación $O_{R.P}$ a partir de la selección y filtrado de las características que describen a cada objeto y las relaciones entre los objetos de agrupación $O_{R.P}$, de referencia O_R y de propuesta O_P.

La página "Explore" permite seleccionar las instancias y las variables que se desean explorar a través de diferentes funcionalidades (figura A.7). Seleccionando "Show raw data selection", se pueden ver los datos originales en formato tabular; seleccionando "Show selection box plot graphic" (figura A.8a), una representación gráfica en forma de cajas y bigotes (box-plot), donde se muestra el valor máximo, mínimo y los cuartiles Q_1, Q_2 y Q_3 de una variable; seleccionando "See selection aggregation graphic", una representación gráfica de varias variables seleccionando los valores del eje X, eje Y, el color y la función de agregación a utilizar entre 5 posibles: suma, conteo, valor medio, mínimo y máximo (figura A.8b); o seleccionando "Show selection in context via 3D Slicer" (figura A.9), ver una selección de objetos en su contexto a través de la aplicación externa 3D Slicer.

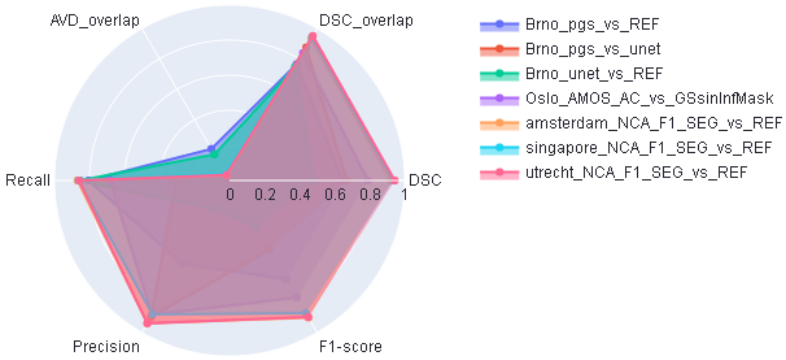


Figura A.6: Ejemplo de figura tipo radar

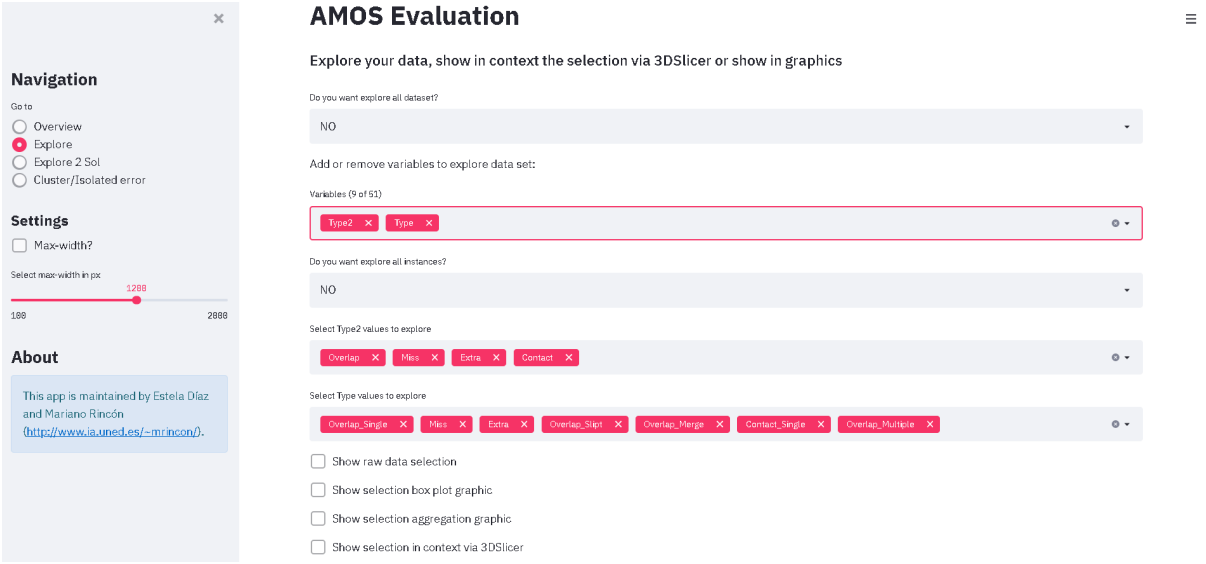
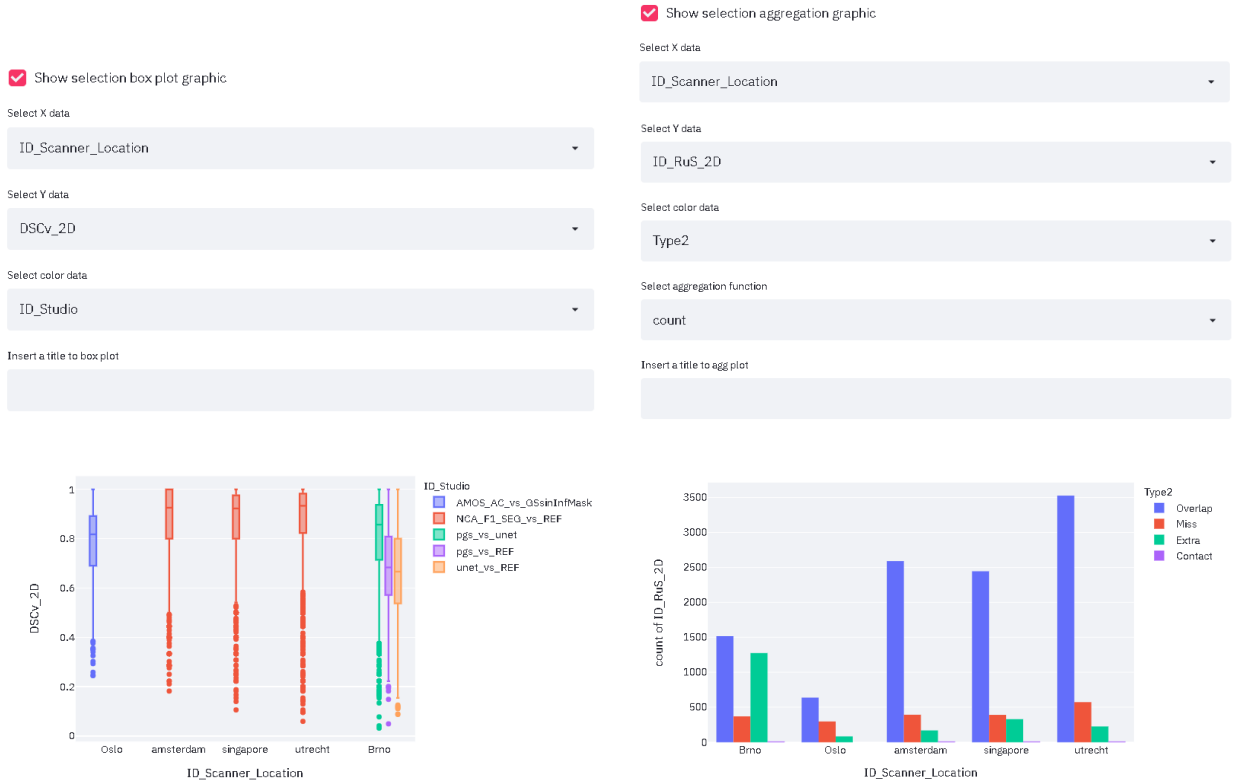
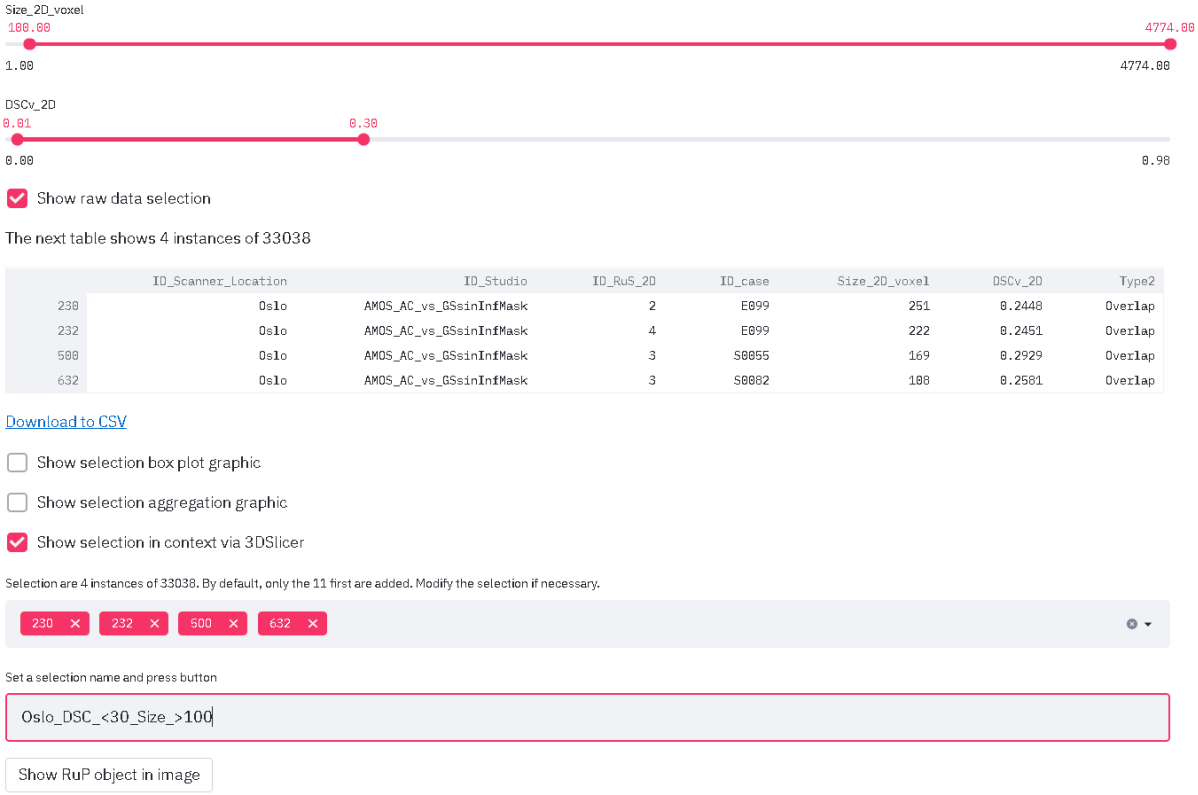


Figura A.7: Página de exploración de datos del modo individual

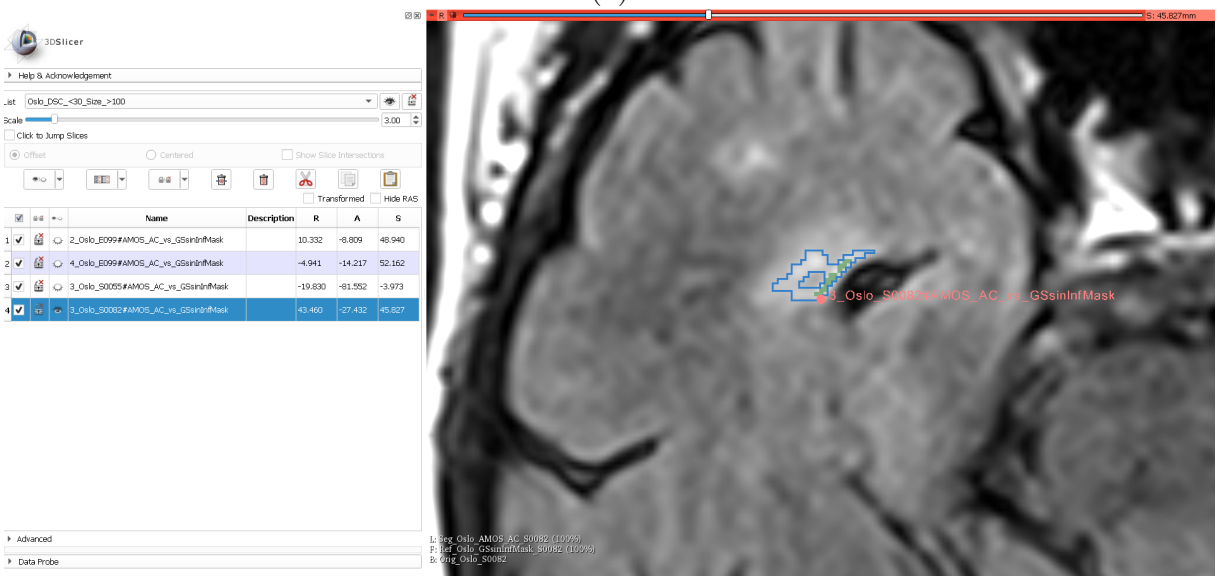


(a) (b)

Figura A.8: Ejemplos de funcionalidad de la página de exploración individual: gráficos box-plot y gráficos de barras



(a)



(b)

Figura A.9: Ejemplos de funcionalidad de la página de exploración individual: visualización de objeto en su contexto

La página "Explore 2 sol" (figura A.10) permite explorar los comportamientos de tres segmentaciones, O_P, O_A y O_R, a partir del enlazado de información a partir de los objetos de agrupación $O_{R.P}$, $O_{R.A}$ y $O_{A.P}$. Mediante la relación entre sus objetos componentes se pueden crear varias soluciones (tablas LOJ) que contiene información útil como el tipo de error, el porcentaje de solape, el tamaño o la posición que permiten detectar comportamientos enlazados de error. En el selector se puede seleccionar el fichero a explorar, donde el nombre del fichero contiene la información de los objetos enlazados. Su orden es importante ya que los identificadores de los objetos componentes de la primera agrupación son los que se utilizan como claves para la reunión de la información.

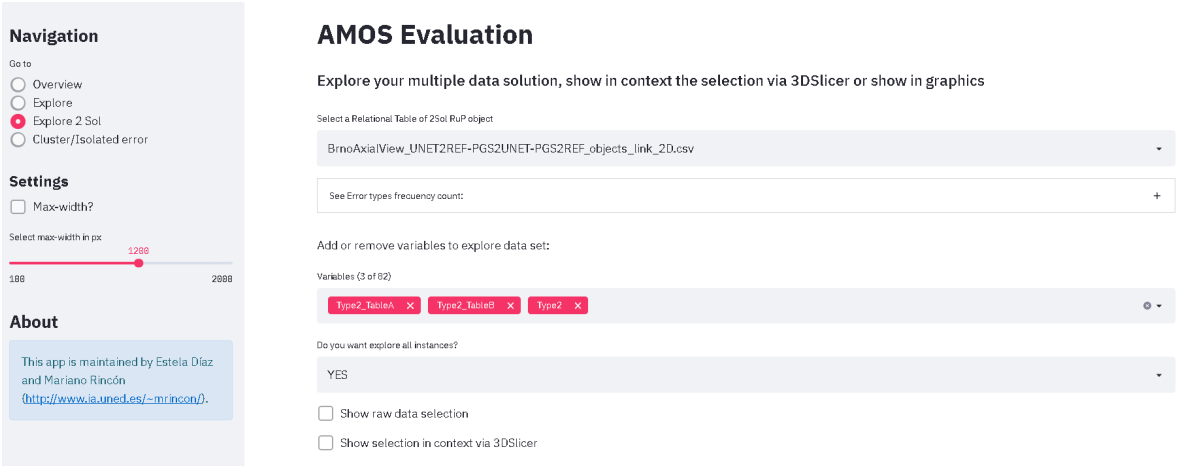


Figura A.10: Página de exploración de datos del análisis comparado

Para conocer los distintos comportamientos de error enlazados (ver sección 3.3.4) que se dan en el fichero explorado, se puede desplegar el panel "See Error types frecuency count" y ver las tres variables que describen el tipo de error de los objetos, con sus valores y su frecuencia, como se muestra en la figura A.11.

Para analizar comportamientos concretos hay que seleccionar las variables a analizar (por defecto el tipo de error), seleccionar sus rangos o valores y, en función de la visualización deseada, seleccionar la opción correspondiente para visualizarlos en forma tabular o en su contexto con 3D Slicer.

Agrupaciones y casos aislados

La página "Cluster / Isolated error" permite visualizar los resultados del análisis de evaluación realizado sobre los conjuntos de datos multidimensionales. Se utiliza una selección de atributos y un umbral de solapamiento según el campo DSC_2D para clasificar los objetos con solape como objetos imperfectos quedando los objetos de agrupación $O_{R.P}$ clasificados por tipo de error como Miss, Extra, Imperfect y Success. Para ello, como se muestra en la figura A.12 hay que elegir el directorio de resultados,

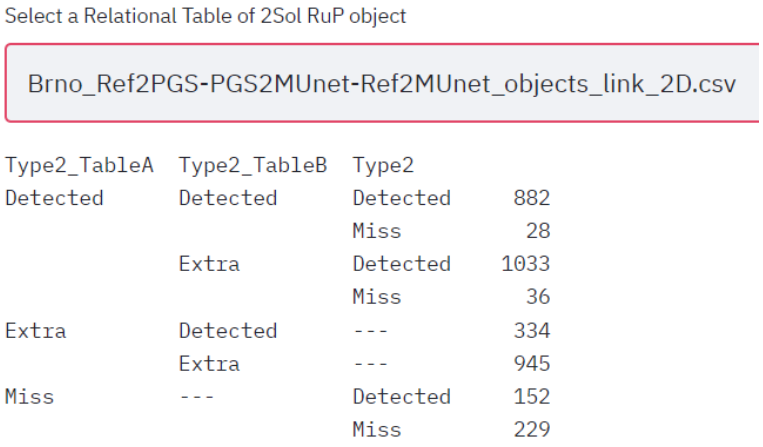


Figura A.11: Ejemplos de funcionalidad de la página de exploración comparada: frecuencia de los comportamientos

seleccionar la configuración utilizada en selección de atributos y el umbral de objeto imperfecto.

En la evaluación se han realizado dos tipos de análisis por tipo de error, búsqueda de agrupaciones y casos aislados, en inglés clustering y outliers, respectivamente. Para realizar el clustering se utiliza el algoritmo CLIQUE (ver 3.3.2.1) con dos parámetros configurables, el número de divisiones de la rejilla y el umbral de densidad mínima, para detectar agrupaciones de objetos de error.

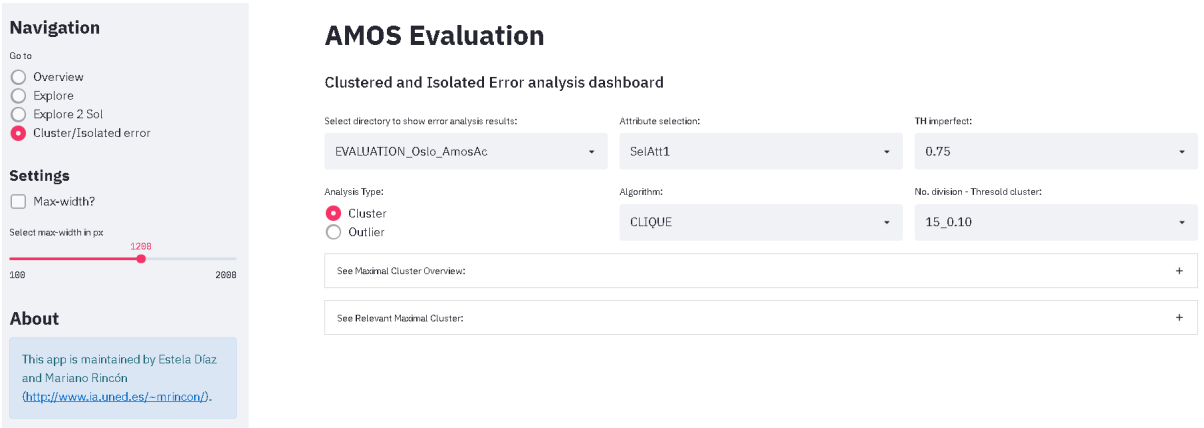


Figura A.12: Página de visualización hallazgos de error: agrupaciones (cluster) y anomalías (outlier)

En el panel "See Maximal Cluster Overview" se pueden observar las múltiples agrupaciones detectadas por cada tipo de error, las variables que la conforman, su tamaño y su relevancia respecto a los objetos de éxito (Success) según se seleccione en el selector de umbral ESR como se muestra en la figura A.13. Cada gráfica muestra en el eje X un identificador de las variables del conjunto de datos analizados, en esta caso su posición,

y en el eje Y las distintas agrupaciones detectadas ordenadas por su dimensión (abajo las de menor número de variables). El color de fondo indica la dimensionalidad de la agrupación y los bloques azules oscuros indican las variables que las describen. Cuando son agrupaciones relevantes, es decir, que superan el umbral ESR seleccionado, se marcan con bloques rojos.

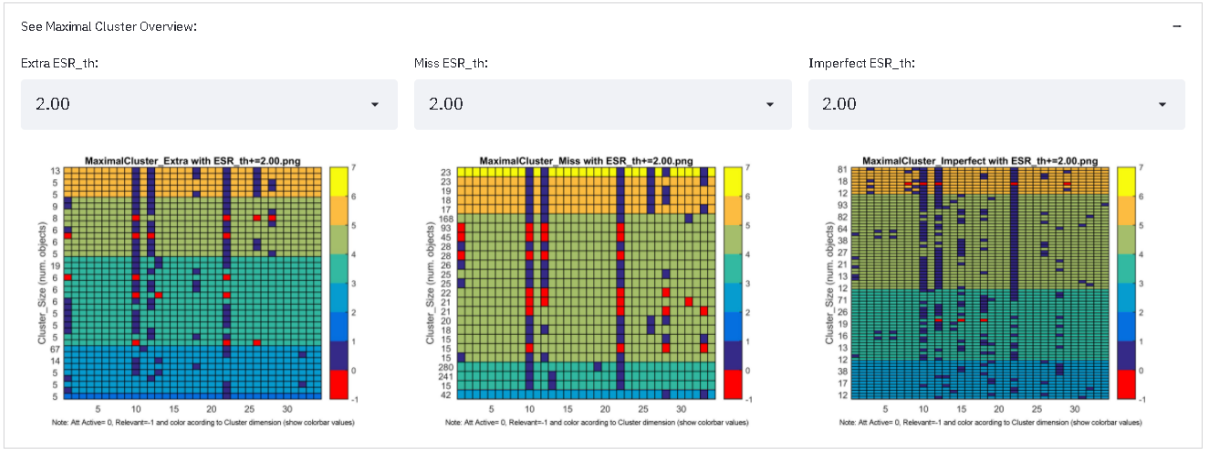
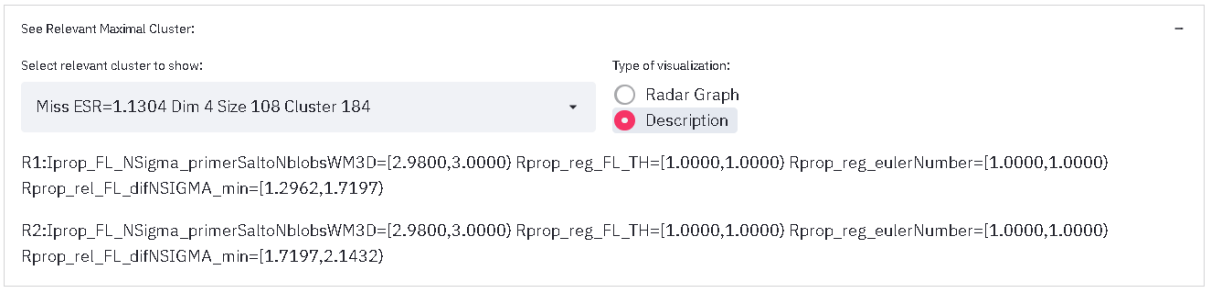


Figura A.13: Ejemplo de funcionalidad de la página de hallazgos de error: resumen de agrupaciones maximales de tipo Extra, Miss e Imperfect

En el panel "See Relevant Maximal Cluster" se puede seleccionar una agrupación relevante y ver su descripción tanto en formato gráfico con un diagrama de araña donde se muestran las variables activas sobre el total de variables y sus rangos según la configuración utilizada (figura A.14a) o de forma textual con una descripción de cada variable y sus rangos activos (figura A.14b).

En el caso de la visualización de los resultados de la búsqueda de casos aislados existen dos tipos de análisis. El análisis unidimensional utilizando el método del rango intercuartil (Test Tukey Tukey [1977]) o el análisis multidimensional utilizando el paquete R "OutliersO3" . Para el primer caso hay que seleccionar el factor aplicado para considerar los datos como valor atípico, donde se utiliza un factor de 1.5 para los casos leves y de 3 para los casos extremos. Las variables que tengan datos fuera de dicho rango aparecen en el selector y se puede elegir el tipo de gráfico para ver su comportamiento junto a la descripción textual (figura A.15a). Para el segundo caso hay que seleccionar la configuración utilizada que indica el método de selección de variables y el nivel de tolerancia para cada uno de los tres algoritmos utilizados (HDo, adjOut y MCD). Cada caso considerado outlier se puede visualizar en su contexto junto a su descripción (figura A.15b).

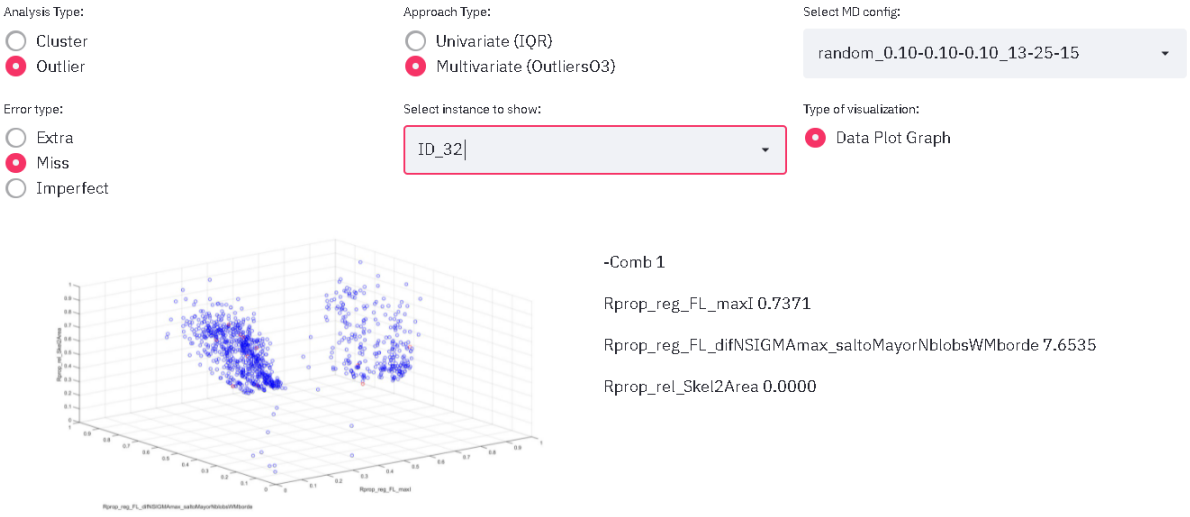
(a)



(b)

Figura A.14: Ejemplo de funcionalidad de la página de hallazgos de error: descripción de una agrupación relevante en formato gráfico (a) o en formato texto (b)

(a)



(b)

Figura A.15: Ejemplo de funcionalidad de la página de hallazgos de error: descripción de anomalías de forma invariante (a) y multivariante (b)

Apéndice B

Las características de los objetos de agrupación $O_{R.P}$

B.1. Descripción de características utilizadas

En el módulo de descripción de objetos de la sistema AMOSE^2 se calculan y se incorporan múltiples características para crear un gran vector por cada objeto de agrupación. En él, se agrega información identificativa de los objetos y de su contexto, características de la agrupación tanto visuo-espaciales, de composición como de medidas de similitud espacial. Por último, si el SVC es de caja gris y ofrece información de los objetos segmentados en etapas internas y/o información del proceso de computo, también se agrega su información al vector de características.

Para el estudio de evaluación "Oslo_AMOS-2D" con el sistema AMOSE^2, tenemos un vector de características de 186 variables. Las variables 1-7 son atributos descriptivos, los atributos 8-24 son atributos visuo-espaciales de los objetos de agrupación, los atributos 25-167 provienen del algoritmo AMOS-2D (son características de los objetos ancestros) y los atributos de 168-186 se corresponden con características topológicas y medidas de similitud espaciales de los objetos de agrupación. Se presentan en dos tablas la descripción de las variables debido a la longitud de los nombres de las variables. En la tabla B.1 se presenta el identificador del atributo y su nombre, mientras que en B.2 el identificador y la descripción almacenada en las ontologías utilizadas. Las características identificativas comienzan por "ID_", las de los objetos de agrupación por "AGprop_" y las que proceden del sistema AMOS-2D comienzan por "Iprop_" si describen características de toda la imagen o por "Rprop_" si describen características del objeto segmentado. Para el resto de estudios, como proceden de sistemas de caja gris, el número de atributos que describen a los objetos de agrupación es menor, sin las características del SVC solo se tiene en la versión actual del prototipo un vector de 44 características.

En la tabla B.1 se ha añadido la columna "ID_filtered" para indicar la correspondencia entre el identificador y la variable que se utiliza en el módulo de caracterización del error. El filtrado de variables es necesario, por un lado, para preparar los conjuntos de datos y por otro lado, para analizar la relevancia de las variables. Para separar el conjunto de datos por tipo de error, en la preparación de los datos se utiliza el atributo DSCv, y para aplicar aprendizaje no supervisado en el análisis de relevancia de características, hay que eliminar atributos redundantes y los que no aporten información, por ello, se eliminan las variables identificativas, variables duplicadas y las medidas de similitud espacial.

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	**Id_filtered**	**Nombre**
1		ID_Scanner_Location
2		ID_Studio
3		TH_SEG
4		ID_Config
5		ID_RuS_2D
6		ID_RuS_3D
7		ID_case
8		AGRprop_Position_3D_1
9		AGRprop_Position_3D_2
10		AGRprop_Position_3D_3
11	1	AGRprop_Position_1er_voxel_3D_1
12	2	AGRprop_Position_1er_voxel_3D_2
13	3	AGRprop_Position_1er_voxel_3D_3
14		AGRprop_IJK_1
15		AGRprop_IJK_2
16		AGRprop_IJK_3
17		AGRprop_Size_2D_mm3
18	4	AGRprop_AGRprop_Size_2D_voxel
19	5	AGRprop_MajorAxisLength_2D
20	6	AGRprop_MinorAxisLength_2D
21	7	AGRprop_Orientation_2D
22	8	AGRprop_MeanIntensity_2D
23	9	AGRprop_MinIntensity_2D
24	10	AGRprop_MaxIntensity_2D
25	11	Iprop_meanWM

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	Id_filtered	Nombre
26	12	Iprop_stdWM
27	13	Iprop_WMsize
28	14	Iprop_WMsizeL
29	15	Iprop_WMsizeR
30	16	Iprop_numBlobs
31	17	Iprop_FL_NSigma_max
32	18	Iprop_FL_NSigma_saltoMayorNblobsWM
33	19	Iprop_FL_NSigma_saltoMayorNblobsWMborde
34	20	Iprop_FL_NSigma_primerSaltoNblobsWM
35	21	Iprop_FL_NSigma_primerSaltoNblobsWMborde
36	22	Iprop_FL_NSigma_areasegWM
37	23	Iprop_FL_NSigma_areasegWMborde
38	24	Iprop_FL_NSigma_saltoMayorNblobsWM3D
39	25	Iprop_FL_NSigma_saltoMayorNblobsWMborde3D
40	26	Iprop_FL_NSigma_primerSaltoNblobsWM3D
41	27	Iprop_FL_NSigma_primerSaltoNblobsWMborde3D
42	28	Iprop_FL_NSigma_areasegWM3D
43	29	Iprop_FL_NSigma_areasegWMborde3D
44	30	Rprop_reg_FL_meanI
45	31	Rprop_reg_FL_stdI
46	32	Rprop_reg_FL_minI
47	33	Rprop_reg_FL_maxI
48	34	Rprop_reg_FL_cmI
49	35	Rprop_reg_FL_NSIGMA_max
50	36	Rprop_reg_FL_NSIGMA_min
51	37	Rprop_reg_FL_NSIGMA_mean
52	38	Rprop_reg_FL_NSIGMAdistrib
53	39	Rprop_reg_FL_difNSIGMAmax_saltoMayorNblobsWM
54	40	Rprop_reg_FL_difNSIGMAmean_saltoMayorNblobsWM
55	41	Rprop_reg_FL_difNSIGMAmin_saltoMayorNblobsWM
56	42	Rprop_reg_FL_difNSIGMAmax_saltoMayorNblobsWMborde
57	43	Rprop_reg_FL_difNSIGMAmean_saltoMayorNblobsWMborde
58	44	Rprop_reg_FL_difNSIGMAmin_saltoMayorNblobsWMborde

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	Id_filtered	Nombre
59	45	Rprop_reg_FL_difNSIGMAmax_primerSaltoNblobsWM
60	46	Rprop_reg_FL_difNSIGMAmean_primerSaltoNblobsWM
61	47	Rprop_reg_FL_difNSIGMAmin_primerSaltoNblobsWM
62	48	Rprop_reg_FL_difNSIGMAmax_primerSaltoNblobsWMborde
63	49	Rprop_reg_FL_difNSIGMAmean_primerSaltoNblobsWMborde
64	50	Rprop_reg_FL_difNSIGMAmin_primerSaltoNblobsWMborde
65	51	Rprop_reg_FL_difNSIGMAmax_areasegWM
66	52	Rprop_reg_FL_difNSIGMAmean_areasegWM
67	53	Rprop_reg_FL_difNSIGMAmin_areasegWM
68	54	Rprop_reg_FL_difNSIGMAmax_areasegWMborde
69	55	Rprop_reg_FL_difNSIGMAmean_areasegWMborde
70	56	Rprop_reg_FL_difNSIGMAmin_areasegWMborde
71	57	Rprop_reg_FL_difNSIGMAmax_saltoMayorNblobsWM3D
72	58	Rprop_reg_FL_difNSIGMAmean_saltoMayorNblobsWM3D
73	59	Rprop_reg_FL_difNSIGMAmin_saltoMayorNblobsWM3D
74	60	Rprop_reg_FL_difNSIGMAmax_saltoMayorNblobsWMborde3D
75	61	Rprop_reg_FL_difNSIGMAmean_saltoMayorNblobsWMborde3D
76	62	Rprop_reg_FL_difNSIGMAmin_saltoMayorNblobsWMborde3D
77	63	Rprop_reg_FL_difNSIGMAmax_primerSaltoNblobsWM3D
78	64	Rprop_reg_FL_difNSIGMAmean_primerSaltoNblobsWM3D
79	65	Rprop_reg_FL_difNSIGMAmin_primerSaltoNblobsWM3D
80	66	Rprop_reg_FL_difNSIGMAmax_primerSaltoNblobsWMborde3D
81	67	Rprop_reg_FL_difNSIGMAmean_primerSaltoNblobsWMborde3D
82	68	Rprop_reg_FL_difNSIGMAmin_primerSaltoNblobsWMborde3D
83	69	Rprop_reg_FL_difNSIGMAmax_areasegWM3D
84	70	Rprop_reg_FL_difNSIGMAmean_areasegWM3D
85	71	Rprop_reg_FL_difNSIGMAmin_areasegWM3D
86	72	Rprop_reg_FL_difNSIGMAmax_areasegWMborde3D
87	73	Rprop_reg_FL_difNSIGMAmean_areasegWMborde3D
88	74	Rprop_reg_FL_difNSIGMAmin_areasegWMborde3D
89	75	Rprop_reg_FL_TH
90	76	Rprop_reg_T1_meanI
91	77	Rprop_reg_T1_stdI

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	Id_filtered	Nombre
92	78	Rprop_reg_T1_minI
93	79	Rprop_reg_T1_maxI
94	80	Rprop_reg_T1_cmI
95	81	Rprop_reg_DistToVent_meanI
96	82	Rprop_reg_DistToVent_stdI
97	83	Rprop_reg_DistToVent_minI
98	84	Rprop_reg_DistToVent_maxI
99	85	Rprop_reg_DistToVent_cmI
100	86	Rprop_reg_DistToBorder_meanI
101	87	Rprop_reg_DistToBorder_stdI
102	88	Rprop_reg_DistToBorder_minI
103	89	Rprop_reg_DistToBorder_maxI
104	90	Rprop_reg_DistToBorder_cmI
105	91	Rprop_reg_DistToSkel_meanI
106	92	Rprop_reg_DistToSkel_stdI
107	93	Rprop_reg_DistToSkel_minI
108	94	Rprop_reg_DistToSkel_maxI
109	95	Rprop_reg_DistToSkel_cmI
110	96	Rprop_reg_DistToWMBorder_meanI
111	97	Rprop_reg_DistToWMBorder_stdI
112	98	Rprop_reg_DistToWMBorder_minI
113	99	Rprop_reg_DistToWMBorder_maxI
114	100	Rprop_reg_DistToWMBorder_cmI
115	101	Rprop_reg_DistToCHULL_meanI
116	102	Rprop_reg_DistToCHULL_stdI
117	103	Rprop_reg_DistToCHULL_minI
118	104	Rprop_reg_DistToCHULL_maxI
119	105	Rprop_reg_DistToCHULL_cmI
120	106	Rprop_reg_DistToRH_meanI
121	107	Rprop_reg_DistToRH_stdI
122	108	Rprop_reg_DistToRH_minI
123	109	Rprop_reg_DistToRH_maxI
124	110	Rprop_reg_DistToRH_cmI

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	Id_filtered	Nombre
125	111	Rprop_reg_DistToLH_meanI
126	112	Rprop_reg_DistToLH_stdI
127	113	Rprop_reg_DistToLH_minI
128	114	Rprop_reg_DistToLH_maxI
129	115	Rprop_reg_DistToLH_cmI
130	116	Rprop_reg_DistToCM_meanI
131	117	Rprop_reg_DistToCM_stdI
132	118	Rprop_reg_DistToCM_minI
133	119	Rprop_reg_DistToCM_maxI
134	120	Rprop_reg_DistToCM_cmI
135	121	Rprop_reg_area
136	122	Rprop_reg_Perimeter
137	123	Rprop_reg_Kforma
138	124	Rprop_reg_eccentricity
139	125	Rprop_reg_centroidx
140	126	Rprop_reg_centroidy
141	127	Rprop_reg_majorAxisLength
142	128	Rprop_reg_minorAxisLength
143	129	Rprop_reg_equivDiameter
144	130	Rprop_reg_solidity
145	131	Rprop_reg_extent
146	132	Rprop_reg_eulerNumber
147	133	Rprop_reg_intP
148	134	Rprop_reg_distx
149	135	Rprop_reg_disty
150	136	Rprop_reg_distz
151	137	Rprop_reg_inWM
152	138	Rprop_rel_Skel2Area
153	139	Rprop_reg_anchoWM
154	140	Rprop_rel_coefDistToSkelmin_anchoWM
155	141	Rprop_reg_maxProfundidad
156	142	Rprop_reg_Nsigma
157	143	Rprop_rel_FL_difMeanI

Tabla B.1: Relacion entre identificador y nombre de las características de los objetos de agrupacion

Id	**Id_filtered**	**Nombre**
158	144	Rprop_rel_FL_difMeanI_norm
159	145	Rprop_rel_FL_difNSIGMA_max
160	146	Rprop_rel_FL_difNSIGMA_min
161	147	Rprop_rel_FL_difNSIGMA_mean
162	148	Rprop_rel_FL_difNSIGMA_max_slice
163	149	Rprop_rel_FL_difNSIGMA_mean_slice
164	150	Rprop_rel_T1_difMeanI
165	151	Rprop_rel_T1_difMeanI_norm
166		Rprop_C
167		AGRprop_TPv_2D_mm3
168		AGRprop_FNv_2D_mm3
169		AGRprop_FPv_2D_mm3
170		AGRprop_TPv_2D_voxel
171		AGRprop_FNv_2D_voxel
172		AGRprop_FPv_2D_voxel
173		AGRprop_DSCv_2D
174	152	AGRprop_TPRv_2D
175		AGRprop_OSRv_2D
176		AGRprop_USRv_2D
177		AGRprop_TPoc_2D
178		AGRprop_FNoc_2D
179		AGRprop_FPoc_2D
180		AGRprop_FNocd_2D_min
181		AGRprop_FNocd_2D_max
182		AGRprop_FPocd_2D_min
183		AGRprop_FPocd_2D_max
184		AGRprop_Num_obj_SEG
185		AGRprop_Num_obj_REF
186		AGRprop_Type

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
1	Identifier of adquisition entity location
2	Identifier of studio/experiment
3	Value of threshold to binarize image
4	Identifier of SVC configuration
5	Identifier of agrupation object in 2D (default)
6	Identifier of agrupation object in 3D
7	Identifier of patien
8	Value of agrupation object location in brain, 1st dim of centroid
9	Value of agrupation object location in brain, 2st dim of centroid
10	Value of agrupation object location in brain, 3st dim of centroid
11	Value of agrupation object location in brain, 1st dim of pixel location list
12	Value of agrupation object location in brain, 2st dim of pixel location list
13	Value of agrupation object location in brain, 3st dim of pixel location list
14	Value of agrupation object location in brain, 1st dim of pixel location list in Image coordinate system
15	Value of agrupation object location in brain, 2st dim of pixel location list Image coordinate system
16	Value of agrupation object location in brain, 3st dim of pixel location list Image coordinate system
17	Value of agrupation object size in mm3
18	Value of agrupation object size in voxel
19	Value of agrupation object major axis length
20	Value of agrupation object minos axis length
21	Value of agrupation object orientation
22	Value of agrupation object mean intensity
23	Value of agrupation object min intensity
24	Value of agrupation object max intensity
25	Value of Brain White Matter CubeImage using FL data and select mean.
26	Value of Brain White Matter CubeImage using FL data and select standard deviation.
27	Value of size of Brain White Matter on CubeImage.
28	Value of size of Brain White Matter Left Hemisfere on CubeImage.
29	Value of size of Brain White Matter Right Hemisfere on CubeImage.
30	Value of number of blobs before the classifier on SliceImage.
31	Value of CubeImage intensity using FL normalized data and select maximum.
32	Value of intensity threshold of SliceImage with maximum increase of the number of segmented object using FL normaliced data on Brain White Matter, form [3.6;-0.1:2.8]

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
33	Value of intensity threshold of SliceImage with maximum increase ofthe number of segmented object using FL normaliced data on Brain White Matter Border, form [3.6;-0.1:2.8]
34	Value of intensity threshold of SliceImage that first increase the number of segmented object using FL normaliced data on Brain White Matter, form [3.6;-0.1:2.8]
35	Value of intensity threshold of SliceImage that first increase the number of segmented object using FL normaliced data on Brain White Matter Border, form [3.6;-0.1:2.8]
36	Value of intensity threshold of SliceImage that segment at least one object with 20 points using FL normaliced data on Brain White Matter.Thdefault=2.
37	Value of intensity threshold of SliceImage that segment at least one object with 20 points using FL normaliced data on Brain White Matter Border.Thdefault=2.
38	Value of median intensity threshold of CubeImage with maximum increase ofthe number of segmented object using FL normaliced data on Brain White Matter, form [3.6;-0.1:2.8]
39	Value of median intensity threshold of CubeImage with maximum increase ofthe number of segmented object using FL normaliced data on Brain White Matter Border, form [3.6;-0.1:2.8]
40	Value of median intensity threshold of CubeImage that first increase the number of segmented object using FL normaliced data on Brain White Matter, form [3.6;-0.1:2.8]
41	Value of median intensity threshold of CubeImage that first increase the number of segmented object using FL normaliced data on Brain White Matter Border, form [3.6;-0.1:2.8]
42	Value of median intensity threshold of CubeImage that segment at least one object with 20 points using FL normaliced data on Brain White Matter.Thdefault=2.
43	Value of median intensity threshold of CubeImage that segment at least one object with 20 points using FL normaliced data on Brain White Matter Border.Thdefault=2.
44	Value of object intensity using FL data and select mean.
45	Value of object intensity using FL data and select standard deviation.
46	Value of object intensity using FL data and select minimum.
47	Value of object intensity using FL data and select maximum.
48	Value of object intensity using FL data on mass center point.
49	Value of object intensity using FL normalized data and select maximum.
50	Value of object intensity using FL normalized data and select minimum.
51	Value of object intensity using FL normalized data and select mean.
52	Value of object intensity using FL normalized data and select maximum-mean/mean-min.
53	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmasaltoMayorNblobsWM.
54	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmasaltoMayorNblobsWM.

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
55	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmasaltoMayorNblobsWM.
56	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmasaltoMayorNblobsWMBorde.
57	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmasaltoMayorNblobsWMBorde.
58	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmasaltoMayorNblobsWMBorde.
59	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaprimerSaltoNblobsWM.
60	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaprimerSaltoNblobsWM.
61	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaprimerSaltoNblobsWM.
62	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaprimerSaltoNblobsWMBorde.
63	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaprimerSaltoNblobsWMBorde.
64	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaprimerSaltoNblobsWMBorde.
65	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaareasegWM.
66	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaareasegWM.
67	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaareasegWM.
68	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaareasegWMBorde.
69	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaareasegWMBorde.
70	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaareasegWMBorde.
71	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmasaltoMayorNblobsWM3D.
72	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmasaltoMayorNblobsWM3D.
73	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmasaltoMayorNblobsWM3D.
74	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmasaltoMayorNblobsWMBorde3D.
75	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmasaltoMayorNblobsWMBorde3D.
76	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmasaltoMayorNblobsWMBorde3D.
77	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaprimerSaltoNblobsWM3D.
78	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaprimerSaltoNblobsWM3D.
79	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaprimerSaltoNblobsWM3D.
80	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaprimerSaltoNblobsWMBorde3D.
81	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaprimerSaltoNblobsWMBorde3D.
82	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaprimerSaltoNblobsWMBorde3D.
83	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaareasegWM3D.
84	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaareasegWM3D.

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
85	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaareasegWM3D.
86	Value of difference between RpropregFLNSIGMAmax and IpropFLNSigmaareasegWMborde3D.
87	Value of difference between RpropregFLNSIGMAmean and IpropFLNSigmaareasegWMBorde3D.
88	Value of difference between RpropregFLNSIGMAmin and IpropFLNSigmaareasegWMBorde3D.
89	Control variable with value 1 if segmented with 1er threshold or 2 if second.
90	Value of object intensity using T1 data and select mean.
91	Value of object intensity using T1 data and select standard deviation.
92	Value of object intensity using T1 data and select minimum.
93	Value of object intensity using T1 data and select maximum.
94	Value of object intensity using T1 data on mass center point.
95	Value of distance from object central point to Brain Ventricule and select mean.
96	Value of distance from object central point to Brain Ventricule and select standar deviation value.
97	Value of distance from object central point to Brain Ventricule and select minimum.
98	Value of distance from object central point to Brain Ventricule and select maximum.
99	Value of distance from object central point to Brain Ventricule and select mass center.
100	Value of distance from object central point to Brain Border and select mean.
101	Value of distance from object central point to Brain Border and select standar deviation value.
102	Value of distance from object central point to Brain Border and select minimum.
103	Value of distance from object central point to Brain Border and select maximum.
104	Value of distance from object central point to Brain Border and select mass center.
105	Value of distance from object central point to Brain Sekeleton and select mean.
106	Value of distance from object central point to Brain Sekeleton and select standar deviation value.
107	Value of distance from object central point to Brain Sekeleton and select minimum.
108	Value of distance from object central point to Brain Sekeleton and select maximum.
109	Value of distance from object central point to Brain Sekeleton and select mass center.
110	Value of distance from object central point to Brain White Matter Border and select mean.
111	Value of distance from object central point to Brain White Matter Border and select standard-deviation value.
112	Value of distance from object central point to Brain White Matter Border and select minimum.
113	Value of distance from object central point to Brain White Matter Border and select maximum.
114	Value of distance from object central point to Brain White Matter Border and select mass center value.
115	Value of distance from object central point to Brain Convex Hull and select mean.
116	Value of distance from object central point to Brain Convex Hull and select standar deviation value.

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
117	Value of distance from object central point to Brain Convex Hull and select minimum.
118	Value of distance from object central point to Brain Convex Hull and select maximum.
119	Value of distance from object central point to Brain Convex Hull and select mass center.
120	Value of distance from object central point to Brain Right Hemisfere and select mean.
121	Value of distance from object central point to Brain Right Hemisfere and select standar deviation value.
122	Value of distance from object central point to Brain Right Hemisfere and select minimum.
123	Value of distance from object central point to Brain Right Hemisfere and select maximum.
124	Value of distance from object central point to Brain Right Hemisfere and select mass center.
125	Value of distance from object central point to Brain Left Hemisfere and select mean.
126	Value of distance from object central point to Brain Left Hemisfere and select standar deviation value.
127	Value of distance from object central point to Brain Left Hemisfere and select minimum.
128	Value of distance from object central point to Brain Left Hemisfere and select maximum.
129	Value of distance from object central point to Brain Left Hemisfere and select mass center.
130	Value of distance from object central point to Brain Mass Centerl and select mean.
131	Value of distance from object central point to Brain Mass Centerl and select standar deviation value.
132	Value of distance from object central point to Brain Mass Centerl and select minumum.
133	Value of distance from object central point to Brain Mass Centerl and select maximum.
134	Value of distance from object central point to Brain Mass Centerl and select mass center.
135	Value of object size in points.
136	Value of distance around the boundary of the object.
137	Value of size relation between object perimeter and area, it is related to roundness.
138	Value of eccentricity of object.
139	Value of object centroid of 1er dimension on SliceImage space (R->L);
140	Value of object centroid of 2er dimension on SliceImage space (A->P);
141	Value of maximum axis length of object.
142	Value of minimum axis length of object.
143	Value of equivalent diameter of object.
144	Value of proportion of the pixels in the convex hull that are also in the object. Computed as Area/ConvexArea
145	Value of extent property of object, it is related to size relation between object area and bounding box area.
146	Value of Euler number property of object, it is related to holes of object. 1=No holes, 0=1 hole, -1= 2 holes,. . .
147	Value of position of object point wrt SliceImage space.

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
148	Value of relative position of object wrt WM space on 1er dimension of Axial View (R->L).
149	Value of relative position of object wrt WM space on 2er dimension of Axial View (A->P).
150	Value of relative position of object wrt WM space on 3er dimension of Axial View (I->S).
151	Value of relation between object size in Brain White Matter and object size.
152	Value of relation between object skeleton size and object area.
153	Value of distance to farther WhiteMatter pixel. REVIEW: Not well implemented.
154	Value of relation between RpropregDistToSkelminI and RpropreganchoWM.
155	Value of maximum distance between an object point and object periphery point.
156	Value of mean intensity of object using FL normaliced data wrt Brain White Matter.
157	Value of difference between RpropregFLmeanI and RpropperFLmeanI.
158	Value of difference between RpropregFLmeanI and RpropperFLmeanI, first value normaliced.
159	Value of difference between RpropregFLNSIGMAmax and RpropperFLNSIGMAmax, first value normaliced.
160	Value of difference between RpropregFLNSIGMAmin and RpropperFLNSIGMAmin, first value normaliced.
161	Value of difference between RpropregFLNSIGMAmean and RpropperFLNSIGMAmean, first value normaliced.
162	Value of difference between IpropFLNsigmamax and RpropregFLNSIGMAmax, first value normaliced.
163	Value of difference between IpropFLNsigmamean and RpropregFLNSIGMAmean, first value normaliced.
164	Value of difference between RpropregT1meanI and RpropperT1meanI.
165	Value of difference between RpropregT1meanI and RpropperT1meanI, first value normaliced.
166	Variable with information about an internal classifier. If the value is 1 then object is selected, other cases is discarted.
167	Value of True Positive elements in mm3
168	Value of False Negative elements in mm3
169	Value of False Positive elements in mm3
170	Value of True Positive elements in number of voxel
171	Value of False Negative elements in number of voxel
172	Value of False Positive elements in number of voxel
173	Value of Dice coeficient at pixel-level
174	Value of True Positive rate at pixel-level
175	Value of OverSegmentation Rate rate at pixel-level
176	Value of UnderSegmentation Rate rate at pixel-level
177	Value of True Positive object components

Tabla B.2: Descripción de las características de los objetos de agrupacion

Id	Descripción
178	Value of False Negative object componets
179	Value of False Positive object components
180	Value of False Negative object components min distance
181	Value of False Negative object components max distance
182	Value of False Positive object components min distance
183	Value of False Positive object components max distance
184	Value of number of proposed segmentation objects (OP)
185	Value of number of reference segmentation objects (OR)
186	Error type of agrupation object: [Detected, Miss, Extra]/[Ovelap, Contact, Miss, Extra]/[Success, Imperfect, Miss, Extra]

Apéndice C

Otros resultados con la metodología AMOSE^2

C.1. Descripción de agrupaciones de objetos de otros estudios realizados

En este trabajo, se ha aplicado la metodología AMOSE^2 a otros estudios, en concreto, se han analizado las segmentaciones obtenidas usando el dataset de Brno con el algoritmo AMOS-2D y las obtenidas usando el dataset de Brno, Amsterdam, Singapore y Utrecht con el algoritmo M-Unet.

Los resultados a nivel de objetos, tras aplicar el MDE, se muestran en la tabla C.1. En primer lugar, se presenta la información a nivel de número objetos, donde encontramos los objetos de la referencia "$\#(O_R)$", los objetos de la propuesta "$\#(O_P)$" y los objetos de la agrupación "$\#(O_{R.P})$" resultantes tras la tarea de Generación. Los objetos clasificados como detectados, sin contacto con origen en la referencia y sin contacto con origen en la propuesta designados, como "$\#(O_{Detected})$", "$\#(O_{Miss})$" y "$\#(O_{Extra})$", respectivamente. Además, como información adicional, se aportan los valores relativos de éstos últimos (en %) respecto al número de objetos de agrupación. Hay que destacar que, como estos valores se calculan respecto al número de objetos de agrupación, hay que tener cuidado a la hora de interpretar estudios con porcentajes elevados de objetos extra, ya que desvirtúan la información de las métricas.

En segundo lugar, se presentan los resultados a nivel del volumen de los objetos, es decir, a nivel de vóxeles. Se muestran el número de vóxeles de la referencia "$V(O_R)$", el número de vóxeles de la propuesta "$V(O_P)$" y el número de vóxeles de la agrupación "$V(O_{R.P})$". La información sobre similitud espacial se da mediante el número de vóxeles solapados "$V(O_{Overlap})$", el número de vóxeles infrasegmentados

		AMOS-2D	**M-Unet**		
		Brno	**Amsterdam**	**Singapore**	**Utrech**
Núm. Objeto	$\#(O_R)$	2129	3025	2875	4125
	$\#(O_P)$	1147	2836	2953	3915
	$\#(O_{R.P})$	2458	3152	3165	4322
	$\#(O_{Detected})$	740 (30.1 %)	2590 (82.2 %)	2444 (77.2 %)	3523 (81.5 %)
	$\#(O_{Miss})$	1338 (54.4 %)	393 (12.5 %)	391 (12.4 %)	573 (13.3 %)
	$\#(O_{Extra})$	380 (15.5 %)	169 (5.4 %)	330 (10.4 %)	226 (5.2 %)
Núm. Voxel	$V(O_R)$	33154	61223	127283	166550
	$V(O_P)$	38494	58747	121752	158822
	$V(O_{R.P})$	53360	64781	132641	171880
	$V(O_{Overlap})$	18288 (34.3 %)	55189 (85.2 %)	116394 (87.8 %)	153492 (89.3 %)
	$V(O_{UnderSeg})$	4767 (8.9 %)	4990 (7.7 %)	9817 (7.4 %)	10942 (6.4 %)
	$V(O_{OverSeg})$	6879 (12.9 %)	3144 (4.9 %)	4434 (3.3 %)	4833 (2.8 %)
	$V(O_{Miss})$	10099 (18.9 %)	1044 (1.6 %)	1072 (0.8 %)	2116 (1.2 %)
	$V(O_{Extra})$	13327 (25 %)	414 (0.6 %)	924 (0.7 %)	497 (0.3 %)
Métricas	DSCv	0.5105	0.92	0.9348	0.9435
	$DSCv_{Detected}$	0.7585	0.9314	0.9423	0.9511
	Recall	0.3561	0.8683	0.8621	0.8601
	Precision	0.6607	0.9387	0.881	0.9397
	F1-score	0.4628	0.9021	0.8715	0.8982

Tabla C.1: Resultados generales de las soluciones comparadas respecto a la referencia y entre ellas

"$V(O_{UnderSeg})$", el número de vóxeles sobresegmentados "$V(O_{OverSeg})$", el número de vóxeles no detectados "$V(O_{Miss})$" y el número de vóxeles extra detectados "$V(O_{Extra})$". Además, como información adicional se aportan los valores relativos (en %) respecto al volumen de los objetos de agrupación de todos ellos.

Las medidas anteriores permiten conocer el comportamiento general de la solución, pero son difíciles de manejar para realizar comparaciones o rankings. Para solucionarlo se utilizan métricas clásicas de evaluación supervisada para resumir las medidas basadas en vóxeles y las basadas en objetos. Para las primeras, se utiliza el coeficiente Sørensen-Dice (DSC) y la diferencia media volumétrica (AVD de sus siglas en inglés) y para las segundas, el valor F ("F1-score") y sus componentes, la exhaustividad ("Recall") y la precisión ("Precision").

De esta forma, con pocas métricas se puede conocer el comportamiento general de un estudio. En la ultima parte de la tabla se muestran la media de los DSCs agrupados por caso ("DSC") y la media del DSC agrupado por caso de los objetos con contacto ("$DSC_{Overlap}$"). En el caso de la diferencia media volumétrica (AVD) sólo se calcula para los casos de los objetos de agrupación con contacto ("$AVD_{Overlap}$").

Al observar las tablas se puede ver que, respecto al porcentaje de detección de objetos (“ $\%\#O\_D$”), se tienen resultados muy diversos, con valores que varían entre un 18 % y un 80 % (procede de estudio “Brno-M_Unet” explicado en capítulo 4), que reflejan los diferentes grados de madurez de los SVC. En cuanto a la calidad de la delineación de los objetos detectados según su porcentaje de infra y sobresegmentación (“ $\%V(O_{UnderSeg})$” y “ $\%V(O_{OverSeg})$”), se observa que en todos los estudios es inferior al 15 % y que el tamaño de los objetos sin contacto, tanto de tipo Extra como de tipo Miss, es bajo.

C.2. Análisis comparado entre una segmentación no binaria y la referencia

En este estudio se analizan los resultados del algoritmo de segmentación de caja negra M-Unet sobre tres conjuntos de datos: “Amsterdam”, “Utrecht” y “Singapore”. Se trata de un estudio donde se dispone de las imágenes originales, FLAIR y T1, de 60 pacientes y de las imágenes de una segmentación de referencia de cada uno de ellos. El algoritmo proporciona un volumen de segmentación no binario, con valores entre 0 y 1, por lo que se analiza su comportamiento para diferentes umbrales de binarización respecto a la segmentación de referencia, para una caracterización avanzada del error. Al ser un algoritmo de caja negra sólo se dispone de los atributos básicos para el modelado del error, lo que supone una disminución de la capacidad de detección de patrones de error y errores aislados. También se estudia mediante la forma de uso de comparación enlazada la diferencia de comportamientos del error al seleccionar dos valores de umbral en la segmentación predicha, con el objetivo de conocer más detalles su diferencias.

En la tabla C.2 se muestran las diferentes métricas de evaluación de segmentación de objetos para un conjunto de cinco umbrales. Éstos se han utilizado para crear las máscaras binarias que delimitan las lesiones hiperintensas y realizar los análisis de evaluación respecto a la segmentación de referencia. Se muestra el coeficiente Dice (DSC) a nivel de vóxel para los objetos con solape, el valor F1 a nivel de objetos y el número de objetos extra detectados (N_EXTRA) y no detectados (N_MISS).

Umbral	Promedio de DSC	Promedio de F1Score	Suma de N_EXTRA	Suma de N_MISS
0.1	0.9222	0.8636	825	1325
0.3	0.9215	0.8656	725	1363
0.5	0.9205	0.8653	681	1406
0.7	0.9189	0.8656	632	1436
0.9	0.9169	0.8653	554	1494

Tabla C.2: Análisis de diferentes umbrales al binarizar la propuesta respecto a la referencia

De los valores umbral analizados, el umbral que en promedio mejor detecta las manchas hiperintensas tanto a nivel de vóxeles como a nivel de objetos es el valor de umbral igual a 0.3. Este valor se utiliza para estudiar de forma profunda el error que se produce en los resultados de la segmentación de objetos del algoritmo M-Unet.

Los métricas tradicionales de evaluación se muestra en la figura C.1, separando los estudios por cada conjunto de datos. Se observa que los resultados son buenos, con valores de DSC de los objetos con solape y valores de F1-score mayores a 0.85 en todos los casos. También se observa que el conjunto de datos "singapore" presenta peor comportamiento, con mayor porcentaje de objetos no detectados (MISS).

ID_Scanner_Location	ID_Studio	#O_RuP	#O_REF	#O_PRE	#O_Detected	%#_D	#O_Miss	%#_M	#O_Extra	%#_E
amsterdam	NCAF1_30_vs_REF	3152	3025	2836	2590	82.2%	393	12.5%	169	5.4%
singapore	NCAF1_30_vs_REF	3165	2875	2953	2444	77.2%	391	12.4%	330	10.4%
utrecht	NCAF1_30_vs_REF	4322	4125	3915	3523	81.5%	573	13.3%	226	5.2%

ID_Scanner_Location	ID_Studio	V_RuP	V_REF	V_PRE	V_Overlap	%V_O	V_UnderSeg	%V_Under	V_OverSeg	%V_Over	V_Miss
amsterdam	NCAF1_30_vs_REF	64781	61223	58747	55189	85.2%	4990	7.7%	3144	4.9%	1044
singapore	NCAF1_30_vs_REF	132641	127283	121752	116394	87.8%	9817	7.4%	4434	3.3%	1072
utrecht	NCAF1_30_vs_REF	171880	166550	158822	153492	89.3%	10942	6.4%	4833	2.8%	2116

ID_Scanner_Location	ID_Studio	DSC_meanByRuP	DSC	DSC_overlap	AVD_overlap	Recall	Precision	F1-score
amsterdam	NCAF1_30_vs_REF	0.7197	0.9200	0.9314	0.0307	0.8683	0.9387	0.9021
singapore	NCAF1_30_vs_REF	0.6698	0.9348	0.9423	0.0427	0.8621	0.8810	0.8715
utrecht	NCAF1_30_vs_REF	0.7114	0.9435	0.9511	0.0372	0.8601	0.9397	0.8982

Figura C.1: Resumen de resultados de evaluación

Al analizar cómo se comporta cada conjunto de datos de forma individual, por caso, como se muestra en la figura C.2, se pueden conocer comportamientos de error relevantes mediante la composición de los objetos de agrupación y del valor del coeficiente Dice de los objetos con solape. Las gráficas de arriba pertenecen al conjunto de datos "amsterdam", las del medio a "singapore" y las de abajo a "utrecht". En ellas se puede observar un predominio de objetos con solape y un valor medio del coeficiente Dice alto, lo que significa buena similitud. Pero si se centra el análisis en los casos de error, se observa en "singapore" caso "7_train" un número muy elevado de objetos de agrupación de tipo Extra. En el caso de objetos de agrupación de tipo Miss se observa que tiene valores elevados el caso "9_train" de "utrecht". Al observar los objetos en su contexto, en la figura C.3, se detecta que el caso "singapore&7_train" tiene una orientación de la imagen distinta al resto de imágenes de otros casos y en el caso "utrech&9_train" se observa un caso con un tamaño muy elevado de los ventrículos y un gran número de hiperintensidades.

En las gráficas de la figura C.2 también se puede observar que aparecen objetos de agrupación de tipo "Contact", es decir, de objetos con contacto sin solape, en los tres conjuntos de datos. En concreto 3 objetos de contacto en "amsterdam", 2 objetos en "singapore" y 1 objeto en "utrecht", como se muestra en la tabla C.3, siendo todos ellos de

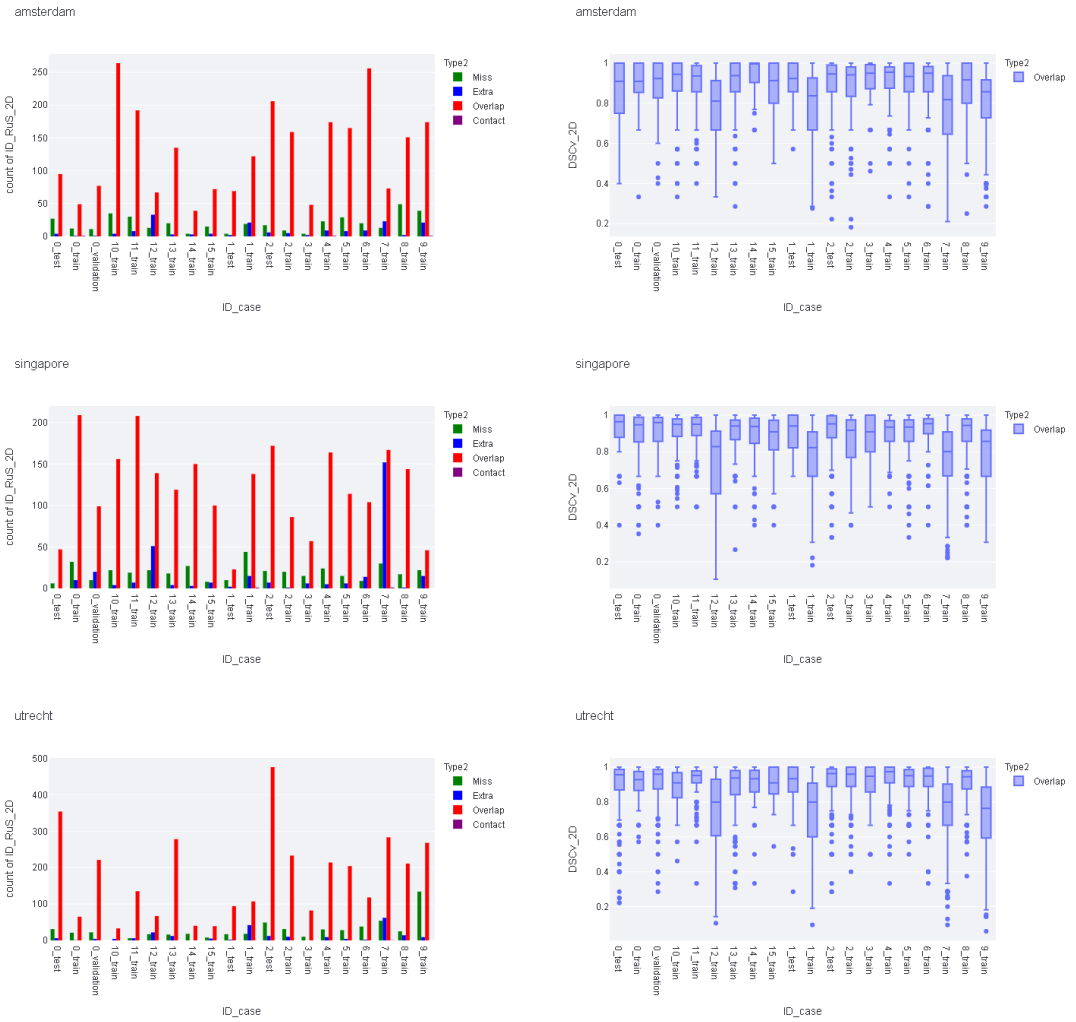


Figura C.2: Distribución del tipo de composición de los objetos de agrupación (izda) y distribución del DSC de objetos de solape (dcha) para los tres conjuntos de datos del challenge

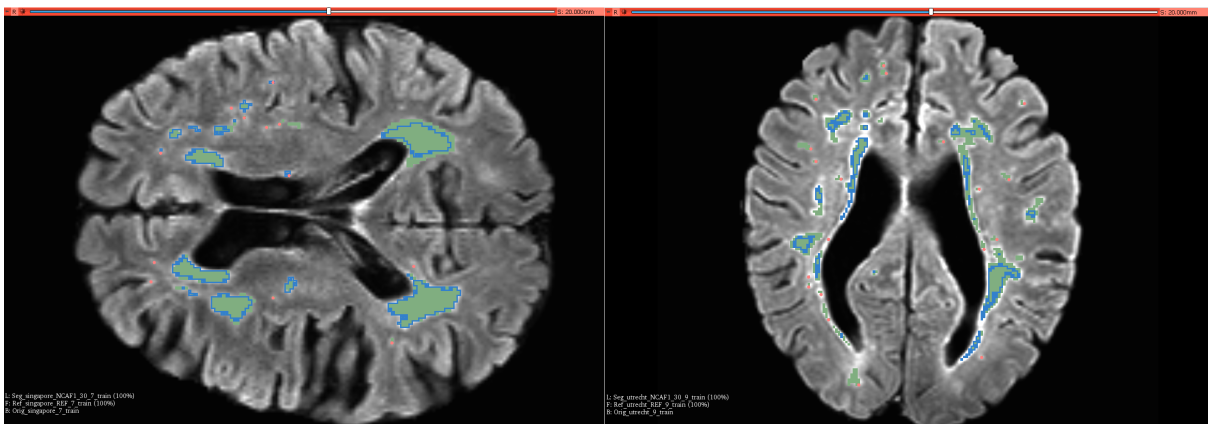


Figura C.3: Casos con objetos de error que destacan respecto al resto

tamaño muy pequeño. En un tipo de error poco común y no deseable, que puede provocar funcionamientos incorrectos, por lo que se deben conocer y eliminar.

ID_Scanner _Location	ID_ RuP _2D	ID _case	RuP _Type	Size _2D _voxel	Size_ R _voxel	Size_P _voxel	Min Intensity _2D	Max Intensity _2D
amsterdam	43	0_train	Contact	3	2	1	0.2707	0.3284
amsterdam	198	4_train	Contact	3	2	1	0.2372	0.2468
amsterdam	197	9_train	Contact	2	1	1	0.3535	0.3622
singapore	6	1_train	Contact	3	2	1	0.2342	0.2566
singapore	104	7_train	Contact	5	3	2	0.2217	0.3203
utrecht	121	7_train	Contact	3	2	1	0.1136	0.1214

Tabla C.3: Información sobre los objetos de agrupación con contacto sin solape

Algunos de estos casos en su contexto se muestran en la figura C.4, lo que permite conocer su ubicación, forma y niveles de intensidad. En ellos se observa que el objeto de segmentación de la propuesta no se ubica en el vóxel de mayor intensidad de su vecindad, probablemente debido a los procesos de convolución del algoritmo utilizado para la segmentación.

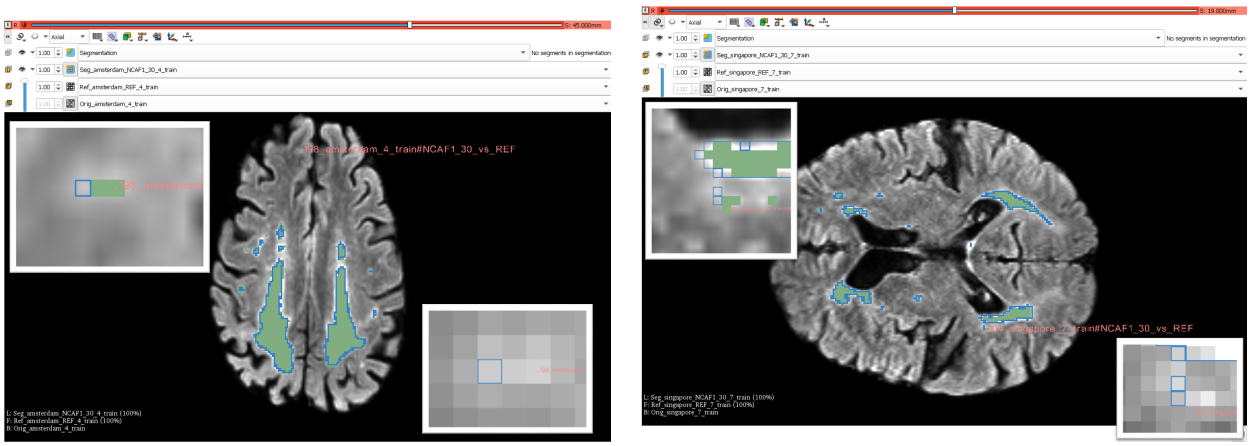


Figura C.4: Objetos con contacto y sin solape en su contexto

Para descubrir nuevas características de los errores, su localización y comportamiento se analizan las imágenes del conjunto de datos “Challenge” con los métodos propuestos para búsqueda de agrupaciones de error y errores aislados. De un total de 10639 objetos de agrupación pertenecientes a tres conjuntos de datos, se aplica como umbral de objeto de agrupación de éxito un valor de 0.75 y se obtienen 7204 objetos de éxito, 1347 objetos imperfectos, 1363 objetos miss y 725 objetos extra. Su distribución por cada localización se muestra en la tabla C.4.

Por otro lado, aplicando la comparación enlazada a dos valores de umbral, para binarizar la imagen de segmentación probabilística, junto a la segmentación de referencia,

Tipos de error de $O_{R.P}$

	Extra	Imperfect	Miss	Success	Total
Amsterdam	169	384	396	2203	3152
Singapore	330	406	393	2036	3165
Utrecht	226	557	574	2965	4322
	725	**1347**	**1363**	**7204**	**10639**

Tabla C.4: Distribución por tipo de error de los objetos de agrupación del conjunto de datos "Challenge"

se pueden estudiar qué diferencias existen, dónde se dan y cómo son. Algunas preguntas que surgen a la hora de seleccionar un umbral respecto a otro pueden ser respecto a los vóxeles de baja probabilidad, ¿cómo son? ¿están en la periferia de objetos segmentados o son objetos individuales? Para ejemplificar su uso, se analiza el conjunto de datos de "amsterdam" y se analiza mediante la metodología propuesta la segmentación probabilística con dos valor de umbral de 0.3 para O_{Pa} y de 0.7 para O_{Pb}. Se comparan entre ellas y respecto a la segmentación de referencia O_R. El resumen de comportamientos de este análisis se muestra en la tabla C.5 para las tres posibles reuniones definidas en la metodología AMOSE^2: $CE_{1ºPb,2ºPa->R}$,$CE_{1ºPb,2ºR->Pa}$,$CE_{1ºPa,2ºR->Pb}$.

P0.3vP0.7-REFvP0.7-REFvP0.3

ErrType_A	ErrType_B	ErrType_C	Frec
Miss	---	Extra	21
Miss	---	Overlap	12
Overlap	Contact	Contact	3
Overlap	Extra	Extra	148
Overlap	Extra	Overlap	2
Overlap	Overlap	Overlap	2798

REFvP0.7-P0.3vP0.7-REFvP0.3

ErrType_A	ErrType_B	ErrType_C	Frec
Contact	Overlap	Contact	3
Extra	Overlap	---	150
Miss	---	Miss	393
Miss	---	Overlap	7
Overlap	Overlap	Overlap	2885

REFvP0.3-P0.3vP0.7-REFvP0.7

ErrType_A	ErrType_B	ErrType_C	Frec
Contact	Overlap	Contact	3
Extra	Miss	---	21
Extra	Overlap	---	148
Miss	---	Miss	393
Overlap	Miss	Miss	7
Overlap	Miss	Overlap	7
Overlap	Overlap	Overlap	2898

Tabla C.5: Resumen de los comportamientos detectados en la comparación enlazada para el conjunto de datos "Amsterdam"

A continuación se analiza en detalle una de ellas, por ejemplo, la tabla de la izquierda que muestra la comparación enlazada $CE_{1ºPb,2ºPa->R}$. En primer lugar, se observa que hay 33 objetos de tipo Miss (21+12), que indica que son objetos que se detectan en $O_{P0.3}$ pero que no están en $O_{P0.7}$, es decir, son objetos de baja probabilidad. Consultado en la app web se comprueba que todos ellos son de tamaño muy pequeño entre 1 y 2 vóxeles. De ellos, hay 21 que no se detectan en la referencia (Miss-*-Extra) por lo que son objetos de error extra detectados y 12 objetos que si detectan en la referencia (Miss-*- Overlap). De este último caso, se muestra un objeto en su contexto en la figura C.5. En la imagen se muestran tres segmentaciones, con borde azul se muestra la segmentación P0.3 y con color salmón la segmentación P0.7 de la propuesta y con color verde la segmentación de referencia R. En una ventana se muestra ampliada la zona del objeto con este tipo de error

y unas flechas amarillas para indicar su ubicación exacta. Se observa que a la izquierda no hay solape, es un objeto Miss mientras a la derecha si lo hay, es un objeto Overlap. En este ejemplo el objeto de baja probabilidad se encuentra en la periferia del objeto de la referencia.

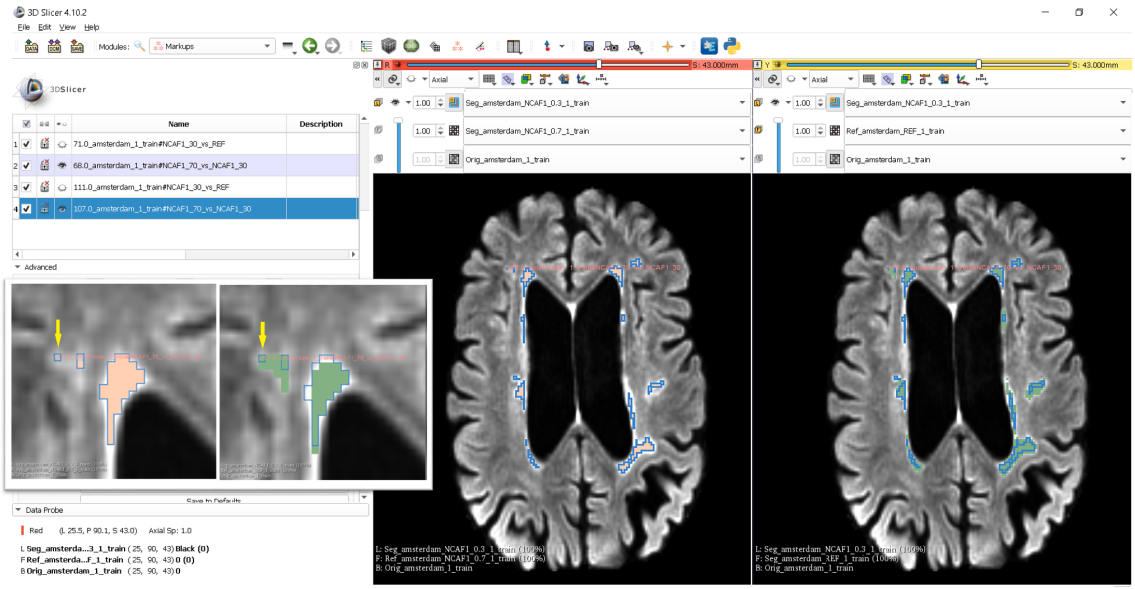


Figura C.5: Objeto Miss-*-Overlap en la comparación enlazada de $LOJ_{P0.3.P0.7}$

La elección de un valor de umbral más alto puede dar lugar a que aparezcan objetos de agrupación granulares, es decir, de tipo N-M con un número de objetos de la referencia y/o de la propuesta mayor que 1. Para comprobar este comportamiento, se estudia el tipo Overlap-Extra-Overlap de la comparación enlazada y se muestran dos ejemplos en la figura C.6. En ambos casos son objetos granulares de tipo 1-2 en el objeto de agrupación $O_{P0.3.P0.7}$ con un porcentaje de solape el muy alto a nivel vóxel, Respecto a la referencia existe error, que se da sobre todo en la periferia, a la izquierda de sobresegmentación y a la derecha de infrasegmentacion.

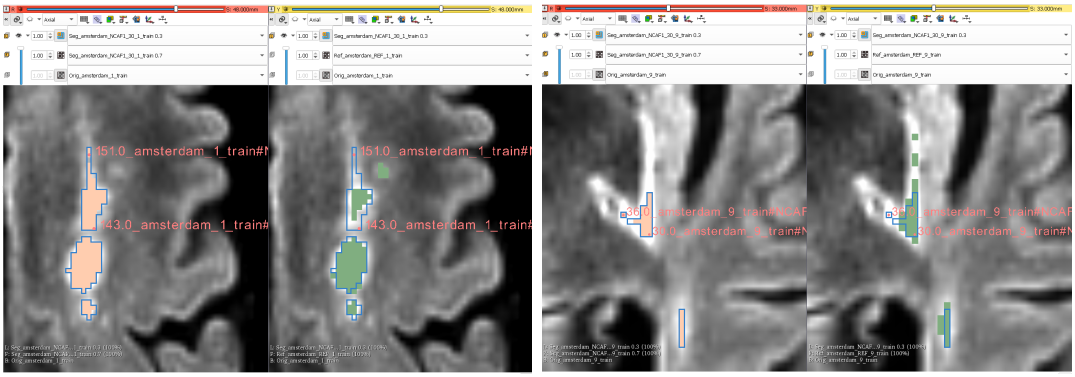


Figura C.6: Objetos Overlap-Extra-Overlap en la comparación enlazada de $LOJ_{P0.3.P0.7}$

Otro comportamiento de error relevante es el de los objetos de contacto sin solape. En la comparativa que se está analizando se observa que hay tres objetos cuyo tipo de error es Overlap-Contact-Contact, lo que significa que tanto la propuesta P0.3 como la propuesta P0.7 tienen solape, al menos en un vóxel, y respecto a la segmentación de referencia sólo tiene contacto. En la figura C.7 se muestran dos ejemplos en su contexto. Este hallazgo podría llevar a los expertos a modificar la segmentación de referencia, agrandando el objeto segmentado, ya que hay dos propuestas que indican que es una zona hiperintensa y además hay contacto con la referencia. Dicho refinamiento eliminaría este comportamiento de error.

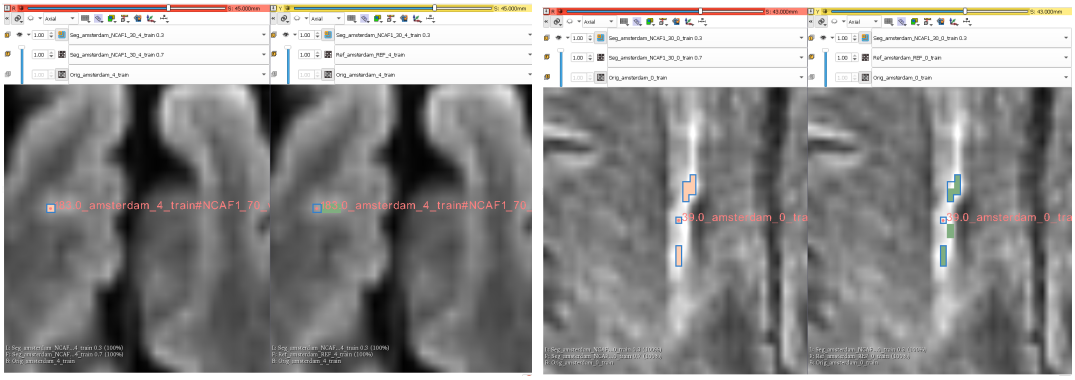


Figura C.7: Objetos Overlap-Contact-Contact en la comparación enlazada de $\text{LOJ}_{\text{P0.3.P0.7}}$

Referencias

Sistemas cognitivos: qué son, objetivos y aplicaciones. *UNIR Revista -INGENIERÍA Y TECNOLOGÍA-*, 2021. URL https://www.unir.net/ingenieria/revista/sistemas-cognitivos/.

Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall CRC, 1st edition, 2013. ISBN 1466558210, 9781466558212.

Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, June 1998. ISSN 0163-5808. doi: 10.1145/276305.276314. URL http://doi.acm.org/10.1145/276305.276314.

Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, May 2015. ISSN 1573-756X. doi: 10.1007/s10618-014-0365-y. URL https://doi.org/10.1007/s10618-014-0365-y.

Petronella Anbeek, Koen L. Vincken, Glenda S. van Bochove, Matthias J.P. van Osch, and Jeroen van der Grond. Probabilistic segmentation of brain tissue in mr imaging. *NeuroImage*, 27(4):795–804, 2005. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2005.05.046. URL https://www.sciencedirect.com/science/article/pii/S1053811905003228.

Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, and Peter M. Atkinson. Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1424, 2021. doi: https://doi.org/10.1002/widm.1424. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1424.

Xabier Artaechevarria, Arrate Muñoz-Barrutia, and Carlos Ortiz de Solorzano. Combination strategies in multi-atlas image segmentation: Application to brain mr data. *IEEE Trans. Med. Imaging*, 28(8):1266–1277, 2009.

Hans E. Atlason, Askell Love, Sigurdur Sigurdsson, Vilmundur Gudnason, and Lotta M. Ellingsen. Segae: Unsupervised white matter lesion segmentation from brain mris using a cnn autoencoder. *NeuroImage: Clinical*, 24:102085, 2019. ISSN 2213-1582. doi: https://doi.org/10.1016/j.nicl.2019.102085. URL https://www.sciencedirect.com/science/article/pii/S2213158219304322.

R. Balakrishnan, M.D.C. Valdés Hernández, and A.J. Farrall. Automatic segmentation of white matter hyperintensities from brain magnetic resonance images in the era of deep learning and big data - a systematic review. *Computerized Medical Imaging and Graphics*, 88, 2021. doi: 10.1016/j.compmedimag.2021.101867. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099710372&doi=10.1016%2fj.compmedimag.2021.101867&partnerID=40&md5=7ad3720afd696e7f562a9a784a8635f4. cited By 0.

María Verónica Barrera and Cayana Natalia Cabezas. *Prevalencia de la enfermedad de pequeño vaso y características clínicas que se asocian a mayor deterioro funcional, cognitivo y afectivo en adultos mayores con enfermedad cerebrovascular atendidos en el servicio de neurología del hospital Carlos Andrade Marín en el periodo 2020 - 2021*. PhD thesis, Pontificia universidad católica del ecuador. Facultad de medicina., 2021.

V. R. Basil and A. J. Turner. Iterative enhancement: A practical technique for software development. *IEEE Transactions on Software Engineering*, SE-1(4):390–396, Dec 1975. ISSN 0098-5589. doi: 10.1109/TSE.1975.6312870.

Feras A. Batarseh, Laura Freeman, and Chih-Hao Huang. A survey on artificial intelligence assurance. *Journal of Big Data*, 8(1):60, Apr 2021. ISSN 2196-1115. doi: 10.1186/s40537-021-00445-7. URL https://doi.org/10.1186/s40537-021-00445-7.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

P. Berkhin. A survey of clustering data mining techniques. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg, 2006. URL http://dx.doi.org/10.1007/3-540-28349-8_2.

Emmanuel Blanchard, Mounira Harzallah, and Pascale Kuntz. A generic framework for comparing semantic similarities on a subsumption hierarchy. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 20–24, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press. ISBN 978-1-58603-891-5. URL http://dl.acm.org/citation.cfm?id=1567281.1567291.

Barry W. Boehm. A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes*, 11(4):14–24, August 1986. ISSN 0163-5948. doi: 10.1145/12944.12948. URL http://doi.acm.org/10.1145/12944.12948.

Christian Bohm, Karin Kailing, Hans-Peter Kriegel, and Peer Kroger. Density connected clustering with local subspace preferences. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 27–34, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2142-8. URL http://dl.acm.org/citation.cfm?id=1032649.1033433.

Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors, 2020.

Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Universiteit Twente, Enschede, September 1997. URL http://doc.utwente.nl/17864/.

Pierre Bourque and Richard E. Fairley, editors. *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, CA, version 3.0 edition, 2014. ISBN 978-0-7695-5166-1. URL http://www.swebok.org/.

C. Brewster, O'Hara K., S. Fuller, Y. Wilks, E. Franconi, M. A. Musen, J. Ellman, and S. B. Shum. Knowledge representation with ontologies: The present and future. *IEEE Intelligent Systems*, 19(1):72–81., 2004.

Adam M. Brickman, Joel R. Sneed, Frank A. Provenzano, Ernst Garcon, Lauren Johnert, Jordan Muraskin, Lok-Kin Yeung, Molly E. Zimmerman, and Steven P. Roose. Quantitative approaches for assessment of white matter hyperintensities in elderly populations. *Psychiatry Research: Neuroimaging*, 193(2):101–106, 2011. ISSN 0925-4927. doi: https://doi.org/10.1016/j.pscychresns.2011.03.007. URL https://www.sciencedirect.com/science/article/pii/S0925492711001168.

Anna Brugulat-Serrat, Gemma Salvadó, Grégory Operto, Raffaele Cacciaglia, Carole H. Sudre, Oriol Grau-Rivera, Marc Suárez-Calvet, Carles Falcon, Gonzalo Sánchez-Benavides, Nina Gramunt, Carolina Minguillon, Karine Fauria, Frederik Barkhof, José L. Molinuevo, Juan D. Gispert, and ALFA Study. White matter hyperintensities mediate gray matter volume and processing speed relationship in cognitively unimpaired participants. *Human Brain Mapping*, 41(5):1309–1322, 2020. doi: https://doi.org/10.1002/hbm.24877. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.24877.

Juan Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle Karhohs, Matthieu Broisin, Csaba Molnar, Claire McQuin, Shantanu Singh, Fabian Theis, and Anne Carpenter. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95, 07 2019. doi: 10.1002/cyto.a.23863.

Maria Eugenia Caligiuri, Paolo Perrotta, Antonio Augimeri, Federico Rocca, Aldo Quattrone, and Andrea Cherubini. Automatic detection of white matter hyperintensities in healthy aging and pathology using magnetic resonance imaging: A review. *Neuroinformatics*, 13(3):261–276, July 2015. ISSN 1559-0089. doi: 10.1007/s12021-015-9260-y. URL https://doi.org/10.1007/s12021-015-9260-y.

Fabrice Camous. *Ontology-based Document Represent at ion for Biomedical Informat ion Retrieval*. PhD thesis, Dublin City University, 2007.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3), jun 2011. ISSN 0004-5411. doi: 10.1145/1970392.1970395. URL https://doi.org/10.1145/1970392.1970395.

Aaron Carass, Snehashis Roy, Amod Jog, Jennifer L. Cuzzocreo, Elizabeth Magrath, Adrian Gherman, Julia Button, James Nguyen, Ferran Prados, Carole H. Sudre, Manuel Jorge Cardoso, Niamh Cawley, Olga Ciccarelli, Claudia A.M. Wheeler-Kingshott, Sébastien Ourselin, Laurence Catanese, Hrishikesh Deshpande, Pierre Maurel, Olivier Commowick, Christian Barillot, Xavier Tomas-Fernandez, Simon K. Warfield, Suthirth Vaidya, Abhijith Chunduru, Ramanathan Muthuganapathy, Ganapathy Krishnamurthi, Andrew Jesson, Tal Arbel, Oskar Maier, Heinz Handels, Leonardo O. Iheme, Devrim Unay, Saurabh Jain, Diana M. Sima, Dirk Smeets, Mohsen Ghafoorian, Bram Platel, Ariel Birenbaum, Hayit Greenspan, Pierre-Louis Bazin, Peter A. Calabresi, Ciprian M. Crainiceanu, Lotta M. Ellingsen, Daniel S. Reich, Jerry L. Prince, and Dzung L. Pham. Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. *NeuroImage*, 148:77–102, 2017. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2016.12.064. URL https://www.sciencedirect.com/science/article/pii/S1053811916307819.

Aaron Carass, Snehashis Roy, Adrian Gherman, Jacob C. Reinhold, Andrew Jesson, Tal Arbel, Oskar Maier, Heinz Handels, Mohsen Ghafoorian, Bram Platel, Ariel Birenbaum, Hayit Greenspan, Dzung L. Pham, Ciprian M. Crainiceanu, Peter A. Calabresi, Jerry L. Prince, William R. Gray Roncal, Russell T. Shinohara, and Ipek Oguz. Evaluating white matter lesion segmentations with refined sørensen-dice analysis. *Scientific Reports*, 10(1):8242, May 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-64803-w. URL https://doi.org/10.1038/s41598-020-64803-w.

Balakrishnan Chandrasekaran. Design problem solving: A task analysis. *AI Magazine*, 11 (4):59, Dec. 1990. doi: 10.1609/aimag.v11i4.857. URL https://ojs.aaai.org/index.php/aimagazine/article/view/857.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation, 2021.

Regis Clouard, Arnaud Renouf, and Marinette Revenu. An ontology-based model for representing image processing application objectives. *International Journal of Pattern Recognition and Artificial Intelligence*, 24(8):1181–1208, 2010. doi: 10.1142/S0218001410008354.

Alistair Cockburn. *Agile Software Development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. ISBN 0-201-69969-9.

Olivier Commowick, Audrey Istace, Michaël Kain, Baptiste Laurent, Florent Leray, Mathieu Simon, Sorina Camarasu Pop, Pascal Girard, Roxana Améli, Jean-Christophe Ferré, Anne Kerbrat, Thomas Tourdias, Frédéric Cervenansky, Tristan Glatard, Jérémy Beaumont, Senan Doyle, Florence Forbes, Jesse Knight, April Khademi, Amirreza Mahbod, Chunliang Wang, Richard McKinley, Franca Wagner, John Muschelli, Elizabeth Sweeney, Eloy Roura, Xavier Lladó, Michel M. Santos, Wellington P. Santos, Abel G. Silva-Filho, Xavier Tomas-Fernandez, Hélène Urien, Isabelle Bloch, Sergi Valverde, Mariano Cabezas, Francisco Javier Vera-Olmos, Norberto Malpica, Charles Guttmann, Sandra Vukusic, Gilles Edan, Michel Dojat, Martin Styner, Simon K. Warfield, François Cotton, and Christian Barillot. Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure. *Scientific Reports*, 8(1):13650, Sep 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-31911-7. URL https://doi.org/10.1038/s41598-018-31911-7.

P. L. Correia and F. Pereira. Objective evaluation of video segmentation quality. *IEEE Trans Image Process*, 12(2):186–200, 2003.

W.R. Crum, O. Camara, and D.L.G. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *Medical Imaging, IEEE Transactions on*, 25 (11):1451–1461, Nov 2006. ISSN 0278-0062. doi: 10.1109/TMI.2006.880587.

S. Damangir, A. Manzouri, K. Oppedal, S. Carlsson, M.J. Firbank, H. Sonnesyn, O. Tysnes, O'Brien J.T., M.K. Beyer, E. Westman, D. Aarsland, L. Wahlund, and G. Spulber. Multispectral mri segmentation of age related white matter changes using a cascade of support vector machines. *Journal of the Neurological Sciences*, 2012.

Renske de Boer, Henri A. Vrooman, Fedde van der Lijn, Meike W. Vernooij, M. Arfan Ikram, Aad van der Lugt, Monique M.B. Breteler, and Wiro J. Niessen. White matter lesion extension to automatic brain tissue segmentation on mri. *NeuroImage*, 45(4):1151–1161, 2009. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2009.01.011. URL https://www.sciencedirect.com/science/article/pii/S1053811909000561.

Thamiris de Souza Alves, Caterine Silva de Oliveira, Cesar Sanin, and Edward Szczerbicki. From knowledge based vision systems to cognitive vision systems: A review. *Procedia Computer Science*, 126:1855–1864, 2018. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.08.077. URL https://www.sciencedirect.com/science/article/pii/S1877050918313553. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

Stéphanie Debette and H S Markus. The clinical importance of white matter hyperintensities on brain magnetic resonance imaging: systematic review and meta-analysis. *BMJ*, 341, 2010. ISSN 0959-8138. doi: 10.1136/bmj.c3666. URL https://www.bmj.com/content/341/bmj.c3666.

Charles DeCarli, Joseph Massaro, Danielle Harvey, John Hald, Mats Tullberg, Rhoda Au, Alexa Beiser, Ralph D'Agostino, and Philip A. Wolf. Measures of brain morphology and infarction in the framingham heart study: establishing what is normal. *Neurobiology of Aging*, 26(4):491–510, 2005. ISSN 0197-4580. doi: https://doi.org/10.1016/j.neurobiolaging.2004.05.004. URL https://www.sciencedirect.com/science/article/pii/S0197458004001988.

L.R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26: 297–302, 1945.

F. K. Došilović, M. Brčić, and N. Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215, May 2018. doi: 10.23919/MIPRO.2018.8400040.

P. Duque, J. M. Cuadra, E. Jiménez, and Mariano Rincón-Zamorano. Data preprocessing for automatic wmh segmentation with fcnns. In José Manuel Ferrández Vicente, José Ramón Álvarez-Sánchez, Félix de la Paz López, Javier Toledo Moreo, and Hojjat Adeli, editors, *From Bioinspired Systems and Biomedical Applications to Machine Learning*, pages 452–460, Cham, 2019. Springer International Publishing. ISBN 978-3-030-19651-6.

Pablo Duque, Mariano Rincón, and Jose Manuel Cuadra. Modified u-net for wmh segmentation. WMH Segmentation Challenge, 2020. URL https://wmh.isi.uu.nl/results/uned/.

Tom Eelbode, Jeroen Bertels, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B. Blaschko. Optimization for medical image segmentation: Theory and practice when evaluating with dice score or jaccard index. *IEEE Transactions on Medical Imaging*, 39(11):3679–3690, 2020. doi: 10.1109/TMI.2020.3002417.

Aitor Elorriaga. Sistemas cognitivos en el aseguramiento de la calidad de software. 2018. URL https://www.mtp.es/blog/testing-software/sistemas-cognitivos-calidad-software/.

Mark Everingham, Mark Everingham, Henk Muller, Henk Muller, Barry Thomas, and Barry Thomas. Evaluating image segmentation algorithms using the pareto front. In *Proc. Seventh European Conf. Computer Vision*, volume 4, pages 34–48, 2002.

M. Fatimaezzahra, E. Abdelaziz, S. Mohamed, and B. Loubna. Towards domain ontology creation based on a taxonomy structure in computer vision. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 7(2):269–278, 2016. ISSN 2158-107X.

Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V. Miller, Steve Pieper, and Ron Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic resonance imaging*, 30(9):1323–1341, Nov 2012. ISSN 1873-5894. doi: 10.1016/j.mri.2012.05.001. URL https://pubmed.ncbi.nlm.nih.gov/22770690.22770690.

Enzo Ferrante. Inteligencia artificial para el análisisi de imágenes médicas. In *Seminarios de informática en salud*. Departamento de Informática en Salud Hospital Italiano - Instituto Universitario del Hospital Italiano, 2021. URL https://www.youtube.com/watch?v=ulX6fZzILdM&t=2131s.

Benedikt M. Frey, Marvin Petersen, Carola Mayer, Maximilian Schulz, Bastian Cheng, and Götz Thomalla. Characterization of white matter hyperintensities in large-scale mri-studies. *Frontiers in Neurology*, 10, 2019. ISSN 1664-2295. doi: 10.3389/fneur.2019.00238. URL https://www.frontiersin.org/article/10.3389/fneur.2019.00238.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 166–181, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44268-5. URL http://dl.acm.org/citation.cfm?id=645362.650863.

Aimee Cecilia Hernández García, Mireya Tovar, and José de Jesús Lavalle-Martínez. Medidas de similitud semántica aplicadas a una ontología de dominio. *Res. Comput. Sci.*, pages 119–131, 2018.

V. Garcia, H. De Jesus Ochoa Dominguez, and B. Mederos. Analysis of discrepancy metrics used in medical image segmentation. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 13(1):235–240, Jan 2015. ISSN 1548-0992. doi: 10.1109/TLA.2015.7040653.

Ruth Geraldes, Maciej Juryńczyk, Giordani Rodrigues dos Passos, Alexander Pichler, Karen Chung, Marloes Hagens, Serena Ruggieri, Cristina Auger, Jaume Sastre-Garriga, Christian Enzinger, Declan Chard, Frederik Barkhof, Claudio Gasperini, Alex Rovira, Gabriele DeLuca, Jacqueline Palace, and on behalf of the MAGNIMS study group. The role of pontine lesion location in differentiating multiple sclerosis from vascular risk factor-related small vessel disease. *Multiple Sclerosis Journal*, 27(6): 968–972, 2021. doi: 10.1177/1352458520943777. URL https://doi.org/10.1177/1352458520943777. PMID: 32757905.

Julie Gerlings, Arisa Shollo, and Ioanna Constantiou. Reviewing the need for explainable artificial intelligence (xai), 2021.

Eduardo Gil. La matriz de confusión: ¡claramente explicada! 2022. URL https://profesordata.com/.

Ashok K. Goel, Swaroop Vattam, Bryan Wiltgen, and Michael Helms. Cognitive, collaborative, conceptual and creative - four characteristics of the next generation of knowledge-based cad systems: A study in biologically inspired design. *Comput. Aided Des.*, 44(10):879–900, oct 2012. ISSN 0010-4485. doi: 10.1016/j.cad.2011.03.010. URL https://doi.org/10.1016/j.cad.2011.03.010.

Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 1846283965.

Jorge Fernández González. *Introducción alas metodologías ágiles. Otras formas de analizar y desarrollar*. Fundación para la Universitat Oberta de Catalunya, 2013.

A. B. Goumeidane, M. Khamadja, B. Belaroussi, H. Benoit-Cattin, and C. Odet. New discrepancy measures for segmentation evaluation. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 2, pages II–411–14 vol.3, Sept 2003. doi: 10.1109/ICIP.2003.1246704.

Ludovica Griffanti, Giovanna Zamboni, Aamira Khan, Linxin Li, Guendalina Bonifacio, Vaanathi Sundaresan, Ursula G. Schulz, Wilhelm Kuker, Marco Battaglini, Peter M. Rothwell, and Mark Jenkinson. Bianca (brain intensity abnormality classification algorithm): A new tool for automated segmentation of white matter hyperintensities. *NeuroImage*, 141:191–205, 2016. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2016.07.018. URL https://www.sciencedirect.com/science/article/pii/S1053811916303251.

Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Adquisition*, 5:199–220, 1993.

Bert Guindon and Ying Zhang. Application of the dice coefficient to accuracy assessment of object-based image classification. *Canadian Journal of Remote Sensing*, 43(1): 48–61, 2017. doi: 10.1080/07038992.2017.1259557. URL https://doi.org/10.1080/07038992.2017.1259557.

Gan Guojun, Chaoqun Ma, and Jianhong Wu. *Data Clustering. Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.

Chih-Ying Gwo, David C. Zhu, and Rong Zhang. Brain white matter hyperintensity lesion characterization in t(2) fluid-attenuated inversion recovery magnetic resonance images: Shape, texture, and potential growth. *Frontiers in neuroscience*, 13:353–353, Apr 2019. ISSN 1662-4548. doi: 10.3389/fnins.2019.00353. URL https://pubmed.ncbi.nlm.nih.gov/31057353. 31057353[pmid].

V.C. Hachinski, P. Potter, and H. Merskey. Leuko-araiosis. *Arch. Neurol. 44*, 1:21–23, 1987. doi: https://doi.org/10.1001/archneur.1987.00520130013009.

J. Hao, Y. Shen, H. Xu, and J. Zou. A region entropy based objective evaluation method for image segmentation. In *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, pages 373–377, May 2009. doi: 10.1109/IMTC.2009.5168478.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation, 2014. URL https://arxiv.org/abs/1407.1808.

D. M Hawkins. *Identification of Outliers*. Chapman and Hall, London - New York, 1980.

Rutger Heinen, M.D. Steenwijk, Frederik Barkhof, and J. Matthijs Biesbroek. Performance of five automated white matter hyperintensity segmentation methods in a multicenter dataset. *Scientific Reports*, 9(1):16742, Nov 2019. ISSN 2045-2322. doi: https://doi.org/10.1038/s41598-019-52966-0. URL https://doi.org/10.1038/s41598-019-52966-0.

Robert Hoehndorf, Michel Dumontier, and Georgios V. Gkoutos. Evaluation of research in biomedical ontologies. *Brief Bioinform*, 14(6):696–712, January 2013. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbs053. URL http://bib.oxfordjournals.org/content/14/6/696. PMID: 22962340.

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 340–353, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33712-3.

Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1312, 2019. doi: 10.1002/widm.1312. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1312.

Isabel Hotz, Pascal Frédéric Deschwanden, Franziskus Liem, Susan Mérillat, Brigitta Malagurski, Spyros Kollias, and Lutz Jäancke. Performance of three freely available methods for extracting white matter hyperintensities: Freesurfer, ubo detector, and bianca. *Human Brain Mapping*, 43(5):1481–1500, 2022. doi: https://doi.org/10.1002/hbm.25739. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.25739.

Mia Hubert, Michiel Debruyne, and Peter J. Rousseeuw. Minimum covariance determinant and extensions. *WIREs Computational Statistics*, 10(3):e1421, 2018. doi: https://doi.org/10.1002/wics.1421. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1421.

Jens Hühn and Eyke Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, Dec 2009. ISSN 1573-756X. doi: 10.1007/s10618-009-0131-8. URL https://doi.org/10.1007/s10618-009-0131-8.

Ivana Isgum, Marius Staring, Annemarieke Rutten, Mathias Prokop, Max A. Viergever, and Bram van Ginneken. Multi-atlas-based segmentation with local decision fusion - application to cardiac and aortic segmentation in ct scans. *IEEE Trans. Med. Imaging*, 28(7):1000–1010, 2009. URL http://dblp.uni-trier.de/db/journals/tmi/tmi28.html#IsgumSRPVG09.

V. Ithapu, V. Singh, C. Lindner, B. P. Austin, C. Hinrichs, C. M. Carlsson, and S. C. Johnson. Extracting and summarizing white matter hyperintensities using supervised segmentation methods in alzheimer's disease risk and aging studies. *Human Brain Mapping*, 35(8):4219–4235, 2014. doi: http://doi.org/10.1002/hbm.22472.

Qiang Ji. Combining knowledge with data for efficient and generalizable visual learning. *Pattern Recognition Letters*, 124:31–38, 2019. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2017.11.013. URL https://www.sciencedirect.com/science/article/pii/S0167865517304270. Award Winning Papers from the 23rd International Conference on Pattern Recognition (ICPR).

Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *4th SIAM International Conference on Data Mining (SDM)*, pages 246–257, 2004.

H. J. Kuijf, J. M. Biesbroek, J. De Bresser, R. Heinen, S. Andermatt, M. Bento, M. Berseth, M. Belyaev, M. J. Cardoso, A. Casamitjana, D. L. Collins, M. Dadar, A. Georgiou, M. Ghafoorian, D. Jin, A. Khademi, J. Knight, H. Li, X. Lladó, M. Luna, Q. Mahmood, R. McKinley, A. Mehrtash, S. Ourselin, B. Park, H. Park, S. H. Park, S. Pezold, E. Puybareau, L. Rittner, C. H. Sudre, S. Valverde, V. Vilaplana, R. Wiest, Y. Xu, Z. Xu, G. Zeng, J. Zhang, G. Zheng, C. Chen, W. van der Flier, F. Barkhof, M. A. Viergever, and G. J. Biessels. Standardized assessment of automatic segmentation of white matter hyperintensities and results of the wmh segmentation challenge. *IEEE Transactions on Medical Imaging*, 38(11):2556–2568, Nov 2019. ISSN 1558-254X. doi: 10.1109/TMI.2019.2905770.

Craig Larman and Victor R. Basili. Iterative and incremental development: A brief history. *Computer*, 36(6):47–56, June 2003. ISSN 0018-9162. doi: 10.1109/MC.2003.1204375. URL http://dx.doi.org/10.1109/MC.2003.1204375.

Juan J. Lastra-Díaz and Ana García-Serrano. A novel family of ic-based similarity measures with a detailed experimental survey on wordnet. *Eng. Appl. Artif. Intell.*, 46 (PA):140–153, November 2015. ISSN 0952-1976. doi: 10.1016/j.engappai.2015.09.006. URL http://dx.doi.org/10.1016/j.engappai.2015.09.006.

Xinxin Li, Yu Zhao, Jiyang Jiang, Jian Cheng, Wanlin Zhu, Zhenzhou Wu, Jing Jing, Zhe Zhang, Wei Wen, Perminder S. Sachdev, Yongjun Wang, Tao Liu, and Zixiao Li. White matter hyperintensities segmentation using an ensemble of neural networks. *Human Brain Mapping*, 43(3):929–939, 2022. doi: https://doi.org/10.1002/hbm.25695. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.25695.

Acervo Lima. TÉcnicas de selecciÓn de funciones en el aprendizaje automÁtico. 2019. URL https://es.acervolima.com/tecnicas-de-seleccion-de-funciones-en-el-aprendizaje-automatico/.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL http://dl.acm.org/citation.cfm?id=645527.657297.

Bing Liu, Yiyuan Xia, and Philip S. Yu. Clustering through decision tree construction. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, pages 20–29, New York, NY, USA, 2000. ACM. ISBN 1-58113-320-0. doi: 10.1145/354756.354775. URL http://doi.acm.org/10.1145/354756.354775.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

Pauline Maillard, Nicolas Delcroix, Fabrice Crivello, Carole Dufouil, Sebastien Gicquel, Marc Joliot, Nathalie Tzourio-Mazoyer, Annick Alpérovitch, Christophe Tzourio, and Bernard Mazoyer. An automated procedure for the assessment of white matter hyperintensities by multispectral (t1, t2, pd) mri and an evaluation of its between-centre reproducibility based on two large community databases. *Neuroradiology*, 50 (1):31–42, Jan 2008. ISSN 1432-1920. doi: 10.1007/s00234-007-0312-3. URL https://doi.org/10.1007/s00234-007-0312-3.

Nicolas Maillot and Monique Thonnat. Ontology based complex object recognition. *Image and Vision Computing, Elsevier*, 26(1):pp 102–113, January 2008. URL https://hal.inria.fr/inria-00502361.

Jose V. Manjón, Pierrick Coupé, Parnesh Raniga, Ying Xia, Patricia Desmond, Jurgen Fripp, and Olivier Salvado. Mri white matter lesion segmentation using an ensemble of neural networks and overcomplete patch-based voting. *Computerized Medical Imaging and Graphics*, 69:43–51, 2018. ISSN 0895-6111. doi: https://doi.org/10.

1016/j.compmedimag.2018.05.001. URL https://www.sciencedirect.com/science/article/pii/S0895611118302866.

Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2020.103655. URL https://www.sciencedirect.com/science/article/pii/S1532046420302835.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423 vol.2, 2001. doi: 10.1109/ICCV.2001.937655.

J. Mauriño Donato and J. Álvarez-Sabin. Lesiones de la sustancia blanca cerebral: significado clínico y mecanismos fisiopatológicos. *Hipertensión y Riesgo Vascular*, 21(1):38–42, 2004. ISSN 1889-1837. doi: https://doi.org/10.1016/S1889-1837(04)71447-6. URL https://www.sciencedirect.com/science/article/pii/S1889183704714476.

Riccardo Mazza. *Introduction to Information Visualization*. Springer Publishing Company, Incorporated, 1 edition, 2009. ISBN 1848002181.

Boris Mirkin and Eugene V. Koonin. A top-down method for building genome classification trees with linear binary hierarchies. In *Bioconsensus, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, 2001*, pages 97–112, 2001.

Apoorva Mishra and Deepty Dubey. A comparative study of different software development life cycle models in different scenarios. *International Journal of Advance Research inComputer Science and Management Studies*, 2013.

Himanshu Mittal, Avinash Chandra Pandey, Mukesh Saraswat, Sumit Kumar, Raju Pal, and Garv Modwel. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools and Applications*, Feb 2021. ISSN 1573-7721. doi: 10.1007/s11042-021-10594-9. URL https://doi.org/10.1007/s11042-021-10594-9.

Farmeena Khan Mohd Ehmer Khan. A comparative study of white box, black box and grey box testing techniques. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 3(6), 2012. URL http://ijacsa.thesai.org/.

G. Moise, J. Sander, and M. Ester. P3c: A robust projected clustering algorithm. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 414–425, Dec 2006. doi: 10.1109/ICDM.2006.123.

Mariana Morales and Ever Augusto Torres. Resonancia mágnetica nuclear, 2008. URL https://es.slideshare.net/cheo.torres/resonancia-magnetica-nuclear-rmn?from_action=save.

Mark A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1 (4):4–12, jun 2015. doi: 10.1145/2757001.2757003. URL https://doi.org/10.1145/2757001.2757003.

Harsha S. Nagesh, Sanjay Goil, and Alok N. Choudhary. Adaptive grids for clustering massive data sets. In *SDM*, 2001.

Ying-Hwey Nai, Bernice W. Teo, Nadya L. Tan, Sophie O'Doherty, Mary C. Stephenson, Yee Liang Thian, Edmund Chiong, and Anthonin Reilhac. Comparison of metrics for the evaluation of medical segmentations using prostate mri dataset. *Computers in Biology and Medicine*, 134:104497, 2021. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2021.104497. URL https://www.sciencedirect.com/science/article/pii/S0010482521002912.

J.C. Nascimento and J.S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, 2006. doi: 10.1109/TMM.2006.876287.

Sira Vegas Natalia Juristo, Ana M. Moreno. Técnicas de evaluación de software. Technical report, Universidad Politécnica de Madrid, 2006.

Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9, New York, NY, USA, 2001. ACM. ISBN 1-58113-377-4. doi: 10.1145/505168.505170. URL http://doi.acm.org/10.1145/505168.505170.

Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In Kohei Arai and Supriya Kapoor, editors, *Advances in Computer Vision*, pages 128–144, Cham, 2020. Springer International Publishing. ISBN 978-3-030-17795-9.

Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979. ISSN 0018-9472. doi: 10.1109/tsmc.1979.4310076. URL http://dx.doi.org/10.1109/tsmc.1979.4310076.

Gilsoon Park, Jinwoo Hong, Ben A. Duffy, Jong-Min Lee, and Hosung Kim. White matter hyperintensities segmentation using the ensemble u-net with multi-scale highlighting foregrounds. *NeuroImage*, 237:118140, 2021. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2021.118140. URL https://www.sciencedirect.com/science/article/pii/S1053811921004171.

Viktor Pekar and Steffen Staab. Taxonomy learning: Factoring the structure of a taxonomy into a semantic classification decision. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1072228.1072318. URL http://dx.doi.org/10.3115/1072228.1072318.

P. Phillips, Carina Hahn, Peter Fontana, Amy Yates, Kristen Greene, David Broniatowski, and Mark Przybocki. Four principles of explainable artificial intelligence, September 2021. URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=933399.

Mark Polak, Hong Zhang, and Minghong Pi. An evaluation metric for image segmentation of multiple objects. *Image Vision Comput.*, 27(8):1223–1227, July 2009. ISSN 0262-8856. doi: 10.1016/j.imavis.2008.09.008. URL http://dx.doi.org/10.1016/j.imavis.2008.09.008.

Jordi Pont-Tuset and Ferran Marques. Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1465–1478, 2016.

Keith Price. Annotated computer vision bibliography, 2022. URL http://www.visionbib.com/bibliography/contents.html.

J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0. URL http://portal.acm.org/citation.cfm?id=152181.

R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 17–30, 1989.

David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 165–176, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 1558602623.

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL http://dl.acm.org/citation.cfm?id=1625855.1625914.

K. Riku, F. Ryo, O. Tomonobu, and O. Takenao. A method for supporting retrieval of articles on protein structure analysis considering users ' intention. *BMC Bioinformatics*, 12(Suppl 1):S42, 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-S1-S42. URL http://www.biomedcentral.com/1471-2105/12/S1/S42http://www.doaj.org/doaj?func=abstract&id=702860.

M. Rincón, E. Díaz-López, P. Selnes, K. Vegge, M. Altmann, T. Fladby, and A. Bjornerud. Improved automatic segmentation of white matter hyperintensities in mri based on multilevel lesion features. *Neuroinformatics*, 2017.

D M Rivera, S Puentes, and L Caballero. Resonancia magnética cerebral: secuencias básicas e interpretación. *Universitas Medica*, 2011. ISSN 0041-9095. URL https://www.redalyc.org/articulo.oa?id=231022506005.

D. Rivière, D. Geffroy, I. Denghien, N. Souedet, and Y. Cointepas. Brainvisa: an extensible software environment for sharing multimodal neuroimaging data and processing tools. *NeuroImage*, 47, 2009.

Torsten Rohlfing and Calvin R. Maurer, Jr. Shape-based averaging. *IEEE Transactions on Image Processing*, 16(1):153–161, January 2007. doi: 10.1109/TIP.2006.884936. URL http://ieeexplore.ieee.org.laneproxy.stanford.edu/iel5/83/4032799/04032827.pdf?isnumber=4032799&prod=JNL&arnumber=4032827&arnumber=4032827&arSt=153&ared=161&arAuthor=Rohlfing%2C+T.%3B+Maurer%2C+Jr.%2C+C.+R.

M. T. Romá Ferri. *OntoFIS: tecnología ontológica en el dominio farmacoterapéutico*. PhD thesis, Universidad de Alicante, 2009.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Mert R. Sabuncu, B. T. Thomas Yeo, Koenraad Van Leemput, Bruce Fischl, and Polina Golland. A generative model for image segmentation based on label fusion. *IEEE Trans. Med. Imaging*, 29(10):1714–1729, 2010. URL http://dblp.uni-trier.de/db/journals/tmi/tmi29.html#SabuncuYLFG10.

Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 08 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm344. URL https://doi.org/10.1093/bioinformatics/btm344.

Gemma Salvadó, Anna Brugulat-Serrat, Carole H. Sudre, Oriol Grau-Rivera, Marc Suárez-Calvet, Carles Falcon, Karine Fauria, M. Jorge Cardoso, Frederik Barkhof, José Luis Molinuevo, Juan Domingo Gispert, and A. L. F. A. Study. Spatial patterns of white matter hyperintensities associated with alzheimer's disease risk factors in a cognitively healthy middle-aged cohort. *Alzheimer's research & therapy*, 11(1): 12–12, Jan 2019. ISSN 1758-9193. doi: 10.1186/s13195-018-0460-1. URL https://pubmed.ncbi.nlm.nih.gov/30678723. 30678723[pmid].

David Sánchez, Montserrat Batet, David Isern, and Aida Valls. Ontology-based semantic similarity: A new feature-based approach. *Expert Syst. Appl.*, 39(9):7718–7728, July 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.01.082. URL http://dx.doi.org/10.1016/j.eswa.2012.01.082.

Paul Schmidt. *Bayesian inference for structured additive regression models for large-scale problems with applications to medical imaging*. PhD thesis, LMU Munich, January 2017. URL http://nbn-resolving.de/urn:nbn:de:bvb:19-203731.

Paul Schmidt, Christian Gaser, Milan Arsic, Dorothea Buck, Annette Förschler, Achim Berthele, Muna Hoshi, Rüdiger Ilg, Volker J. Schmid, Claus Zimmer, Bernhard Hemmer, and Mark Mühlau. An automated tool for detection of flair-hyperintense white-matter lesions in multiple sclerosis. *NeuroImage*, 59(4):3774–3783, 2012. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2011.11.032. URL https://www.sciencedirect.com/science/article/pii/S1053811911013139.

Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237, Jan 2014. ISSN 1573-756X. doi: 10.1007/s10618-012-0300-z. URL https://doi.org/10.1007/s10618-012-0300-z.

Streamlit Documentation. Release 0.82.0. Streamlit Inc., https://docs.streamlit.io/_/downloads/en/0.82.0/pdf/, May 2021.

A. A. Taha, A. Hanbury, and O. A. J. del Toro. A formal method for selecting evaluation metrics for image segmentation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 932–936, Oct 2014. doi: 10.1109/ICIP.2014.7025187.

Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, Aug 2015. ISSN 1471-2342. doi: 10.1186/s12880-015-0068-x. URL https://doi.org/10.1186/s12880-015-0068-x.

Philippe Tran, Urielle Thoprakarn, Emmanuelle Gourieux, Clarisse Longo dos Santos, Enrica Cavedo, Nicolas Guizard, François Cotton, Pierre Krolak-Salmon, Christine Delmaire, Damien Heidelberg, Nadya Pyatigorskaya, Sébastian Ströer, Didier Dormont, Jean-Baptiste Martini, and Marie Chupin. Automatic segmentation of white matter hyperintensities: validation and comparison with state-of-the-art methods on both multiple sclerosis and elderly subjects. *NeuroImage: Clinical*, 33:102940, 2022. ISSN 2213-1582. doi: https://doi.org/10.1016/j.nicl.2022.102940. URL https://www.sciencedirect.com/science/article/pii/S2213158222000055.

J. W. Tukey. Exploratory data analysis. *Addison-Wesley*, 1977.

R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1046.

Antony Unwin. Multivariate outliers and the o3 plot. *Journal of Computational and Graphical Statistics*, 28(3):635–643, 2019. doi: 10.1080/10618600.2019.1575226. URL https://doi.org/10.1080/10618600.2019.1575226.

M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, June 1996.

Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:7068349, Feb 2018. ISSN 1687-5265. doi: 10.1155/2018/7068349. URL https://doi.org/10.1155/2018/7068349.

Jiguang Wang, Qiang Huang, Zhi-Ping Liu, Y. Wang, Ling-Yun Wu, Luonan Chen, and X. Zhang. Noa: a novel network ontology analysis method. *Nucleic Acids Research*, 39: e87 – e87, 2011.

Lichao Wang, K. Lekadir, S. Lee, R. Merrifield, and Guang-Zhong Yang. A general framework for context-specific image segmentation using reinforcement learning. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 32(5):943–956, 2013. ISSN 0278-0062. doi: 10.1109/TMI.2013.2252431.

Yanbo Wang, Joseree Ann Catindig, Saima Hilal, Hock Wei Soon, Eric Ting, Tien Yin Wong, Narayanaswamy Venketasubramanian, Christopher Chen, and Anqi Qiu. Multi-stage segmentation of white matter hyperintensity, cortical and lacunar infarcts. *NeuroImage*, 60(4):2379–2388, 2012. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2012.02.034. URL https://www.sciencedirect.com/science/article/pii/S105381191200211X.

Zhaobin Wang, E. Wang, and Ying Zhu. Image segmentation evaluation: a survey of methods. *Artificial Intelligence Review*, 53(8):5637–5674, Dec 2020. ISSN 1573-7462. doi: 10.1007/s10462-020-09830-9. URL https://doi.org/10.1007/s10462-020-09830-9.

Joanna M. Wardlaw, Maria C. Valdés Hernández, and Susana Muñoz-Maniega. What are white matter hyperintensities made of? relevance to vascular cognitive impairment. *Journal of the American Heart Association*, 4(6):001140–001140, Jun 2015. ISSN 2047-9980. doi: 10.1161/JAHA.114.001140. URL https://pubmed.ncbi.nlm.nih.gov/26104658. 26104658[pmid].

S. K. Warfield, K. H. Zou, and W. M. Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging*, 23(7):903–921, 2004. doi: 10.1109/TMI.2004.828354.

Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL http://dx.doi.org/10.3115/981732.981751.

Luren Yang, Fritz Albregtsen, Tor Lønnestad, and Per Grøttum. A supervised approach to the evaluation of image segmentation methods. In *BP 101 - 54602 Villers-ls-Nancy Cedex (France) Unit de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France) Unit de recherche INRIA Rhne-Alpes : 655,*

avenue de l'Europe - 38330 Montbonnot-St-Martin (France) Unit d, pages 759–765. Springer, 1995.

M. Yoshita, E. Fletcher, D. Harvey, M. Ortega, O. Martinez, D.M. Mungas, B.R. Reed, and C.S. DeCarli. Extent and distribution of white matter hyperintensities in normal aging, mci, and ad. *Neurology*, 67(12):2192–2198, 2006.

Alan Yuille and Aude Oliva. Frontiers in computer vision: Nsf white paper. *Frontiers in Computer Vision*, 2010.

Hui Zhang, Jason E. Fritts, and Sally A. Goldman. An entropy-based objective evaluation method for image segmentation. *Proc. SPIE*, 5307:38–49, 2003. doi: 10.1117/12.527167. URL http://dx.doi.org/10.1117/12.527167.

Hui Zhang, Jason E. Fritts, and Sally A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, May 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.08.003. URL http://www.sciencedirect.com/science/article/pii/S1077314207001294.

Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1995.

A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer. Morphometric analysis of white matter lesions in mr images: method and validation. *IEEE Transactions on Medical Imaging*, 13(4):716–724, Dec 1994. ISSN 0278-0062. doi: 10.1109/42.363096.

A.P. Zijdenbos, R. Forghani, and A.C. Evans. Automatic "pipeline.ªnalysis of 3-d mri data for clinical trials: application to multiple sclerosis. *IEEE Transactions on Medical Imaging*, 21(10):1280–1291, 2002. doi: 10.1109/TMI.2002.806283.

Kelly H. Zou, Simon K. Warfield, Aditya Bharatha, Clare M. C. Tempany, Michael R. Kaus, Steven J. Haker, 3rd Wells, William M, Ferenc A. Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic radiology*, 11(2):178–189, Feb 2004. ISSN 1076-6332. doi: 10.1016/s1076-6332(03)00671-8. URL https://pubmed.ncbi.nlm.nih.gov/14974593. 14974593[pmid].