TESIS DOCTORAL

PLATAFORMA DE ANÁLISIS DE IMÁGENES SATELITALES PARA EL DESCUBRIMIENTO DE RECURSOS HÍDRICOS MEDIANTE LA APLICACIÓN DE TÉCNICAS BASADAS EN INTELIGENCIA ARTIFICIAL.

VNiVERSiDAD D SALAMANCA

DEPARTAMENTO DE INFORMÁTICA Y AUTOMÁTICA

FACULTAD DE CIENCIAS

UNIVERSIDAD DE SALAMANCA

AUTOR:

ANTÍA CARMONA BALEA

La memoria titulada "PLATAFORMA DE ANÁLISIS DE IMÁGENES SATELITALES PARA EL DESCUBRIMIENTO DE RECURSOS HÍDRICOS MEDIANTE LA APLICACIÓN DE TÉCNICAS BASADAS EN INTELIGENCIA ARTIFICIAL" que presenta D. Antía Carmona Balea para optar al Grado de Doctor por la Universidad de Salamanca ha sido realizada bajo la dirección del profesor Dr. Juan Francisco de Paz Santana, Catedrático de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca, el profesor Dr. Gabriel Villarrubia González, Profesor Titular de Departamento de Informática y Automática de la universidad de Salamanca.

Salamanca, octubre 2023

El Doctorando/a

Fdo: Antía Carmona Balea

Los directores

Fdo.: Dr. Juan Francisco de Paz Santana

Catedrático de Universidad

Departamento de Informática y Automática, Universidad de Salamanca

DE PAZ SANTANA JUAN FRANCISCO - 76125754D

Firmado digitalmente por DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-76125754D, givenName=JUAN FRANCISCO, sn=DE PAZ SANTANA, cn=DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Fecha: 2023.12.11 11:18:47 +01'00'

Fdo.: Dr. Gabriel Villarrubia González

Profesor Titular de Universidad

Departamento de Informática y Automática, Universidad de Salamanca

VILLARRUBIA GONZALEZ GABRIEL - 71932845T

Firmado digitalmente por VILLARRUBIA GONZALEZ GABRIEL - 71932845T
Fecha: 2023.12.10 16:05:02 +01'00'

Investment in reliability will increase until it exceeds the probable cost of errors, or until someone insists on getting some useful work done."

(Glib's Fourth Law of Unreliability)

Resumen

España es el segundo país de Europa con más piscinas. Sin embargo, la literatura jurídica estima que el 20% de las piscinas no están declaradas de forma legal o son irregulares.

La Administración cuenta con un cuerpo de personas que analizan mediante procedimientos manuales, imágenes de satélite o de drones para detectar estructuras ilegales o irregulares. Este método es costoso en términos de esfuerzo, implicación de recursos humanos y tiempo, además de ser un método basado en la subjetividad de la persona que lo lleva a cabo.

La propuesta de este trabajo de investigación pretende diseñar una plataforma basada en sistemas multiagente que incluya técnicas de visión artificial y que permita la detección automática de estructuras ilegales, pudiendo destacar, por ejemplo, la detección de balsas irregulares. Para la consecución exitosa de este trabajo, se emplearán herramientas de información geográfica (SIG) basadas en ortofotografía, combinadas con técnicas avanzadas de visión artificial basadas en redes convolucionales para la detección de objetos.

Además, el uso de una arquitectura multiagente permitirá que el sistema diseñado sea modular, con la posibilidad de que las diferentes partes del sistema trabajen conjuntamente, equilibrando la carga de trabajo. El sistema propuesto ha sido validado mediante pruebas en diferentes ciudades de España.

El sistema ha mostrado resultados prometedores en la realización de esta tarea, con una tasa de acuerdo superior al 97%.

Abstract

Spain stands as the second-ranked European nation in terms of the abundance of swimming pools. However, it has come to light in legal circles that a substantial 20% of these aquatic facilities either evade declaration or exist in an irregular manner.

To tackle this issue, the governing bodies employ a team of individuals who manually scrutinize satellite and drone imagery. Their objective is to pinpoint structures that run afoul of legality or convention. This approach demands significant expenditure of both labor and time, compounded by the inherent subjectivity associated with human interpretation.

This proposal sets forth the ambition to craft a platform capable of autonomously identifying aberrant pools. This endeavor draws upon geographical information systems (GIS) grounded in orthophotography, coupled with cutting-edge machine learning methodologies for precise object detection.

Moreover, a multi-agent architecture comes into play, introducing modularity into the system's framework. This modular design facilitates the collaborative functioning of distinct system components, enabling the equitable distribution of workloads.

The efficacy of the proposed system has been established through rigorous testing across various municipalities in Spain. Encouragingly, the system has yielded promising outcomes in its execution of this task, boasting an impressive F1-Score of 97.1%

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que, con su inquebrantable respaldo y constante aliento, han desempeñado un papel esencial en la realización de este logro significativo en mi trayectoria. Estas palabras están dedicadas con sincero afecto a cada uno de vosotros.

En primer lugar, deseo extender mi agradecimiento al Dr. Gabriel Villarrubia por su incansable dedicación y motivación a lo largo de estos años. Su colaboración fue fundamental para la conclusión de esta tesis, y sin su apoyo incondicional, este logro no habría sido posible. También quiero reconocer al Dr. Juan Francisco de Paz por su codirección y orientación; sus valiosos consejos y asistencia desempeñaron un papel crucial en el éxito de esta investigación doctoral.

En segundo lugar, quiero expresar mi profunda gratitud a mi familia por su apoyo continuo durante todo este tiempo. Especialmente, agradezco a mis padres por ser una fuente constante de inspiración a lo largo de los años y por permitirme alcanzar mis metas. Su respaldo en cada paso ha sido esencial. A Ares le agradezco su paciencia y confianza incuestionable en mi capacidad para lograr lo que me proponga. Desde mi abuela hasta mis tíos y primos, todos ellos son modelos que seguir. Quiero honrar especialmente la memoria de aquellos que ya no están, pero que siguen siendo una parte esencial de mi ser.

Un agradecimiento especial se dirige a mis amigos, quienes siempre respaldan mis proyectos, nuestra relación de apoyo mutuo ha sido crucial. Quiero destacar de manera especial a Marina, quien fue el impulso para iniciar esta tesis.

Asimismo, deseo expresar mi gratitud a mis colegas del grupo de investigación ESALab, ya que su colaboración resultó ser fundamental para el desarrollo de esta investigación.

Cada uno de vosotros habéis dejado una marca indeleble en este viaje académico, y estoy profundamente agradecida por haber contado con vuestro apoyo y contribuciones invaluables.

Contenido

Índice de Figuras

Índice de Tablas

CAPÍTULO I

Introducción

En este capítulo, se introduce el trabajo de investigación, donde se establece la hipótesis principal y se describen los objetivos iniciales. En esta investigación, se presenta un análisis exhaustivo de la aplicación de las redes neuronales convolucionales, en combinación con sistemas multiagente, con el propósito de automatizar la detección y validación de recursos hídricos a partir de imágenes de satélite. Se aborda en detalle el alcance del estudio y sus características más destacadas, así como los componentes y conceptos tecnológicos que sustentan el sistema propuesto.

1 Introducción

1.1 Introducción

La cartografía es la ciencia que produce, difunde y estudia los mapas, en la última década ha experimentado una gran evolución debido a las nuevas tecnologías que han ayudado a la automatización de estos procesos, ya que, hasta hace pocos años, los procesos de revisión cartográfica se han llevado a cabo de forma manual, principalmente aquellos procesos destinados a comprobaciones del área fiscal.

Estos procesos hasta la actualidad requerían importantes inversiones en aviones o helicópteros para lograr obtener imágenes desde altura, lo que hacía que los procesos cartográficos fueran costosos. Debido a esto los municipios no podían realizar levantamientos cartográficos con frecuencia. Por eso uno de los campos de investigación más actuales es la aplicación de las capacidades tecnológicas de las autoridades locales para realizar levantamientos detallados del territorio de los municipios a un coste razonable.

Uno de los hitos más importantes que la cartografía permite es afinar los datos fiscales o verificar la información geográfica en la que se basan los impuestos locales. Un aspecto que tienen en cuenta los ayuntamientos en un proceso de fiscalización es el tamaño de las parcelas, modificaciones en el tamaño de las construcciones y la construcción de piscinas.

Sin embargo, este proceso de detección y validación no es fácil, ya que hay muchos aspectos a tener en cuenta con la cartografía, como la ubicación, la hora del día cuando tomamos la imagen, lo cerca o lejos que está tomada la imagen o los diferentes obstáculos que están presentes.

Además, el resultado de este proceso debe ser evaluado por una persona, una tarea costosa que depende de la subjetividad del individuo, ya que es él quien debe detectar las estructuras construidas en una zona. Las características de la imagen en los procesos

cartográficos pueden llevar a errores en la determinación de la existencia de dichas estructuras.

En esta investigación se hace foco en la detección y validación de forma automática de recursos hídricos en imágenes satelitales, ya que la gestión y el control de recursos hídricos, sobre todo aquellos considerados limitados e indispensables, como el agua potable, ha causado preocupación en los últimos años.

También es de vital importancia conocer las zonas con grandes volúmenes de agua, ya que, en caso de incendio en una zona cercana, los equipos de bomberos pueden hacer uso de ella (Tien, 2007).

Conocer la ubicación exacta de las piscinas es crucial a efectos de recaudación de impuestos y por razones ecológicas. En concreto, la construcción de piscinas en verano repercute en la demanda de agua municipal. Por lo tanto, es comprensible que el gobierno local pida una contribución extra de los propietarios de piscinas en forma de impuesto.

Un tercer problema eminente son las enfermedades transmitidas por mosquitos, que afectan a muchas personas en todo el mundo, sobre todo en países tropicales y subtropicales como Brasil. El agua de las piscinas de los hogares desocupados puede no estar filtrada adecuadamente, y el agua de lluvia acumulada junto con las hojas en descomposición, son el hábitat ideal que garantiza el ciclo de vida de los mosquitos (Passos, 2020).

1.2 Hipótesis y objetivos

Podemos mencionar que el hito principal de este trabajo hace foco en la detección automática de recursos hídricos mediante la aplicación de algoritmos inteligentes con objeto de mejorar y optimizar los procedimientos que se emplean en la actualidad.

Debido a ello, se investigará, la creación de un sistema inteligente que permita detectar depósitos artificiales de agua de manera automática empleando inteligencia artificial y en particular la detección de patrones en imágenes, mediante una arquitectura de agentes que pueda ser fácilmente adaptada a todo tipo de escenarios.

Como punto de partida de esta tesis se parte de la hipótesis de que el uso de algoritmos de aprendizaje profundo, en concreto las redes neuronales convolucionales en combinación con los sistemas multiagente pueden permitir la automatización en la

detección y validación de recursos hídricos a partir de imágenes de satélite facilitando de esta manera la gestión y control de los recursos hídricos de manera automática.

Debido a ello, se proponen los siguientes objetivos:

- Investigar en el diseño y desarrollo de un sistema que pueda generar imágenes satelitales de un área grande utilizando varias fuentes de datos.
- Investigar en los sistemas de detección y clasificación de recursos hídricos a partir de imágenes satelitales generadas utilizando algoritmos de aprendizaje profundo.
- Verificar en los sistemas oficiales si los recursos detectados forman parte de las bases de datos municipales y están registradas de forma legal.
- Diseñar una arquitectura multiagente para interconectar las diferentes partes del sistema que facilite el manejo de varias fuentes de información.

1.3 Motivación

El cambio climático conlleva consigo muchos problemas, y una de las principales preocupaciones en los últimos años es la gestión y el control de recursos limitados e indispensables como es el agua potable (Gleick, 1998). Las regiones que sufren largos períodos de sequía deben concienciarse sobre el uso moderado de este valioso recurso y esforzarse por controlar estrictamente su despilfarro.

En este contexto, la explotación de imágenes por satélite para el estudio y localización de características naturales, como costas, cursos fluviales y bosques, está adquiriendo relevante importancia (Carolyn, 2000; Khanna, and Kondawar, 1991). Este enfoque se vuelve especialmente significativo en las zonas turísticas, donde la demanda de agua es considerable durante el verano, por ejemplo, para campos de golf, piscinas, parques acuáticos, entre otros.

La utilización de tecnología espacial para el monitoreo y gestión de recursos hídricos se convierte así en una herramienta crucial para abordar el desafío de la escasez de agua en regiones afectadas por el cambio climático. Esto permite una mejor planificación y conservación de este recurso tan esencial, y contribuye a la sostenibilidad ambiental y económica de estas áreas.

Teniendo en cuenta este problema, con esta tesis doctoral se pretende investigar en el diseño y desarrollo de técnicas para la detección automática de estructuras que se puede aplicar a diferentes casos de estudio como el de las piscinas ilegales. Esto permitiría a las autoridades locales mantener un inventario de las piscinas situadas en su territorio, pudiendo de esta manera controlar e imponer multas a quienes derrochen agua.

No obstante, contar con un registro actualizado de las piscinas resulta valioso no solo en términos de conservación de agua, ya que el contexto de cambio climático está generando condiciones ambientales propicias para la aparición y propagación de incendios cada vez más devastadores durante la temporada estival y en respuesta al aumento de las temperaturas y las intensas olas de calor. Ante esta situación, las autoridades de Protección Civil suelen activar alertas por riesgo de incendios forestales, buscando enfrentar situaciones de emergencia declaradas. En tales circunstancias, el personal dedicado a la lucha contra incendios necesita acceso a recursos esenciales, ya sean de propiedad pública o privada, los helicópteros que combaten contra incendios están facultados para extraer agua, cuando sea necesario, de cualquier reservorio que se considere adecuado para sus operaciones.

Para ver la relevancia de lucha contra los incendios y la necesidad de tener localizados los recursos hídricos, en el año 2022, en España se registraron oficialmente 55 Grandes Incendios Forestales (GIF), siendo definidos como aquellos que afectan una superficie forestal de al menos 500 hectáreas, o en el caso de las Islas Canarias 250 hectáreas. El Sistema Europeo de Información sobre Incendios Forestales (EFFIS) identificó 61 incendios con áreas superiores a las 500 hectáreas en el año 2022. Los focos más críticos en la península ibérica se pueden observar en la figura 1.

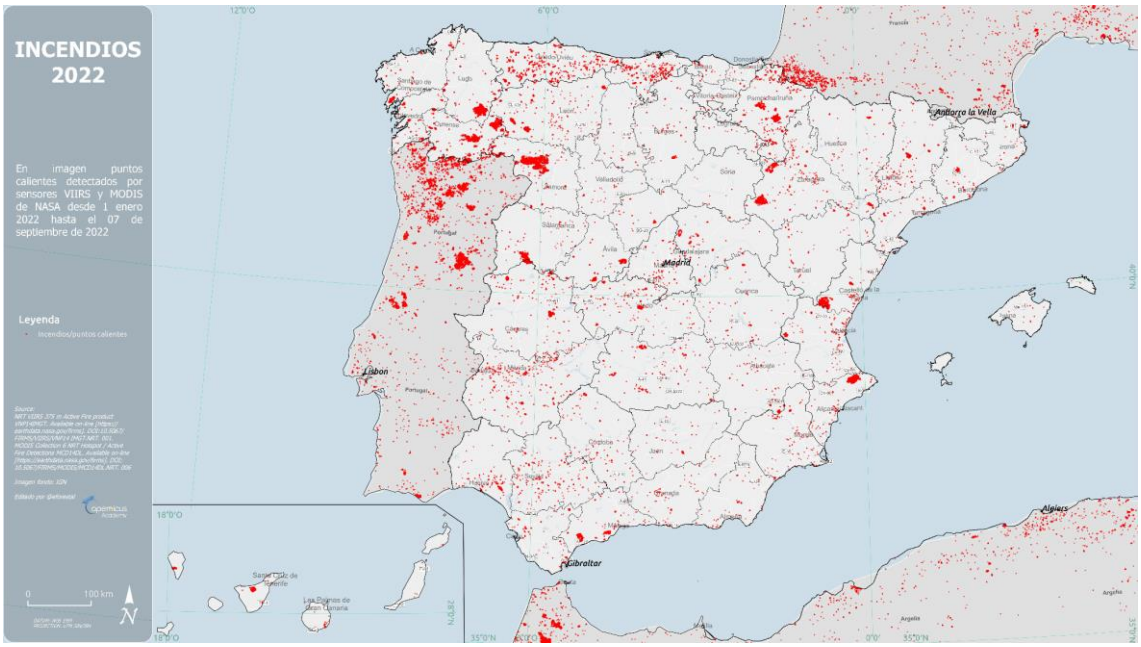


Fig 1 Puntos calientes detectados en España. Fuente: MODIS-VIIRS, NASA

El fenómeno del cambio climático no solo está propiciando un entorno favorable para la propagación de incendios forestales. Actualmente, uno de los problemas de salud más destacados en las regiones tropicales se vincula con la proliferación del mosquito Aedes aegypti. Este insecto es capaz de transmitir enfermedades virales como el dengue, zika y chikungunya. Su ciclo biológico se ve altamente influenciado por las condiciones ambientales. Es lícito afirmar que las transformaciones pronosticadas en el planeta ejercerán un impacto significativo en el panorama epidemiológico de las enfermedades transmitidas por este vector, ya que paulatinamente el clima se volverá más cálido y húmedo (López-Latorre 2016).

Actualmente, la Organización Mundial de la Salud (OMS) está considerando el control y la eliminación de posibles criaderos de mosquitos una medida preventiva fundamental para hacer frente a las enfermedades para ellos (MS, 2020). Sin embargo, las autoridades de salud a menudo ven este desafío incrementado, especialmente en áreas donde una parte considerable de la población no tiene acceso a estructuras de vivienda adecuadas o servicios básicos bien gestionados, como agua, saneamiento y eliminación de residuos sólidos, que son factores clave para riesgo de enfermedades transmitidas por mosquitos (OMS, 2017; MSB, 2020), por lo cual llevar un registro y detectar balsas de agua, es clave para tener conocimiento sobre uno de los principales focos de proliferación del Aedes aegypti y poder contribuir a la prevención del dengue, zika y chikungunya.

1.4 Metodología propuesta

Para llevar a cabo este estudio de investigación, que ha culminado en la elaboración de la presente tesis, se ha empleado el enfoque metodológico de Investigación-Acción (R. O'Brien 1998), reconocido por su orientación hacia la acción y el cambio. Este enfoque permite al investigador abordar problemas claramente definidos con el objetivo de generar nuevos conocimientos, basándose en trabajos previos a lo largo del tiempo. En la actualidad, constituye un enfoque común en la investigación empírica. El proceso metodológico se desglosa de la siguiente manera: (1) identificación precisa del problema, (2) exploración de hipótesis potenciales, selección de una hipótesis y formulación de una propuesta, (3) verificación de la hipótesis elegida y (4) obtención de conclusiones a partir de la evaluación de los resultados obtenidos.

Siguiendo esta metodología, se desarrolla una propuesta orientada a abordar la problemática identificada. En la fase final del estudio, se derivan conclusiones sustanciales a partir de la evaluación de los resultados obtenidos en la investigación. Estos hallazgos se revisan en colaboración con el tutor, lo que a su vez permite sugerir las siguientes etapas del proceso investigativo.

Para lograr esto, se establecen actividades específicas que permiten alcanzar los objetivos propuestos y, al mismo tiempo, validar la hipótesis planteada. A continuación, se detallan las actividades programadas a lo largo de esta investigación con dicho propósito:

- Identificación de la problemática: Iniciar con la presentación del problema en su contexto, lo que facilitará la definición de objetivos e hipótesis.
- Exploración del estado actual de la investigación: Analizar la problemática y las soluciones propuestas por otros investigadores en contextos similares. Este análisis debe ser constante durante toda la investigación.
- Formulación de modelos y validación incremental de los objetivos conforme se avanza en la definición de sus componentes. La división de los modelos en componentes facilitará la validación y enriquecerá el proceso de investigación.
- Evaluación de resultados enfrentándolos con procedimientos existentes para determinar el logro de los objetivos y la hipótesis planteada.
- Divulgación de resultados a través de presentaciones y participación en congresos y publicaciones en revistas científicas. La participación en congresos es crucial para el intercambio de ideas de manera directa.

Estas actividades se llevan a cabo de manera iterativa a lo largo de todo el proceso investigativo, lo que confiere a este proceso un carácter incremental y repetitivo, en sintonía con enfoques ingenieriles como el proceso unificado en la ingeniería del software.

En esta metodología, es fundamental organizar reuniones con los directores de la investigación para mantenerse al tanto del progreso. Una vez que se haya definido la problemática inicial, es relevante verificar la pertinencia de la estrategia actual, considerar posibles ajustes e incluso explorar nuevas técnicas o métodos que puedan acercarnos mejor a los objetivos planteados. El objetivo es permitir que la investigación se beneficie de factores como:

- Adaptabilidad a modificaciones: La capacidad de responder ante cambios y la incorporación de tecnologías o técnicas novedosas.
- Eficiencia: La flexibilidad en la planificación del trabajo conlleva ventajas, ya que se pueden añadir nuevos objetivos que inicialmente no se consideraban al definir el problema.
- Gestión del tiempo: Mantener un control sobre el tiempo dedicado a cada etapa nos permite evaluar la presencia o ausencia de riesgos en la planificación.

1.5 Estructura de la tesis

Esta tesis doctoral se presenta dividida en diez capítulos, con el fin de explicar y desarrollar la hipótesis inicial para poder lograr los objetivos establecidos.

Los dos próximos capítulos se dedican a un análisis exhaustivo del estado actual del conocimiento en relación con la problemática en cuestión, así como a la exploración de diversas técnicas de Inteligencia Artificial que puedan aportar soluciones a dicho problema.

El segundo capítulo introduce los conceptos de agentes y sistemas multiagentes, destacando particularmente los sistemas multiagentes basados en agentes virtuales, entre los cuales se destaca PANGEA. En los capítulos tres y cuatro se aborda el tema de las redes neuronales como base del Aprendizaje Profundo (Deep Learning) y su contribución a la detección de objetos en imágenes. El capítulo cinco proporciona un repaso de los trabajos más recientes en la detección de recursos hídricos en imágenes, así como en el uso de redes convolucionales. El capítulo seis desarrolla la arquitectura del sistema multiagente propuesto para la detección de recursos hídricos en imágenes satelitales. El capítulo siete presenta un caso de estudio específico relacionado con la

detección de piscinas, con el propósito de determinar si las piscinas identificadas cumplen o no con las normativas legales y así poder evaluar la solución propuesta. Los resultados obtenidos se exponen en el capítulo ocho, que a su vez presenta las conclusiones derivadas del análisis realizado en el transcurso de este trabajo. También se abordan posibles direcciones futuras de investigación, conformando el capítulo nueve. Finalmente, el capítulo diez incluye las referencias bibliográficas que han sido consultadas como base durante el desarrollo de este trabajo. El capítulo once es un glosario de las siglas usadas en esta tesis.

CAPÍTULO II

AGENTE Y SISTEMAS MULTIAGENTE

VNiVERSiDAD
D SALAMANCA

En este capítulo, se introducen los conceptos generales relacionados con el uso de agentes y sistemas multiagente, destacando sus atributos y ventajas fundamentales que los convierten en herramientas especialmente valiosas en el diseño de sistemas con requisitos de heterogeneidad, modularidad, escalabilidad, paralelismo y flexibilidad. En el contexto de esta investigación, se dedica una atención especial a las tendencias contemporáneas en el desarrollo de sistemas de este tipo desde una perspectiva organizacional.

2 Agentes y sistemas multiagente

El concepto de agente en el ámbito de la Inteligencia Artificial emergió en los últimos años de la década de 1970 y comenzó a adquirir importancia en la década de 1980. A partir de la década de 1990, la tecnología de agentes atrajo la atención tanto en el ámbito académico de la investigación como en la industria (Geneserech, 1994). Esta tecnología introdujo un paradigma innovador en la ingeniería de software (Jennings, 1997), presentando enfoques originales para el análisis, diseño y desarrollo de sistemas de software (Nwana, 1990).

Un Sistema Multiagente (SMA) puede definirse como el conjunto de agentes que operan de forma autónoma para la consecución exitosa de un objetivo común. Estos agentes colaboran para abordar tareas y se distinguen por su flexibilidad, que proviene de su inherente capacidad de aprendizaje y la toma de decisiones autónomas. A través de interacciones con agentes vecinos o de su entorno, los agentes adquieren nuevos conocimientos sobre contextos y acciones. Posteriormente, aplican estos conocimientos para tomar decisiones y ejecutar acciones en el entorno, con el propósito de resolver las tareas asignadas. Es precisamente esta flexibilidad la que permite a los SMA abordar problemas en diversos campos de manera efectiva (López 2018).

2.1 Agente inteligente

El concepto de agente inteligente no se limita a una definición estática. La literatura abarca diversas interpretaciones, que abarcan desde las más simples hasta las más precisas. Estas definiciones se ven influenciadas por diferentes disciplinas, como la ingeniería de software, la inteligencia artificial, la ciencia cognitiva y la informática en general. En lugar de enumerar de manera exhaustiva diversas definiciones, se expondrán dos definiciones generales de agentes propuestas por Russell (Russell, 1995) y Maess (Maess, 1995), las cuales son ampliamente aceptadas en varias comunidades de investigación.

Según Russell, se puede definir un agente como una identidad que recibe información de su entorno a través de sensores y que puede actuar en ese entorno mediante efectores. Bajo esta perspectiva, un agente puede ser cualquier entidad, ya sea física o virtual, que interactúa con el entorno mediante percepciones y acciones. Podemos definir entonces que un agente es un componente software que recibe información del entorno y genera una respuesta que desencadena una acción en el sistema. En muchos casos, un agente representa la combinación de elementos físicos (infraestructura informática) y virtuales (software que opera en esa infraestructura).

Por su parte, Maess ofrece una definición más detallada de agente que amplía la definición previamente citada: "Los agentes **autónomos** son sistemas **computacionales** los cuales operan en entornos dinámicos y complejos. Estos sistemas perciben y actúan de manera autónoma en su entorno, intentado cumplir una serie de **objetivos** o tareas predefinidos". Elementos cruciales en esta definición incluyen, la autonomía, la naturaleza computacional y la presencia de objetivos. El término "Autonomía" hace referencia a la capacidad de los agentes computacionales de operar sin necesidad de intervención directa de otras entidades, manteniendo cierto grado de control sobre sus acciones. Computacionalmente se hace una distinción entre los agentes de interés en ingeniería (agentes computacionales) de los agentes biológicos (humanos, animales, bacterias), una distinción que no siempre es evidente desde la definición original. La asignación de objetivos a los agentes indica que estos interactúan con su entorno con el fin de alcanzar metas específicas, demostrando un comportamiento racional al minimizar o maximizar sus medidas de rendimiento según el contexto. En este contexto, el comportamiento hace referencia a las acciones que los agentes emprenden en respuesta a estímulos sensoriales o secuencias de tales estímulos.

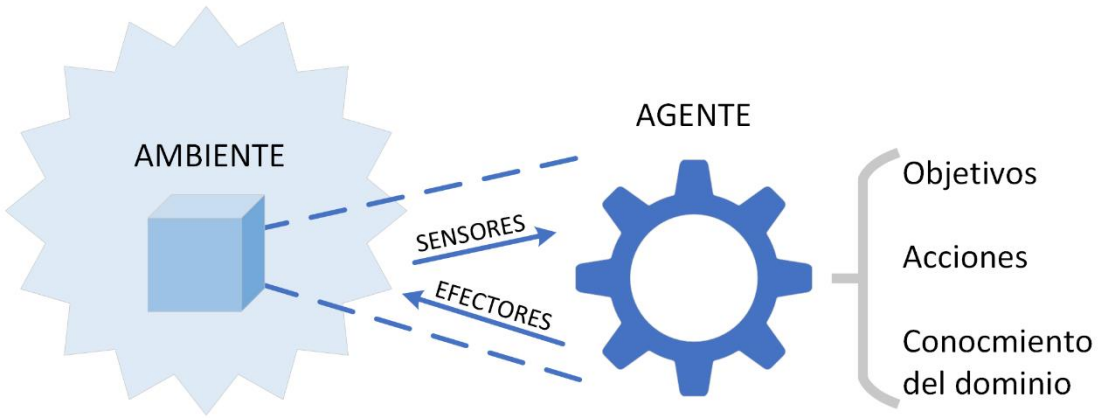


Fig 2 Interacción entre un agente y su entorno

Como se ilustra en la Figura 2, además de las entradas sensoriales, las acciones y los objetivos, un agente puede incorporar un conocimiento de dominio (entendimiento acerca de un entorno específico o un problema por resolver). Este conocimiento puede abarcar enfoques algorítmicos, basados en técnicas de Inteligencia Artificial (IA) (como reglas, lógica difusa, redes neuronales, aprendizaje automático), heurísticas, entre otros. En el ámbito de la IA, a menudo se refiere a un agente como un agente inteligente. A diferencia de los objetos (que se definen mediante atributos y métodos), un agente se define a través de su propio comportamiento. Varias propiedades que se atribuyen a los agentes, en diferentes grados según la situación problemática, son:

- Autonomía: Un agente inteligente actúa independientemente, manteniendo de esta forma control sobre su estado interno.
- Reactividad: Un agente se considera reactivo cuando interactúa con su entorno y es capaz de adaptarse y responder a los cambios del entorno de manera oportuna.
- Proactividad: Un agente es proactivo cuando puede establecer y perseguir metas, no solo reaccionar ante eventos, demostrando iniciativa.
- Habilidad Social: Los agentes tienen algún lenguaje de comunicación con el fin de interactuar y tal vez cooperar con otros agentes (posiblemente humanos). Esto se conoce como capacidad de comunicación del agente, permitiendo que el agente inteligente obtenga información de diversas fuentes.
- Capacidad de Cooperación: Esto implica que un agente inteligente coopera con otros agentes para alcanzar objetivos específicos.
- Habilidad de Razonamiento: Los agentes inteligentes pueden inferir y extrapolar según el conocimiento y las experiencias actuales.
- Comportamiento Adaptativo: Los agentes inteligentes aprenden o modifican su comportamiento con base en experiencias previas.
- Confiabilidad: Los usuarios deben confiar en que sus agentes actuarán y proporcionarán información de manera precisa y actuarán en su mejor interés.

2.2 Clasificaciones de agente

A la hora de clasificar los agentes y tras una revisión detallada, en la literatura pueden encontrase diferentes ejemplos de clasificaciones (Russell and Norvig 1995), (Nwana 1996) o (Cvetković and Parmee 2002). Las características clave de los agentes pueden ayudarnos a clasificar agentes de forma útil. La tabla 1 enumera varias de las propiedades mencionadas.

Descripción	Nombre	Propiedad
Respuesta rápida a los cambios en el medio.	Identificación y actuación	Reactivo
Actúa en respuesta al entorno	Proactivo con propósito	Orientado a un fin
Se comunica con otros agentes, incluso con personas	Social	Comunicativo
Cambios adaptativos en base a la experiencia		Aprendizaje
Continuo proceso en ejecución		Temporalmente continuo
Capaz de transportarse a sí mismo de una máquina a otra.		Móvil
Ejerce control sobre sus propias acciones		Autónomo
Las Acciones no están programadas		Flexible
Creíble “personalidad” y “estado emocional”		Personaje

Tabla 1 Descripción de las propiedades de los agentes

2.3 Sistemas multiagente

Los sistemas formados por múltiples agentes que se relacionan entre sí se denominan Sistema Multiagente. Los problemas complejos que no puede abordar un único programa informático o algoritmo aislado suelen abordarse mediante el diseño de soluciones tecnológicas que emplean diseños de sistemas multiagente.

La definición de un agente es aplicable de manera similar en el contexto de un sistema multiagente. Diversas definiciones han sido propuestas, dependiendo de la disciplina de investigación a la que pertenezcan. De nuevo, el propósito aquí no es enumerar y analizar múltiples definiciones, sino seleccionar una definición que parezca ser amplia y cercana a la comunidad de investigación en la rama de la ingeniería. Según Stone (Stone y Veloso, 2000), un SMA puede describirse como "una red de entidades de resolución de problemas (agentes) débilmente acoplados, que colaboran para encontrar soluciones a problemas que no se podrían abordar mediante las capacidades o el conocimiento individual de cada entidad (agente)".

La colaboración entre los agentes en un SMA implica la existencia de algún tipo de cooperación entre los agentes individuales. La tipología de cooperación propuesta por Doran (Doran et al., 1996) se presenta de manera esquemática en la Figura 3.

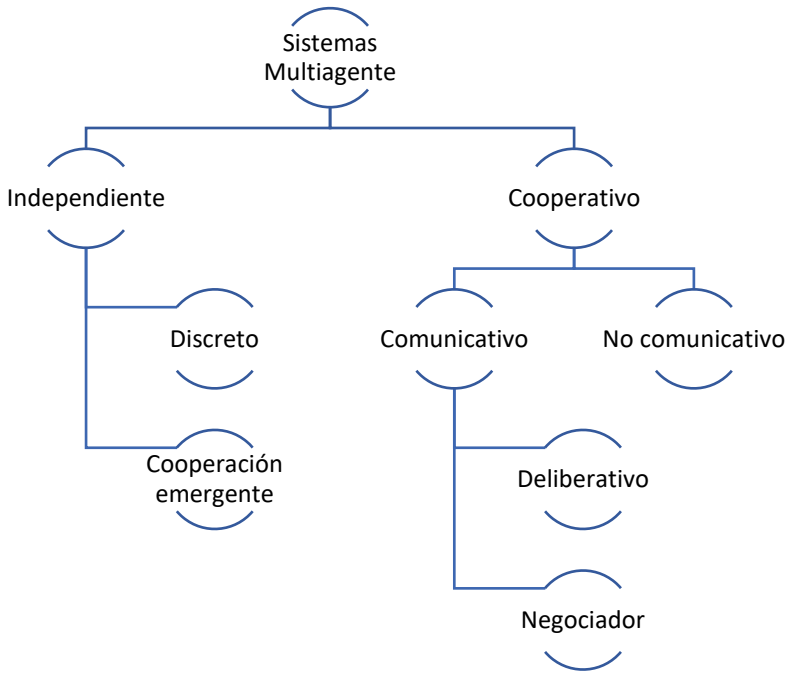


Fig 3 Tipología de cooperación SMA (Doran et al., 1996)

Un SMA se considera independiente si cada agente individual persigue sus propios objetivos de manera independientemente a los demás agentes que forman parte del sistema. Un SMA es "discreto" cuando sus agentes son independientes y sus objetivos no están interrelacionados. En el caso de los SMA discretos, la cooperación no está implícita. Sin embargo, los agentes pueden cooperar sin tener la intención de hacerlo, lo que resulta en una cooperación emergente.

En contrapartida, los SMA cooperativos son configuraciones en las cuales los agentes incorporan en sus itinerarios algún tipo de colaboración con otros agentes. Esta colaboración puede ser de naturaleza comunicativa, implicando la emisión y recepción deliberada de mensajes. La comunicación deliberada puede adoptar dos modalidades: negociación o deliberación. Es esencial discernir el uso del término "deliberativo" en la Figura 3 del concepto de "agente deliberativo". En este contexto, un SMA deliberativo alude a la noción de que los agentes planifican colectivamente sus acciones para fomentar la cooperación mutua. Los sistemas de negociación se asemejan a los deliberativos, pero incluyen un componente adicional de competencia.

Adicionalmente, los SMA cooperativos pueden adoptar una modalidad no comunicativa, en la cual los agentes coordinan sus actividades cooperativas al observar y reaccionar al comportamiento de los demás agentes. Varias propiedades definen a un SMA (Sycara, 1998) de la forma siguiente:

- Los agentes de forma individual poseen un conocimiento parcial para abordar los objetivos a los que se enfrentan, lo que resulta en una perspectiva limitada.
- Los datos se distribuyen de forma descentralizada.
- Los cálculos se ejecutan de manera asincrónica.
- No hay una supervisión global del sistema ni un diseño centralizado.
- Estos sistemas suelen ser abiertos, lo que implica la posibilidad de que nuevos agentes se unan al sistema o que agentes existentes lo abandonen.

2.3.1 Plataformas sistemas multiagente

Actualmente las líneas de investigación se orientan en crear sistemas cada vez más abiertos y dinámicos. Esto implica añadir nuevas capacidades como la adaptación, la reorganización, el aprendizaje, la coordinación, etc.

Pensar en términos de diseño organizativo difiere del enfoque centrado en el agente que ha sido dominante durante muchos años. Un SMA orientado a la organización no se considera en términos de estados mentales, sino sólo de capacidades y limitaciones, de conceptos organizativos como roles (o función, o posición), grupos (o comunidades), tareas (o actividades) y protocolos de interacción (o estructura de diálogo), por tanto, en lo que relaciona la estructura de una organización con el comportamiento observable externamente de sus agentes.

Las Organizaciones de Agentes virtuales (VOs) (Feber 2004, Foster 2001) surgieron como respuesta a esta idea; incluyen un conjunto de agentes con funciones y normas que determinan su comportamiento, y representan un lugar donde estas nuevas capacidades asumirán un papel crítico. Posibles topologías organizativas y aspectos como los mecanismos de comunicación y coordinación determinan en gran medida la flexibilidad, apertura y dinamismo que puede ofrecer un sistema multiagente.

Existen muchas plataformas diferentes para crear SMA que facilitan el trabajo del agente; sin embargo, las que permiten la creación de VO son muchas menos, y es difícil encontrar una única plataforma que contenga todos los requisitos de una VO.

Todas las plataformas de creación de SMA existentes hasta la fecha deben estudiarse de acuerdo con dos categorías principales: las que simplemente soportan la creación e interacción de agentes sin que se incorporen aspectos organizativos, y las que permiten la creación de organizaciones virtuales con conceptos clave como normas y roles (De Paz, J. et al., 2014).

2.3.1.1 Plataformas que no incorporan aspectos organizativos.

Dentro del ámbito de los SMA, han surgido diversas plataformas diseñadas para agilizar la generación y administración de agentes, cada una con enfoques y características particulares. Estas plataformas desempeñan un papel esencial al proporcionar un entorno propicio para la implementación y operatividad de SMA en diversas aplicaciones. A continuación, se examinarán algunas de estas plataformas junto con sus principales contribuciones:

Una de las primeras plataformas que merece atención es FIPA-OS, una derivación directa del estándar FIPA (Foundation for Intelligent Physical Agents) (O'Brien, 1998). La FIPA se dedica a promover la estandarización y el desarrollo de tecnologías para agentes inteligentes. A través de FIPA-OS, se busca implementar y aprovechar las

directrices establecidas por el estándar FIPA para lograr interacciones y comunicaciones eficientes entre agentes.

En el panorama de las plataformas de agentes, se destaca la April Agent Platform (AAP) (Dale, 2011), que implementa el lenguaje April en lugar del más común Java. Una de sus fortalezas radica en su habilidad para facilitar el desarrollo y la implementación de agentes en el entorno de Internet. Además, su compatibilidad con los estándares de Servicios Web y Web Semántica añade un nivel adicional de versatilidad.

Otra plataforma relevante es JAvaScript Object Notation (JASON) (Bordini, 2005; Bordini, 2007). Su contribución significativa radica en la fácil implementación de agentes basados en la arquitectura de Bases de Datos Interoperantantes (BDI) (Rao, 1991), que se enfoca en creencias, deseos e intenciones como componentes centrales. El núcleo de la plataforma JASON contiene AgentSpeak, un intérprete de agentes que expande el lenguaje (Rao, 1996).

En el ámbito práctico, la plataforma Java Agent DEvelopment Framework (JADE) (Bellifemine, 1999) ha demostrado ser valiosa para el desarrollo de SMA en escenarios reales. Con su enfoque en la creación y administración de agentes, JADE proporciona herramientas y estructuras que simplifican la creación y el funcionamiento de SMA.

Por su parte, la plataforma JADE, una de las más conocidas y empleadas en la actualidad, se enfoca en la implementación del modelo de referencia FIPA, ofreciendo una sólida infraestructura de comunicación y servicios de plataforma. A través de JADE, se facilita la gestión de agentes y se proporciona un conjunto de herramientas para el desarrollo y la depuración de SMA.

Dentro del contexto de los sistemas de agentes basados en el modelo creencia-deseo-intención o BDI, se puede destacar el de Jadex (Braubach, 2004). Esta plataforma ofrece un marco de software para la creación de agentes que se alinean con el modelo BDI, permitiendo una transición fluida desde el desarrollo de agentes convencionales hacia la adopción de este enfoque.

Aunque estas plataformas comparten la capacidad de crear agentes y gestionar su comunicación y servicios, es fundamental destacar que, en el caso de las VO, se debe prestar atención especial a los aspectos normativos y organizativos que deben estar integrados en la propia plataforma. Estas plataformas no proporcionan exclusivamente herramientas técnicas, sino que también desempeñan un rol crucial en la creación de entornos donde los agentes pueden colaborar de manera efectiva y coherente para lograr objetivos específicos.

2.3.1.2 Plataformas que soportan aspectos organizativos

MadKit (Hübner 2007) surgió como una de las primeras plataformas en considerar aspectos organizativos fundamentales. La arquitectura de la plataforma se basa en el modelo AGR (agente-grupo-rol) (Gutknecht 1997). Sin embargo, aunque puede manejar el concepto de rol, no lo aborda como una entidad de clase en sí, y el comportamiento asociado al rol se implementa directamente en el agente que adopta ese rol. Los roles están estrechamente vinculados a las arquitecturas de agentes. Este enfoque afecta negativamente la reusabilidad y el modularidad de las organizaciones (Gaud 2008).

Otra plataforma pionera en términos de aspectos estructurales fue Jack Teams. JACK Teams es una extensión de JACK Intelligent Agents (Busetta 1998), que proporciona un marco de modelado orientado a equipos. Ambos son ampliaciones del lenguaje de programación Java. El código fuente implementado se compila primero en código Java convencional antes de ejecutarse.

S-MOISE+ se presenta como un middleware organizativo que sigue el modelo MOISE (Hübner 2009). Constituye una ampliación de SACI, en la que los agentes poseen una arquitectura consciente de la organización. Se han desarrollado sistemas en colaboración con JASON, utilizando S-MOISE+ como middleware para lograr un modelo más completo (Hübner 2009). El resultado fue J-Moise+ (Hübner 2007), que comparte conceptos generales del sistema con S-MOISE+. La distinción principal radica en la programación de los agentes: en S-MOISE+, los agentes se programan en Java (mediante una arquitectura de agentes simplificada), mientras que en J-MOISE+ se programan en AgentSpeak.

2.3.1.3 Plataformas de agentes virtuales

Uno de los enfoques actuales de investigación en SMA busca desarrollar sistemas cada vez más abiertos y dinámicos, adaptables al contexto. Esto conlleva la adición de nuevas habilidades como adaptación, reorganización, aprendizaje y coordinación. En respuesta a esta idea, surgieron las organizaciones de agentes virtuales (Foster et al., 2001; Ferber et al. ,2004), las cuales comprenden un grupo de agentes con roles y reglas que definen su conducta, y se convierten en el entorno donde estas habilidades innovadoras desempeñarán un rol esencial. Las diversas formas de organizar los aspectos relativos a los mecanismos de comunicación y coordinación tienen un impacto significativo en la flexibilidad, apertura y dinamismo que puede ofrecer un sistema multiagente.

Las plataformas existentes hasta la fecha para la creación de SMA pueden clasificarse en dos categorías principales: aquellas que se enfocan en brindar soporte para la creación e interacción de agentes, y aquellas que permiten establecer organizaciones virtuales con conceptos fundamentales como normas y roles.

A pesar de la disponibilidad de varias plataformas distintas para la creación de SMA, las que permiten la formación de VO son considerablemente menos numerosas, y encontrar una plataforma única que satisfaga todos los requisitos para tal fin es un desafío.

Una de las principales desventajas de las plataformas orientadas a VO es la ligera pérdida del concepto de servicio y, en consecuencia, la gestión de estos servicios y del Directory Facilitator (DF) descrito en el estándar FIPA. THOMAS se basa en la idea de que no existen agentes internos y los servicios de arquitectura se ofrecen como servicios web finales. De este modo, el producto final es totalmente independiente de cualquier plataforma de agentes y se encuentra totalmente orientado a SMA abiertos (Ginet 2010).

En esta investigación, se ha optado por la utilización de un MAS denominado Platform for Automatic coNstruction of orGanizations of intElligent Agents (PANGEA), que se distingue de otros sistemas preexistentes como SPADE, la Biblioteca de Python, JADE u osBrain por su capacidad para generar organizaciones virtuales. PANGEA adopta una arquitectura BDI y de Sistemas de Razonamiento basado en Casos (CBR)-BDI, se adhiere al estándar FIPA-ACL, facilita la difusión de mensajes según roles, suborganizaciones o directamente entre agentes, y su programación se realiza en Java.

2.3.1.4 PANGEA

Los trabajos que se han analizado en la literatura sobre PANGEA (Zato, 2012) desarrollan una arquitectura que permite que los diferentes agentes se adecuen a las necesidades computacionales del sistema de forma dinámica. La plataforma que puede crear, gestionar y controlar integralmente VOs, tiene las siguientes características principales:

- Modelos de agentes distintos, que incluyen arquitectura BDI y CBR-BDI.
- Control del ciclo de vida de los agentes mediante herramientas gráficas.
- Protocolo de comunicación que habilita la difusión, multidifusión basada en roles o suborganizaciones, y comunicación entre agentes.
- Herramienta de depuración.
- Módulo para interactuar con agentes FIPA-ACL.

- Gestión de herramientas y servicios para el descubrimiento de servicios.
- Web Services.
- Soporte para organizaciones de cualquier topología.
- Gestión de organizaciones.
- Servicios para la reorganización dinámica de la organización.
- Servicios para la distribución de tareas y equilibrio de la carga de trabajo.
- Motor de reglas de negocio para asegurar la conformidad con estándares establecidos.
- Programado en Java y altamente extensible.
- Capacidad de tener agentes en diversas plataformas (Windows, Linux, macOS, Android e iOS).
- Interfaz para la supervisión de las organizaciones.

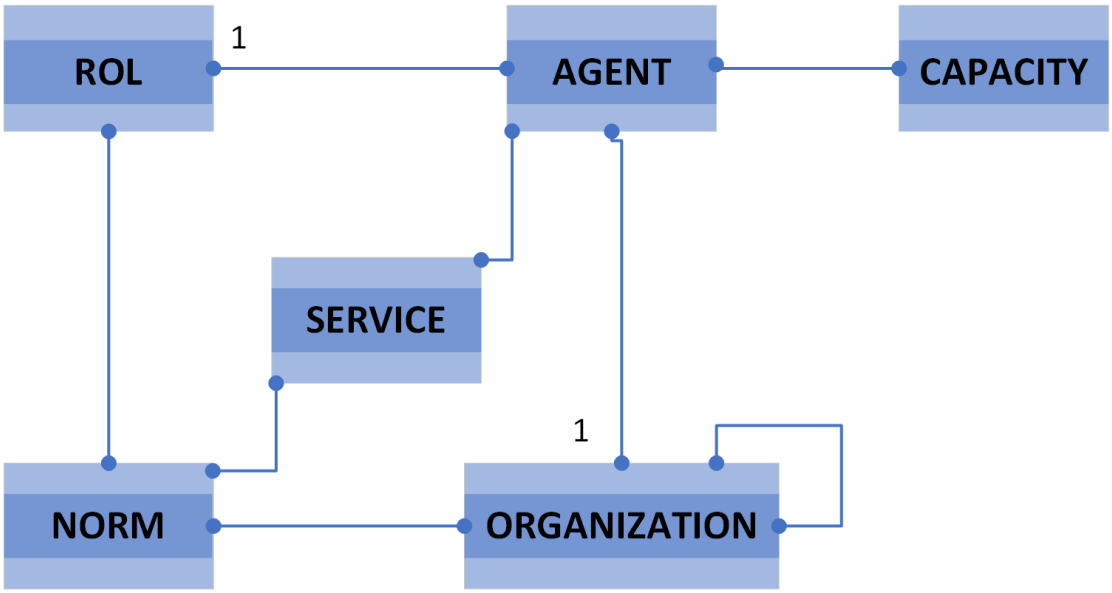


Fig 4 Principales clases del sistema PANGEA

La Figura 4 muestra las entidades principales del sistema y cómo los roles, normas y organizaciones son clases que facilitan la inclusión de aspectos organizativos. Los servicios también se incluyen como entidades separadas de los agentes, lo que facilita su flexibilidad y adaptación.

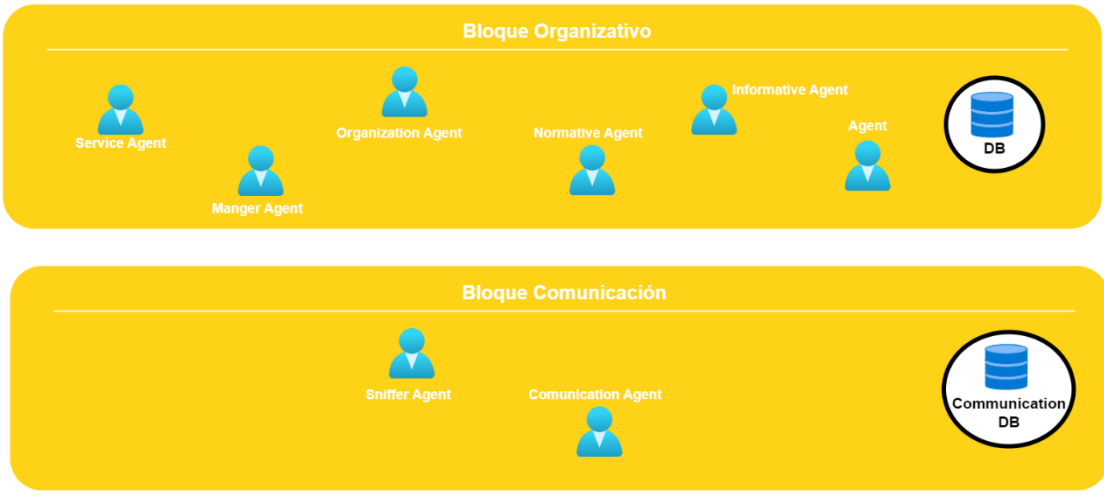


Fig 5 Arquitectura PANGEA

Al lanzar la ejecución, se inicia el Bloque de comunicación, y la plataforma de agentes proporciona automáticamente los siguientes agentes para facilitar el control de la organización:

- Organization Manager: responsable de la gestión real de organizaciones y suborganizaciones, verifica la entrada/salida de agentes y asigna roles.
- Information Agent: accede a la base de datos con toda la información relevante del sistema.
- Service Agent: registra y controla la operación de los servicios ofrecidos por los agentes.
- Norm Agent: garantiza el cumplimiento de las normas refinadas en la organización.
- Communication Agent: controla la comunicación entre agentes y registra las interacciones.
- Sniffer: administra el historial de mensajes y filtra la información.

La plataforma ofrece dos modos de operación: En el primer modo todos los agentes se encuentran en la misma máquina, en el segundo modo los agentes están distribuidos en máquinas diferentes para conseguir una mayor distribución y escalabilidad del sistema. En ambos modos se busca maximizar la distribución de recursos mediante servicios web que implementan los servicios finales de las aplicaciones de usuario. Esto permite emular

una arquitectura cliente-servidor, donde los agentes proveedores y consumidores interactúan.

Cada suborganización recibe automáticamente un OrganizationAgent durante su creación, que controla la suborganización y se comunica con el OrganizationManager si es necesario. La comunicación se basa en el protocolo Internet Relay Chat (IRC), permitiendo una comunicación en tiempo real, que es gestionada mediante los agentes CommunicationAgent y el Sniffer.

Con esta arquitectura modular basada en agentes con roles totalmente definidos, la plataforma apuesta por su elevada flexibilidad, facilitando la creación, gestión y control de Organizaciones Virtuales. Otra ventaja de esta arquitectura es que permite que los servicios del sistema aumenten bajo demanda. Cuando un agente se une la plataforma, debe comunicar qué servicios están disponibles y cuales son aquellos que puede ofrecer a otras entidades.

CAPÍTULO III

MACHINE LEARNING

VNiVERSiDAD
D SALAMANCA

En este capítulo se presenta el aprendizaje automático haciendo hincapié en las redes neuronales y el aprendizaje de estas ya que forman la base de la visión artificial necesaria para identificar recursos hídricos en imágenes.

3 Machine Learning

En este apartado ser va a introducir los conceptos claves sobre el aprendizaje automático, el cual es considerado un subcampo de la inteligencia artificial que se basa en métodos estadísticos para obtener la capacidad de realizar tareas específicas mediante la observación de muchos ejemplos precisos acerca de esas tareas, también se conoce como ML, por sus siglas en inglés.

Los métodos tradicionales de aprendizaje automático que son ampliamente utilizados en visión por ordenador requieren de un procesamiento preliminar de imágenes y una transformación extensa de los datos. Estas técnicas suelen estar caracterizadas por presentar dificultades a la hora de tratar datos naturales en su etapa inicial o sin procesar. Durante muchos años, el hecho de desarrollar sistemas automáticos de reconocimiento de patrones , era una tarea compleja, ya que requería de un sistema que pudiera extraer las características de las imágenes, con el fin de convertir los datos sin procesar (como los valores de cada píxel de una imagen) a un formato adecuado, como pueden ser los vectores de características, que puedan ser utilizados por un sistema de aprendizaje, generalmente un clasificador, con el fin de clasificar o detectar patrones en la las imágenes de entrada. Todo esto implicaba necesariamente el diseño e implementación de un conjunto de métodos que permita a las máquinas descubrir automáticamente las representaciones necesarias para la detección o clasificación utilizando datos sin procesar, a este conjunto se le conoce como aprendizaje de representaciones.

3.1 Deep Learning

Desde 2006, los métodos de aprendizaje profundo han adquirido una mayor importancia (Hinton y Salakhutdinov, 2006). La relevancia de este método en el reconocimiento de imágenes puede atribuirse a varios factores, pudiendo destacar entre ellos, la aparición de conjuntos de datos de entrenamiento ampliamente anotados, como ImageNet (J. Deng et al., 2009), que han demostrado plenamente las poderosas capacidades de aprendizaje de estos métodos. Además, la utilización de estos sistemas ha ido de la mano del rápido desarrollo de sistemas informáticos altamente paralelos, como los clústeres

de unidades de procesamiento de gráficos (GPU). Esto ha favorecido la optimización de los tiempos empleados en los procedimientos de entrenamiento no supervisado, basado en clases, guiado por un codificador automático (Deng et al., 2010) o una máquina Boltzmann restringida (Dahl et al., 2010). La implementación de técnicas como la eliminación de datos y el aumento de datos han resuelto el problema del sobre aprendizaje durante el entrenamiento (Hinton et al., 2012; Krizhevsky et al., 2012).

La adición de la normalización por lotes, Batch Normalization (BN) tiene un gran impacto en la eficiencia del entrenamiento de redes neuronales profundas (Ioffe y Szegedy, 2015). Además, se ha realizado un estudio en profundidad de diferentes arquitecturas de red, como AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2013), Overfeat (Sermanet et al., 2013), Grupo de Geometría Visual. (VGG) (Simonyan y Zisserman, 2014) y Residual Network (ResNet) (K. He et al., 2016), con la finalidad de mejorar su rendimiento.

En el contexto del Aprendizaje Profundo, se puede resumir los esfuerzos que numerosos grupos de investigación realizan para que los niveles de representación emerjan a través de la combinación de módulos simples, con funcionamiento no lineal, los cuales transforman gradualmente la representación desde los niveles originales, como entrada sin formato, a niveles superiores de abstracción. Esta serie de transformaciones están permitiendo en el campo del aprendizaje de funciones cada vez más complejas y precisas. En las labores de clasificación, las clases de mayor representación enfatizan los aspectos relevantes de la entrada original, reduciendo los sesgos irrelevantes.

3.2 Redes neuronales

En la década de 1940 (Pitts and McCulloch, 1947), surgen las redes neuronales con el objetivo inicial de emular el sistema cerebral humano para resolver problemas de aprendizaje generales basados en principios. Estas redes experimentaron un auge en las décadas de 1980 y 1990 con la introducción de un algoritmo mediante el cual se minimizan los errores en el aprendizaje automático mediante la retropropagación de los errores de las capas ocultas a la capa de salida (Rumelhart et al. ,1986).

Sin embargo, limitaciones como el sobreajuste, la disponibilidad limitada de datos a gran escala, la capacidad de cálculo restringida y el rendimiento comparativo moderado en comparación con otras técnicas de aprendizaje automático, condujeron a una disminución de su popularidad a principios de la década de 2000.

Las redes neuronales se asemejan a la función de las neuronas del cerebro humano, permitiendo de este modo que las redes neuronales en una computadora reconozcan patrones y den solución a problemas comunes en áreas como la inteligencia artificial, el aprendizaje automático y el aprendizaje automático profundo.

También conocidas como redes neuronales artificiales (ANN) o redes neuronales simuladas (SNN), las redes neuronales son un subconjunto del aprendizaje automático y son la base de los algoritmos de aprendizaje profundo. Su estructura y nomenclatura están inspiradas en el cerebro humano, simulando cómo las neuronas biológicas se transmiten señales entre sí.

Una red neuronal artificial está formada por capas y nodos, los cuales están formados por una capa inicial o, de entrada, seguida de una o varias capas ocultas y finalmente una capa de salida. Cada neurona artificial denominada nodo, tiene un peso y un umbral, que son los factores mediante los cuales se establecen conexiones con otros nodos.

Estas redes se entrenan con grandes volúmenes de datos para mejorar la precisión con el tiempo. Cuando se ajustan adecuadamente, se convierten en potentes motores de inteligencia artificial que permiten una rápida clasificación y agrupación de datos. En las redes neuronales multicapa las capas ocultas aprenden a representar las entradas de la red para ayudar a predecir las salidas objetivo (Bengio, 2001).

3.2.1 Tipos de redes neuronales

3.2.1.1 Redes neuronales perceptores multicapa

Un perceptrón multicapa o red neuronal de propagación directa (MLP) está formada normalmente por neuronas sigmoideas, debido a la naturaleza no lineal de la gran parte de los problemas que se presentan en el mundo real. Estos modelos a menudo se alimentan con datos de entrenamiento y forman la base de campos como la visión artificial y el procesamiento del lenguaje natural, además de sustentar otras redes neuronales (Martínez et al., 2023).

Desde el comienzo del reconocimiento de patrones (Selfridge, 1958; Rosenblatt, 1957), los investigadores han buscado reemplazar las características diseñadas a mano con redes de tipo multicapa entrenables. Hasta mediados de los años 1980 esta solución no fue ampliamente entendida. Durante los últimos años, se ha descubierto que las arquitecturas multicapa se pueden entrenar mediante procesos de descenso de gradiente estocástico ya que optimizan la función objetivo sustituyendo el gradiente real por una

estimación, esto es posible siempre que los módulos sean funciones con pendientes moderadamente suaves de las entradas y sus pesos internos, la pendiente se puede calcular mediante propagación hacia adelante. Varios grupos independientes descubrieron esta idea durante los años 1970 y 1980 (Werbos, 1974; Parker, 1985; LeCun, 1985; Rumelhart et al., 1986).

La propagación hacia adelante para calcular la pendiente de la función objetivo ponderada de un conjunto de módulos se realiza mediante el empleo de la regla de la serie para la derivada. La derivada (o pendiente) del objetivo referente a la entrada del módulo se puede calcular devolviendo la derivada (o pendiente) del objetivo con respecto a la salida del siguiente módulo (o con la entrada de corriente del módulo). Esta ecuación de propagación hacia adelante se aplica de forma iterativa para propagar el gradiente a través de todos los módulos, tomando como inicio la salida superior, dónde se produce la predicción, siendo el destino la entrada inferior, la cual toma la entrada externa. Una vez calculados los gradientes, es fácil calcular los gradientes ponderados de cada módulo.

Las arquitecturas de redes neuronales de propagación directa se utilizan en muchas aplicaciones de aprendizaje profundo debido a su papel fundamental en las redes neuronales artificiales. En esta configuración, la información se comunica en una sola dirección, desde la capa de entrada a través de una o más capas ocultas hasta la capa de salida, sin retroalimentación directa (Figura 6). Estas redes están compuestas por nodos llamados neuronas, organizados en capas.

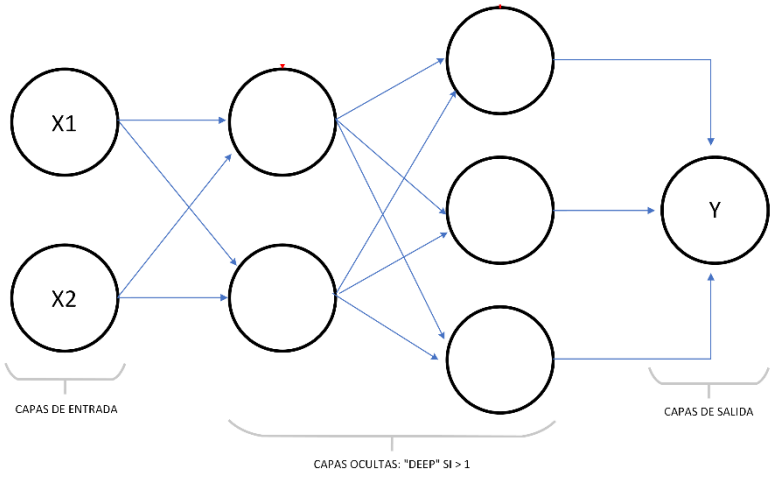


Fig 6 Red neuronal de perceptores multicapa

Cada neurona de la red neuronal recibe una entrada ponderada de las neuronas de la capa anterior. Posteriormente realiza una suma ponderada de estas entradas y aplica una función de activación para determinar su salida. El paso de una capa a la siguiente se produce cuando un conjunto de unidades calcula la suma ponderada de las entradas de la capa anterior y comunica el resultado a través de una función no lineal.

Actualmente, la función de activación no lineal más utilizada es la unidad lineal rectificada (ReLU), que se define como f(z) = max (z, 0). Aunque en décadas anteriores se han utilizado no linealidades más ligeras como tanh(z) o 1/(1+exp(-z)), ReLU tiende a aprender más rápido en redes multicapa, permitiendo el entrenamiento de redes profundas supervisadas sin requerir un entrenamiento previo no supervisado. Se denominan unidades ocultas a aquellas unidades que no pertenecen a la capa de entrada o salida. Estas unidades tienen la función de distorsionar la entrada de forma no lineal para lograr una separación lineal de las categorías en la capa final. En otras palabras, estas capas permiten a la red encontrar patrones y características complejas en los datos para realizar tareas de clasificación o predicción. Por lo que se puede considerar que las capas ocultas deforman la entrada de forma no lineal, consiguiendo que las categorías sean linealmente separables en la capa final.

A finales de la década de 1990, las redes neuronales y los métodos basados en la propagación directa fueron abandonados en gran medida porque no era posible aprender técnicas de aprendizaje si el conjunto de datos era muy reducido. En particular, se piensa que el simple descenso de gradiente queda estancado en perfiles de peso donde ningún cambio pequeño puede reducir el error medio.

El interés en las redes de propagación directa revivió alrededor de 2006 (Hinton et al., 2005; Bengio et al., 2006; Hinton et al., 2006; Ranzato et al., 2006) por un grupo de investigadores convocados por el Instituto Canadiense de Estudios Avanzados (CIFAR).

Los investigadores introdujeron procesos de aprendizaje no supervisados que pueden generar clases de detectores de características sin datos etiquetados previos. El objetivo de aprendizaje de cada capa del detector de características es poder reproducir o modelar el comportamiento del detector de características (o entrada sin procesar) de la capa inferior. Al entrenar previamente varias capas de detectores de características cada vez más complejos utilizando este objetivo de reconstrucción, los pesos de la red profunda se pueden inicializar con valores razonables.

La capa de unidad de salida final puede luego agregarse a la parte superior de la red y sintonizar todo el sistema en profundidad utilizando la propagación directa estándar (Hinton, 2005; Bengio et al., 2006; Hinton et al., 2006; Ranzato et al, 2006), lo cual

ofrece buenos resultados en determinados escenarios como son el reconocimiento de dígitos escritos a mano o detección de peatones, especialmente cuando la cantidad de datos etiquetados es muy limitada (Sermanet, 2013).

La primera aplicación importante de este método de preentrenamiento fue en el reconocimiento de voz y fue posible gracias a la llegada de unidades de procesamiento gráfico, que son fáciles de programar (Raina et al, 2009) y permiten a los investigadores entrenar la red 10 o 20 veces más rápido.

En 2009, este método se utilizó para asignar ventanas de coeficientes de tiempo cortos extraídas de ondas sonoras a un conjunto de probabilidades para diferentes segmentos del habla que pueden representarse mediante cuadros en el medio de la ventana. Estos métodos alcanzaron resultados sin precedentes en una prueba estándar de reconocimiento de voz utilizando un vocabulario reducido (Mohamed et al, 2012) y se amplió rápidamente para lograr resultados inimaginables en una sola tarea.

Aun así, ha surgido una clase específica de red profunda multicapa, la cual tiene una mayor facilidad de entrenamiento y exhibe una mejor generalización que las redes que poseen conectividad total entre capas adyacentes. Esta innovación se denomina red neuronal convolucional (ConvNet o CNN). En una época en la que las redes neuronales aún no se habían adoptado ampliamente, las CNNs lograron un éxito considerable en la práctica. Recientemente, esta arquitectura ha sido adoptada extensamente por la comunidad de visión artificial.

3.2.1.2 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) se caracterizan por su estructura de retroalimentación. Estos algoritmos de aprendizaje se emplean principalmente en el análisis de datos de series temporales para realizar predicciones sobre eventos futuros, como pronósticos del mercado de valores o estimaciones de ventas. Las RNN procesan una secuencia de entrada y en sus capas ocultas mantienen un vector de estado que encapsula datos históricos de la totalidad de los elementos previos en la secuencia. Se puede conceptualizar las salidas de las unidades ocultas como si fueran las salidas de diversas neuronas en una red multicapa profunda (Figura 7).

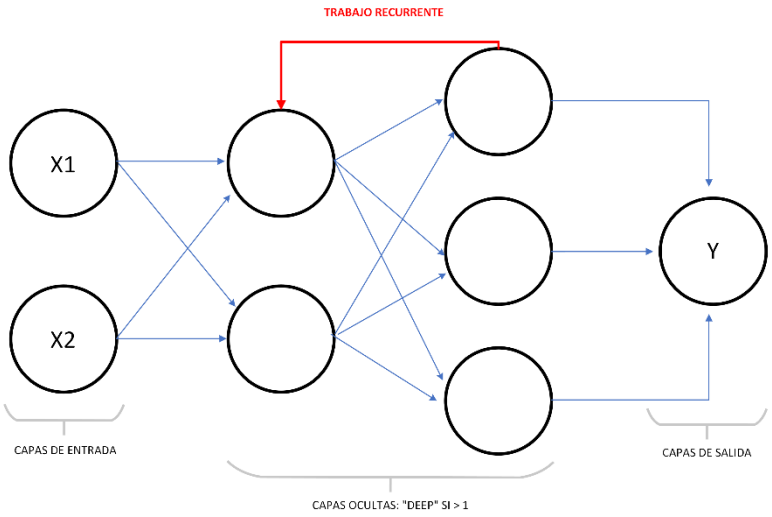


Fig 7 Red neuronal recurrente

Las RNN son sistemas dinámicos altamente poderosos, pero han presentado un desafío significativo debido a los problemas de gradiente que tienden a crecer o desvanecer en cada paso de tiempo, lo que puede llevar a la explosión o desvanecimiento de los gradientes (Hochreiter, 1991; Bengio et al., 1994). No obstante, gracias a los avances en su arquitectura (ElHihi and Bengio, 1995; Hochreiter and Schmidhuber, 1997) y en las técnicas de entrenamiento (Sutskever, 2012; Pascanu et al., 2013), se ha descubierto que las RNN son particularmente eficaces para predecir el siguiente carácter en un texto (Sutskever et al., 2011) o la siguiente palabra en una secuencia (Mikolov et al., 2013).

3.2.1.3 Redes neuronales convucionales

Las redes neuronales convolucionales constituyen un tipo de redes de propagación hacia adelante, como se mencionó previamente, pero su aplicación principal recae en el reconocimiento de imágenes, detección de patrones y visión artificial (Lecun et al., 1998; Nebauer, 1998). Estas redes operan bajo los principios del álgebra lineal. Las CNN procesan datos en matrices multidimensionales (por ejemplo, 1D: secuencias de texto; 2D: imágenes o audio; 3D: video), como una imagen en color formada por tres matrices que contienen las intensidades de los píxeles en los canales de color.

Mediante el uso de ventanas deslizantes, exploran estas matrices transformando los datos brutos en características a lo largo de múltiples capas de creciente abstracción, permitiendo detectar patrones dentro de una imagen.

Las CNNs se sustentan en cuatro conceptos clave que capitalizan las particularidades de las señales naturales: conexiones locales, pesos compartidos, agrupación y el empleo de múltiples capas. La arquitectura de una CNN típica (Fig. 8) se organiza en etapas secuenciales.

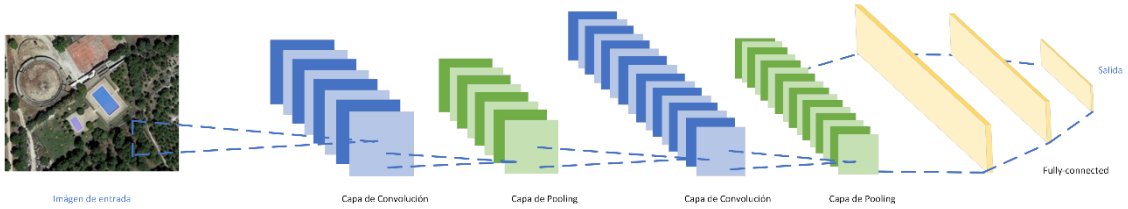


Fig 8 Red neuronal convucional

Como puede verse en la figura 8, las redes neuronales convolucionales están formadas por dos tipos de capas: las capas convolucionales y las capas de agrupación, que se explicarán en los siguientes subapartados.

3.2.1.3.1 Capa Convolucional

En una capa convolucional, las unidades se agrupan en mapas de características, donde cada unidad está asociada a regiones locales de los mapas de características de la capa previa mediante un conjunto de pesos conocido como banco de filtros (Ciresçan, 2011). El producto de la combinación local estos valores ponderados pasan a través de una función no lineal, como la ReLU.

En esta arquitectura, las unidades de un mismo mapa de características se distribuyen en un único banco de filtros, aunque los diversos mapas de características en una capa emplean bancos de filtros distintos. La razón detrás de esta estructura se debe a dos motivos fundamentales:

- En las imágenes o matrices de datos, los grupos locales de valores tienden a ser altamente correlacionados, lo que conduce a la detección de patrones locales fácilmente discernibles.
- En las imágenes las propiedades estadísticas locales y otras señales no varían según la ubicación. En otras palabras, si un patrón es detectable en una región de la imagen, podría aparecer en cualquier otra parte. Por lo tanto, la idea de compartir los mismos pesos entre unidades ubicadas en diferentes posiciones permite detectar patrones idénticos en varias partes de la matriz. Esta operación de filtrado es una convolución discreta en términos matemáticos, de ahí su nombre.

La convolución, en esencia, conserva las relaciones espaciales entre los píxeles mientras aprende las características propias de cada imagen mediante pequeñas ventanas de datos de entrada. Aunque se exploren los detalles matemáticos de la convolución, se explica de manera comprensiva cómo opera en las imágenes (Sánchez-Alor, 2020). Cada imagen puede representarse como una matriz de valores de píxeles. Si se considera una imagen de 5 x 5 y una matriz de 3 x 3, tal como se ilustra en la Figura 9.

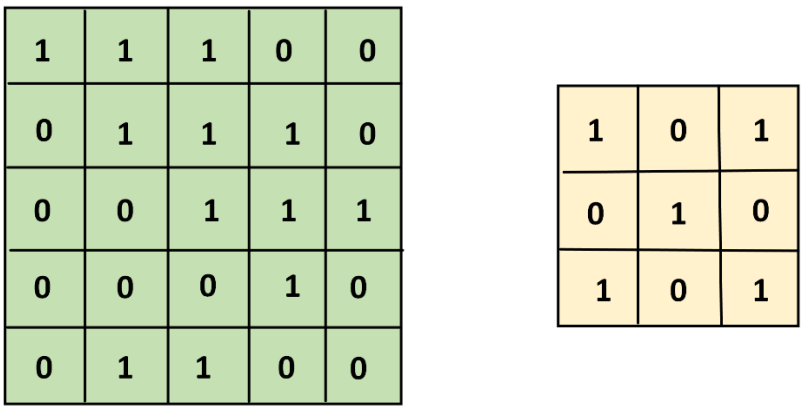


Fig 9 Imagen 5x5 y matriz 3x3

El proceso de cálculo de la convolución entre una imagen de 5 x 5 y una matriz de 3 x 3 puede llevarse a cabo tal como se exhibe en la Figura 10, el proceso sigue los pasos detallados a continuación. Mediante un desplazamiento gradual, la matriz amarilla se desliza sobre la imagen original (representada en verde) píxel a píxel. Se ejecuta una multiplicación elemento por elemento entre ambas matrices para cada posición, posteriormente se suman los productos obtenidos para obtener un único valor que

determina el contenido de un elemento específico en la matriz resultante (resaltada en azul). Es importante notar que la matriz 3 × 3 "observa" únicamente una porción de la imagen de entrada en cada iteración.

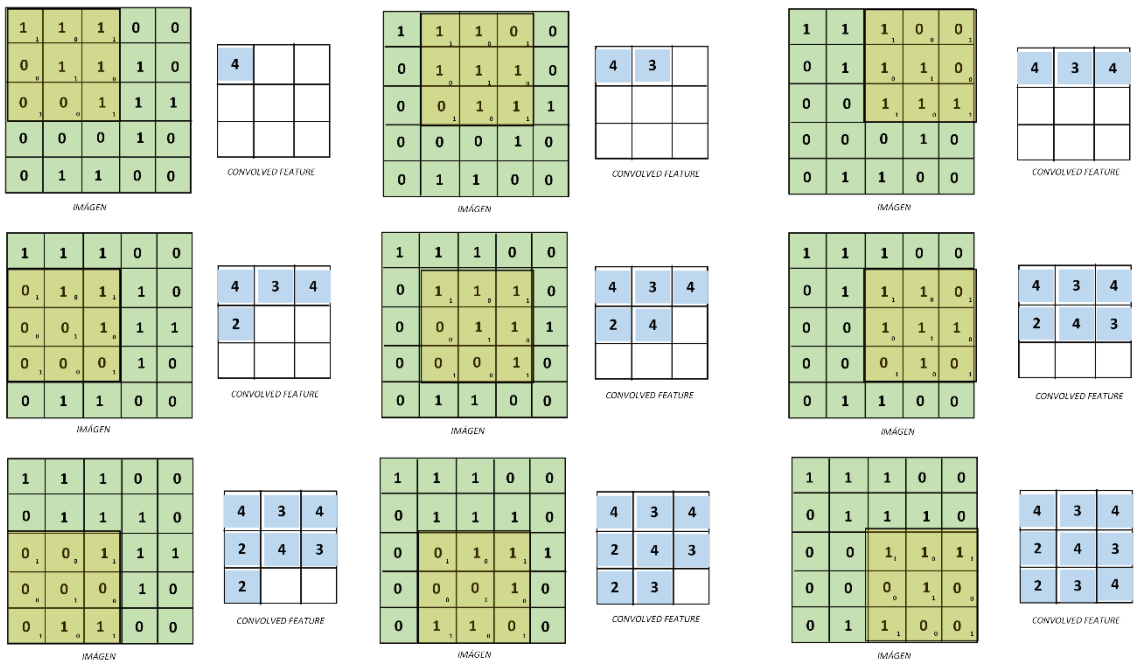


Fig 10 Ejemplo Convolución imagen 5 x 5 y la matriz 3 x 3.

En la terminología de las CNN, la matriz 3 × 3 es conocida como 'detector de características', 'filtro' o 'kernel' y la matriz resultante después de deslizarla sobre la imagen y realizar el producto escalar es denominada Mapa de Características o matriz convolucionada (Ciresçan, 2011).

Al observar la representación de la Figura 10 se hace evidente que distintos tamaños en la matriz del kernel darán origen a diversos mapas de características para una misma imagen de entrada. En el contexto del aprendizaje profundo, una CNN aprende de manera autónoma durante el proceso de entrenamiento (aunque es necesario definir parámetros como el número de filtros, el tamaño del filtro, la estructura de la red, entre otros, previamente al proceso de entrenamiento). A medida que incrementamos la cantidad de filtros, logramos extraer más características de la imagen, mejorando así la capacidad de la red para reconocer patrones en las imágenes.

Para controlar el tamaño de la matriz convolucionada que constituye el mapa de características es fundamental determinar estos tres parámetros antes de ejecutar la operación de convolución:

- Profundidad (Depth): Esta se relaciona con la cantidad de filtros utilizados en la operación de convolución.
- Zero Padding: Ocasionalmente, es beneficioso rellenar la matriz de entrada con ceros alrededor de su borde, permitiendo aplicar el filtro a los elementos adyacentes de la matriz de imagen. Un aspecto favorable del zero-padding es que otorga control sobre las dimensiones de los mapas de características. Agregar relleno también se conoce como "wide convolution" (convolución ancha), mientras que no emplear relleno es denominado "narrow convolution" (convolución estrecha).
- Stride: La "zancada" define cuántos píxeles se desliza el filtro por la matriz de entrada. Si se establece en 1, los filtros avanzan un píxel a la vez. Aumentar este valor implica un mayor espaciado entre cada aplicación del filtro, resultando en mapas de características más pequeños.

3.2.1.3.2 No linealidad (ReLU)

Finalizada la operación de convolución es necesario aplicar una operación adicional llamada Rectified Linear Unit (ReLU) (Agarap, 2018), cuya expresión matemática se puede ver en la ecuación 1, siendo una operación no lineal. Su salida se puede observar en la Figura 11.

$$R(z) = max\{0, z\}$$

Ecuación 1 ReLu

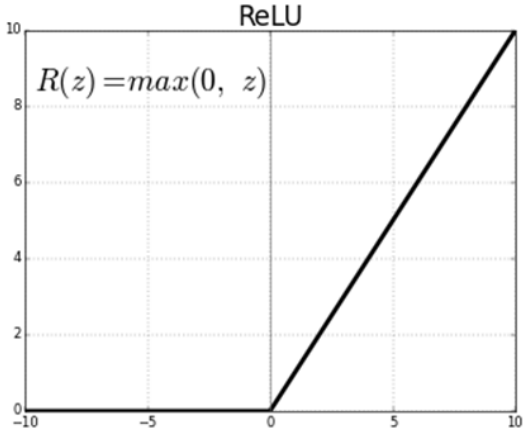


Fig 11 Operación no lineal ReLU

La ReLU es una operación a nivel de elemento la cual se aplica individualmente a cada píxel en un mapa de características y tiene como objetivo reemplazar por cero todos los valores de píxeles que sean negativos en el mapa. La función principal de ReLU es incorporar la propiedad de no linealidad en nuestras redes neuronales convolucionales, esto es necesario ya que la mayoría de los datos tienden a ser de naturaleza no lineal debido a que pertenecen al mundo real. Dado que la convolución en sí misma es una operación lineal que involucra multiplicación y suma de elementos de matrices, incorporamos ReLU para introducir la componente no lineal en el proceso (esto es necesario porque la convolución en su estado puro es una operación lineal). Si bien que es cierto que existen otras funciones no lineales disponibles, como sigmoides o tangentes hiperbólicas, en la mayoría de los casos ReLU tiende a funcionar de una manera más eficaz. Esto se debe a su simplicidad y a su habilidad para solventar problemas como el desvanecimiento de gradientes, lo que lo convierte en una elección preferida en muchas situaciones.

3.2.1.3.3 Capa de agrupación o Pooling

A pesar de que la función principal de la capa convolucional es detectar combinaciones locales de rasgos de la capa anterior, la capa de agrupación cumple con la tarea de combinar características semánticamente similares en una sola entidad. Dado que las posiciones relativas de los rasgos que forman un patrón pueden variar en cierta medida, aplicar una agrupación con una granularidad más amplia a la posición de cada rasgo puede ser más efectivo para identificar el patrón de manera confiable.

La operación de agrupación espacial (spatial pooling) tiene como objetivo reducir la dimensionalidad de cada mapa de características, al mismo tiempo que preserva la información crucial. Hay varios métodos de agrupación espacial disponibles, como Max Pooling, Average Pooling, Sum Pooling, entre otros (Scherer et al. 2010).

En el caso particular del Max Pooling (Nagi et al., 2011), se establece una ventana de vecindad espacial, por ejemplo, una ventana de tamaño 2 × 2 sirve para seleccionar el valor mayor del mapa de características rectificado incluido dentro del tamaño de esa ventana. En vez de elegir el valor más grande, también podría calcularse el promedio o la suma de todos los elementos incluidos en esa ventana. No obstante, al llevar a cabo de forma práctica diferentes procedimientos, se ha observado que el Max Pooling tiende a brindar mejores resultados.

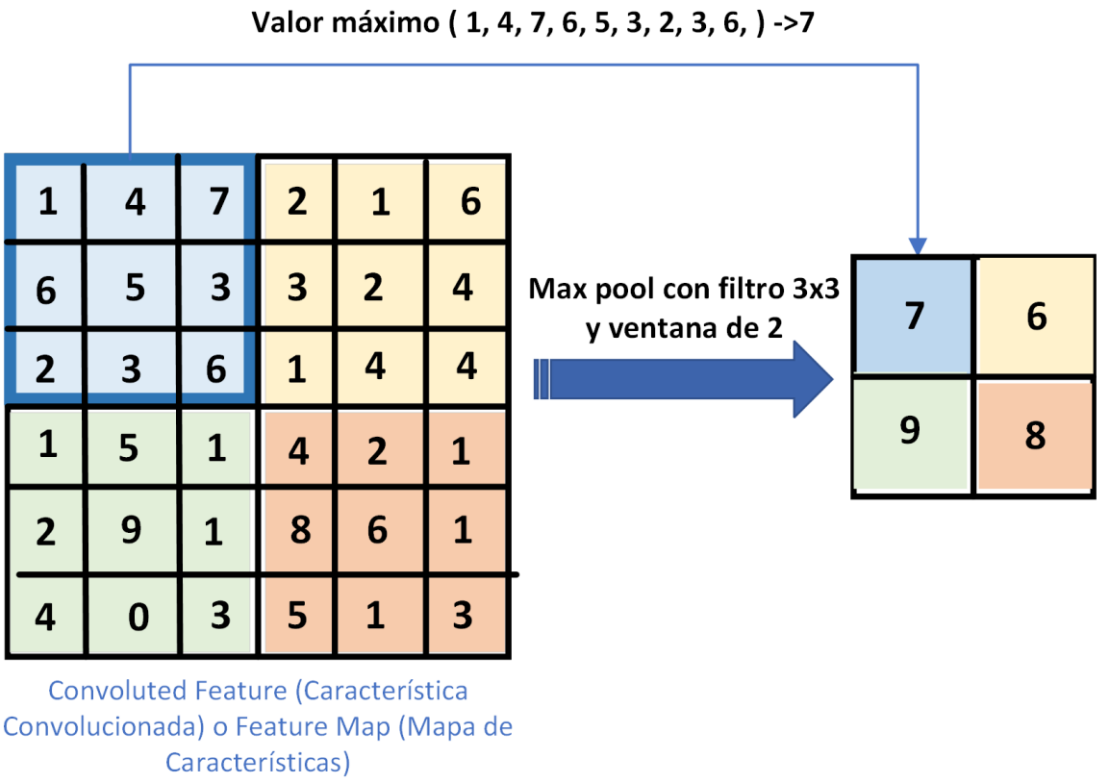


Fig 12 Ejemplo Max Pooling filtro 3x3 y ventana de 2x2

La Figura 12 ofrece de forma visual un ejemplo de la operación de Max Pooling en un mapa de características rectificadas (obtenido tras la convolución y posterior ReLU) utilizando una ventana de tamaño 3 × 3.

Las capas de agrupación se dedican a calcular el valor máximo en una zona local de unidades dentro de un mapa de características (o en múltiples mapas de características). Es importante notar que las capas de agrupación colindantes obtienen información de parches desplazados en más de una columna o una fila, lo que conlleva a una reducción en la dimensión, generando una invariancia ante desplazamientos y deformaciones pequeños.

Una típica red convolucional está compuesta por dos o tres etapas consecutivas de convolución, no linealidad y agrupación. En conjunto, las características esenciales de las imágenes son extraídas por estas capas, las cuales también incorporan la no linealidad necesaria en la red y disminuyen la dimensión de las características extraídas.

La entrada a la capa completamente conectada (fully connected layer), tema que se abordará en la siguiente sección, es la salida obtenida tras la segunda capa de agrupación.

3.2.1.3.4 Capa totalmente conectada

El término de capa completamente conectada se refiere a un Perceptrón multicapa convencional, que utiliza la función de activación Softmax en su capa de salida (Ciresçan, 2011). Esta función(σ) transforma un vector (z) de K dimensiones compuesto por valores reales en otro vector de K dimensiones, donde cada componente está en el rango de [0,1], y la suma total de todas las componentes es igual a 1. El término "completamente conectado" indica que cada neurona en la capa anterior establece conexiones con cada neurona en la capa subsiguiente.

$$\sigma: \mathbb{R}^K \longrightarrow [0,1]^K$$
$$\sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, para\ j = 1, \dots, K$$

Ecuación 2 Función de activación Softmax

El objetivo de la capa completamente conectada radica en aprovechar las características de alto nivel adquiridas a través de las etapas de convolución y agrupación para llevar a cabo la tarea de clasificación de imágenes en múltiples categorías, conforme a los datos de entrenamiento disponibles. Por ejemplo, en un escenario de clasificación de imágenes, se contemplan cuatro resultados posibles, como se detalla en el lado derecho de la Figura 8 (las conexiones específicas entre los nodos en la capa completamente conectada no se presentan aquí).

Esta capa también ofrece la posibilidad de aprender combinaciones no lineales de las características extraídas, de manera eficaz. Aunque las características individuales provenientes de las capas de convolución y agrupación resultan beneficiosas para la clasificación, las combinaciones entre estas características podrían generar un impacto aún mayor. Para asegurar que las probabilidades de salida sumen 1, se implementa la función Softmax en la capa de salida de esta capa completamente conectada.

El proceso de propagación de gradientes en una CNN es igual de fluido que en una red profunda tradicional, lo que simplifica el entrenamiento de todos los pesos asociados a los bancos de filtros. Las redes neuronales profundas aprovechan la jerarquía inherente en muchas señales naturales, donde las características de nivel superior emergen de las de nivel inferior. En el contexto de imágenes, la combinación local de bordes conforma

patrones, estos patrones se amalgaman en partes y las partes, a su vez, constituyen objetos. De manera análoga, en el ámbito del habla y el texto, se observan jerarquías desde los sonidos hasta las palabras y las frases. La agrupación garantiza la robustez de las representaciones ante variaciones en la posición y el aspecto de los elementos presentes en la capa anterior.

Las capas convolucionales y de agrupación en CNN se inspiran directamente en conceptos de células simples de la neurociencia visual (Hubel and Wiesel, 1962), mientras que su estructura global guarda paralelismos con la organización de las vías visuales LGN-V1-V2-V4-IT en la corteza visual (Felleman and Essen, 1991).

Los avances significativos en visión computarizada y la detección de objetos han sido impulsados en gran medida por las redes neuronales convolucionales (He, Zhang, Ren, and Sun, 2016; Krizhevsky, Sutskever, and Hinton, 2012; Long, Shelhamer, and Darrell, 2015).

3.2.2 Aprendizaje de las redes neuronales

El método de entrenamiento más frecuentemente empleado en las redes neuronales es el aprendizaje supervisado. Si se desea desarrollar un sistema que clasifique imágenes que contengan elementos como casas, coches, personas o mascotas. En este proceso, se recolecta un extenso conjunto de datos que consisten en imágenes de casas, coches, personas y mascotas, cada una etiquetada con su categoría correspondiente. Durante el período de entrenamiento, se presenta una imagen a la máquina y esta produce un resultado en forma de un vector de puntuaciones, uno por cada categoría. El objetivo es que la categoría correcta obtenga la puntuación más alta entre todas las categorías.

Esto se logra, calculando una función objetivo que cuantifica el error (o distancia) entre el patrón de puntuaciones deseado y las puntuaciones de salida. A continuación, la máquina ajusta sus parámetros internos adaptables con el fin de reducir este error. Estos parámetros adaptables, a menudo llamados pesos, son valores numéricos que actúan como "controles" que definen la relación de entrada y salida de la máquina. En un contexto típico de aprendizaje profundo, podría haber cientos de millones de estos pesos adaptables y una cantidad similar de ejemplos etiquetados para entrenar la máquina.

Para ajustar con precisión el conjunto de pesos, el algoritmo de aprendizaje calcula un vector de gradiente que, para cada peso, señala cómo cambiaría el error si el peso

aumentara ligeramente. A continuación, se modifica el conjunto de pesos en la dirección opuesta al vector de gradiente.

La función objetivo, evaluada en todos los ejemplos de entrenamiento, puede concebirse como una suerte de paisaje en un espacio de alta dimensión compuesto por los valores de peso. El vector de gradiente en sentido contrario indica la dirección más pronunciada de descenso en este paisaje, moviéndose hacia un mínimo en el que el error de salida tiende a ser bajo en promedio.

Para optimizar las funciones objetivo se emplea un enfoque conocido como descenso de gradiente estocástico (SGD), el cual implica presentar el conjunto de entrada de varios ejemplos, calcular los resultados y errores correspondientes, determinar el gradiente promedio de dichos ejemplos y ajustar los pesos en función de ello. Esta secuencia se repite para múltiples grupos pequeños de ejemplos hasta que la media de la función objetivo ya no disminuya.

La denominación estocástica se debe a que cada conjunto pequeño de ejemplos brinda una estimación con ruido del gradiente medio de todos los ejemplos, este algoritmo evalúa la función objetivo en todas las observaciones muestrales de forma aleatoria antes de terminar cada ciclo de corrección de propagación hacia atrás y hacia adelante o epoch.

Este proceso relativamente simple suele derivar en la obtención de un conjunto de pesos eficaz en comparación con técnicas de optimización mucho más elaboradas (Bottou, 2007). Luego de completar el entrenamiento, se evalúa el rendimiento del sistema en un conjunto de ejemplos diferente denominado conjunto de prueba. Esto permite examinar la habilidad de generalización de la máquina, es decir, su capacidad de proporcionar respuestas coherentes ante nuevos datos que no fueron presentados durante el entrenamiento.

En muchas aplicaciones actuales del aprendizaje automático, se utilizan clasificadores lineales basados en características diseñadas manualmente. Un clasificador lineal de dos clases realiza un cálculo de suma ponderada de los componentes del vector de características. Si esta suma ponderada supera cierto umbral, la entrada se clasifica en una categoría específica.

Desde la década de 1960, se sabe que los clasificadores lineales solo pueden dividir el espacio de entrada en regiones muy simples, es decir, regiones que son semiespacios separados por un hiperplano. No obstante, problemas como el reconocimiento de imágenes y de habla exigen que la relación entrada-salida sea insensible a variaciones irrelevantes en la entrada, como cambios en la posición, orientación o iluminación de un

objeto, o variaciones en el tono o acento del habla. Además, debe ser altamente sensible a cambios sutiles específicos.

Un clasificador lineal, o cualquier otro tipo de clasificador "superficial" necesitan un extractor de características competente que resuelva el desafío de selectividad-invarianza. Este extractor de características debe crear representaciones selectivas de los elementos clave de la imagen, pero que sean insensibles a aspectos irrelevantes como la orientación del objeto.

Para potenciar los clasificadores, se pueden emplear características no lineales genéricas, como en los enfoques basados en métodos kernel (Schölkopf and Smola, 2003). Sin embargo, características genéricas, como las generadas por el kernel gaussiano, no permiten una generalización sólida fuera del conjunto de ejemplos de entrenamiento (Bengio et al, 2005). La solución convencional es diseñar a mano extractores de características efectivos, una tarea que exige una amplia comprensión de la ingeniería y experiencia en el campo. Sin embargo, esta complejidad puede obviarse si se pueden aprender automáticamente buenas características mediante un proceso de aprendizaje generalizado. Esta es la principal ventaja que ofrece el aprendizaje profundo.

Una arquitectura de aprendizaje profundo se compone de una pila de módulos simples de múltiples capas, la mayoría de los cuales están sometidos al proceso de aprendizaje. Muchos de estos módulos realizan mapeos no lineales de entrada a salida. Cada uno de estos módulos transforma su entrada para aumentar tanto la selectividad como la invarianza en la representación. Con varias capas no lineales, por ejemplo, con una profundidad de 5 a 20 capas, un sistema puede expresar funciones altamente complejas de sus entradas que son simultáneamente sensibles a minuciosos detalles e insensibles a variaciones significativas e irrelevantes, como el fondo, la pose, la iluminación y los objetos circundantes.

A lo largo del tiempo, las técnicas de aprendizaje automático han desempeñado un papel fundamental y, en muchos casos, centran la evolución de algoritmos de visión artificial. La disciplina de la visión por computadora, en la década de 1970, emergió de campos como la inteligencia artificial, el procesamiento digital de imágenes y el reconocimiento de patrones (hoy conocido como aprendizaje automático). El procesamiento de imágenes, la interpolación de datos dispersos, la minimización de energía variacional y las estrategias de modelos gráficos han sido herramientas fundamentales en la visión artificial a lo largo de las últimas cinco décadas (Szeliski, 2022).

En la actualidad y a modo de resumen, las redes neuronales profundas son los modelos de aprendizaje automático más populares y ampliamente utilizados en visión por

computadora. Estas redes no solo se aplican a tareas de clasificación y segmentación semántica, sino que también se emplean en labores de nivel inferior, como la mejora de imágenes, la estimación de movimientos y la recuperación de profundidad (Bengio, LeCun y Hinton, 2021)

CAPÍTULO IV

ALGORITMOS DE APRENDIZAJE USADOS EN LA DETECCIÓN DE OBJETOS EN IMÁGENES

VNiVERSiDAD
D SALAMANCA

En este capítulo se realiza una revisión de la evolución de los diferentes algoritmos de aprendizaje usados en las redes convolucionales para la detección de objetos en imágenes. Con el fin de comprender su funcionamiento y conocer las ventajas que conllevan el uso de cada uno. Así como entender los conceptos clave necesarios para evaluar su rendimiento. Siendo estos conocimientos fundamentales para abordar la detección de recursos hídricos en imágenes.

4 Algoritmos de aprendizaje utilizados para la detección de objetos en imágenes.

4.1 Detección de Objetos en imágenes

Antes del auge del Aprendizaje Profundo, la detección de objetos en imágenes seguía un proceso tedioso que constaba de varios pasos. Inicialmente, se identificaban los bordes y se extraían características utilizando métodos como SIFT (Scale Invariant Feature Transform) (Piccinini et al., 2012) y HOG (Histograma de Gradientes Orientados) (Mizuno et al., 2012). Estas características se comparaban con modelos de objetos predefinidos en diferentes escalas, con el fin de detectar y localizar los objetos en la imagen. Actualmente, sin embargo, la detección de objetos se ha transformado debido a la proliferación de las CNN que implementan varios modelos y algoritmos para llevar a cabo esta tarea.

Cada capa de CNN se denomina mapa de características. En la capa original, el mapa de características es una matriz tridimensional que representa las intensidades de los píxeles en diferentes canales de color (por ejemplo, RGB). A medida que se avanza en las capas, cada mapa de características es una imagen multicanal generada, donde cada píxel puede interpretarse como un objeto específico. Cada neurona está conectada a una pequeña porción de neuronas vecinas en la capa anterior, llamada campo receptivo.

En estos mapas de características, se aplican diversas transformaciones (Krizhevsky et al., 2012; Oquab et al., 2014), como filtrado y agrupación. La operación de filtrado (convolución) se produce cuando la matriz de filtro (con pesos ajustados) se desliza sobre los valores de un campo receptivo de neuronas, a este proceso le sigue una función no lineal (por ejemplo, la sigmoidea (Wadley, 1952) o ReLU), para obtener las respuestas finales. La operación de agrupación sirve para transformar las respuestas de un campo receptivo en un único valor, generando descripciones de características más sólidas, dentro de estas operaciones encontramos la agrupación máxima, promedio, L2 o normalización de contraste local (Kavukcuoglu et al., 2009).

A través de la combinación de convolución y agrupación, se construye una jerarquía inicial de características que se puede ajustar de manera supervisada agregando varias capas completamente conectadas (FC) en función de las tareas visuales específicas. Particularmente, se añade una capa final dependiendo de la tarea, con diferentes funciones de activación (Krizhevsky et al., 2012), para obtener una probabilidad condicional específica para cada neurona de salida. Las redes CNN pueden optimizar su capacidad de predicción mediante funciones objetivo (como el error cuadrático medio o la pérdida de entropía cruzada) y el uso de SGD. Las ventajas inherentes de las CNN sobre los métodos convencionales se pueden resumir de la siguiente manera:

- Profundidad de expresión: a diferencia de los modelos de superficie tradicionales, la arquitectura más profunda en las CNN proporciona una expresividad significativamente ampliada, lo que mejora la representación y el aprendizaje (Girshick et al., 2014; Kavukcuoglu et al., 2010).
- Uso de tareas relacionadas: La estructura CNN permite la optimización general de varias tareas relacionadas. Por ejemplo, en el caso de R-CNN rápido, la clasificación y la regresión del cuadro delimitador se combinan mediante el método de aprendizaje multitarea (Girshick et al., 2014; Kavukcuoglu et al., 2010).
- Transformaciones de datos de alta dimensión: las CNN profundas pueden resolver los desafíos clásicos de la visión por computadora reorganizándolos en problemas de transformación de datos de alta dimensión. Esto proporciona una nueva perspectiva para la resolución de problemas desde un enfoque diferente (Girshick et al., 2014; Kavukcuoglu et al., 2010).

Gracias a estas ventajas, las CNN se ha convertido en una herramienta imprescindible en muchos campos de la investigación. Estas áreas van desde la reconstrucción de imágenes y la súper resolución (Zeiler et al., 2010; Noh et al., 2015), hasta la clasificación de imágenes (Zhao et al., 2014; Jia et al., 2014), recuperación de imágenes (Babenko et al., 2014; Wan et al., 2014), reconocimiento facial (Yang y Nevatia, 2016), detección de peatones (Tomè et al., 2016; Xiang et al., 2017; Zhao et al., 2017) y vídeo análisis (Ngiam et al., 2011; Wu et al., 2015).

A continuación, se profundizará en la clasificación de algoritmos, concentrándonos en tres específicos utilizados en este estudio: Mask R-CNN, Detectron2 y Yolov4. No obstante, antes de explorar estos detectores, resulta fundamental establecer algunos

conceptos clave necesarios para evaluar su rendimiento, como el IoU y el Promedio de Precisión en Múltiples Escalas (mAP).

4.1.1 Índice de Intersección sobre Unión (IoU)

La métrica o Índice Jaccard, también conocida como IoU, es una métrica de evaluación que se utiliza para medir la precisión del detector de objetos en un conjunto de datos particular. Por lo tanto, cualquier algoritmo que genere cajas o cuadros delimitadores se puede evaluar usando IoU.

$$\mathcal{J}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Ecuación 3 IoU

Dónde A es el cuadro delimitador de referencia y B son los cuadros delimitadores previstos por el modelo, por lo que para evaluar un algoritmo detector de objetos mediante la métrica IoU se requiere:

- Cuadro delimitador de referencia (Ground-Truth): estos cuadros delimitadores representan áreas previamente etiquetadas en el conjunto de prueba. Estas áreas indican la posición exacta de los objetos en la imagen.
- Cuadros delimitadores previstos por el modelo: estos cuadros delimitadores son generados por el modelo que se está evaluando

La Figura 13 muestra una representación visual que ilustra la comparación entre el cuadro delimitador de referencia y el cuadro predicho por el modelo. También se muestra el área de intersección y asociación entre estos cuadros, y la calificación de calidad de la métrica de IoU se obtiene mediante calificación: pobre, buena y excelente.

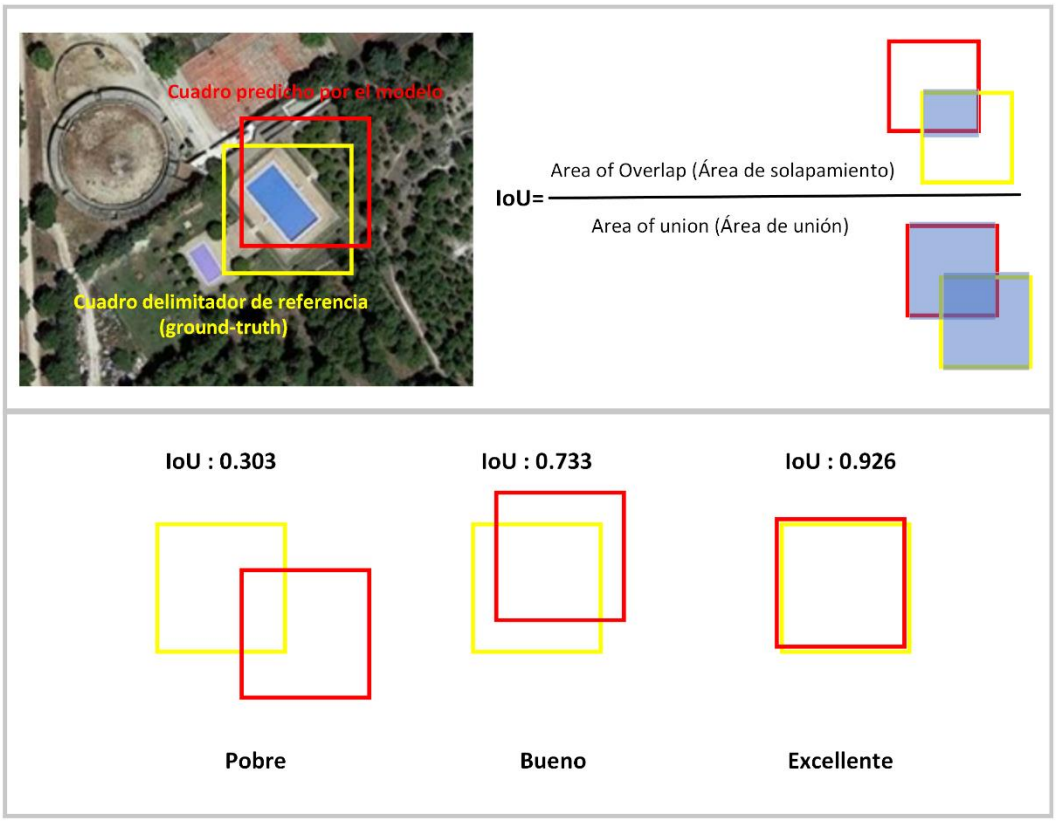


Fig 13 Representación visual IoU

Observando la ecuación 1, queda claro que el IoU se fundamenta esencialmente en una relación entre las áreas de los cuadros o cajas usados en la evaluación. En el numerador, se calcula el área de superposición entre el cuadro delimitador de predicción y el cuadro delimitador de referencia. El denominador es la región delimitada, que incluye tanto el cuadro delimitador previsto como el cuadro delimitador de referencia. Al dividir la superposición por la unión, se obtiene el punto final: el índice de intersección en la unión.

Cualitativamente, los cuadros delimitadores que se predijo que tendrían una superposición significativa con los cuadros delimitadores de referencia obtuvieron puntuaciones más altas que aquellos con menos superposición. Esto convierte a IoU en una métrica muy eficaz para evaluar detectores de objetos personalizados.

En la práctica, es muy difícil que las coordenadas (x, y) del cuadro delimitador previsto coincidan con las coordenadas (x, y) del cuadro delimitador de referencia de forma idéntica. Sin embargo, el objetivo no es lograr coincidencias exactas de coordenadas, sino garantizar que nuestros cuadros delimitadores previstos coincidan con la mayor precisión posible. Las cifras de IoU reflejan fielmente esta opinión.

4.1.2 Promedio de Precisión en Múltiples Escalas (mAP)

Para evaluar el desempeño de un modelo de clasificación, se utilizan medidas comunes como la precisión y la sensibilidad (también llamada recall). Para comprender el concepto del mAP, es esencial comenzar analizando la precisión y la sensibilidad. La precisión de una clase particular en la clasificación, también conocida como valor predictivo positivo, se calcula como la relación entre los verdaderos positivos (TP) y el número total de predicciones incluidos los TP y los falsos positivos (FP).

$$Precision = \frac{TP}{TP + FP}$$

Ecuación 4 Precisión

De manera similar, la sensibilidad, a menudo denominada tasa de verdaderos positivos (TPR), para una clase particular en la clasificación, se define como la relación entre el número de TP y positivos totales según la verdad fundamental (ground truth), siendo este valor es igual a la suma de TP y falsos negativos (FN).

$$Recall = \frac{TP}{TP + FN}$$

Ecuación 5 Recall

Si consideramos las ecuaciones, está claro que en cualquier modelo de clasificación existe un equilibrio intrínseco entre precisión y sensibilidad. En el contexto de las redes neuronales, esta compensación se puede ajustar utilizando un umbral softmax en la capa final del modelo.

Para lograr una alta precisión, es necesario reducir el número de FP, lo que puede provocar una disminución de la sensibilidad. De manera similar, al reducir los falsos negativos FN, se aumenta la sensibilidad, pero se reduce la precisión. En muchas situaciones de detección de objetos y recuperación de información, lograr una alta precisión es un objetivo importante.

Por lo general, la precisión y la sensibilidad se combinan con otras medidas, como la exactitud (accuracy), el Valor-F (F1-score), la especificidad (Tasa de verdaderos

negativos, TNR), la curva ROC (Características operativas del receptor), la curva de ganancia y la curva de elevación. Sin embargo, estas métricas no siempre brindan una perspectiva completa al evaluar la eficacia de un modelo en tareas de recuperación de información o detección de objetos. De ahí el concepto de mAP. Es importante tener en cuenta que los cálculos de mAP pueden variar entre las tareas de detección de objetos y recuperación de información. En este contexto, la presente tesis se centrará en la detección de objetos, que es nuestro principal tema de análisis.

Para calcular mAP, primero se deben establecer umbrales los umbrales de IoU para determinar si el cuadro delimitador de predicción (BB) se clasifica como TP, FP o FN. A continuación, se presenta la relación del valor de IoU y su designación:

- TP: IoU mayor a 0.5.
- FP: que abarca dos escenarios: IoU menor a 0.5 o múltiples BB para un mismo objeto.
- FN: sin BB o IoU mayor a 0.5 pero con la selección de clase incorrecta.

Una vez que TP, FP y FN se definen formalmente, la precisión y sensibilidad de la detección se pueden calcular para una clase particular en todo el conjunto de pruebas. A cada BB se le asigna un nivel de confianza, generalmente derivado de su clase softmax, que se utiliza para clasificar los hallazgos de mayor a menor confianza. Esta confianza se utiliza para construir un gráfico de precisión y sensibilidad similar en forma al de la Figura 14 (Hui, 2021). Por lo tanto, la definición general de precisión media (AP) se basa en el cálculo del área bajo la curva de recuperación de precisión (PR), la expresión matemática usada en el cálculo de AP se muestra en la ecuación 6.

$$AP = \int_0^1 p(r)dr$$

Ecuación 6 Precisión promedio

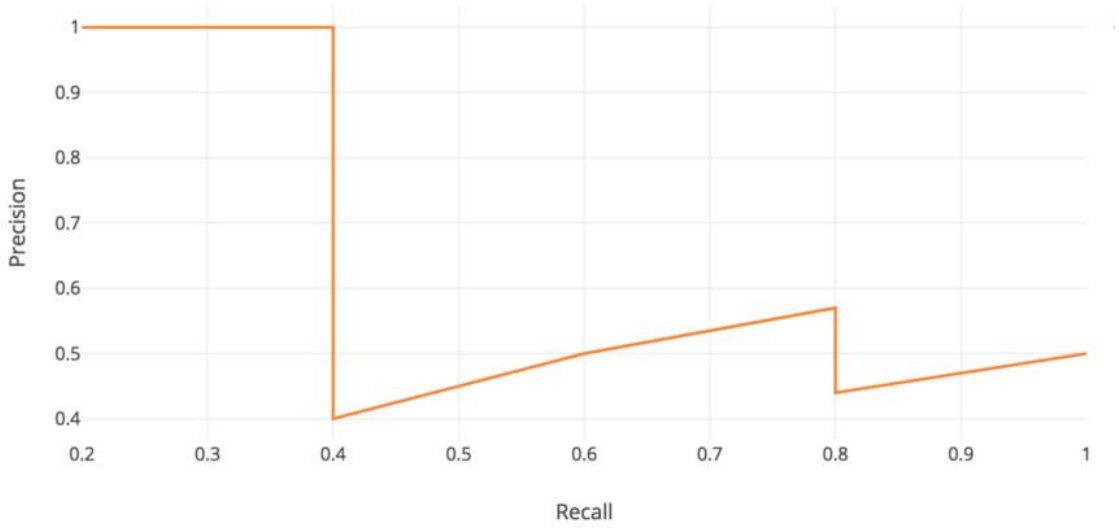


Fig 14 Ejemplo gráfica de la Precisión frente a la Sensibilidad. Fuente: Hui, 2021.

Tanto la precisión como la sensibilidad siempre oscilan entre 0 y 1. En consecuencia, el AP también se encuentra en el rango de 0 a 1. Es común aplicar un suavizado al patrón zigzag, antes de calcular el AP para la detección de objetos, el suavizado se realiza de forma simple, se asigna el valor máximo de precisión al intervalo, con el fin de mantenerlo constante. Consideración todos los elementos anteriormente expuestos, ahora estamos en posición de calcular el mAP para la detección de objetos, simplemente promediando los AP calculados para todas las clases.

4.2 Clasificación algoritmos detección de objetos en imágenes

Existen principalmente dos tipos básicos de métodos genéricos de detección de objetos (ver Figura 15), cada uno con un enfoque característico.

Por un lado, están los métodos de generación de propuestas de regiones, conocidos como Two-stage, que divide el proceso de detección de objetos en dos etapas diferentes. En estos métodos en el primer paso, se generan recomendaciones para áreas que pueden contener objetos en la imagen. Estas proposiciones se consideran objetos posibles y se obtienen mediante algoritmos específicos, como el método de la ventana deslizante. En el segundo paso, cada recomendación de área se procesa y clasifica en diferentes tipos de objetos mediante un clasificador. Este enfoque tiene como objetivo lograr una alta precisión de detección, ya que todas las propuestas se examinan minuciosamente, lo que permite una evaluación en profundidad de cada área de los objetos.

Por otro lado, existen métodos de regresión o clasificación, también conocidos como Single Stage. En estos métodos, la detección de objetos se aborda directamente como un problema de regresión o clasificación. En lugar de generar recomendaciones regionales en una etapa temprana, en este método la predicción tanto de las categorías de objetos como de las ubicaciones se realiza en un único proceso mediante una red neuronal. Esta red realiza una regresión para ajustar cuadros delimitadores alrededor de los objetos detectados y efectuar una clasificación para asignar etiquetas a cada objeto. Aunque la precisión puede verse ligeramente reducida en comparación con los métodos de generación de recomendaciones zonales, el método de regresión o clasificador es más rápido y eficiente, lo que lo hace particularmente adecuado para aplicaciones en tiempo real.

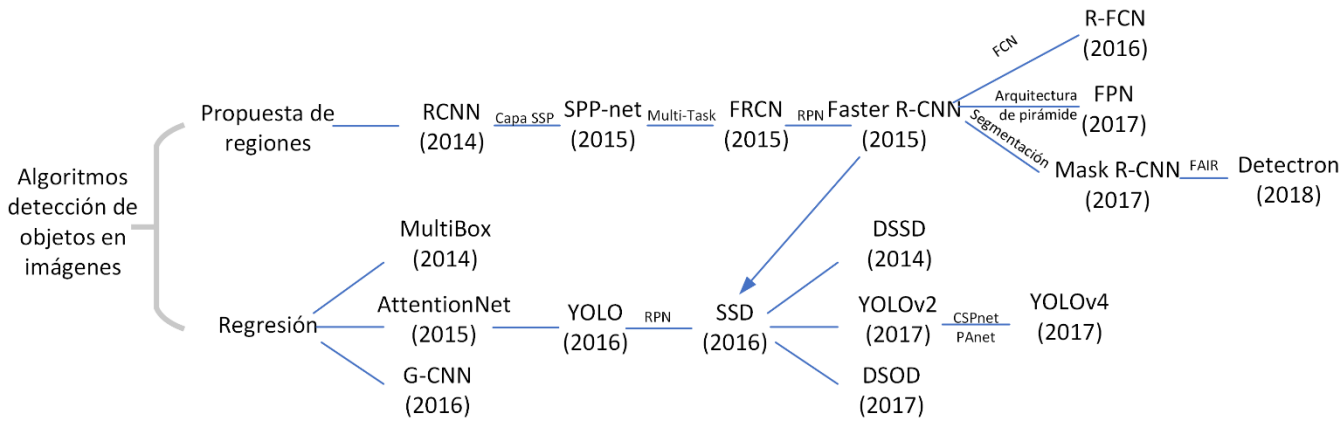


Fig 15 Clasificación algoritmos detección de objetos en imágenes

Existen distintos enfoques en la detección de objetos, cada uno con sus propias características. Los métodos basados en propuestas de regiones abarcan R-CNN (Girshick, 2016), Fast R-CNN (Girshick, 2015), Faster R-CNN (Rampersad, 2020), MASK-CNN (Doll, 2010), Detectron2 (Girshick, 2018) y otros. En contraste, los métodos basados en regresión/clasificación engloban MultiBox (Erhan et al., 2014), AttentionNet (Yoo et al., 2015), G-CNN (Najibi et al, 2016), YOLO (Redmon et al., 2016), Single Shot MultiBox Detector (SSD) (Liu et al, 2016), single shot deconvolucional (DSSD) (Fu et al.,2017), y detectores de objetos profundamente supervisados (DSOD) (Shen et al., 2017). Cabe señalar que los anclajes introducidos en Faster R-CNN conectan de alguna manera estas dos clasificaciones, estableciendo ciertas correlaciones entre ellas.

4.3 Arquitectura de propuesta de regiones

El uso de ventanas deslizantes junto con una CNN fue uno de los primeros métodos empleados con el fin de detectar objetos en imágenes, sirviendo para sentar las bases para algoritmos posteriores con una estructura de dos pasos.

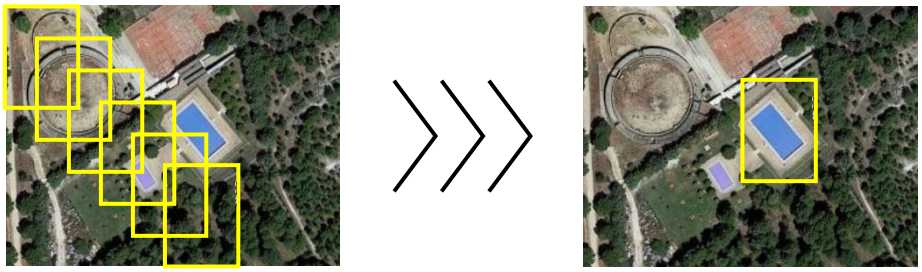


Fig 16 Ventanas deslizantes

El funcionamiento de las ventanas deslizantes es relativamente simple, se puede ver sintetizado en la figura 16, en la cual se muestra una ventana rectangular que se desplaza por toda la imagen, explorando de esta manera varias posiciones y escalas. La ventana se puede ajustar a diferentes escalas, lo que permite procesar objetos de diferentes tamaños mediante la extracción de características a través de una red convolucional. Estas características posteriormente se utilizan como entrada en un clasificador, por ejemplo, una máquina de vectores de soporte (SVM), con el fin de asignar la clase del objeto e informar a un regresor lineal delimitado por caja del objeto.

A pesar de sus ventajas, como la facilidad de implementación y la capacidad de detectar objetos en variedad de escalas y posiciones, operar únicamente con ventanas deslizantes tiene limitaciones significativas. En particular, requiere una enorme carga computacional, ya que la clasificación debe realizarse para cada posición y escala de ventana, lo que requiere una gran sobrecarga computacional. La elección de la forma y el tamaño de la ventana es un hándicap, especialmente cuando los objetos de interés son significativamente diferentes en tamaño y forma.

Fig 17 Búsqueda selectiva

Para abordar las limitaciones inherentes al empleo de las ventanas deslizantes, han surgido métodos de regiones de interés (ROI) para resolver el problema de reducir el número de regiones a clasificar. Estos métodos se centran exclusivamente en las regiones con mayor probabilidad de incluir un objeto de interés, entre estos enfoques se incluye la búsqueda selectiva, representada en la figura 17.

La búsqueda selectiva trata a cada píxel de una imagen como un grupo individual. Posteriormente se evalúan las texturas de estos grupos y se combinan aquellos que poseen similitudes notables. Para evitar el dominio de los grupos grandes sobre los grupos más pequeños, es preferible fusionar los grupos pequeños en grupos grandes. Repitiendo este proceso de fusión hasta que se alcanza el criterio de convergencia predefinido. La Figura 17 ilustra la formación de grupos en la primera imagen, mientras que la segunda imagen muestra rectángulos amarillos que representan regiones de interés que se pueden generar en cada etapa de la búsqueda selectiva.

En el contexto de la arquitectura integrada de dos niveles de búsqueda selectiva, se destaca la serie R-CNN (Region-Based Convolutional Neural Network). Esta familia incluye tres algoritmos principales: R-CNN, Fast R-CNN y Faster R-CNN. Cada variante representa una evolución significativa con respecto a su predecesora en el campo de la detección de objetos.

4.3.1 R-CNN

La idea original de R-CNN se atribuye a un grupo de investigadores entre los que se encontraban Ross Girshick, Jeff Donahue, Trevor Darrell y Jitendra Malik en 2014 (Girshick et al., 2014). Esta propuesta pretende abordar el reto que suponen las ventanas deslizantes, que generan un gran número de zonas candidatas. La estrategia presentada

por R-CNN se basa en dos factores principales, como se muestra en la Figura 18. Primero, aborda la generación de recomendaciones de regiones mediante la implementación de un método de búsqueda selectiva. Con este proceso, se generan aproximadamente dos millares de regiones sugeridas para cada imagen, identificando así posibles regiones de interés. La segunda parte se centra en utilizar CNN para extraer características de cada área propuesta. Durante este proceso, cada área propuesta se escalará a un tamaño constante, lo que facilita el procesamiento a través de CNN. A continuación, las características extraídas se introducen en un clasificador SVM, que determina la clase de entidad contenida en la región propuesta, y en un regresor lineal, que se utiliza para refinar los límites del cuadro delimitador.

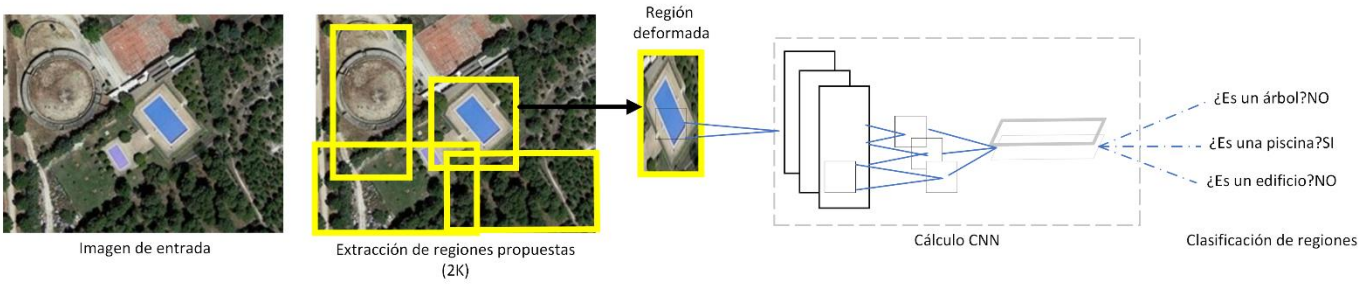


Fig 18 R-CNN

Sin embargo, a pesar de los esfuerzos por mejorar la eficiencia, R-CNN todavía tiene algunas limitaciones. Aunque uno de sus puntos fuertes es su velocidad de procesamiento, ejecutar R-CNN puede ser relativamente lento en términos de velocidad de detección, dado que CNN debe procesar cada recomendación de área individualmente.

4.3.2 Fast R-CNN

Fast R-CNN es una evolución de la arquitectura R-CNN original, propuesta por Ross Girshick en 2015 (Girshick, 2015). Este nuevo concepto supera algunas limitaciones de R-CNN y logra buenos resultados en el procesamiento de aplicaciones locales.

El funcionamiento de Fast R-CNN se muestra en la Figura 19. Se basa en la detección de objetos utilizando funciones determinadas. Sin embargo, no usa una red neuronal convolucional para cada concepto individual. Fast R-CNN genera una única CNN para toda la imagen. Luego, para cada segmento seleccionado, se extraen vectores de características de la salida de CNN utilizando características concatenadas. Varias partes

de este vector se concatenan para formar todas las partes de la región y formar los cuadros necesarios.

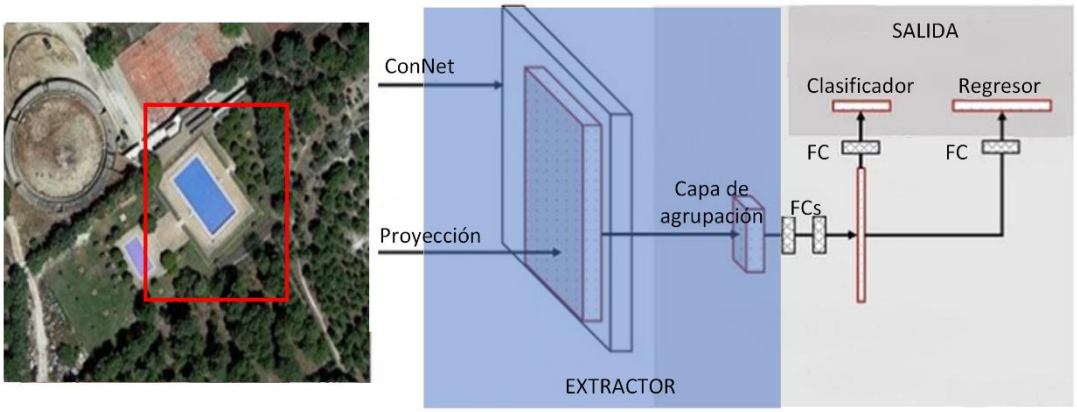


Fig 19 Fast R-CNN

En última instancia, R-CNN Fast logra una aceleración significativa en comparación con R-CNN al evitar el acceso individual a los miles de observaciones locales por imagen. No obstante, a pesar de estos logros, se presentan desafíos. Uno de ellos radica en la dependencia continua de estudios seleccionados para la creación del dominio deseado, lo cual puede obstaculizar el avance y no integrarse en el proceso de entrenamiento del modelo. Además, aunque Fast R-CNN es más eficiente en términos de velocidad que su predecesor, en algunas situaciones puede no ser la opción más veloz para aplicaciones en tiempo real.

4.3.3 Faster R-CNN

Faster R-CNN representa un avance sustancial con respecto a Fast R-CNN, enfocándose en el aspecto crítico de la generación de propuestas de regiones. Su innovación principal radica en la inclusión de una capa adicional denominada Red de Propuestas de Región (RPN, por sus siglas en inglés), la cual comparte gran parte de sus cálculos con la CNN utilizada para la extracción de características. La RPN aprovecha el mapa de características de la CNN para generar un conjunto de propuestas de regiones, cada una de ellas con una puntuación asignada al objeto correspondiente. Estas propuestas se emplean posteriormente tanto en la capa de agrupación de regiones como en el resto de la arquitectura de Fast R-CNN para llevar a cabo las tareas de clasificación y ajuste de la caja delimitadora.

La incorporación de RPN otorga a Faster R-CNN la capacidad de generar propuestas de regiones como parte integral del proceso de entrenamiento, mejorando así la eficiencia

y la coherencia entre las propuestas y la detección de objetos. Además, la RPN y la CNN comparten una parte sustancial de sus cálculos, lo que resulta en un rendimiento más rápido en comparación con Fast R-CNN.
A pesar de sus notables avances en velocidad y precisión en comparación con Fast R-CNN, Faster R-CNN todavía no cumple completamente con los requisitos necesarios para su implementación en aplicaciones de tiempo real.

4.3.4 Mask R-CNN

En el panorama de los algoritmos sobresalientes, surge Mask R-CNN. Esta aproximación extiende los fundamentos de Faster R-CNN y se distingue por integrar una rama adicional para la predicción de máscaras de objetos de forma paralela con la rama preexistente para la detección de los cuadros delimitadores. Un elemento crítico en el enfoque de Mask R-CNN es la alineación píxel a píxel, una característica esencial que estaba ausente en Fast/Faster R-CNN.
Mask R-CNN sigue el esquema de dos etapas, siendo la primera etapa idéntica a la de RPN. Sumado a la predicción de clase y el desplazamiento del cuadro en la segunda etapa, Mask R-CNN introduce la generación de máscaras binarias para cada región de interés (RoI).
La particularidad de este enfoque radica en su diferenciación de sistemas convencionales, donde la clasificación depende de las predicciones de máscara. La ejecución y el entrenamiento de la máscara R-CNN se simplifican gracias al marco R-CNN, que posibilita la implementación de diversos diseños arquitectónicos.
Además, la inclusión de la rama de máscaras añade una carga computacional moderada, permitiendo mantener el sistema ágil y fomentando la experimentación eficiente. La Figura 20 proporciona una representación visual que ilustra la segmentación lograda por este algoritmo.

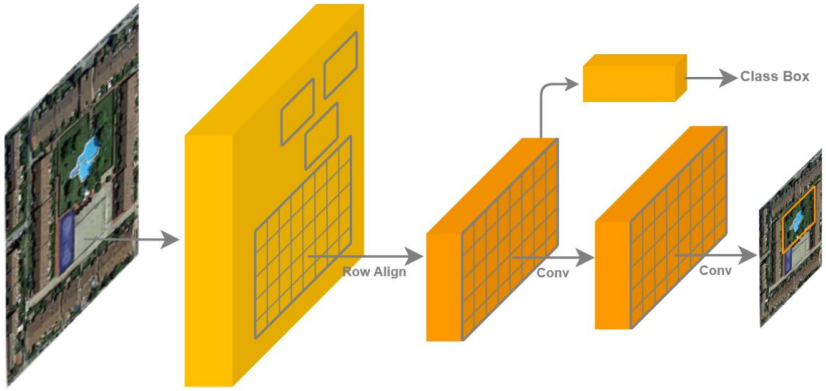


Fig 20 Mask R-CNN

4.3.5 Detectron2

En el mismo contexto, surge Detectron (Girshick, 2018), un software creado por el equipo de investigación de Facebook AI Research (FAIR) que abarca algoritmos de detección de objetos, incluyendo Mask R-CNN. Detectron tiene como objetivo proporcionar un alto nivel de calidad y rendimiento para la investigación en detección de objetos. La arquitectura de este algoritmo se ha diseñado de manera flexible para permitir la implementación y evaluación ágil de nuevas investigaciones en este campo. Sin embargo, surge la versión Detectron2 (Wu, 2019) como una mejora de Detectron. La distinción notable entre ambas versiones radica en que la última versión presenta un diseño más modular, adaptable y extensible, lo que permite un procesamiento más rápido, especialmente en sistemas con GPU. Detectron2 incorpora implementaciones de alta calidad de los algoritmos de detección más avanzados, como DensePose, redes piramidales con características panópticas y diversas variantes de la familia pionera de modelos Mask R-CNN, también desarrollada por el equipo de FAIR. Los creadores del algoritmo han reproducido las características de ResNet-50-FPN junto con el algoritmo Scale Jitter

4.4 Arquitecturas de Regresión

En la sección previa, se ha explorado la evolución de las arquitecturas de detección o regresión en dos pasos, desde su inicio con R-CNN hasta sus iteraciones más avanzadas en la actualidad. Aunque estas redes son consideradas como estándares en términos de precisión para la clasificación y localización de objetos, enfrentan restricciones en lo que

respecta a su velocidad, lo que podría ser insuficiente para aplicaciones de detección en tiempo real.

Debido a la creciente demanda de sistemas capaces de realizar detecciones en tiempo real, han surgido arquitecturas de un solo paso. Los expertos en este campo han optado por simplificar el proceso al reducir el número de etapas, lo que ha dado lugar a algoritmos capaces de calcular directamente las coordenadas de las cajas delimitadoras y las probabilidades de clasificación a partir de una sola imagen.

4.4.1 Single Shot Detector (SSD)

El Detector SSD, también conocido como Single Shot MultiBox Detector (Liu et al., 2016), representa un avance significativo en comparación con modelos anteriores ya que, sin sacrificar notablemente la precisión, logra una mayor velocidad y eficiencia.

La característica más destacada de la arquitectura de SSD en comparación con sus antecesores radica en su enfoque de detección de objetos en una sola pasada, eliminando la necesidad de depender de una red de propuestas de regiones para seleccionar áreas de interés y luego clasificarlas, lo que le otorga su nombre. Este enfoque se traduce en un notable aumento de velocidad y eficiencia en comparación con los modelos de dos etapas como R-CNN y sus variantes.

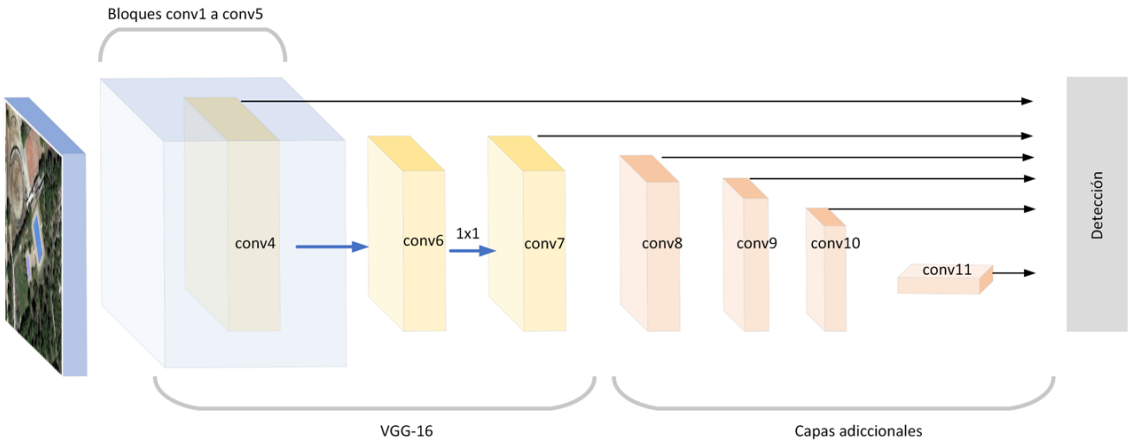


Fig 21 Estructura SSD

La arquitectura de SSD se basa en una red neuronal convolucional, como VGG16, a la cual se le agregan capas convolucionales adicionales con tamaños decrecientes. Esta configuración facilita la detección de objetos de diversas dimensiones. Las capas

convolucionales iniciales, de alta resolución, son especialmente eficientes en la identificación de objetos pequeños en la imagen, capturando detalles minuciosos. A medida que avanzamos en la red, las capas convolucionales se vuelven más pequeñas, reduciendo la resolución, pero capturando información más general de la imagen, lo que las hace aptas para detectar objetos más grandes en la escena.

En el algoritmo SSD, la imagen de entrada se divide en una cuadrícula de celdas de diferentes dimensiones, como se describe en la Figura 22. Estas celdas pueden ser, por ejemplo, de 9x9, 6x6, 4x4, entre otras. Cada celda abarca una escala diferente de la imagen, lo que permite a SSD detectar objetos de varios tamaños. Las celdas más grandes se utilizan para reconocer objetos más grandes y se procesan mediante las últimas capas convolucionales, mientras que las celdas más pequeñas son procesadas por las capas convolucionales iniciales para detectar objetos de menor tamaño.

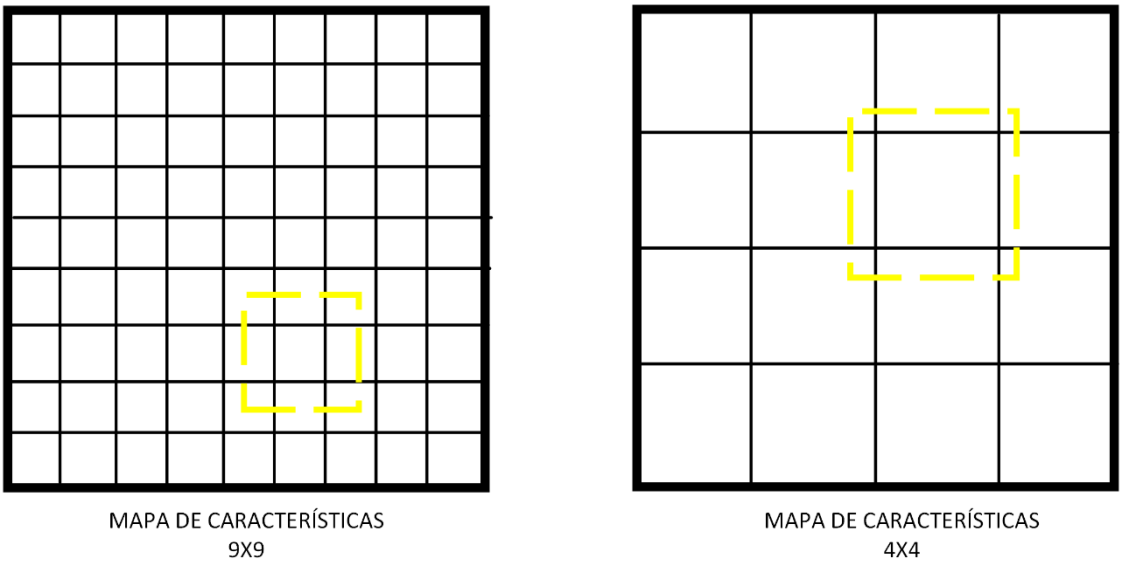


Fig 22 Celdas de malla

El SSD genera una agrupación de cajas ancla en cada celda de esta malla. Estas cajas ancla son rectángulos con diversos tamaños y escalas, definidos por su altura (h) y anchura (w), centrados en la celda correspondiente de la malla, definidos por x e y. La función de cada caja ancla es realizar una propuesta de región para detectar un posible objeto, como se muestra en la Figura 23.

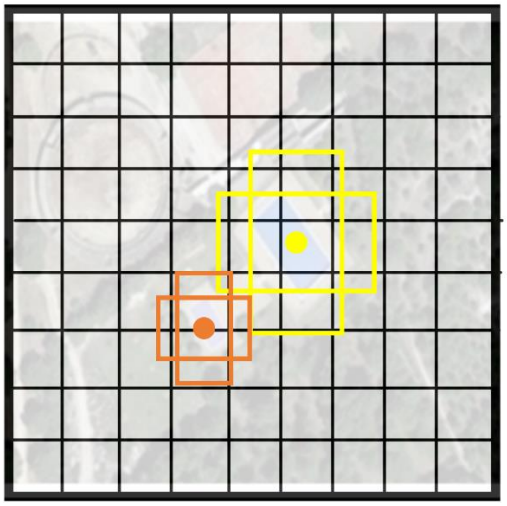


Fig 23 Detección mediante SSD

En la Figura 23, se proporciona una representación visual que brinda una comprensión más profunda del proceso llevado a cabo por SSD. En este contexto, es evidente que para cada caja ancla, el algoritmo realiza dos tipos esenciales de predicciones: en primer lugar, asigna una puntuación de confianza a cada posible clase de objeto; y, en segundo lugar, realiza ajustes dirigidos para delimitar la caja de manera precisa.

Específicamente, las puntuaciones de confianza desempeñan un papel crucial al indicar la posibilidad en términos de probabilidad de que una clase particular de objeto forme parte de la caja ancla considerada. Estas puntuaciones representan la confianza de la red en la presencia de cada categoría y son esenciales para la posterior clasificación de objetos.

Por otro lado, los ajustes destinados a delimitar la caja permiten que SSD mejore la posición y las dimensiones de la caja ancla original. Esta optimización resulta en una adaptación más precisa de la caja al objeto detectado. Estos parámetros se expresan mediante escalas y desplazamientos relativos a la caja ancla, garantizando que la caja final se ajuste de manera óptima alrededor del objeto identificado. Este proceso de refinamiento es fundamental para asegurar una localización precisa y efectiva de los objetos en la imagen.

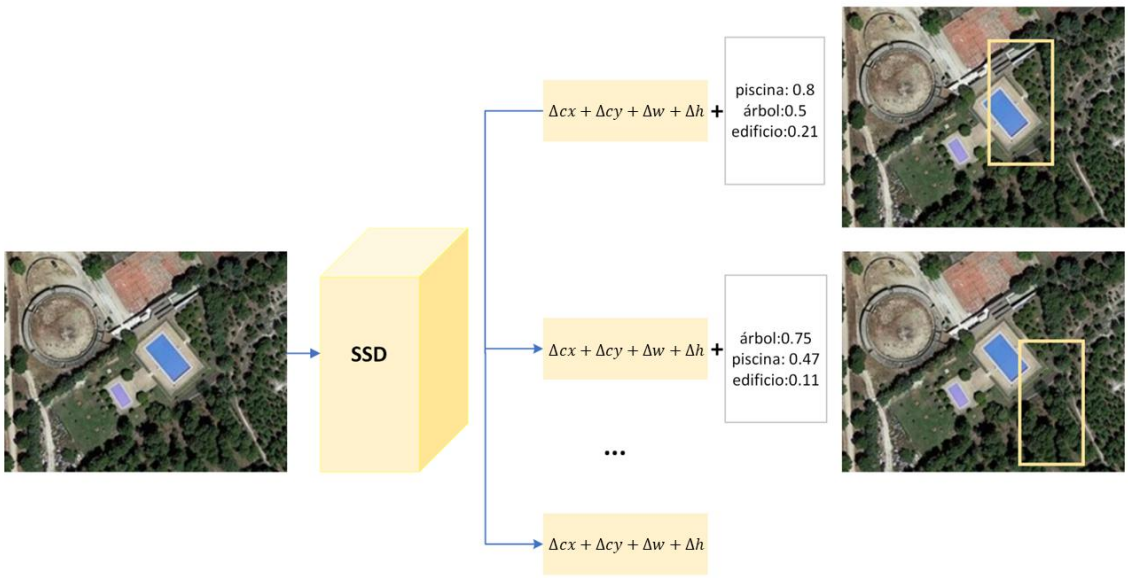


Fig 24 Predicción de puntuación ajustes de caja

En la figura 24 se resume el enfoque de SSD, el cual genera conjuntos de cajas ancla en cada celda de una cuadrícula, definidas por x, y, h, w, seguido por la predicción de puntuaciones de confianza y ajustes para las cajas delimitadoras de cada caja de anclaje. Esta característica dota a SSD de la capacidad para detectar una amplia variedad de tamaños y formas de objetos en una sola pasada a través de la red neuronal convolucional.

A pesar de las ventajas mencionadas, es crucial reconocer que SSD también presenta ciertas limitaciones. En particular, su desempeño podría ser menor en la detección de objetos pequeños en comparación con enfoques de dos etapas. Esto se debe a la utilización de cajas de anclaje con dimensiones predefinidas en SSD, lo cual puede no ser óptimo para capturar objetos de menor tamaño. Además, aunque SSD logra una mayor velocidad en comparación con los métodos de dos etapas, en varias aplicaciones de tiempo real, especialmente en dispositivos que cuentan con recursos limitados, puede no llegar a alcanzar la velocidad requerida para un rendimiento óptimo.

4.4.2 YoloV4

YOLO, acrónimo de You Only Look Once" (Redmon, 2016), representa un avance significativo en términos de velocidad en comparación con enfoques anteriores. De

manera similar a SSD, YOLO también elimina la necesidad de usar regiones de interés para posteriormente clasificarlas, como se realiza en R-CNN. En su lugar, YOLO, que significa "You Only Look Once", se distingue por su enfoque de detección de objetos en una sola pasada, lo que justifica su nombre. Sin embargo, YOLO introduce una distinción crucial en comparación con SSD en la forma en que selecciona las cajas delimitadoras para la detección de objetos. Esta estrategia simplifica el proceso de detección y proporciona a YOLO una mayor velocidad en comparación con SSD.

El algoritmo YOLO toma una imagen como entrada y emplea una red neuronal convolucional profunda y sencilla para identificar objetos en la imagen. A continuación, se presenta la arquitectura de la red CNN que constituye la base de YOLO.

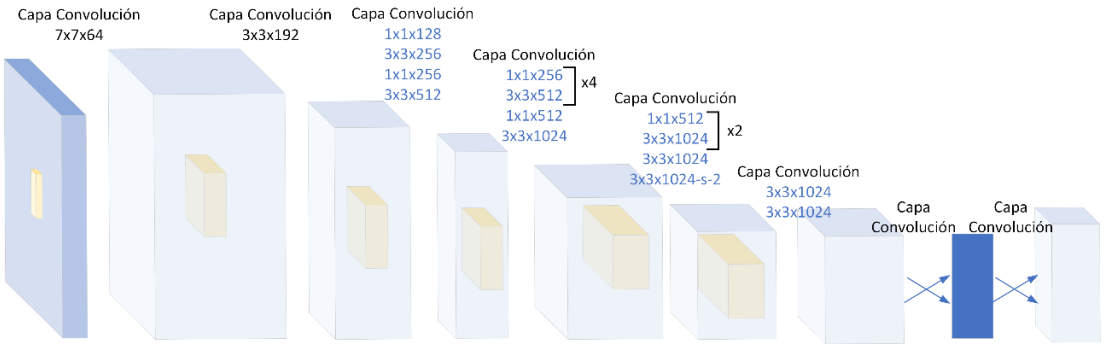


Fig 25 Arquitectura YOLO

La estructura se conforma por la unión de capas convolucionales (24 capas), seguidas de un conjunto de capas completamente conectadas (2 capas). Estas capas convolucionales implementan filtros de dimensiones 3x3 y 1x1, y capas de agrupación, reduciendo de este modo la dimensión en su recorrido. Mientras las capas convolucionales extraen las características de la imagen, son las capas totalmente conectadas las que ejecutan las predicciones.

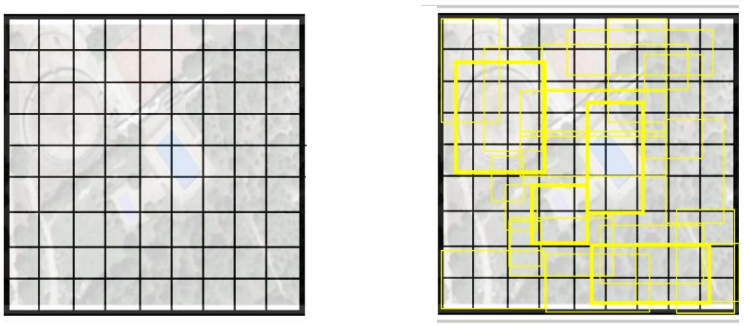


Fig 26 Predicciones YOLO

La red recibe una imagen como entrada y produce una matriz de dimensiones SxSx(B*5+C) en la salida, donde SxS representa la cantidad de celdas en las que se divide la imagen. B representa el número de cajas delimitadoras por cada celda, 5 corresponde al número de elementos que cada celda predice, y C denota la cantidad de clases a detectar. En esta matriz, cada celda en la cuadrícula anticipa B cajas delimitadoras, cada una con sus respectivas puntuaciones de confianza, y además estima una distribución de probabilidad para las clases. Cada predicción de caja delimitadora comprende cinco atributos: las coordenadas (x, y) del centro de la caja (en relación con la posición de la celda en la cuadrícula), el ancho (W) y alto de la caja (h) (en relación con el tamaño de la imagen), junto con una puntuación de confianza que indica la probabilidad de que la caja contenga un objeto y cuán adecuadamente se ajusta la caja delimitadora a ese objeto.

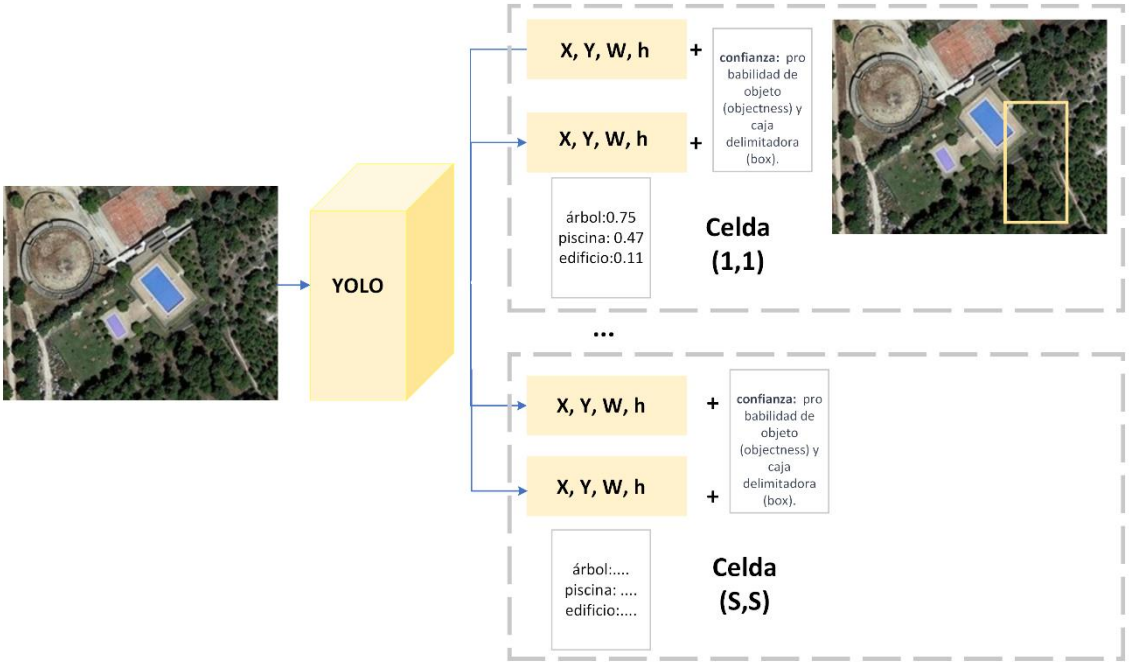


Fig 27 Funcionamiento YOLO

En el contexto de la comparación entre SSD y YOLO en la generación de cajas delimitadoras, se observa una diferencia en sus metodologías. En SSD, cada celda contiene un conjunto de cajas ancla, las cuales se ajustan en tamaño y desplazamiento para obtener las cajas delimitadoras finales. En cambio, YOLO genera directamente las cajas delimitadoras. Si se generan múltiples cajas en una celda de YOLO, por ejemplo, cinco cajas, solo una de ellas es responsable de detectar el objeto, específicamente aquella

con la mayor IoU con la caja real del objeto. Las otras cajas generadas se descartan. En contraste, SSD utiliza todas las cajas ancla para hacer pronósticos.

En el proceso inmerso utilizado en YOLO, la confianza en la detección de la caja principal se multiplica por las probabilidades pronosticadas para cada clase de objeto, lo que deriva en la probabilidad final de cada categoría. Luego, YOLO detecta la clase con la probabilidad más alta. Para finalizar, YOLO utiliza la supresión no máxima para eliminar cajas redundantes. Esta operación es crucial cuando múltiples celdas detectan un mismo objeto. YOLO retiene solo la caja con la mayor probabilidad de clase y descarta las demás cajas. Por lo tanto, YOLO, al predecir directamente las cajas delimitadoras, logra una velocidad considerable en comparación con SSD. Sin embargo, este enfoque también implica una disminución en la precisión. Aunque YOLO sacrifica precisión al predecir directamente las cajas, SSD refina sus pronósticos ajustando las cajas ancla durante la detección. Además, SSD tiene la capacidad de realizar pronósticos en distintos niveles de la imagen, lo que resulta en una detección más eficiente de objetos de diferentes tamaños.

En la evolución de YOLO, se han desarrollado varias versiones, como YOLOv2 (Redmon, 2017), YOLOv3 (Redmon, 2018) y YOLOv4 (Bochkovskiy, 2020). Además, se han creado variaciones de estas versiones para adaptarlas a diversos contextos y mejorar su rendimiento (Chen, 2021), (Silva, 2020). Con la introducción de estos diversos algoritmos de detección de objetos en imágenes, el siguiente capítulo se centrará en hacer una revisión detallada del estado de la literatura acerca de la detección de recursos hídricos y piscinas mediante la utilización de imágenes satelitales e inteligencia artificial.

CAPÍTULO V

ESTADO DEL ARTE EN LA DETECCIÓN DE RECURSOS HÍDRICOS EN IMÁGENES SATELITALES

VNiVERSiDAD D SALAMANCA

Uno de los hitos más importante de este trabajo es investigar acerca de las técnicas que posibiliten la detección automática de recursos hídricos en imágenes haciendo uso de técnicas basadas en inteligencia artificial. Por lo tanto, para poder llevar a cabo este trabajo, es necesario realizar una revisión detalla del estado de la literatura.

5 Estado del arte en la detección de piscinas en imágenes satelitales

La teledetección es una subdisciplina del campo más amplio de la geomática, dedicada a recolectar, almacenar y analizar información geoespacial y que complementa las disciplinas de SIG, topografía y cartografía. La teledetección es la ciencia, tecnología y arte de obtener información sobre objetos desde una distancia, permitiendo diferenciar patrones y relaciones no evidentes en imágenes convencionales. Las imágenes capturadas por sensores digitales de cámaras aéreas y satélites son utilizadas extensivamente para cartografiar la superficie terrestre, recursos naturales e infraestructura urbana. Estas imágenes también se emplean en la evaluación de eventos de desastre y la planificación de actividades de respuesta de emergencia

La ortofotografía facilita ligeramente el reconocimiento de las piscinas al aire libre. Entre las posibilidades existentes para resolver este problema está el uso de imágenes de satélite en combinación con técnicas de aprendizaje automático (Habibie, 2020).

Con la proliferación y evolución de los vehículos aéreos no tripulado ("Unmanned Aerial Vehicle", UAV) y sus sistemas de control, denominados "RPAS" por sus siglas en ingles "Remotely Piloted Aircraft System" (Sistema de Aeronave Pilotada a Distancia) el proceso de detección de imágenes es mucho más rápido y rentable que hace una década (Gray, 2019).

El principal problema en estos procesos es relacionar las imágenes recogidas por satélites o drones con un sistema de detección de piscinas y la correspondiente verificación de estas dentro de las bases de datos de los sistemas locales. La solución para el problema de detección siguió los avances en la literatura de algoritmos de aprendizaje automático (Goodfellow, 2016).

5.1 Trabajos anteriores

Los trabajos relacionados que figuran en esta sección son recientes y presentan pruebas y aportaciones relevantes para el área. Sin embargo, son limitados y no se focalizan en los puntos analizados necesarios para el control y gestión de la verificación de la legalidad, detección de objetos y clasificación de imágenes.

El trabajo propuesto por Tien (Tien, 2007), se centra en la localización de fuentes de agua para los servicios de emergencia, se procesaron imágenes de satélite y, a continuación, las imágenes se segmentaron. En este trabajo se introduce las SVM para clasificar masas de agua de pequeño tamaño, concretamente piscinas. Los SVM son algoritmos de aprendizaje estadístico que construyen un hiperplano para separar dos clases con un margen máximo. Pueden manejar datos no linealmente separables transformando los datos en un espacio de características de mayor dimensión. En este trabajo, se resuelve un problema de optimización para encontrar el hiperplano separador. Las SVM son aplicadas para identificar objetos de interés a partir de valores de píxeles. Se utilizan funciones kernel para mapear los datos en un espacio dimensional superior, permitiendo la separación no lineal. El procedimiento experimental para clasificar piscinas en imágenes satelitales utilizando SVM involucra la elección de una imagen de entrenamiento y prueba donde se extraen valores RGB de los píxeles y se preprocesan con umbrales para generar archivos de clasificación. Los archivos de entrenamiento y prueba se utilizan para ejecutar el SVM, que clasifica automáticamente los píxeles en piscina y no piscina. Los resultados son utilizados para crear imágenes aisladas de piscinas.

En este estudio se logró identificar cuerpos de agua en imágenes Satelitales. Aunque no se logró determinar con precisión las dimensiones de las piscinas, sí se pudo identificarlas en las imágenes. Como líneas futuras, se planea la realización de más experimentos con diferentes texturas para mejorar los resultados y estudiar la identificación de cuerpos de agua en diferentes condiciones climáticas. Así como incluir la adquisición de datos de sensores digitales y la identificación de piscinas en imágenes multibanda.

En el trabajo de Galindo (Galindo, 2009), los autores propusieron un sistema que aborda específicamente la detección de piscinas en imágenes en color de áreas urbanas capturadas por el satélite Quickbird. La motivación principal es monitorear y localizar piscinas llenas durante períodos de sequía para que las autoridades locales puedan tomar medidas. Durante este trabajo, se presenta el desafío de detectar piscinas con agua en imágenes satelitales de alta resolución. Se discuten las complicaciones relacionadas con

la resolución de las imágenes en color, la forma de las piscinas y el color del agua. Estos factores hacen que la detección de piscinas sea más compleja de lo que parece a simple vista. El enfoque propuesto consta de dos etapas. Primero, se realiza un análisis de color para identificar regiones homogéneas que puedan corresponder a piscinas, para ello se utiliza el espacio de color C1C2C3 para detectar áreas de agua en la imagen. Se menciona la normalización de la banda C1 y la aplicación de umbralización de Otsu para obtener una imagen binaria con posibles piscinas. Posteriormente, se utilizan técnicas de contornos activos para refinar la forma de las piscinas. Mediante el uso de algoritmos Snake se ajustan los contornos de las regiones candidatas de piscinas. Las pruebas se realizaron en imágenes de la Costa del Sol en España. El algoritmo propuesto demuestra un alto grado de precisión en la detección de piscinas llenas, con tasas de éxito que superan el 93%. Específicamente, de un total de 250 piscinas llenas evaluadas en el conjunto de datos, el algoritmo logró detectar correctamente 234 piscinas, lo que representa un índice de precisión del 93.6%. Además de su capacidad para identificar con precisión piscinas llenas, el algoritmo también se destaca por su habilidad para estimar con exactitud el volumen de agua contenido en cada piscina. Las estimaciones de volumen obtenidas a través del algoritmo presentaron un desvío promedio del 6.2% en comparación con mediciones manuales realizadas en el terreno.

La motivación de Kim et al. (Kim, 2011) se debe a la preocupación por la detección piscinas abandonadas en California debido a su relación con la proliferación de insectos, una problemática también observada en otros países (Passos, 2020), En este trabajo se utiliza el satélite GeoEye-1 para obtener imágenes pancromáticas y multiespectrales. La imagen pancromática tenía una mayor resolución espacial, pero carecía de información de color, mientras que la imagen multiespectral proporcionaba colores, aunque con una resolución espacial más baja. Con el fin de combinar resolución y color, aplicaron el proceso de pansharpening mediante un filtro de paso alto (HPF), lo que mejoró la calidad de las imágenes en comparación con las originales. Para identificar visualmente piscinas privadas en el área de estudio, se utilizaron fotografías aéreas de Google Earth® y las imágenes pansharpened de GeoEye-1. En este trabajo, se excluyeron las piscinas que estaban cubiertas por árboles y sombras. En total, se identificaron visualmente 822 piscinas que luego se utilizaron como referencia para validar la precisión de la clasificación. Lo autores optaron por el enfoque de Análisis Geográfico Basado en Objetos de Imagen (GEOBIA) para extraer piscinas privadas de las imágenes pansharpened de GeoEye-1. GEOBIA consta de dos etapas: segmentación de imágenes y clasificación. La segmentación generó límites vectoriales para los objetos de imagen individuales, lo que permitió la extracción de atributos espaciales y espectrales para cada objeto. Para distinguir las piscinas de otras características, se empleó el Índice de

Diferencia Normalizada de Agua (NDWI) para identificar cuerpos de agua. Utilizaron el valor espectral de la imagen NDWI y atributos espaciales como el ajuste rectangular (RF) y el tamaño de los objetos de imagen. Durante la elaboración de este trabajo, se recolectaron 800 muestras aleatorias que se utilizaron para evaluar la precisión general y las precisiones individuales en las fases de validación.

El método de fusión HPF resultó en una imagen pansharpened con coeficientes de correlación de 0.93 para las cuatro bandas espectrales. El coeficiente de correlación evalúa la calidad del color del pansharpening en comparación con la imagen multiespectral original. Utilizaron una imagen NDWI calculada a partir de las imágenes pansharpened para identificar piscinas. Luego, aplicaron un ajuste de densidad para mejorar la identificación de las piscinas y reducir los valores negativos en la imagen NDWI. Para la clasificación de piscinas, emplearon el algoritmo de Segmentación de Multiresolución, una componente del enfoque GEOBIA. Adicionalmente, probaron diferentes parámetros de escala para optimizar la segmentación, lo que inicialmente resultó en 18,666 objetos de imagen, sin embargo, aplicando criterios basados en tamaño y forma, redujeron los segmentos a 987 para excluir sombras.

En conclusión, Kim et al. logran identificar un porcentaje significativo de piscinas con precisión utilizando la metodología GEOBIA y empleando diversas estrategias para mejorar la precisión de la clasificación, incluida la exclusión de sombras y la consideración del tamaño y la forma de los objetos. Aunque la detección automática alcanzó una tasa de precisión del 94 por ciento, recomendaron combinar GEOBIA con interpretación manual para lograr una detección más completa. En este trabajo como líneas futuras se propone la incorporación de conjuntos de datos adicionales, como LIDAR (Light Detection and Ranging) y datos de edificios/carreteras, ya que, podrían mejorar la precisión de la clasificación y la separación de piscinas de cuerpos de agua más grandes.

Rodríguez-Cuenca (Rodriguez-Cuenca et al., 2014), presenta una metodología semiautomática para determinar las ubicaciones de las piscinas existentes en un entorno urbano utilizando imágenes aéreas y datos LIDAR. Este artículo presenta un enfoque semiautomático para detectar piscinas en áreas urbanas. La identificación y extracción comienzan con la lectura de la imagen aérea y la rasterización de los datos LIDAR, para ello, se leen las bandas de la imagen aérea y se obtiene información de altura e intensidad de los datos LIDAR. En este trabajo se rasterizan los datos para generar un modelo digital de superficie (DSM), un modelo digital de terreno (DTM) y un modelo digital de superficie normalizado (nDSM). Luego, se segmenta la imagen aérea en diferentes áreas, aplicando un método de crecimiento de regiones a una banda de la imagen aérea,

utilizando la primera componente del análisis de componentes principales (PCA). Este método, divide la imagen en regiones basadas en similitudes de píxeles vecinos. En un paso posterior, se crea un grafo de adyacencia de regiones (RAG) para gestionar la imagen a nivel de región, utilizándose índices como el NDVI, intensidad LIDAR, nDSM y un índice de piscinas para detectar diferentes tipos de coberturas del suelo. En esta investigación se aplica la teoría de la evidencia de Dempster-Shafer con el objetivo es mapear cinco tipos de coberturas del suelo, principalmente piscinas. La teoría de Dempster-Shafer se usa para asignar probabilidades a categorías. Cada categoría tiene una probabilidad asignada según índices de decisión. Luego de combinar la evidencia y asignar probabilidades, las regiones se asignan a las categorías con la probabilidad más alta. Los resultados contienen falsos positivos en regiones sombreadas, etiquetando zonas oscuras como agua debido a similitudes espectrales. Como extra en este trabajo, se corrige reasignando categorías en áreas sombreadas usando imágenes de sombras generadas desde datos de vuelo, eliminando de esta forma posible falsos positivos y regiones pequeñas. El método propuesto emplea el NDSM basado en la respuesta espectral de piscinas, un grafo de adyacencia de regiones y la teoría de Dempster-Shafer para identificar la ubicación de esta cobertura en áreas urbanas. Durante la ejecución de este trabajo, se ha evaluado junto con otros dos métodos más: clasificaciones supervisadas (Mahalanobis y SVM) y el método NDWI con objeto de detectar piscinas en un conjunto de datos reales de Alcalá de Henares. Los resultados muestran una precisión del 99,86% y una índice kappa de 0,79 para el método propuesto con NDSPI, superando a Mahalanobis y NDWI (90,19% y 99,25%) y cercano al SVM (99,87%). La innovación radica en su independencia de entrenamiento previo y en la necesidad de ajustar umbral de cada índice para una buena detección. Aunque hubo falsos positivos, los errores de comisión fueron menores, y SVM y el método propuesto detectaron casi todas las piscinas.

Ferner et al. (Ferner, 2019) estudia la detección de viviendas con piscinas mediante CNN, aplicadas a mapas de calor de carga construidos a partir de perfiles de carga, es decir analizan representaciones gráficas o conjuntos de datos que muestran cómo varía el consumo de energía eléctrica de un dispositivo, un sistema o un conjunto de usuarios a lo largo del tiempo. Aunque este trabajo no utiliza imágenes de satélite ni aéreas, está relacionado con el presente proyecto, ya que utiliza una CNN.

Los resultados demuestran que las redes neuronales convolucionales son capaces de aprender de manera competitiva la presencia de piscinas, incluso en el caso de un conjunto de datos relativamente pequeño que contiene perfiles de carga de 64 hogares con piscinas. Se evaluaron dos configuraciones: i) usando la CNN tal como está para la

clasificación (CNN pura) y ii) extrayendo las características de la CNN (después de la cuarta capa convolucional) y clasificando estas características mediante un clasificador de vecinos cercanos (CNN+k-NN). La última versión tiene como objetivo estudiar si las características obtenidas por la CNN son adecuadas únicamente en un entorno de CNN o si se generalizan para diferentes clasificadores. Cuando se usa características de la CNN con un clasificador de vecinos cercanos (k-nearest neighbor), se mantiene la precisión de clasificación, pero se pierde precisión. El mejor clasificador en términos de precisión solo logra el 60.6% de precisión. Sin embargo, la configuración completa de la CNN supera a los métodos anteriores y obtiene una precisión del 95.5% y una precisión del 71.9%.

Domozi et al. (Domozi, 2019) se centra en la detección automatizada de piscinas en imágenes aéreas, tomadas por drones, a través de la implementación de redes neuronales, complementada por la utilización de la plataforma de procesamiento de imágenes Pix4D.

El método propuesto implica el entrenamiento de una red neuronal convolucional (CNN) para reconocer patrones y características específicas de piscinas en imágenes de alta resolución. Estas imágenes se generan mediante la plataforma de procesamiento de imágenes Pix4D, que permite la creación de ortofotos precisas a partir de imágenes aéreas y satelitales. Pix4D Capture, una herramienta de planificación de vuelos en la plataforma Pix4D, permite la captura programada de imágenes aéreas desde drones, lo que garantiza la obtención de datos geoespaciales precisos y controlados. Durante el proceso de entrenamiento de la CNN, se emplea un conjunto de muestras de entrenamiento meticulosamente seleccionadas, que representan una amplia gama de formas y condiciones de piscinas, así como elementos no relacionados, como elementos arquitectónicos. Se enfatiza la importancia de la diversidad en las muestras de entrenamiento para lograr una detección robusta y precisa en diversas situaciones. Una vez entrenada, la CNN se aplica a las ortofotos generadas por Pix4D. Estas ortofotos capturan con gran detalle las características geoespaciales de las áreas evaluadas. La red neuronal procesa estas imágenes y genera resultados de detección que indican la presencia de piscinas en ubicaciones específicas. La integración de Pix4D en el proceso de detección agrega un componente esencial de procesamiento de imágenes geoespaciales de alta calidad, lo que contribuye a la precisión y confiabilidad del sistema de detección.

La evaluación de la eficacia del enfoque se realiza mediante la comparación de los resultados de detección con datos manuales de referencia. Se observa que la precisión de la detección está influenciada por varios factores, incluida la calidad de las imágenes de

entrenamiento y la resolución de las imágenes de prueba generadas por Pix4D. En condiciones ideales, la red neuronal logra una tasa de detección cercana al 100%, lo que demuestra su capacidad para reconocer con éxito piscinas en imágenes de alta resolución.

A pesar de los resultados prometedores, se identifican desafíos en la detección de piscinas en condiciones no ideales, como la presencia de cobertores u obstrucciones, lo que puede afectar la tasa de falsos positivos. La adaptabilidad de la red neuronal a diferentes escenarios y condiciones se discute como un área de mejora, y se sugieren enfoques futuros para abordar estos desafíos y optimizar aún más la precisión de la detección, posiblemente aprovechando las capacidades avanzadas de procesamiento de Pix4D.

Passos et al (Passos, 2020) en su trabajo propone un mecanismo para la detección automática de balsas de agua que propician criaderos de mosquitos. La base de datos proporciona el material necesario para entrenar y evaluar el sistema. La base de datos se compone de videos aéreos que capturan diferentes escenarios y objetos de interés. Cada video se clasificó manualmente, cuadro a cuadro, con cajas delimitadoras que identifican la ubicación de los objetos asociados a balsas de agua. La información geográfica obtenida del dron utilizado en la captura de videos permite una localización precisa de cada objeto en la escena. Para la detección de objetos, se emplea arquitectura Faster R-CNN, que ha demostrado un alto rendimiento en la detección de objetos en imágenes y videos. La arquitectura consta de tres componentes principales: un extractor de atributos, una RPN y un módulo de clasificación y regresión. El extractor de atributos, basado en una red neuronal convolucional profunda, procesa las imágenes y extrae mapas de atributos convolucionales. La RPN genera posibles regiones de interés en función de estos mapas, y el módulo de clasificación y regresión se encarga de asignar etiquetas a estas regiones y refinar sus cajas delimitadoras. La técnica de rastreo de objetos mediante Phase Correlation es introducida para mejorar la consistencia de las detecciones en cuadros sucesivos. Esta técnica se aprovecha de la propiedad de translación en el tiempo entre cuadros contiguos. Al calcular la correlación de fase entre los cuadros, se obtiene el desplazamiento espacial entre ellos, permitiendo alinear y asociar las detecciones de objetos. La translación contribuye a reducir los falsos positivos y mejorar la precisión del sistema en la detección de criaderos de mosquitos. En este trabajo, se utilizan varios modelos de Faster R-CNN con diferentes configuraciones de arquitectura y extracción de atributos para llevar a cabo la detección de objetos en los videos aéreos de la base de datos. En la fase de resultados se evalúan métricas estándar como AP50, precisión, revocación y F1-score.

Los resultados cuantitativos muestran que los modelos que utilizan la arquitectura FPN (Fast Page Mode) logran los valores más altos de AP50 y revocación. Sin embargo, estos modelos también tienen tasas más altas de falsos positivos, lo que resulta en una disminución de la precisión. Por otro lado, el modelo que utiliza la arquitectura R101-C4 logra un equilibrio entre precisión y revocación, obteniendo el mayor valor de F1-score. Cuando se analiza el impacto del rastreo de objetos mediante Phase Correlation en las detecciones, se muestra cómo esta técnica reduce significativamente los falsos positivos y mejora la precisión global del sistema. Finalmente, se comparan las detecciones antes y después del rastreo, demostrando claramente los beneficios de esta técnica en la mejora de la consistencia de las detecciones.

Lima et al. (Lima, 2021) propone un sistema de detección y clasificación de construcciones, específicamente piscinas, basado en imágenes aéreas y datos geográficos. El sistema comprende tres componentes principales: procesamiento de imágenes aéreas para detección, integración de sistemas de información municipales y visualización de construcciones ilegales. En este trabajo, se emplean algoritmos de Aprendizaje Profundo (Deep Learning) para la detección de objetos, y se utilizan procesos ETL para la integración con otros sistemas. El objetivo es acelerar la identificación de piscinas no autorizadas, la detección, clasificación y georreferenciación de objetos distintos en imágenes requiere la integración de diferentes sistemas. El sistema desarrollado admite las siguientes características: i) importar imágenes GeoTIFF y procesarlas en un modelo entrenado para detectar objetos clasificados como piscinas; ii) georreferenciar las coordenadas de los objetos detectados en estas imágenes; e iii) integrar estos resultados con datos de capas geográficas relacionadas con licencias de construcción y parcelas en la municipalidad para validar si hay una licencia para ese propósito.

El proceso general del trabajo de Lima et al. involucró dos fases principales:

1. Detección y clasificación de piscinas en imágenes aéreas municipales

   Para crear un sistema de análisis de piscinas, es necesario contar con un conjunto de datos que permita entrenar a los algoritmos. Para esta investigación, no fue posible obtener un conjunto de datos preparado completo para piscinas, por lo que se comenzó uno nuevo a partir de un conjunto de datos parcial disponible en la base de datos Open Kaggle. Dado que este conjunto de datos aún no estaba catalogado, fue necesario realizar un preprocesamiento para etiquetar la clase de

piscinas, así como las cajas delimitadoras de los objetos contenidos en cada imagen. La preparación del conjunto de datos incluyó un conjunto de imágenes de fotografías aéreas existentes de la zona de Redlands en California.

El modelo de entrenamiento se realizó en Faster R-CNN Inception V2 con un mAP de 69%, YOLOv3-SPP con núcleo darknet53 con un mAP de 80%, y Keras-Retinanet con un mAP de 83%.

Para el proceso de detección de piscinas, fue necesario convertir las coordenadas de ubicación (x, y) a un Sistema de Referencia de Coordenadas (CRS, por sus siglas en inglés) relativo al mapa ortofotográfico. Con el fin de representar el mundo real, los SIG funcionan como una pila de capas. En este trabajo, se desarrolló un método para identificar las coordenadas del objeto en el sistema decimal (longitud, latitud) que recibe como parámetros de entrada el mosaico de la imagen donde se detectó el objeto, las coordenadas del cuadro delimitador del objeto en la imagen, el índice de generación y el valor de confianza del porcentaje de detección del objeto. El uso de la API OSGEO completó el proceso de georreferenciación.

2. Integración de esos datos con otros sistemas de información de la municipalidad.

   La integración con sistemas municipales involucró un proceso ETL ("Extract, Transform, Load") para fusionar datos de parcelas, construcciones y piscinas. Adicionalmente, se validó la detección verificando intersecciones y se generó un archivo GeoJson con detalles de propietarios, validaciones y confianza en la detección. Se desarrolló una interfaz de usuario que permite visualizar y analizar resultados en mapas interactivos. Se implementó una agrupación de piscinas para mejorar la visualización y se proporcionó información detallada al hacer clic en los marcadores.

En esta sección, para ofrecer un análisis más detallado y claro, analizamos diferentes trabajos realizados. Todos los métodos explicados en esta sección se resumen en la Tabla 2.

Tabla 2 Trabajos anteriores

Autor	Año	Algoritmo	Imágenes	Precisión
Tien *et al.*	2007	SVM	Imágenes satelitales	-
Galindo *et al.*	2009	Análisis de color y segmentación	Imágenes satelitales	93%
Kim *et al.*	2011	Pan-sharpening	Imágenes satelitales	94%
Rodríguez-Cuenca *et al.*	2014	SVM	Imágenes aéreas + LIDAR	99,86%
Ferner *et al.*	2019	CNN/ Vecinos Cercanos	-	71,9%
Domozi *et al.*	2019	R-CNN	Imágenes de dron	99%
Passos *et al.*	2020	Faster R-CNN/ResNet-101-C4	Imágenes de dron	74%
Lima *et al.*	2021	Faster R-CNN	Imágenes GeoTIFF	83%

CAPÍTULO VI

ARQUITECTURA PROPUESTA

VNiVERSiDAD
D SALAMANCA

En los capítulos anteriores se ha realizado una introducción a los sistemas multiagente, a las técnicas de reconocimiento de objetos en imágenes, así como las propuestas actuales en materia de reconocimiento de recursos hídricos. En este capítulo se presenta una plataforma que trata de dar solución a la detección y validación de recursos hídricos de forma automática.

6 Plataforma de análisis de imágenes satelitales para el descubrimiento de recursos hídricos

6.1 Propuesta

Como uno de los puntos fuertes de este trabajo de investigación, se presenta una novedosa arquitectura que permite la detección automática de recursos hídricos adquiriendo imágenes de diferentes fuentes externas. El sistema propuesto puede aplicar diferentes algoritmos para determinar los recursos hídricos pudiendo comprobar automáticamente los que están referenciados y los que no lo están en un territorio particular accediendo a las bases de datos de control local. Para que el sistema sea escalable, robusto y capaz de fusionar la información procedente de varias redes neuronales, se utiliza una arquitectura multiagente. Un sistema multiagente permite construir una plataforma reconfigurable dinámicamente que se adapta a las necesidades particulares del contexto posibilitando que recursos y capacidades del sistema se distribuyan uniformemente entre los distintos elementos del sistema. De este modo, se solventan los problemas que suelen producirse en sistemas centralizados, tales como los cuellos de botella o el acceso recurrente a recursos críticos. Además, la eficacia del sistema a la hora de recuperar, filtrar y coordinar información debe asegurar unos tiempos de respuesta mínimos. La cuantificación numérica de recursos hídricos a partir de una gran cantidad de datos es una tarea que requiere de mucho tiempo computacional siendo importante plantear una arquitectura modular basada en capas con capacidades de reorganización a los requerimientos computaciones en cada instante.

Las imágenes de satélite se obtienen por fuentes externas diferentes, lo que dificulta el desarrollo de algoritmos ya que las imágenes difieren en escala, resolución, tipo de sensor, orientación, calidad y condiciones de iluminación ambiental. Además de estas dificultades los recursos hídricos pueden tener estructuras complicadas y pueden estar ocultos por otros edificios o árboles.

6.2 Componentes de la arquitectura propuesta

Para lograr el objetivo de este trabajo de investigación, la detección y verificación automática de la legalidad de las piscinas construidas en espacios privados, es necesario disponer de una arquitectura con características bien definidas para que el sistema pueda funcionar correctamente.

Para ello se ha modelado una arquitectura basada en agentes virtuales donde cada uno de los agentes del sistema trabaja individualmente para conseguir un objetivo común. Una de las principales necesidades que han llevado al diseño del sistema utilizando esta arquitectura ha sido el diseño de un sistema distribuido que realice las diferentes tareas de extracción de imágenes entrenamiento de modelos y clasificación de forma escalable.

En la arquitectura planteada en este trabajo, los agentes se organizan en organizaciones virtuales que componen el sistema. En el diseño del problema particular, se dispone de un módulo dedicado a la adquisición de imágenes satelitales, que una vez obtenidas, son procesadas en un segundo módulo que sirve para la detección de recursos hídricos y la extracción de sus coordenadas. La información resultante es enviada al módulo de validación cuya función es contrastar la localización de los recursos hídricos localizados en el módulo de detección con los registros oficiales. Adicionalmente, la arquitectura dispone un cuarto módulo que constituye la interfaz de la aplicación. La interconexión de los cuatro módulos se realiza mediante PANGEA, la característica esencial de esta arquitectura basada en agentes virtuales es que los agentes se replican en función de la demanda computacional sin afectar al resto del sistema, asegurando la robustez de este.

En las siguientes subsecciones se profundizará en cada módulo, así como en los agentes que los conforman.

6.2.1 Adquisición de imágenes

Esta organización virtual es responsable de obtener las imágenes de satélite que otros módulos usaran para detectar y validar los recursos hídricos. Está formado por los siguientes agentes Maps Service, Tiles Selector y Tiles Downloader.

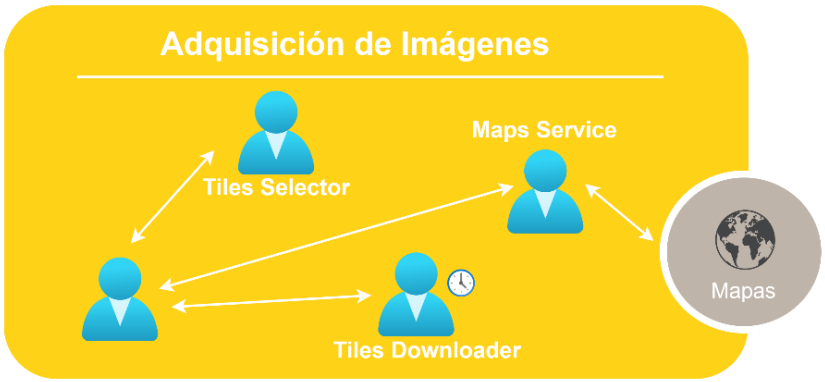


Fig 28 Módulo Adquisición de Imágenes

- Maps Service (Servicio de Mapas): El servicio de mapas es una plataforma en línea que ofrece mapas geográficos y datos de ubicación a los usuarios. Esta plataforma puede admitir imágenes provenientes de diversas fuentes externas. Una de las características destacadas de esta organización es su capacidad para utilizar diversas fuentes cartográficas para descargar imágenes de forma local. Los usuarios pueden acceder a mapas detallados y datos geoespaciales a través de esta plataforma, lo que permite explorar ubicaciones, obtener direcciones, visualizar información geográfica y mucho más.
- Tiles Selector (Selector de Baldosas): Este agente es responsable de calcular y seleccionar las subáreas (fragmentos) específicas de los mapas que deben descargarse. Opera en función de las zonas preseleccionadas por los usuarios del sistema. Este agente emplea algoritmos de cálculo y determina qué partes del mapa son relevantes para la descarga y luego inicia el proceso de obtención de datos.
- Tiles Downloader (Descargador de Baldosas): El Tiles Selector tiene la función de administrar y programar las descargas de las subáreas de mapas. Puede establecer un horario de descarga según la periodicidad indicada, lo que permite actualizar regularmente los datos cartográficos en la plataforma. Este agente garantiza que las áreas cartográficas seleccionadas por el selector se descarguen de manera eficiente y precisa, manteniendo actualizada la información geoespacial disponible para los usuarios.

En conjunto, estos agentes trabajan para proporcionar a los usuarios acceso a mapas actualizados y detallados a través del servicio de mapas. El selector de zonas asegura que se descarguen las partes relevantes del mapa, mientras que el descargador de zonas se

encarga de programar las descargas de manera periódica, manteniendo la información geoespacial al día y brindando una experiencia de usuario efectiva y precisa.

6.2.2 Interfaz de Aplicación

Este organismo es el que permite que la información generada por el sistema sea comprensible por los humanos. Esta organización sirve de interfaz entre el sistema y las diferentes aplicaciones que ofrece el sistema como servicios a usuarios finales. Las aplicaciones que tienen acceso a esta organización pueden acceder o generar datos en el sistema. En este caso, el agente humano con el sistema puede definir qué zonas inspeccionar y revisar los resultados de detección, las alertas o los informes.

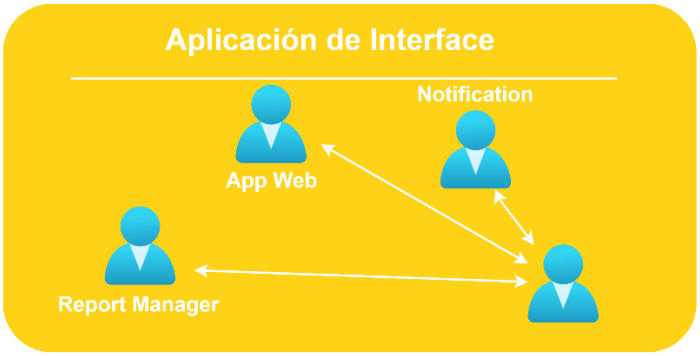


Fig 29 Módulo de Interface

- Notification Agent (Agente de Notificación): Este agente se encarga de gestionar y enviar notificaciones a los usuarios en un sistema o aplicación. Las notificaciones pueden ser de diversos tipos, como alertas, recordatorios, mensajes informativos, actualizaciones de estado, entre otros. El agente de notificación se asegura de que los usuarios reciban la información relevante en tiempo real o según una programación específica. Puede integrarse con diferentes canales de comunicación, como correo electrónico, mensajes de texto, notificaciones push en dispositivos móviles, entre otros. El agente de notificación es esencial para mantener a los usuarios informados y comprometidos con la aplicación.
- Web App Agent (Agente de Aplicación Web): El agente de aplicación web se refiere a los componentes que funcionan en segundo plano para mantener y

gestionar el funcionamiento de una aplicación cliente-servidor. Puede ser responsable de tareas como el procesamiento de datos, la gestión de usuarios, la autenticación y autorización, la interacción con bases de datos, la entrega de contenido dinámico, la seguridad, la administración de sesiones, entre otras. En esencia, el agente de aplicación web es el motor que permite que la aplicación funcione de manera fluida y eficiente, brindando a los usuarios una experiencia interactiva y en tiempo real.

- Report Manager Agent (Agente de Gestión de Informes): El agente de gestión de informes se encarga de administrar y generar informes dentro de un sistema o aplicación. Este agente puede ser parte de un sistema más grande de generación de informes, donde los usuarios pueden solicitar y personalizar informes específicos según sus necesidades. El agente de gestión de informes se comunica con diversas fuentes de datos, extrae información relevante y la presenta en formatos comprensibles, como gráficos, tablas y resúmenes. Este agente es útil para tomar decisiones informadas basadas en datos y para analizar el rendimiento o los patrones dentro de la aplicación o sistema.

6.2.3 Procesado de imagen

Esta organización tiene por objeto realizar tareas relacionadas con el procesamiento de imágenes. Para ello cuenta con la colaboración de los siguientes agentes:

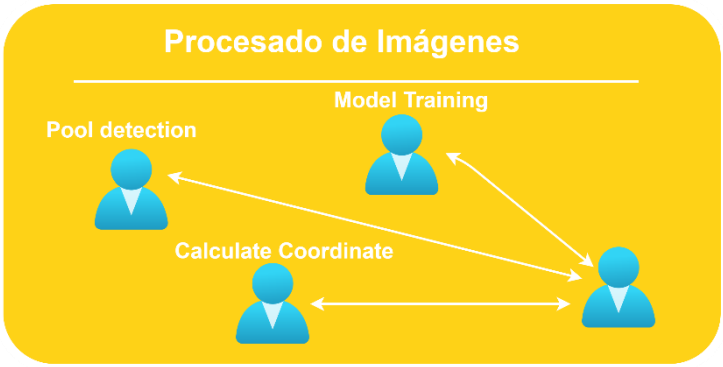


Fig 30 Procesado de imágenes

- Agente de Entrenamiento de Modelos: Este agente es responsable del entrenamiento inicial y del reentrenamiento continuo de los modelos de

detección de objetos en imágenes utilizados en el sistema. Se encarga de procesar nuevas imágenes etiquetadas por el usuario, incorporar estas imágenes etiquetadas a los datos de entrenamiento y ajustar los modelos para mejorar su precisión y capacidad de detección. El agente de entrenamiento de modelos puede usar técnicas de aprendizaje automático y procesamiento de imágenes con objeto de mejorar constantemente la calidad de los modelos empleados en la detección y clasificación.

- Agente Pool Detection: El agente de detección de recursos hídricos (Pool Detection) se dedica a identificar nuevas imágenes que requieren clasificación. Este agente monitorea constantemente la llegada de nuevas imágenes al sistema, y cuando detecta una imagen nueva que necesita ser procesada, utiliza los modelos preentrenados para clasificar y etiquetar las piscinas en las imágenes. Puede aplicar técnicas de procesamiento de imágenes y análisis para detectar y localizar con precisión las piscinas en las imágenes.
- Agente de Cálculo de Coordenadas Globales: Este agente tiene la función principal de convertir las coordenadas locales de los recursos hídricos detectadas por el agente Pool Detection en coordenadas globales en el sistema. Esto implica considerar el contexto espacial y el nivel de zoom de imagen. El agente calcula las coordenadas globales al tener en cuenta las coordenadas relativas al recurso hídrico presente en una imagen particular y luego las ajusta para que se correspondan con las coordenadas geográficas en el sistema global.

En conjunto, estos agentes trabajan en colaboración para realizar el procesamiento completo de la detección, clasificación y ubicación de recursos hídricos en imágenes. Cada agente desempeña un papel clave en el flujo de trabajo y contribuye a la precisión y la funcionalidad general del sistema de detección y análisis de piscinas en el contexto de la arquitectura multiagentes.

6.2.4 Validar Recurso hídrico

Esta organización tiene como objetivo detectar qué piscinas están legalmente registradas en las bases locales de la administración. Para ello, esta organización utiliza un Webservice gubernamental para obtener los datos asociados a cada una de las parcelas. La información devuelta por el servicio indica si los recursos hídricos están registrados o su presencia puede ser considerada como ilegal.

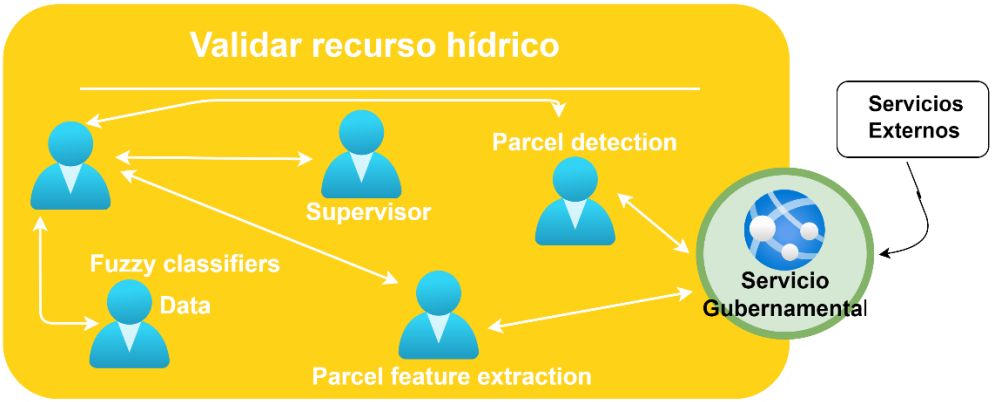


Fig 31 Módulo de validación de recursos hídricos

- Agente de Detección de Parcelas, utiliza servicios externos para detectar parcelas y asignar un identificador único a cada una. Su objetivo principal es identificar y etiquetar las parcelas presentes en las imágenes.
- Supervisor: es una entidad que monitorea y coordina las actividades de los demás agentes y se encarga de la gestión general del sistema, asegurando que todos los componentes funcionen correctamente y se cumplan los objetivos.
- Agente de Extracción de Características, este agente se encarga de obtener información específica sobre las parcelas detectadas, en particular, si un recurso hídrico ya está registrado de forma jurídica en la parcela.
- Agente de Datos Clasificadores Difusos, agente que tiene la tarea de detectar si los recursos hídricos se han localizado de manera duplicada en baldosas adyacentes o a una distancia mínima. Utiliza técnicas de clasificación difusa para realizar esta tarea, lo que significa que puede manejar información ambigua o incierta y tomar decisiones basadas en grados de pertenencia.

6.2.5 Organización del sistema multiagente Pangea

El principal hito de esta organización, que se compone de los agentes mínimos necesarios para que PANGEA sea ejecute de forma ordinaria, es llevar a cabo las tareas de organización de las organizaciones virtuales y la comunicación entre los agentes responsables de cada organización. A continuación, puede ver los agentes que forman parte de esta organización:

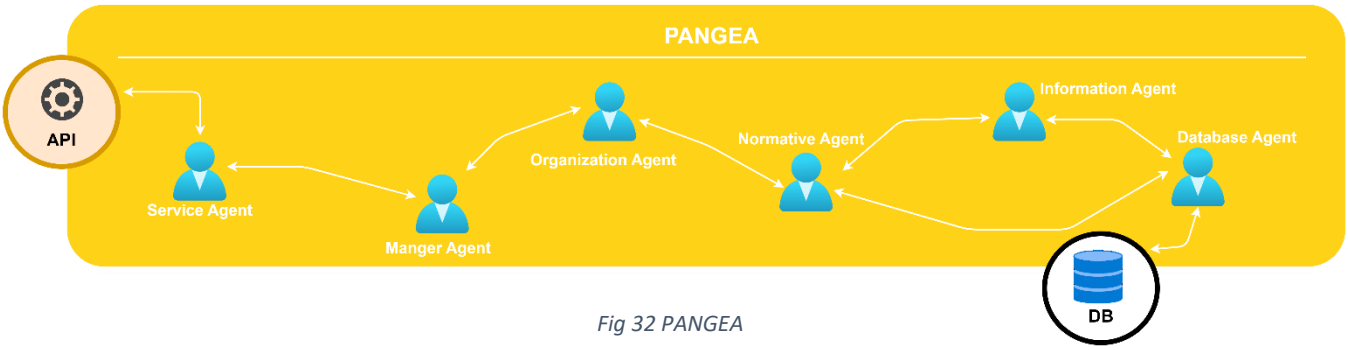


Fig 32 PANGEA

- Agente de Servicio, agente que expone funcionalidad a través de servicios web como interfaz de comunicación entre los agentes externos de la organización y los de la propia organización. Esta interfaz permite la creación de agentes independientes del lenguaje de programación o de ejecución para que pueda ser ofrecido a las diferentes aplicaciones de usuario.
- Agente Gestor, es el responsable de comprobar periódicamente el estado del sistema, detectando sobrecargas y posibles fallos que puedan producirse en los agentes de cada una de las organizaciones.
- Agente de Organización, este agente se encarga de verificar las operaciones de las organizaciones virtuales, garantizando la seguridad y el balanceo de carga. Este agente también proporciona servicios de cifrado.
- Agente normativo, Se encarga de hacer cumplir el cumplimiento de las normas en las comunicaciones entre agentes.
- Agente de Base de Datos, este agente es el único agente de la organización que tiene permisos de acceso a la base de datos. Es el encargado de almacenar la información de estado del sistema, analizar la persistencia y consistencia de los datos capacidades.
- Agente de Información, se encarga de gestionar los servicios disponibles dentro de las organizaciones virtuales, indicando qué servicios están disponibles para

cada uno de los agentes. Cuando un agente se incorpora al sistema, debe indicar qué servicios ofrece al resto de elementos de la arquitectura. De este modo, cuando otro agente solicite un servicio, debe consultar a este agente para saber qué entidad se encargada de ofrecerlo.

Para el correcto funcionamiento y escalabilidad de la arquitectura, el sistema utiliza diferentes bases de datos de forma distribuida; El sistema utiliza la base de datos dentro de la organización PANGEA para almacenar la información del sistema, que está compuesta por los agentes del sistema, los servicios que presta cada uno y las tareas que puede realizar cada agente. Además, el sistema cuenta con una base de datos adicional que se utiliza para almacenar la información específica del caso de estudio, las zonas seleccionadas para la inspección y las piscinas detectadas.

Una de las ventajas que ofrece esta arquitectura es la búsqueda simple de servicios, donde un agente externo puede buscar y ejecutar un servicio ofrecido por un agente dentro de la arquitectura. Para ello, el agente externo debe enviar un mensaje al Agente Gestor, indicando el servicio requerido con los parámetros necesarios para dicho servicio. En colaboración con el resto de los agentes de la organización PANGEA, Agente de Organización, Agente de y Agente de Base de Datos, este agente responde con una lista de agentes disponibles que pueden realizar el servicio solicitado.

Finalmente, el agente que va a realizar la tarea debe aceptar la propuesta para realizar la tarea. La Figura 33 muestra un ejemplo de solicitud del servicio de extracción de características de un paquete por parte de un agente externo.

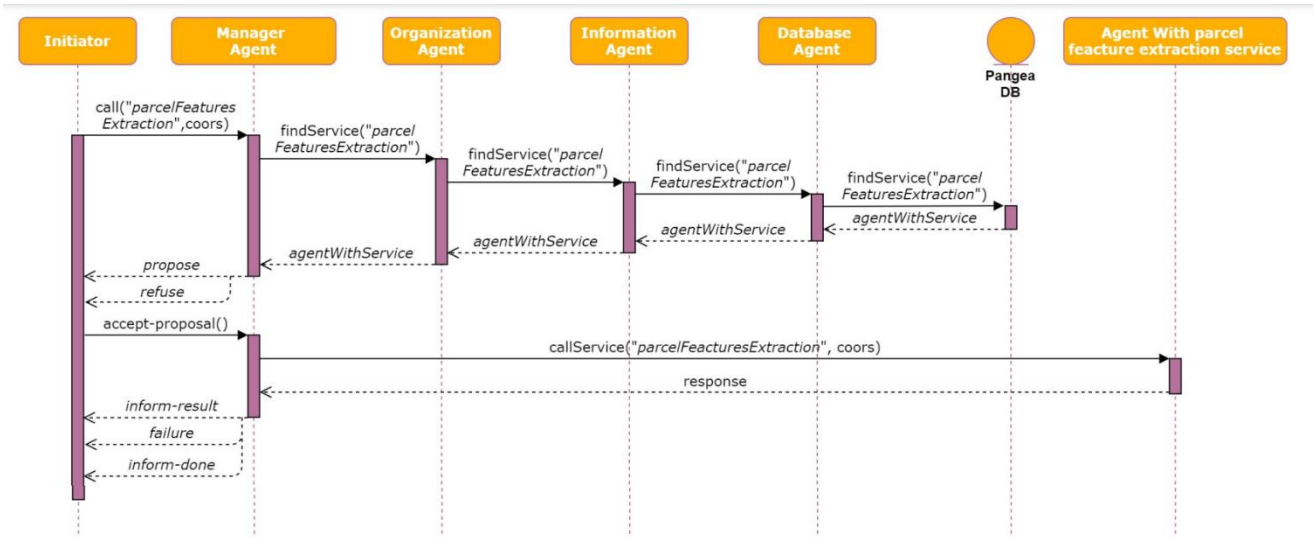


Fig 33 Diagrama de secuencia del sistema

CAPÍTULO VII

CASO DE ESTUDIO

VNiVERSiDAD
D SALAMANCA

Una vez desarrollada la arquitectura propuesta en el capítulo VI en el presente capítulo se detalla el caso de estudio práctico, desarrollando tres bloques principales interconectados mediante la arquitectura multiagente PANGEA.

7 Caso de estudio

El principal reto de este trabajo es diseñar una plataforma capaz de detectar automáticamente piscinas ilegales mediante una arquitectura distribuida y que aumente la eficiencia de los procedimientos que son llevados a cabo en la actualidad y de una forma totalmente manual (Sánchez San Blas et al, 2023). Este proceso requiere fuentes de datos actualizadas con frecuencia y con un coste razonable. Debido a ello se emplea el sistema propuesto en el capítulo 6 que permite la modelización de problemas complejos mediante la utilización de sistemas multiagente. A continuación, se muestra una imagen de las diferentes organizaciones virtuales que conforman el sistema diseñado en este caso de estudio.

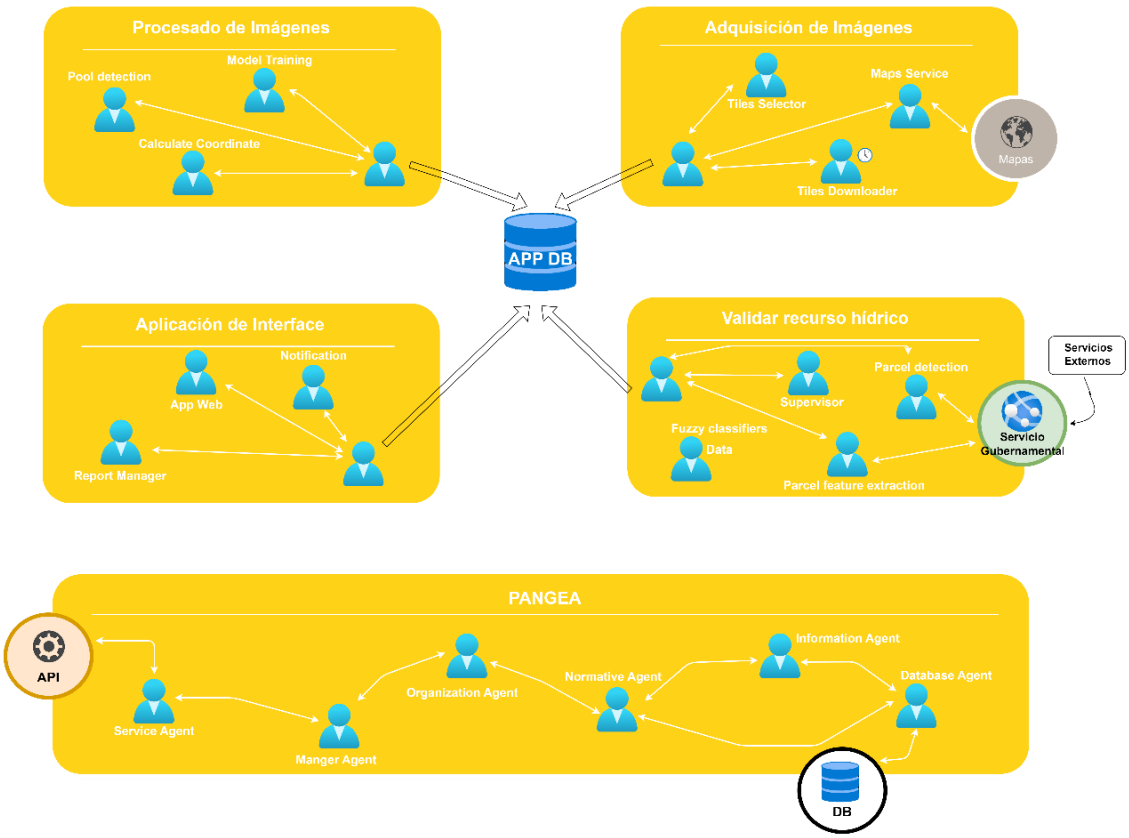


Fig 34 Arquitectura propuesta

Los métodos empleados por las administraciones locales que verifican si las piscinas están autorizadas o no son totalmente precarios y realizándose a mano lo que implica mucha demora de tiempo. Es por ello por lo que es necesario incorporar un sistema de adquisición de imágenes georreferenciados que sea sencillo de utilizar por parte de los trabajadores de las administraciones públicas.

Teniendo en cuenta que la principal fuente de datos obtenidos son imágenes satelitales, y tras una revisión detallada de la literatura, los sistemas que obtienen los mejores resultados en la clasificación y detección de objetos en imágenes son los algoritmos de Deep Learning. El uso de estos algoritmos es crucial en el desarrollo de esta tarea ya que permite la detección de zonas donde el agua está presente en la imagen. Sin esta capacidad de detección, el sistema automático propuesto no sería posible.

En esta sección presenta el caso práctico basado en la técnica de clasificación de imágenes de subbloques. La Figura 35A muestra un ejemplo de una imagen con un zoom 18, y la Figura 35B muestra un ejemplo con zoom 19.

Fig 35 Ejemplo de Imágenes para la detección: A Zoom18, B Zoom19.

La solución propuesta tiene tres bloques principales, el bloque de generación de imágenes de satélite, el bloque de detección y clasificación de las piscinas a partir de las imágenes generadas, y, por último, la comprobación de su legalidad en las bases de datos

gubernamentales. Además, para la interconexión de las partes, se utiliza la arquitectura multiagente PANGEA.

A continuación, se procede a describir los pasos de la solución propuesta, en primer lugar, se comienza explicando el desarrollo del sistema de búsqueda de imágenes en los mapas, posteriormente, se continua presentando los algoritmos y métodos de clasificación utilizados, a ello se añade el conjunto de datos formado por las imágenes y sus anotaciones, seguido de los algoritmos y finalmente, se termina describiendo el sistema que permite comprobar si una piscina está legalmente registrada por el organismo competente.

7.1.1 Bloque de generación de imágenes

Es necesario disponer de una base de datos que cubra un área extensa y que contenga fotos aéreas para detectar piscinas. Los usuarios pueden construir dicha base de datos con herramientas privadas o públicas y utilizar sistemas de aeronaves.

El bloque propuesto para la generación de imágenes es interoperable y tiene la posibilidad de obtener datos de diferentes proveedores, como como Bing Maps, Google Maps, OpenStreetMaps, ESRI World Imagery, Wikimedia Maps , NASA GIB S , Carto Light , Stamen Toner B and W y el Sentinel . Estos proveedores de imágenes satelitales cubren una gran parte de la cartografía a nivel mundial y su acceso es gratuito en la mayoría de los casos.

En el presente trabajo de investigación, el usuario determina una zona en el mapa para posteriormente inspeccionar la zona. En esta herramienta, es posible configurar, por un lado, el zoom de las imágenes, y por otro, la fuente de datos del mapa, como se muestra en la Figura 36a. A continuación, el sistema genera una cuadrícula con pequeños fragmentos de mapa (tiles) que llenan toda la zona dibujada. La figura 36b ilustra el proceso de transformación del área seleccionada en los mosaicos correspondientes.

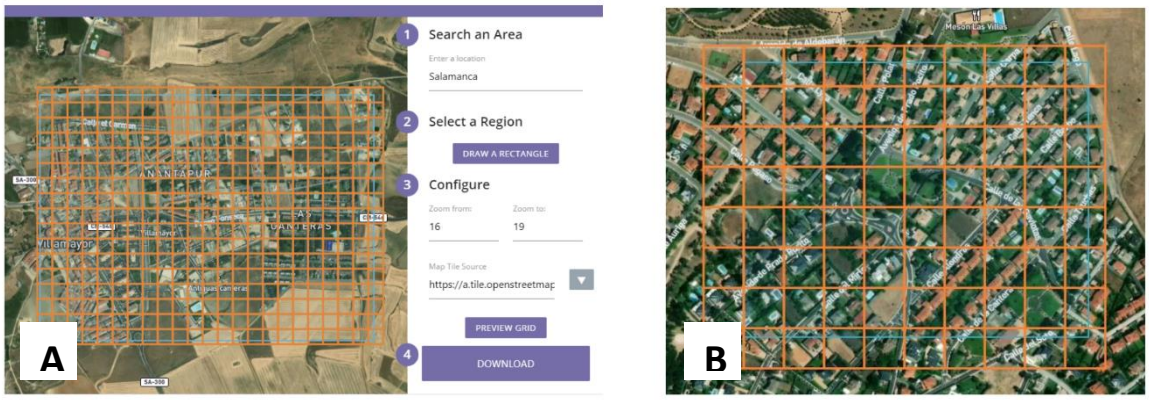


Fig 36 Imagen del bloque degeneración de imágenes.

Los algoritmos utilizados tienen en común que las fases de preprocesamiento de cada una de las imágenes son iguales para cada uno de los métodos aplicados. Todo el preprocesamiento se realiza de forma idéntica y con los mismos parámetros en sus etapas iniciales. Cabe destacar, el proceso de transformación de la imagen a escala de grises y la subdivisión de la imagen en N bloques de igual tamaño.

Posteriormente, para cada uno de estos bloques, se realiza el tratamiento de la imagen para extraer los descriptores de textura. Los descriptores de textura se refieren a la información sobre la disposición espacial del color o las intensidades en una imagen. Los vectores de características, formados a partir de los descriptores de textura, se normalizan y se utilizan en el entrenamiento de la clase. Posteriormente, se normalizan y se utilizan para entrenar a los clasificadores. El lenguaje de programación utilizado para el desarrollo ha sido Python 3.

En particular las bibliotecas de procesamiento de imágenes empleadas durante la ejecución de este caso de estudio han sido: OpenCV, skimage y mahotas. Las librerías de aprendizaje automático que han sido utilizadas son, scikit-learn, TensorFlow, keras y imbalanced-learn.

7.1.2 Bloque de detección y clasificación de las piscinas a partir de las imágenes

El conjunto de datos utilizado para el entrenamiento es un conjunto de datos propio, que ha sido ofrecido a la comunidad científica de manera abierta y gratuita.

El sistema de generación de imágenes explicado en la sección 7.1.1 se ha utilizado para construirlo. Para entrenar el modelo, se obtuvieron 999 imágenes de 512x512 píxeles con un zoom de 18, de la zona geográfica Redlands CA, alrededor de Prospect Park ya que era una zona con alta concentración de piscinas.

El conjunto de imágenes creado consta de una única clase, Piscinas. El conjunto de imágenes tiene 782 imágenes etiquetadas y 219 imágenes que no contienen piscina. En total, hay 2300 etiquetas de la clase piscinas.

La figura 37 muestra un ejemplo de conjunto de etiquetado con la herramienta Roboflow.

Fig 37 Example of labeling a training image.

Para dividir el conjunto de datos se utilizan dos conjuntos (Tabla 3): el conjunto de entrenamiento (80%) con 799 imágenes y 1892 etiquetas, y el conjunto de validación (20%) con 200 imágenes y 408 etiquetas.

Tabla 3 Números de piscinas en cada set de imágenes.

SET	IMÁGENES	PISCINAS
Entrenamiento	799	1892
Validación	200	408
todas	999	2300

El formato de almacenamiento de la información del cuadro delimitador para las notaciones o el etiquetado de este conjunto de datos es diferente para cada modelo de detección de objetos. Se emplean tres conjuntos de datos diferentes con objeto de poder verificar los resultados.

1. **Yolov4**

Para el entrenamiento de la red neuronal YoloV4, se ha utilizado el dispositivo de hardware Jetson Xavier AGX., empleando el modo de anotación Yolo Darknet en formato TXT. Se debe tener en cuenta que se han establecido para trabajar con imágenes de 512x512 píxeles y un máximo de 6000.

2. **MaskRCNN**

Durante el entrenamiento con el algoritmo MaskRCNN se ha utilizado un cuaderno dentro de Google Colab "Train Mask-RCNN Model on Custom Data.". El algoritmo utiliza el formato de anotación Pascal VOC (Everingham, 2015) en formato de archivo xml. Los parámetros de entrenamiento establecidos fueron 150 epochs y una confianza de detección mínima del 70%.

3. **Detectron2**

Finalmente, para el entrenamiento del algoritmo Detectron2, se ha utilizado Google Colab con el libro de trabajo publicado en "Detectron2 Beginner's Tutorial.". Detectron2 utiliza el formato de anotación COCO (Lin, 2014) en formato JSON. Cabe destacar que, para este algoritmo, se pueden incluir anotaciones de segmentación para cada etiqueta. Con objeto de mantener cierta equidad de condiciones con los algoritmos anteriores, no se han incluido anotaciones de segmentación excepto las anotaciones del cuadro delimitador. Este método hace que el algoritmo utilice sólo Faster-RCNN, cuyo archivo de configuración básica se puede encontrar en Detectron2, "faster_rcnn_X_101_32x8d_FPN_3x.". Además, fijamos la tasa de aprendizaje a 0,01 con un máximo de 50.000 iteraciones y pasos en 30.000, 40.000 y 45.000.

7.1.3 Bloque de comprobación del registro legal

Para el registro de propiedades a nivel legal, algunas organizaciones u organismos gubernamentales regionales con competencias pretenden garantizar la seguridad jurídica de las operaciones realizadas en el mercado inmobiliario mediante la creación de catastros.

Se ha de tener en cuenta que la existencia de piscinas afecta al pago del impuesto de bienes inmuebles (IBI), ya que, al igual que otro tipo de construcciones e instalaciones, añade valor añadido a la propiedad. El nombre y la correspondencia de los organismos responsables de garantizar el registro de las obras nuevas varían de un país a otro.

Por ejemplo, en EE.UU., la responsabilidad de estos registros corresponde a los municipios. En España cambio, existe un organismo encargado de esta competencia llamado Dirección General del Catastro. En ambos casos, estos organismos ofrecen un servicio de comprobación de los edificios registrados para un punto determinado en el mapa de coordenadas utilizado en los servicios, como Google Maps.

Por tanto, se propone que el sistema obtenga las coordenadas correspondientes a estas piscinas tras obtener las piscinas detectadas en las imágenes. De este modo, el sistema comprobará si el inmueble correspondiente a las coordenadas obtenidas tiene registrada la piscina detectada.

En este caso, el sistema realiza esta comprobación a través de una petición a un endpoint de la API ofrecida por los servicios de la organización afiliada. A partir de los datos obtenidos como respuesta, será posible comprobar las construcciones registradas, determinando si la piscina está registrada o no.

CAPÍTULO VIII

RESULTADOS

VNiVERSiDAD
D SALAMANCA

En este capítulo se realiza una evaluación de los resultados obtenidos. En primer lugar, se analiza la eficiencia, para un conjunto de evaluación, de los diferentes algoritmos empleados en el bloque de detección y clasificación de las piscinas a partir de las imágenes. Una vez vistos los resultados que obtienen cada uno de los algoritmos será necesario determinar cuál de ellos presenta mejores métricas, para usarlo con el conjunto de prueba.

8 Resultados

Las herramientas utilizadas para el entrenamiento de los modelos disponen de métodos para calcular algunas métricas utilizando el conjunto de validación. Sin embargo, utilizamos un nuevo proceso de evaluación de los algoritmos sin depender de estas herramientas y evaluar algunas cuestiones nuevas de forma particular. Este método consiste en un nuevo conjunto, el conjunto de evaluación, que contiene imágenes representativas del sistema final. Además de comparar los distintos modelos entrenados con los tres algoritmos diferentes, el método compara imágenes ampliadas con dos niveles de zoom diferentes.

8.1 Métricas

Para la evaluación cuantitativa, se han utilizado las siguientes métricas: precisión, es decir, la relación entre TP y las suma de TP junto FP (1); recall, que es la probabilidad de que una imagen se clasifique como positiva y la relación entre los TP y los TP junto con los FNs (2); y F1, que es combinación de las dos métricas anteriores (3).

$$Precision = \frac{TP}{TP + FP} \qquad (1)$$

$$Recall = \frac{TP}{TP + FN} \qquad (2)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \qquad (3)$$

Posteriormente se define la velocidad medida en fotogramas por segundo (FPS); la mAP, calculada por la curva de precisión y recuerdo; y la IoU, que es el área de solapamiento entre el área encontrada en la imagen y el área detectada.

8.2 Conjunto de evaluación

Para la evaluación de los algoritmos, se establece un conjunto de imágenes. Se han elegido ocho localizaciones en la provincia de Salamanca (España), obteniendo una imagen zoom 18 y una imagen zoom 19 para cada localidad. En total 16 imágenes. La Figura 38 muestra una imagen zoom18 de un punto y la Figura 39 muestra una imagen zoom19, con su con sus respectivos datos etiquetados.

Fig 38 Imagen de evaluación zoom 18.

Fig 39 Imagen de evaluación zoom 19.

Las piscinas suelen tener un tamaño diminuto en la imagen, lo que provoca que los expertos humanos no sean capaces de clasificar si son piscinas o no de una forma sencilla. Este conjunto de evaluación contiene en total 85 imágenes. El conjunto se divide en dos grupos 50 para el grupo de imágenes ampliadas a zoom 18 y 35 para el grupo de imágenes ampliadas a zoom 19 (Tabla 4).

Tabla 4 Número de piscinas en cada set de imágenes de evaluación.

SET	IMÁGENES	PISCINAS
Zoom 18	8	50
Zoom 19	8	35
todas	16	85

8.2.1 Yolov4

Como puede verse en la Figura 40, el modelo entrenado con la red neuronal red neuronal YoloV4 consiguió detectar 365 piscinas correctamente y 56 detecciones como falsos positivos utilizando el conjunto de validación. Este modelo ha dado un 89,75% de mAP@0.50, 87% de precisión, 89% de recall y 88% de F1-Score. Estos resultados proceden de un umbral de confianza del 25%.

```
detections_count = 820, unique_truth_count = 408
class_id = 0, name = Piscinas, ap = 89.76%       (TP = 365, FP = 56)

for conf_thresh = 0.25, precision = 0.87, recall = 0.89, F1-score = 0.88
for conf_thresh = 0.25, TP = 365, FP = 56, FN = 43, average IoU = 61.39 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.897555, or 89.76 %
Total Detection Time: 13 Seconds
```

Fig 40 . Validación de resultados con YoloV4

En las Figuras 41 y 42 se puede observar la detección en imágenes del conjunto de evaluación con zoom 18 y zoom 19, respectivamente.

Fig 42 YoloV4 zoom 18.

Fig 41 YoloV4 zoom 19.

En estos ejemplos de detección se utiliza un umbral de confianza del 10%, ya que ha dado los mejores resultados entre los umbrales de confianza 70%, 50%, 25% y 10%. El modelo entrenado con la imagen con zoom 18 ha logrado detectar 8 piscinas de 9, un falso positivo, y una detección descartada, ya que no se sabe si se trata de una piscina o no. En

cuanto a la imagen con zoom 19 ha detectado 5 piscinas de 5 y una detección descartada por el mismo motivo.

La Tabla 5 muestra los resultados obtenidos al utilizar el conjunto de evaluación para la detección con el modelo entrenado en YoloV4 con un umbral de confianza del 10%. Las nomenclaturas utilizadas en la tabla son: TP = Verdadero Positivo, FP = falso positivo, FN = falso negativo, GT = verdad sobre el terreno o total del conjunto de verdades, Prec = Precisión, RC = Recuperación y F1 = Puntuación F1.

Tabla 5 Valores de YoloV4 para las métricas de evaluación

Set	TP	FP	FN	GT	Prec	RC	F1
Z18	43	2	7	50	95.6%	86.0%	90.5%
Z19	33	0	2	35	100%	94.3%	97.1%
All	76	2	9	85	97.4%	89.4%	93.3%

8.2.2 MaskRCNN

El entrenamiento con MaskRCNN devuelve resultados de todos los entrenamientos para cada epoch. Este algoritmo permite evaluar cada modelo resultante de cada epoch individualmente. En este caso, se han elegido los modelos entrenados cuyos resultados destacan en algunas de las métricas devueltas por la herramienta de entrenamiento con el conjunto de validación, como mAP@0.5, precisión, recall y F1-Score. Entre ellos, destaca el modelo resultante de la epoch 46 que ofreció los mejores resultados. Este modelo entrenado destaca respecto a los demás modelos resultantes por su alta precisión, cuyo valor alcanzó el 86,3%. En la Figura 43 se observan los valores de las métricas que ha tenido el modelo epoch 46 con el conjunto de validación. Este modelo ha detectado correctamente 297 piscinas y 47 falsos positivos. Ha alcanzado un mAP@0.5 del 73,54%, una recuperación del 72,79% y una puntuación F1 del 78.98%.

```
==============] - 90s 691ms/step - loss: 0.5755 - val_loss: 1.2065

ference model (last checkpoint of the train model)

map: 0.7354 TP: 297 FP: 47 FN: 111 total: 408 precision: 0.8633 recall: 0.7279 F1: 0.7898
```

Fig 43 Validación de resultados para el modelo entrenado con MaskRCNN

En las Figuras 44 y 45 se puede observar las detecciones en las imágenes del conjunto de evaluación con zoom 18 y zoom 19, respectivamente. En estas detecciones se utiliza un umbral de confianza del 90% porque da mejores resultados y permite filtrar más falsos positivos. El modelo ha detectado 7 piscinas de 9 en la imagen zoom 18 y 4 de 5 en la imagen zoom 19.

Fig 45 MaskRCNN zoom 18.

Fig 44 MaskRCNN zoom 19.

La Tabla 6 muestra los resultados obtenidos al utilizar el conjunto de evaluación para la detección de piscinas con el modelo epoch 46 entrenado en MaskRCNN con un umbral de confianza del 90%.

Tabla 6 Valores de MaskRCNN para las métricas de evaluación

Set	TP	FP	FN	GT	Prec	RC	F1
Z18	29	0	21	50	100%	58.0%	73.4%
Z19	29	2	6	35	93.5%	82.9%	87.9%
All	58	2	27	85	96.7%	68.2%	80.0%

8.2.3 Detectron2

Finalmente, el modelo entrenado con Detectron2 ha obtenido un AP@0.5 del 87,23% con el conjunto de validación, como se puede ver en la Figura 46. En este caso, la herramienta utilizada para el entrenamiento con Detectron2 también calcula el AP@0.5:0.95 con un resultado de 35,06% y el AP@0.75 con un 16,43%.

```
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.440
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.302
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.457
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
[09/02 03:44:51 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl  |
|:------:|:------:|:------:|:------:|:------:|:-----:|
| 35.056 | 87.238 | 16.431 | 20.691 | 36.411 |  nan  |
[09/02 03:44:51 d2.evaluation.coco_evaluation]: Some metrics cannot be computed and is shown as NaN.
[09/02 03:44:51 d2.engine.defaults]: Evaluation results for my_dataset_val in csv format:
[09/02 03:44:51 d2.evaluation.testing]: copypaste: Task: bbox
[09/02 03:44:51 d2.evaluation.testing]: copypaste: AP,AP50,AP75,APs,APm,APl
[09/02 03:44:51 d2.evaluation.testing]: copypaste: 35.0561,87.2379,16.4311,20.6906,36.4108,nan
```

Fig 46 Validación de resultados con Detectron2

Las figuras 47 y 48 muestran las detecciones en las imágenes del conjunto de evaluación con zoom 18 y zoom 19, respectivamente.

Fig 47 Detectron2 zoom 18

Fig 48 Detectron2 zoom 19.

Estas detecciones utilizan un umbral de confianza del 50%. El modelo entrenado en modelo entrenado detecta 7 charcos de 9 en la imagen con zoom 18 y 4 de 5 en la imagen con zoom 19.

La Tabla 7 muestra los resultados obtenidos al utilizar el conjunto de evaluación para la detección de piscinas con el modelo entrenado con Detectron2 con un umbral de confianza del 50%.

Tabla 7 Valores para Detectron2 métricas de evaluación

Set	TP	FP	FN	GT	Prec	RC	F1
Z18	28	5	22	50	84.8%	56.0%	67.5%
Z19	30	3	5	35	90.9%	85.7%	88.2%
All	58	8	27	85	87.9%	68.2%	76.8%

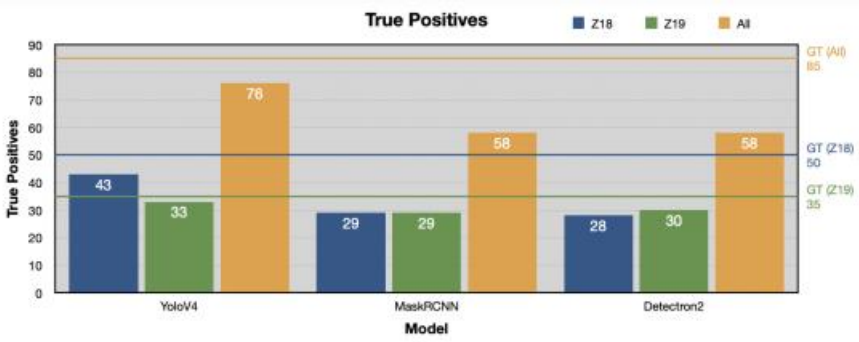
8.2.4 Comparación

Una vez obtenidas las métricas de cada modelo entrenado con el conjunto de evaluación, se compararán los algoritmos. Las Figuras 49a, 49b y 49c muestran gráficos que comparan los modelos con los valores de verdaderos positivos, falsos positivos y falsos negativos, respectivamente. El modelo entrenado con la red neuronal YoloV4 dio resultados mucho mejores que los otros dos algoritmos, ya que, consiguió detectar correctamente muchas más piscinas y, además, sólo detectó 2 piscinas incorrectas. En cambio, si utilizamos un umbral de confianza del 25% o más con el modelo entrenado con YoloV4 conseguimos que el modelo no detecte piscinas incorrectamente. Con el umbral del 25%, se obtienen 61 verdaderos positivos y 0 falsos positivos en todo el conjunto.

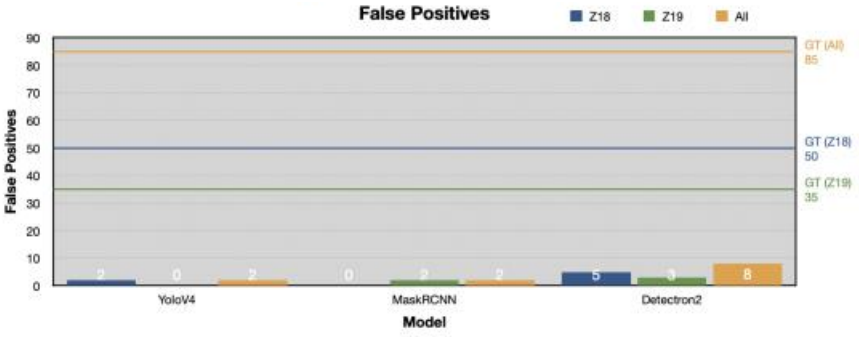
Por último, si comparamos entre el conjunto de imágenes zoom 18 (Z18) con el conjunto de imágenes zoom 19 (Z19) en los tres algoritmos, hay una mayor detección de verdaderos positivos con el conjunto Z19 si lo comparamos con el total de conjuntos existentes (GT).

El algoritmo YoloV4 ha sido capaz de detectar 33 piscinas de 35 en el conjunto Z19, mientras que ha detectado 43 de 50 en el conjunto Z18. Esta diferencia es aún más marcada en los otros dos algoritmos, donde el número de verdaderos positivos es casi

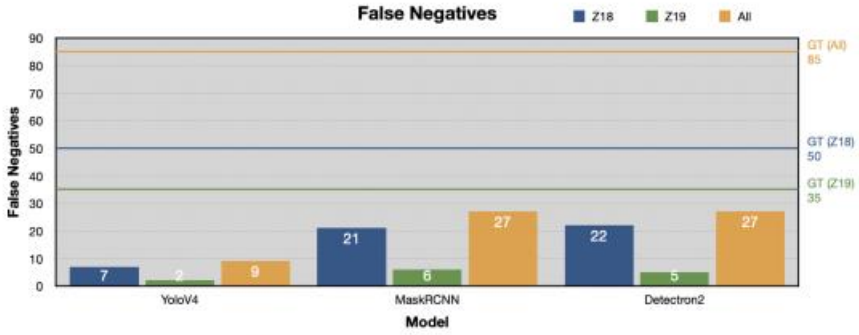
idéntico entre el conjunto Z18 y el conjunto Z19. Sin embargo, hay muchos más verdaderos positivos en Z18 que en Z19.



(a) Comparación TP



(b) Comparación FP



(c) Comparación FN

Fig 49 Comparación entre modelos

Si comparamos la métrica de precisión de cada modelo (Figura 50) el modelo entrenado con YoloV4 es más que los otros dos algoritmos. Además, si aumentamos su umbral de confianza al 25 %, se alcanza una precisión del 100 % en los tres conjuntos: Z18, Z19 y Todos.

Por último, la precisión es menor con el conjunto Z18 en los tres algoritmos en comparación con el conjunto Z19.

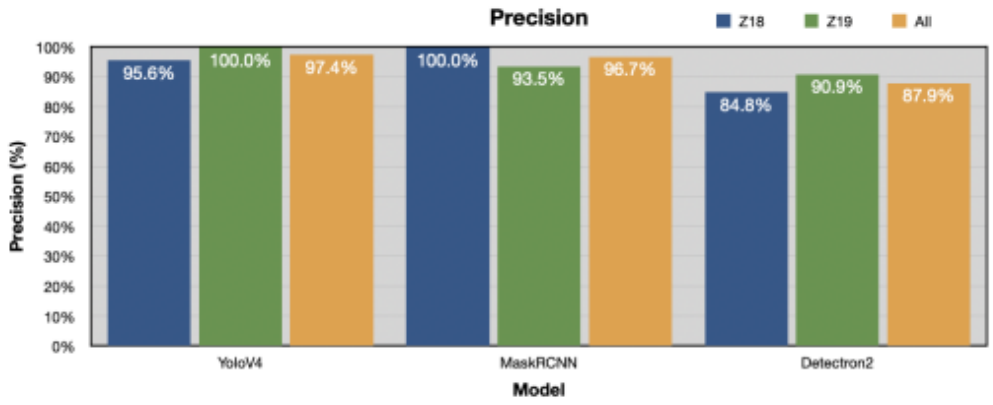


Fig 50 Comparación Precisión

En cuanto a la métrica "recall", el algoritmo YoloV4 ha arrojado mejores resultados, alcanzando el 94,3% en el conjunto Z19. Además, con el conjunto Z19 consiguen una mayor recuperación que con el conjunto Z18, con una diferencia de casi el 30% en el caso del conjunto Z18, con una diferencia de casi el 30% en el caso del algoritmo Detectron2.

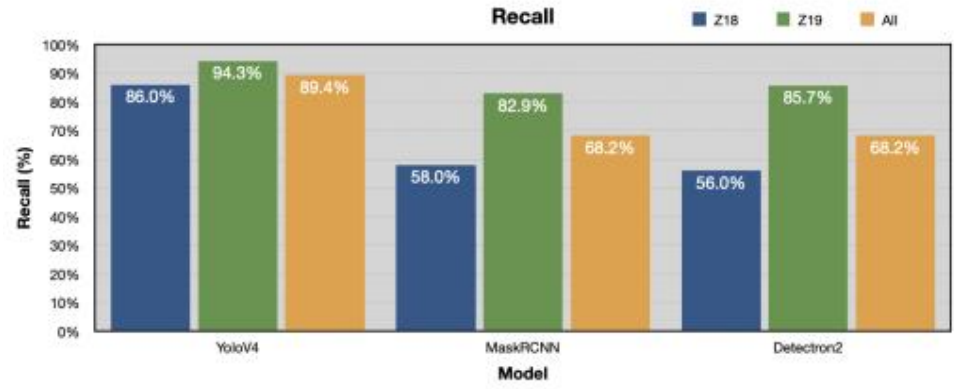


Fig 51. Comparación Recall

Por último, en lo que respecta a la métrica F1-Score, que combina las métricas de precisión y recuerdo, Figura 52, el algoritmo YoloV4 es muy superior a los otros dos algoritmos, alcanzando hasta un 97,1% con el conjunto Z19.

Por otra parte, el conjunto Z19 aporta mejores resultados en comparación con el conjunto Z18.

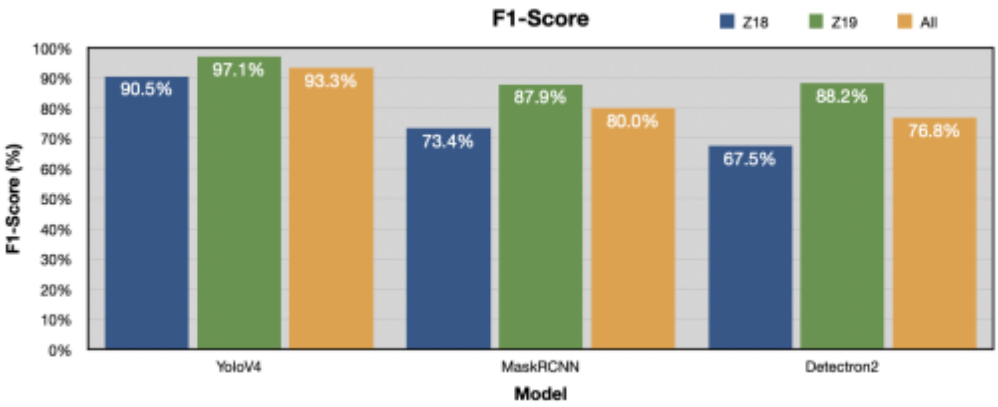


Fig 52 Comparación F1-Score

8.2.5 Caso de prueba del sistema de verificación

Antes de comentar el caso de prueba realizado para el sistema de verificación, se va a describir el mismo.

El caso de estudio se ha desarrollado en un pueblo de Castilla y León situado a 4Km la ciudad de Salamanca. El nombre de este pueblo es Villamayor, se localiza en el extremo meridional de la comarca de La Armuña. según los datos publicados por el Instituto Nacional de Estadística a 1 de enero de 2022 el número de habitantes en esta localidad es de 7.400.

El servicio de la Dirección General del Catastro de esta localidad verifica la información y comprueba los inmuebles y parcelas, ya que la Dirección General del Catastro es el organismo encargado de controlar y gestionar los registros de edificios en España. Se han utilizados los datos del catastro de esta localidad para realizar una comprobación entre las piscinas detectadas por el sistema y aquellas que se encuentran registradas. Para ello el sistema propuesto ha recogido las imágenes y las ha comprobado pasando por el subsistema de detección de piscinas. Se han obtenido las coordenadas de las piscinas detectadas en las imágenes ya que estas se encuentran georeferenciadas. Las coordenadas obtenidas se han comparado de forma automática con las coordenadas de las piscinas que aparecen en el catastro.

El resultado obtenido de este proceso se encuentra en la Figura 53.

Fig 53 Resultados del Sistema de verificación.

La figura 53 muestra tres tipos de puntos que se distinguen por colores:

- Azul: Este punto identifica una piscina detectada por el sistema y que el registro no tiene constancia.

- Verde: Este punto identifica una piscina detectada en el sistema y que fue verificada como registrada en el catastro a través de sus coordenadas geográficas.

- Rojo: Este punto identifica una piscina detectada en el sistema y que no tiene registro asociado para la propiedad correspondiente con sus coordenadas geográficas.

En el pueblo observado de Villamayor, como se observa en la Figura 24, existen en realidad 27 piscinas. De esas 27 piscinas, el sistema ha podido detectar 23 piscinas y no ha podido detectar 5 realmente existentes. De esas 23 piscinas detectadas, 22 de ellas son correctas, sin embargo, una de ellas es incorrecta. De piscinas, 22 están registradas y vinculadas a la propiedad, una no está registrada y otra está pendiente de verificación.

CAPÍTULO IX

CONCLUSIONES

VNiVERSiDAD
D SALAMANCA

A continuación, se describen las principales conclusiones obtenidas tras la consecución satisfactoria de los objetivos marcados en el desarrollo de esta tesis.

9 Conclusiones

Este trabajo pone en valor el potencial que presentan los SMA con agentes virtuales y la detección de recursos hídricos a través de técnicas de aprendizaje profundo. A medida que las últimas décadas se han caracterizado por una rápida evolución tecnológica, la importancia y relevancia de los SMA con agentes virtuales se han destacado en múltiples disciplinas. Durante la ejecución de este trabajo se ha verificado que el uso de CNN es un método válido para la detección de objetos presentes en imágenes que en combinación con los sistemas multiagente permiten la automatización en la detección y validación de recursos hídricos a partir de imágenes de satélite, lo que facilita de la gestión y control de los recursos hídricos de manera automática.

En este contexto, el uso de SMA demuestra ser una solución poderosa para abordar problemas complejos y dinámicos. La capacidad de los agentes virtuales para colaborar, comunicarse y tomar decisiones autónomas ha impulsado la eficiencia y efectividad en la resolución de desafíos que anteriormente resultaban difíciles de abordar de manera individual mediante programas simples y monolíticos. La adaptabilidad inherente de los SMA ha permitido su aplicación en campos diversos, desde la gestión de recursos hasta la planificación logística y la detección de anomalías. La integración de técnicas de aprendizaje profundo en este marco ha marcado un punto de inflexión en la comprensión y resolución de problemas complejos. El aprendizaje profundo ha demostrado su capacidad para analizar grandes cantidades de datos y extraer patrones, mejorando la capacidad predictiva y la calidad de las decisiones. La combinación de estas dos áreas ha dado lugar a sistemas de detección más precisos, capaces de identificar y clasificar objetos con gran precisión. En el transcurso de este trabajo, se ha explorado en profundidad la aplicación de SMA en la detección de recursos hídricos, en imágenes satelitales y de drones. La utilización de la plataforma PANGEA como base para el desarrollo de esta arquitectura ha demostrado ser un enfoque sólido y eficaz para abordar esta compleja tarea.

La arquitectura multiagente implementada ha demostrado ser una solución prometedora para la detección de recursos hídricos. La habilidad de los agentes virtuales para colaborar y comunicarse entre sí, aprovechando la información contextual de la imagen, ha sido crucial para la obtención de resultados precisos y confiables. La

computación distribuida inherente a esta aproximación ha permitido una mayor eficiencia en el proceso de detección y análisis.

La elección de algoritmos de aprendizaje profundo, en particular Mask-RCNN, Detectron2 y YOLOv4, ha sido estratégica y demuestra ser eficaz en la detección de embalses de agua y piscinas en imágenes. El enfoque de detección y segmentación proporcionado por estos algoritmos ha permitido identificar y delinear de manera precisa los recursos hídricos presentes en las imágenes. La superioridad del modelo entrenado con YOLOv4, evidenciada en las métricas de evaluación, ha fortalecido la elección de esta red neuronal para la detección.

Por último, se realiza un caso de prueba para comprobar una población concreta para comprobar el funcionamiento del sistema. De esta forma, a partir de la prueba realizada en la población de Villamayor, se ha verificado que la plataforma funciona de forma exitosa. Esta prueba ha permitido certificar la eficacia del sistema para determinar qué piscinas están registradas o se encuentran no declaradas en los organismos oficiales correspondientes.

El caso de prueba en la población de Villamayor ha sido fundamental para verificar la efectividad y utilidad del sistema en un entorno del mundo real. La capacidad de la arquitectura para corroborar la información sobre piscinas registradas en organismos oficiales refuerza su potencial aplicabilidad en contextos prácticos, demostrando que es posible determinar la presencia de una piscina en una imagen con una precisión superior al 97% utilizando una arquitectura multiagente que permite la computación distribuida y que permite la evaluación de diferentes algoritmos para diferentes procesos de detección de una forma transparente al usuario.

Tras evaluar los algoritmos y compararlos, destacamos el modelo entrenado con la red neuronal YoloV4 que ofrece mejores resultados en todas las métricas. Tras la verificación de resultados, se propone utilizar este algoritmo para la detección de piscinas especificando un umbral de confianza del 10% en caso de que los errores o falsos positivos puedan ser asimilados, permitiendo una mayor detección o recuerdo, o utilizar un umbral de confianza del 25% en caso de que se busque una mayor precisión en la detección de piscinas.

Adicionalmente, para la detección, es mejor utilizar imágenes con zoom 19 frente al zoom 18. Aunque para las imágenes con zoom 19, es necesario procesar cuatro veces más imágenes para la misma área que con imágenes con zoom 18, es muy conveniente sacrificar más recursos computacionales en ares de la detección. Además, en este caso de

estudio, no es vital mostrar las detecciones en tiempo real, por lo que gastar unos segundos extra en el proceso de detección no es una preocupación.

La exploración de los umbrales de confianza ha añadido un nivel adicional de flexibilidad al sistema, permitiendo ajustar la precisión y el recuerdo según los requisitos específicos de la aplicación. La elección entre un umbral del 10% para una mayor detección o un umbral del 25% para una mayor precisión ha abierto la puerta a un análisis más detallado de los resultados en función de las necesidades del usuario y el contexto.

Un aspecto de vital importancia que ha emergido de este estudio es la contribución del sistema propuesto en la verificación de la detección automática de piscinas y la validación de los registros existentes. La capacidad del sistema para confrontar la información detectada con los registros oficiales de piscinas en las entidades pertinentes ha resultado ser una herramienta valiosa en la gestión y control de los recursos hídricos. La detección de piscinas y la posterior comparación con registros oficiales representan un desafío crucial en la administración de estos recursos. La discrepancia entre la información proporcionada por los registros y la realidad del terreno puede generar ineficiencias y errores en la gestión, impactando negativamente en la toma de decisiones y en la planificación de recursos. El sistema desarrollado se erige como una solución efectiva para abordar esta discrepancia, al permitir la verificación cruzada y precisa de los datos.

La habilidad del sistema para identificar automáticamente piscinas y someterlas a un proceso de verificación se traduce en una mejora sustancial en la calidad de la información. El análisis comparativo entre la detección automática y los registros oficiales brinda una doble capa de seguridad y confianza en la información, permitiendo una toma de decisiones informada y precisa. Además, la eficiencia y rapidez con la que el sistema realiza esta comparación se traduce en ahorro de tiempo y recursos, en contraste con los métodos tradicionales de verificación manual.

La utilidad del sistema no se limita únicamente a la verificación de piscinas, sino que sienta las bases para su aplicación en otros contextos y recursos. La capacidad de confrontar la información detectada con registros oficiales puede ser extrapolada a otros escenarios, como la detección de otros tipos de objetos o la verificación de características geográficas. Esto amplía aún más la relevancia y aplicabilidad de la arquitectura multiagente con agentes virtuales desarrollada en este estudio.

En resumen, este estudio ha dejado en claro que la convergencia de SMA y algoritmos de aprendizaje profundo tiene un gran potencial en la detección de recursos hídricos. Los resultados exitosos obtenidos demuestran la capacidad de esta arquitectura para mejorar significativamente la precisión y eficiencia en la identificación de charcos y piscinas. Las

reflexiones técnicas aquí presentadas no solo contribuyen al avance de la investigación en esta área, sino que también abren la puerta a nuevas direcciones y desafíos que requieren una atención continua y una investigación más profunda. El trabajo realizado sienta las bases para futuras mejoras y desarrollos en la detección de recursos hídricos mediante SMA y técnicas de aprendizaje profundo, promoviendo la innovación y el progreso en este campo en constante evolución

CAPÍTULO X

REFERENCIAS

VNiVERSiDAD
D SALAMANCA

10 Referencias

Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). En *arXiv [cs.NE]*. http://arxiv.org/abs/1803.08375

Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). Neural Codes for Image Retrieval. En *Computer Vision – ECCV 2014* (pp. 584–599). Springer International Publishing.

Bellifemine, F., Poggi, A., & Rimassa, G. (1999). JADE - A FIPA-compliant agent framework. En *Proceedings of the Practical Applications of Intelligent Agents*.

Bengio, I., Delalleau, O., & Le Roux, N. (2005). "'e curse of highly variable functions for local kernel machines. En *Proceedings of the Advances in Neural Information Processing Systems* (Vol. 18, pp. 107–114).

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. En *Advances in Neural Information Processing Systems 19* (pp. 153–160). The MIT Press.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural probabilistic language models. En *Innovations in Machine Learning* (pp. 137–186). Springer-Verlag.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. En *arXiv [cs.CV]*. http://arxiv.org/abs/2004.10934

Bordini, R. H., Hübner, J. F., & Wooldridge, M. (2007). *Programming Multi-Agent Systems in Agent Speak Using Jason*. John Wiley & Sons, Ltd.

Bordini, R. H., Hübner, J. F., & Vieira, R. (2005). Jason and the golden fleece of agent-oriented programming. En *Multi-Agent Programming* (pp. 3–37). Springer US.

Bottou, L., & Bousquet, O. (2007). The tradeoffs of large-scale learning. *Proc. Advances in Neural Information Processing Systems*, *20*, 161–168.

Braubach, L., Pokahr, A., & Lamersdorf, W. (2004). Jadex: A Short Overview. *Procceding Main Conference Net. Object Days*, 195–207.

Busetta, P., Rönnquist, R., Hodgson, A., & Lucas, A. (1998). JACK Intelligent Agents - Components for Intelligent Agents in Java. *Agent Oriented Software Pty. Ltd*.

Carolyn J. Merry. (2000). Role of Technology in the Future of Water Resources Remote Sensing Developments. Water Resources IMPACT, 2(5), 13–14.

Chen, S.-H., Wang, C.-W., Tai, I.-H., Weng, K.-P., Chen, Y.-H., & Hsieh, K.-S. (2021). Modified YOLOv4-DenseNet algorithm for detection of ventricular septal defects in ultrasound images. *International Journal of Interactive Multimedia and Artificial Intelligence*, *6*(7), 101. https://doi.org/10.9781/ijimai.2021.06.001

Ciresçan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. En *Proc. of 22nd Intl. Joint Conf. on Artificial Intelligence* (pp. 1237–1242).

Dahl, G. (2010). Phone recognition with the mean-covariance restricted Boltzmann machine. En *Proceedings of NIPS* (pp. 469–477).

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process*, *20*, 33–42.

Dale, J., Knottenbelt, J., Labo, F.: April Agent Platform (2011), http://designstudio.lookin.at/research/relate%20survey/ Survey%20Agent%20Platform/April%20Agent%20Platform.html (accessed November 29, 2011)

De Paz, J. F., Zato, C., Villarrubia, G., Bajo, J., & Corchado, J. M. (2014). Distribution of roles in virtual organization of agents. En *The 8th International Conference on Knowledge Management in Organizations* (pp. 485–497). Springer Netherlands.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Deng, L. (2010). Binary coding of speech spectrograms using a deep auto-encoder. En *Proceedings of INTERSPEECH* (pp. 1692–1695).

Doll, P., Girshick, R., & Ai, F. (2018). Mask R-CNN.

Domozi, Z., & Molnar, A. (2019). Surveying private pools in suburban areas with neural network based on drone photos. *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*.

Doran, J., Franklin, S., Jenkins, N. R., & Norman, T. J. (1996). On cooperation in multi-agent systems". En *UK Workshop on Foundations of Multi-agent Systems*.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136. https://doi.org/10.1007/s11263-014-0733-5

Felleman, D. J., & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, *1*(1), 1–47. https://doi.org/10.1093/cercor/1.1.1-a

Ferber, J., Gutknecht, O., & Michel, F. (2004). From agents to organizations: An organizational view of multi-agent systems. En *Agent-Oriented Software Engineering IV* (pp. 214–230). Springer Berlin Heidelberg.

Ferner, C., Eibl, G., Unterweger, A., Burkhart, S., & Wegenkittl, S. (2019). Pool detection from smart metering data with convolutional neural networks. *Energy Informatics*, *2*(S1). https://doi.org/10.1186/s42162-019-0097-8

Fikes, R. (1991). *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning* (E. Sandewall, Ed.). Morgan Kaufmann publishers, Inc.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the Grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, *15*(3), 200–222. https://doi.org/10.1177/109434200101500302

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional Single Shot Detector. En *arXiv [cs.CV]*. http://arxiv.org/abs/1701.06659

Galindo, C., Moreno, P., & Gonz, J. (2009). *SWIMMING POOLS LOCALIZATION IN COLOUR HIGH-RESOLUTION SATELLITE IMAGES Dept. of System Engineering and Automation*. 510–513.

Gaud, N., Galland, S., Hilaire, V., & Koukam, A. (2009). An Organisational Platform for Holonic and Multiagent Systems. En *Lecture Notes in Computer Science* (pp. 104–119). Springer Berlin Heidelberg.

Geneserech, M. R., & Kecthpel, S. P. (1994). Software Agents. *Communications of ACM*, *37*, 48–53.

Giret, A., Julián, V., Rebollo, M., Argente, E., Carrascosa, C., & Botti, V. (2010). An open architecture for service-oriented virtual organizations. En *Lecture Notes in Computer Science* (pp. 118–132). Springer Berlin Heidelberg.

Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., & He, K. (2018). Detectron. https://github.com/facebookresearch/detectron.

Gleick, P. H. (1998). Water in crisis: Paths to sustainable water use. Ecological Applications, 8(3), 571-579.

González-Briones, A., Villarrubia, G., De Paz, J. F., & Corchado, J. M. (2018). A multi-agent system for the classification of gender and age from images. *Computer Vision and Image Understanding: CVIU*, *172*, 98–106. https://doi.org/10.1016/j.cviu.2018.01.012

Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press.

Gray, P. C., Bierlich, K. C., Mantell, S. A., Friedlaender, A. S., Goldbogen, J. A., & Johnston, D. W. (2019). Drones and convolutional neural networks facilitate automated and accurate cetacean species identification and photogrammetry. *Methods in Ecology and Evolution*, *10*(9), 1490–1500. https://doi.org/10.1111/2041-210x.13246

Gutknecht, O., & Ferber, J. (1997). MadKit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems. *Technical Report R.R. LIRMM*, *9718*.

Habibie, M. I., Ahamed, T., Noguchi, R., & Matsushita, S. (2020). Deep Learning Algorithms to determine Drought prone Areas Using Remote Sensing and GIS. *2020 IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology (AGERS)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hihi, S., & Bengio, Y. (1995). Hierarchical recurrent neural networks for long-term dependencies. En D. Touretzky, M. C. Mozer, & M. Hasselmo (Eds.), *Advances in Neural Information Processing Systems* (Vol. 8). MIT Press.

Hinton, G. E. (2005). What kind of graphical model is the brain? En *Proc. 19th International Joint Conference on Artificial intelligence* (pp. 1765–1775).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006). A fast-learning algorithm for deep belief nets. Neural Comp. 18, 1527–1554

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. En *arXiv [cs.NE]*. http://arxiv.org/abs/1207.0580

Hochreiter, S. (1991). *Untersuchungen zu dynamischen neuronalen Netzen* (T. U. Münich, Ed.).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837

Hübner, J. F., Sichman, J. S., & Boissier, O. (2002). A model for the structural, functional, and deontic specification of organizations in multiagent systems. En *Advances in Artificial Intelligence* (pp. 118–128). Springer Berlin Heidelberg.

Hübner, J.F. (2007). J -Moise+ Programming organisational agents with Moise+ & Jason. Technical Fora Group at EUMAS 2007

Hübner, J.F., Bordini, R.H., Picard, G. (2009). Using Jason and MOISE+ to Develop a Team of Cowboys. In: Hindriks, K.V., Pokahr, A., Sardina, S. (eds.) ProMAS 2008. LNCS, vol. 5442, pp. 238–242. Springer, Heidelberg.

Hui, J., «mAP (mean Average Precision) for Object Detection», Medium, abr. 03, 2019. https://medium.com/@jonathan_hui/map-mean-average-precision-for-objectdetection-5c121a31173

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating deep network training by reducing internal covariate shift. En *arXiv [cs. LG]*. http://arxiv.org/abs/1502.03167

Jennings, N. R., & Wooldridge Najibi, M. (1997). G-CNN: An iterative grid-based object detector. En *Agent Technology: Foundataions, Applications, and Markets* (pp. 2369–2377). Springer.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. En *arXiv [cs.CV]*. http://arxiv.org/abs/1408.5093

Kavukcuoglu, K. (2010). Learning convolutional feature hierarchies for visual recognition. En *Proceedings of NIPS* (pp. 1090–1098).

Kavukcuoglu, K., Ranzato, M., Fergus, R., & LeCun, Y. (2009). Learning invariant features through topographic filter maps. *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Khanna, P., & Kondawar, V. K. (1991). Application of Remote Sensing Techniques for Environmental Impact Assessment. *Current Science*, *61*(3/4), 252–256.

Kim, M., Holt, J. B., Eisen, R. J., & Padgett, K. (2011). *Reisen Detection of swimming pools by geographic object-based image analysis to support west nile virus control efforts*. Photogrammetric Engineering and Remote Sensing, vol. 77, no. 11, pp. 103– 113, doi: 10.14358/pers.77.11.1169.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classifi- cation with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *7*, 1–9.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

LeCun, Y. (1985). *Une procédure d'apprentissage pour Réseau à seuil assymétrique in Cognitiva 85: a la Frontière de l'Intelligence Artificielle, des Sciences de la Connaissance et des Neurosciences 599–604*. 599–604.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE. Institute of Electrical and Electronics Engineers*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lima, B., Ferreira, L., & Moura, J. M. (2021). Helping to detect legal swimming pools with Deep Learning and Data Visualization. *Procedia Computer Science*, *181*, 1058–1065. https://doi.org/10.1016/j.procs.2021.01.301

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. En *Computer Vision – ECCV 2014* (pp. 740–755). Springer International Publishing.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. En *Computer Vision – ECCV 2016* (pp. 21–37). Springer International Publishing.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional net- works for semantic segmentation. En *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

López Barriuso, A. (2018). Sistemas multiagente para la integración de personas discapacitadas.

López-Latorre, M. A., & Neira, M. (2016). Influencia del cambio climático en la biología de Aedes aegypti (Diptera: Culicidae) mosquito transmisor de arbovirosis humanas. *Revista Ecuatoriana de Medicina y Ciencias Biológicas*, *37*(2). https://doi.org/10.26807/remcb.v37i2.2

Maess, P. (1995). Artificial life meets entertainment: Life like autonomous agents". *Communications of the ACM*, *38*(11), 108–114.

Martínez, D. P., López-Batista, V. F., de Paz Santana, J. F., Moreno-García, M. N., & García, F. (2023). Object detection through computer vision. En *Advances in Intelligent Systems and Computing* (pp. 122–130). Springer International Publishing.

Mccabe, F. G., & Clark, K. L. (1995). APRIL-Agent PRocess Interaction Language. En M. J. Wooldridge & N. R. Jennings (Eds.), *Proceedings of the Workshop on Agent Theories, Architectures, and Languages on Intelligent Agents (ECAI 1994)* (pp. 324–340). Springer.

Medium. (s/f). Medium. Recuperado el 22 de agosto de 2023, de https://medium.com/@jonathan_hui/map-mean-average-precision-for-objectdetection-5c121a31173

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. En *arXiv [cs.CL]*. http://arxiv.org/abs/1310.4546

Mizuno, K., Terachi, Y., Takagi, K., Izumi, S., Kawaguchi, H., & Yoshimoto, M. (2012). Architectural study of HOG feature extraction processor for real-time object detection. *2012 IEEE Workshop on Signal Processing Systems*.

Mohamed, A.-R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, *20*(1), 14–22. https://doi.org/10.1109/tasl.2011.2109382

MSB –Ministério da Saúde do Brasil. (2020). Ministério da Saúde lança campanha de combate ao Aedes aegypti. https://antigo.saude.gov.br/saude-de-a-z/combate-aoaedes. Acesso em: 4 jun 2021.

Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, *9*(4), 685–696. https://doi.org/10.1109/72.701181

Ngiam, J. (2011). Multimodal deep learning. En *Proceedings of ICML* (pp. 689–696).

Nagi, J., Ducatelle, F., Di Caro, G. A., Ciresan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*.

Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. *2015 IEEE International Conference on Computer Vision (ICCV)*.

Nwana, H. S., & Ndumu, D. T. (1990). A perpective on software agent research. *Knowledge Engineering Review*, *14*, 125–142.

O'Brien, P.D., Nicol, R.C. (1998). FIPA, Towards a Standard for Software Agents. BT Technology Journal 13(3), 51–59

Obrien, R. (1998). *An Overview of the Methodological Approach of Action Research." Faculty of Information Studies*.

OMS – Organização Mundial de Saúde. (2017). Keeping the vector out: housing improvements for vector control and sustainable development. https://apps.who. int/iris/handle/10665/259404. Acesso em: 4 jun 2021.

Oquab, M. (2014). Weakly supervised object recognition with convolutional neural networks. En *Proceedings of NIPS* (pp. 1–10).

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Parker, D.B. (1985) Learning-Logic: Casting the Cortex of the Human Brain in Silicon. *Technical Report Tr-47, Center for Computational Research in Economics and Management Science. MIT Cambridge, MA*.

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. En *arXiv [cs. LG]*. http://arxiv.org/abs/1211.5063

Passos, W., Silva, E., Netto, S., Martins, J., Costa, Y., Araujo, G., & Lima, A. (2020). Detecção de Potenciais Focos do Aedes aegypti em Vídeos Aéreos Usando Redes Neurais. *Anais de XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*.

Piccinini, P., Prati, A., & Cucchiara, R. (2012). Real-time object detection and localization with SIFT-based clustering. *Image and Vision Computing*, *30*(8), 573–587. https://doi.org/10.1016/j.imavis.2012.06.004

Pitts, W., & McCULLOCH, W. S. (1947). How we know universals, the perception of auditory and visual forms. *The Bulletin of Mathematical Biophysics*, *9*(3), 127–147. https://doi.org/10.1007/bf02478291

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. *Proc. 26th Annual International Conference on Machine Learning*, *873*.

Rampersad, H. (s/f). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Total Performance Scorecard*.

Ranzato, M., Poultney, C., Chopra, S., & Lecun, Y. (2006). Efficient learning of sparse representations with an energy-based model. *Proc. Advances in Neural Information Processing Systems*, *19*, 1137–1144.

Rao, A. S. (1996). Agent Speak(L): BDI agents speak out in a logical computable language. En J. Perram & W. Van De Velde (Eds.), *MAAMAW 1996. LNCS* (Vol. 1038, pp. 42–55). Springer.

Rao, A. S., & Georgeff, M. P. (1991). Modeling Rational Agents within a BDI-Architecture. En J. Allen, R. Fikes, & E. Sandewall (Eds.), *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. En *arXiv [cs.CV]*. http://arxiv.org/abs/1804.02767

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rodríguez-Cuenca, B., & Alonso, M. (2014). Semi-automatic detection of swimming pools from aerial high-resolution images and LIDAR data. *Remote Sensing*, *6*(4), 2628–2646. https://doi.org/10.3390/rs6042628

Rosenblatt, F. (1957). *The Perceptron - A Perceiving and Recognizing Automaton*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Russel, S., & Norvig, P. (1995). *Artificial intelligence - A modern approach*. Prentice Hall.

Sánchez-Alor Expósito, J. (2020). Evaluación de algoritmos de detección de objetos basados en deep learning para detección de incidencias en carreteras.

Sánchez San Blas, H., Carmona Balea, A., Sales, A., Augusto Silva, L., & Villarrubia González, G. (2023). A Platform for Swimming Pool Detection and Legal Verification Using a Multi-Agent System and Remote Image Sensing.

Scherer, D., Muller, A., & Behnke, S. (2010). Evaluation of pooling operationsin convolutional architectures for object recognition. En *Proc. of the Intl.Conf. on Artificial Neural Networks* (pp. 92–101).

Schölkopf, B., & Smola, A. J. (2003). A short introduction to learning with kernels. En *Advanced Lectures on Machine Learning* (pp. 41–64). Springer Berlin Heidelberg.

Selfridge, O. G. (1958). Pandemonium: a paradigm for learning in mechanisation of thought processes. En *Proc. Symposium on Mechanisation of Thought Processes* (pp. 513–526).

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). OverFeat: Integrated recognition, localization and detection using Convolutional Networks. En *arXiv [cs.CV]*. http://arxiv.org/abs/1312.6229

Sermanet, P., Kavukcuoglu, K., Chintala, S., & Lecun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. *2013 IEEE Conference on Computer Vision and Pattern Recognition*.

Shen, Z., Liu, Z., Li, J., Jiang, Y.-G., Chen, Y., & Xue, X. (2017). DSOD: Learning deeply supervised object detectors from scratch. *2017 IEEE International Conference on Computer Vision (ICCV)*.

Silva, L. A., Sanchez San Blas, H., Peral García, D., Sales Mendes, A., & Villarrubia González, G. (2020). An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images. *Sensors (Basel, Switzerland)*, *20*(21), 6205. https://doi.org/10.3390/s20216205

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. En *arXiv [cs.CV]*. http://arxiv.org/abs/1409.1556

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective". *Autonomous Robots*, *8*(3), 345–383.

Sutskever, I. (2012). *Training Recurrent Neural Networks*.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. En *Proc. 28th International Conference on Machine Learning* (pp. 1017–1024).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer.

Tien, D., Rudra, T., & Hope, A. B. (2007). Swimming pool identification from digital sensor imagery using SVM. *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*.

Tomè, D. (2016). Deep convolutional neural networks for pedestrian detection. Signal Process. *Image Commun*, *47*, 482–489.

Wadley, F. M. (1952). Probit Analysis: A Statistical Treatment of the Sigmoid Response Curve. 2nd ed. D. J. Finney. New York-London: Cambridge Univ. Press, 1952. 318 pp. $7.00. *Science (New York, N.Y.)*, *116*(3011), 286–287. https://doi.org/10.1126/science.116.3011.286

Wan, J. (2014). Deep learning for content-based image retrieval: A comprehensive study. En *Proceedings of ACM MM* (pp. 157–166).

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. [Online]. Available: https://github.com/facebookresearch/detectron2.

Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., & Xue, X. (2015). Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *Proceedings of the 23rd ACM international conference on Multimedia*.

Xiang, Y., Choi, W., Lin, Y., & Savarese, S. (2017). Subcategory-aware convolutional neural networks for object proposals and detection. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Yang, Z., & Nevatia, R. (2016). A multi-scale cascade fully convolutional network face detector. *2016 23rd International Conference on Pattern Recognition (ICPR)*.

Yoo, D., Park, S., Lee, J.-Y., Paek, A. S., & Kweon, I. S. (2015). AttentionNet: Aggregating weak directions for accurate object detection. *2015 IEEE International Conference on Computer Vision (ICCV)*.

Zato, C., Villarrubia, G., Sánchez, A., Barri, I., Rubión, E., Fernández, A., Rebate, C., Cabo, J. A., Álamos, T., Sanz, J., Seco, J., Bajo, J., & Corchado, J. M. (2012). PANGEA – platform for automatic coNstruction of orGanizations of intElligent agents. En *Advances in Intelligent and Soft Computing* (pp. 229–239). Springer Berlin Heidelberg.

Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Zhao, Z.-Q., Bian, H., Hu, D., Cheng, W., & Glotin, H. (2017). Pedestrian detection based on fast R-CNN and batch normalization. En *Intelligent Computing Theories and Application* (pp. 735–746). Springer International Publishing.

Zhao, Z.-Q., Xie, B.-J., Cheung, Y.-M., & Wu, X. (2015). Plant leaf identification via a growing convolution neural network with progressive sample learning. En *Computer Vision -- ACCV 2014* (pp. 348–361). Springer International Publishing.

CAPÍTULO XI

GLOSARIO DE SIGLAS

VNiVERSiDAD
D SALAMANCA

11 Glosario de siglas

AAP: April Agent Platform

ACL: Lista de control de acceso

AGR: Agente-Grupo-Rol

ANN: Redes Neuronales Artificiales

AP: Precisión Media

BB: Cuadro delimitador de predicción

BDI: Bases de datos interoperantes

BN: Batch normalization

CBR: Sistemas de razonamiento basado en casos

CIFAR: Instituto Canadiense de Estudios Avanzados

CNN o ConvNet: Red Neuronal Convucional

CRS: Sistema de Referencia de Coordenadas

DF: Directory facilitator

DSM: Modelo digital de superficie

DTM: Modelo digital de terreno

EFFIS: Sistema Europeo de Información sobre Incendios Forestales

ETL: Extract, Transform, Load

FAIR: Facebook AI Research

FC: Capas completamente conectadas

FIPA: Foundation for Intelligent Physical Agents

FN: Falsos negativos

FP: Falsos positivos

FPN: Fast Page Mode

FPS: Fotogramas por segundo

GEOBIA: Análisis Geográfico Basado en Objetos de Imagen

GPU: Unidades de procesamiento de gráficos

GT: Total de conjuntos existentes

HOG: Histograma de Gradientes Orientados

HPF: Filtro de paso alto

IA: Inteligencia Artificial

IBI: Impuesto de bienes inmuebles

IRC: Internet Relay Chat

JADE: Java Agent DEvelopment Framework

JASON: JavaScript Object Notation

LIDAR: Light Detection and Ranging

mAP: Promedio de Precisión en Múltiples Escalas

ML: Machine Learning

MLP: Perceptrón multicapa o red neuronal de propagación directa

nDSM: Modelo digital de superficie normalizado

NDWI: Índice de Diferencia Normalizada de Agua

OMS: Organización Mundial de la Salud

PANGEA: Platform for Automatic coNstruction of orGanizations of intElligent Agents

PCA: Análisis de componentes principales

PR: Curva de recuperación de precisión

ReLU: Unidad lineal rectificada

RF: Ajuste rectangular

RNN: Redes neuronales recurrentes

ROI: Métodos de regiones de interés

RoI: Región de interés

RPAS: Sistema de Aeronave Pilotada a Distancia

RPN: Red de Propuestas de Región

SGD: Descenso de gradiente estocástico

SIFT: Scale Invariant Feature Transform

SIG: Sistemas de Información Geográfica

SMA: Sistemas multiagente

SNN: Redes neuronales simuladas

SSD: Single Shot Detector

SVM: Máquina de vectores de soporte

TAG: Grafo de adyacencia de regiones

TNR: Tasa de verdaderos negativos.

TP: Verdaderos positivos

TPR: Tasa de verdaderos positivos

UAV: Vehículos aéreo no tripulado

VO: Organización de agentes virtuales

YOLO: You Only Look Once