UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

RECONOCIMIENTO DE PATRONES REPETITIVOS EN IMÁGENES DE MOTIVOS DE HERENCIA CULTURAL

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE DATOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

SEBASTIÁN ANDRÉS SEPÚLVEDA ALONSO

PROFESORES GUÍA:
BENJAMÍN BUSTOS CÁRDENAS

PROFESOR CO-GUÍA:
IVÁN SIPIRÁN MENDOZA

COMISIÓN:
VALENTIN CLEMENT BARRIERE

Este trabajo ha sido parcialmente financiado por ANID a través del

Proyecto FONDECYT Regular 1230448

SANTIAGO DE CHILE
2024

RESUMEN DE TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS DE DATOS
RESUMEN DE MEMORIA PARA OPTAR
AL TITULO DE INGENIERO CIVIL EN COMPUTACIÓN
POR: SEBASTIÁN ANDRÉS SEPÚLVEDA ALONSO
FECHA: 2024
PROF. GUÍA: BENJAMÍN BUSTOS CÁRDENAS

RECONOCIMIENTO DE PATRONES REPETITIVOS EN IMÁGENES DE MOTIVOS DE HERENCIA CULTURAL

En el análisis de objetos de herencia cultural uno de los procesos más importantes es la documentación que caracteriza al objeto, desde la procedencia y el tipo de material, hasta detalles como los patrones grabados sobre el objeto. Esta información es utilizada por arqueólogos, antropólogos y otros profesionales para realizar estudios de los objetos y de las culturas que los crearon.

Actualmente, debido al escaso estudio de herramientas de apoyo en la detección de estos objetos, las segmentaciones de patrones grabados sobre objetos de herencia cultural se realizan de manera manual.

El problema que se aborda en esta investigación involucra la detección y segmentación de patrones geométricos y no geométricos repetitivos sobre imágenes de motivos de herencia cultural, específicamente sobre imágenes de texturas de cerámica antigua. El objetivo es realizar la tarea de segmentación de manera automática, colaborando de esta manera en el proceso de recuperación de información manual que actualmente tienen que realizar los arqueólogos y a la vez contribuir con la implementación de nuevas herramientas que permitan realizar esta tarea con otras fuentes de datos o mejorando los resultados que se obtendrán.

Para lograr este objetivo se utiliza el conjunto de datos "*Repetitive Patterns on Textured 3D Surfaces*" [1] (RPT3DS), el cual contiene 81 texturas 2D de modelos 3D de diferentes artefactos de cerámica con motivos pintados sobre su superficie, las cuales pertenecen al Museo Josefina de Ramos, Lima, Perú. Cada patrón repetido del conjunto de imágenes fue segmentado de manera manual por arqueólogos por medio del uso del software compartido por los autores del trabajo.

En esta investigación se utilizaron modelos del estado del arte en detección y segmentación de objetos, los cuales fueron entrenados y evaluados sobre el conjunto de datos *RPT3DS* [1]. Los modelos utilizados involucran algoritmos tradicionales de detección de objetos y algoritmos basados en redes neuronales especializados en tareas de detección de objetos y segmentación. Los resultados mostraron que al realizar fine-tuning con el modelo pre-entrenado de YOLOv8 con el conjunto de datos de COCO [2] se obtiene sobre un 95 % de mAP y es posible detectar patrones que no fueron segmentados de manera manual por los arqueólogos.

Se espera que los resultados permitan realizar la tarea de segmentación en nuevas fuentes de datos de figuras arqueológicas de manera automática y de esta manera colaborar en el proceso de recuperación de información de patrones en este dominio de datos.

A mi amada familia y amigos.

Agradecimientos

Me gustaría expresar mis sinceros agradecimientos al profesor Benjamín Bustos quien ha tenido la confianza en mí para poder realizar este trabajo, además de ser un aporte importante en el desarrollo de mi camino académico, y al profesor Iván Sipirán, quien con sus consejos e ideas me permitieron avanzar sin problemas en los resultados obtenidos en este trabajo.

También me gustaría agradecer a Stefan Lengauer, Reinhold Preiner y Tobias Schreck, investigadores de la Graz University of Technology, Austria, quienes junto al profesor Benjamín Bustos e Iván Sipirán compartieron de manera pública el conjunto de datos utilizado en este trabajo "*Repetitive Patterns on Textured 3D Surfaces*". Estoy seguro de que en un futuro la digitalización y el uso de herramientas computacionales en el análisis de objetos de herencia cultural será un aporte importante en la preservación de la historia de la humanidad.

Este trabajo no hubiera sido posible sin el apoyo del grupo RELELA de la Universidad de Chile, quienes proporcionaron los equipos computacionales necesarios para poder realizar los experimentos y análisis de resultados.

Además, agradezco por el financiamiento de ANID FONDECYT Regular Nº 1230448, liderado por el profesor Benjamín Bustos e Iván Sipirán y de ANID - Millennium Science Initiative Program - Code ICN17_002, liderado por el profesor Benjamín Bustos.

Mi gratitud también se extiende a mis amigos y amigas, quienes siempre me han impulsado a seguir mis sueños, con quienes he compartido momentos inolvidables y quienes también han sabido entender mi ausencia en muchas ocasiones. Su compañía y apoyo han sido fundamentales para poder llegar a este punto de mi vida.

Finalmente agradecer a mi Padre celestial quien me ha dado la oportunidad de estudiar y desarrollarme como persona. A mi familia, quienes siempre han estado preocupando en mis avances y me han dado la fuerza para seguir adelante. Y a mis padres y hermanos, quienes han sido una fuente de inspiración y motivación para poder continuar aprendiendo y poder llegar a ser una mejor persona. Cada uno de ustedes ha contribuido de manera significativa en mi crecimiento académico, y por eso les estaré eternamente agradecido.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

En arqueología los objetos de herencia cultural son piezas de cualquier material fabricado o modificado por una persona en un espacio y momento determinado [3]. Durante el análisis de cualquier artefacto uno de los procesos más importantes es documentar toda la información posible que caracteriza al objeto, desde la procedencia y el tipo de material, hasta detalles como los grabados de letras o figuras que suelen formar patrones y se encuentran grabados sobre el objeto. Estas figuras son conocidos como motivos arqueológicos de herencia cultural [4].

La identificación de motivos y patrones en artefactos arqueológicos juega un rol primordial en la preservación del valor histórico del objeto, ya que permite recuperar información muy propensa a perderse por distintos factores como el deterioro del objeto o la exposición a condiciones ambientales desfavorables.

Actualmente, la tarea de detección de figuras de interés arqueológico se realiza de manera manual con la ayuda de un arqueólogo experto en la cultura a la que pertenece el objeto, quien mediante una inspección visual del contorno del artefacto puede identificar y documentar por medio de dibujos, imágenes y texto la forma de las figuras presentes, el significado que poseen y su relevancia [5]. Este largo proceso, además de ser propenso a errores humanos, genera un problema de escalabilidad debido a que la cantidad de datos a analizar aumenta a medida que se descubren nuevos artefactos.

Por otro lado, en visión computacional la detección y segmentación de objetos en imágenes es una tarea que ha sido ampliamente estudiada en diferentes dominios, como en imágenes médicas [6], en imágenes satelitales [7], entre otros, mostrando resultados prometedores en aquellas áreas. Sin embargo, la investigación en la detección de motivos arqueológicos en imágenes 2D de artefactos arqueológicos aún no ha tenido un gran auge debido a la poca cantidad de datos con anotaciones existentes para su libre uso.

En esta investigación se estudiarán diferentes técnicas de visión computacional para la detección y segmentación en patrones, enfocando el análisis en motivos arqueológicos. Se mostrará la aplicación de algoritmos tradicionales, de redes neuronales convolucionales [8] y de arquitecturas basadas en Vision Transformers (ViT) [9]. De esta manera, se busca entender la complejidad de la tarea y encontrar una técnica o modelo que permita la detección y segmentación de patrones arqueológicos de manera automática, con el fin de proporcionar una alternativa a la anotación manual de este tipo de imágenes.

1.1. Contexto

La herencia cultural se define como el legado de artefactos físicos (bienes culturales) y atributos intangibles de un grupo o sociedad que se considera que son heredados por generaciones sucesivas. Los bienes culturales son objetos de herencia cultural que tienen un valor histórico, artístico, científico, etnológico o antropológico [10].

Los artefactos físicos de herencia cultural varían desde objetos de uso diario como herramientas, armas, joyas, cerámica, monedas, hasta construcciones como edificios, templos, monumentos, entre otros. Estos objetos son utilizados por arqueólogos, antropólogos y otros profesionales para realizar estudios de sus características físicas y de las culturas que los crearon.

Cada artefacto arqueológico físico posee una serie de características que lo hacen único, como su forma, material, color, tamaño, entre otros. Sin embargo, existen artefactos que poseen características similares entre sí, como los motivos arqueológicos, los cuales son figuras geométricas o no geométricas que se encuentran grabadas sobre la superficie del objeto [4]. Este tipo de figuras puede repetirse a lo largo del objeto o puede ser repetido en otro tipo de artefacto, sin importar si posee un tamaño, forma u origen distinto.

Los motivos arqueológicos suelen ser encontrados en una diversa variedad de representaciones artísticas, como en pinturas, superficies arqueológicas, esculturas y especialmente en cerámica, en donde además se encuentra bastante documentación en la literatura al ser uno de los materiales que más diversidad y conservación posee. Por lo mismo, para el estudio del arte y de la historia de distintas culturas la cerámica juega un rol importante para expertos y arqueólogos.

Distintos avances en la tecnología y en la metodología de trabajo en museos han permitido almacenar digitalmente aquellos artefactos elaborados en cerámica, principalmente haciendo uso de escáneres especializados en lograr recuperar la textura y forma de los objetos y de esta manera obtener modelos en 3D que entregan mayor información del objeto que la imagen 2D del mismo. En los últimos años incluso, el uso de modelos de campos de radiación neuronal, o NERFs por sus siglas en inglés, han abierto una nueva forma de poder recopilar masivamente modelos 3D en este tipo de objetos y con un menor costo económico [11].

Una de las ventajas que entrega poder almacenar por medio de imágenes 3D los objetos arqueológicos es poder trasformar la textura en 3D del objeto a una imagen extendida en 2D, lo cual permite una mejor visualización de los motivos arqueológicos que se encuentran en el objeto, y por tanto detectar de manera más fidedigna los patrones que son visibles al ojo humano.

1.2. Objetivos

En la presente Sección se detallan los objetivos que se buscan alcanzar con la realización de esta investigación:

1.2.1. Objetivo general

Desarrollar y validar una herramienta computacional, mediante el uso de modelos del estado del arte en detección y segmentación de imágenes, que permita realizar la identificación automática de patrones de motivos de herencia cultural en imágenes de artefactos arqueológicos peruanos de herencia cultural. La herramienta busca realizar la tarea de detección de patrones de manera automática, proporcionando la mayor información posible sobre la similitud de los patrones encontrados en las imágenes de los artefactos.

1.2.2. Objetivos específicos

OE1. Realizar un estudio de los modelos del estado del arte en detección, segmentación de objetos en imágenes y su rendimiento en patrones de motivos de herencia cultural.

OE2. Obtener, mediante el uso del procesamiento de imágenes, una transformación del conjunto de datos RPT3DS [1] que permita realizar un flujo de trabajo estándar para lograr replicar los experimentos con otras fuentes de datos diferentes a la original.

OE3. Evaluar y comparar el rendimiento mediante la precisión de los modelos de detección y segmentación de objetos en imágenes sobre el conjunto de datos RPT3DS.

OE4. Identificar los principales desafíos en la detección de objetos en imágenes con motivos de herencia cultural utlizando ténicas de procesamiento de imágenes y modelos de redes neuronales para imágenes.

OE5. Proponer mejoras en el conjunto de datos RPT3DS para la detección de objetos en imágenes con motivos de herencia cultural.

1.3. Alcances y limitaciones

En la presente Sección se detallan los alcances y limitaciones que se consideran para la realización de esta investigación:

1.3.1. Alcances

A1. La investigación se enfoca en la detección y segmentación de patrones de motivos de herencia cultural en imágenes de artefactos arqueológicos peruanos de herencia cultural, utilizando técnicas de visión computacional y modelos de redes neuronales.

A2. El estudio abarca el análisis de imágenes 2D de artefactos de cerámica de la cultura peruana que pertenecen a las culturas Chancay y Lurin.

A3. Se espera que los resultados permitan realizar la tarea de detección y segmentación en nuevas fuentes de datos de figuras arqueológicas de manera automática y de esta manera colaborar en el proceso de recuperación de información de patrones en este dominio de datos.

A4. El proyecto es interdisciplinario, involucrando conocimientos de las áreas de arqueología, antropología, visión computacional y aprendizaje automático.

1.3.2. Limitaciones

L1. La investigación está limitada al conjunto de datos RPT3DS, el cual abarca solamente datos de Perú, sin abarcar otras culturas o países. Esto puede limitar la generalización de los resultados obtenidos en la investigación.

L2. La efectividad de los modelos de redes neuronales utilizados en la detección y segmentación de patrones de motivos de herencia cultural está limitada a los recursos de software y hardware disponibles para la realización de esta investigación.

L3. La investigación contempla el uso de un conjunto acotado de técnicas de visión computacional y modelos de redes neuronales del estado del arte en detección y segmentación de objetos en imágenes, por lo que no se consideran otras técnicas o modelos que puedan mejorar los resultados obtenidos.

L4. La interpretación de los patrones artísticos tiene un componente subjetivo, que puede no ser capturado en su totalidad por los modelos de redes neuronales utilizados y queda fuera del alcance de esta investigación.

Capítulo 2

Planteamiento del problema

En el presente capítulo se presentará el problema que se busca resolver en la presente investigación, presentando detalles sobre el problema, la hipótesis de la investigación, los supuestos y las preguntas de investigación.

2.1. Descripción del problema

En el estudio de artefactos arqueológicos, uno de los datos más valiosos es la correcta anotación y análisis de los motivos que adornan la superficie de los objetos (Ver Figura 2.1). La segmentación precisa de estos motivos es fundamental, ya que proporciona información detallada sobre su forma, dimensiones y ubicación en el objeto. Esta tarea se complica debido a la diversidad de estilos y la evolución de los patrones a lo largo de diferentes periodos culturales

Figura 2.1: Selección de imágenes del conjunto de datos "*Repetitive Patterns on Textured 3D Surfaces*" [1]. Las imágenes muestran la segmentación de los motivos arqueológicos en la superficie de una muestra de objetos del conjunto de datos.

La tecnología actual de procesamiento de imágenes ofrece una solución prometedora a este desafío. En particular, la anotación por medio de polígonos que representan la forma de los patrones en la textura de las vasijas arqueológicas. Este proceso permite la extracción de información adicional, como las cajas delimitadoras o *bounding boxes*. Estas cajas delimitadoras son esenciales para los modelos de visión por computadora enfocados en tareas de detección de objetos, ya que señalan las regiones de interés específicas dentro de la imagen.

Sin embargo, la implementación de estas tecnologías enfrenta varios desafíos. Primero, la

precisión en la segmentación es crítica; cualquier error podría resultar en una interpretación errónea de los datos, perdiendo detalles clave que son esenciales para comprender el significado cultural de los patrones. Además, los motivos en estas vasijas a menudo se caracterizan por su complejidad geométrica y variedad estilística, lo que requiere un enfoque sofisticado y adaptable en la tecnología de procesamiento de imágenes. Este enfoque debe ser lo suficientemente robusto para manejar variaciones en la textura, el color y la forma.

Otro aspecto crucial es la interpretación cultural de los datos obtenidos. Si bien la tecnología puede proporcionar una segmentación precisa y detallada, la interpretación de estos datos en un contexto cultural y histórico sigue requiriendo la perspectiva experta de arqueólogos y antropólogos. Por lo tanto, una colaboración efectiva entre tecnólogos y expertos en patrimonio cultural es fundamental para garantizar que los hallazgos tecnológicos se traduzcan en una comprensión más profunda del patrimonio cultural peruano.

Finalmente, la integración de estas tecnologías en la práctica arqueológica plantea desafíos prácticos, incluyendo la necesidad de recursos adecuados, formación técnica y adaptación de las metodologías de investigación existentes. La superación de estas barreras es esencial para aprovechar plenamente el potencial de las tecnologías de procesamiento de imágenes en la arqueología y la conservación del patrimonio cultural.

2.2. Hipótesis

La hipótesis de investigación se plantea como: La implementación de un sistema de detección automática de patrones utilizando técnicas avanzadas de procesamiento de imágenes y aprendizaje supervisado utilizando redes neuronales mejora significativamente la precisión y eficiencia en el análisis de patrones geométricos en vasijas peruanas, en comparación con los métodos tradicionales de análisis manual.

2.3. Supuestos

S1. Los patrones de motivos arqueológicos en imágenes de artefactos arqueológicos peruanos de herencia cultural pueden ser detectados y segmentados por medio de técnicas de procesamiento de imágenes y aprendizaje supervisado.

S2. Los patrones geométricos en las vasijas peruanas pueden ser cuantificados y clasificados eficientemente mediante algoritmos de aprendizaje automático.

S3. Las imágenes de artefactos arqueológicos peruanos de herencia cultural poseen características que permiten aplicar técnicas de procesamiento de imágenes y aprendizaje supervisado.

S4. Las anotaciones de los patrones de motivos arqueológicos en imágenes de artefactos arqueológicos peruanos de herencia cultural son correctas y representan de manera fiel los patrones presentes en las imágenes.

S5. Las segmentaciones de los motivos arqueológicos en el conjunto de datos a utilizar consideran solamente los patrones de relevancia arqueológica, por lo que patrones adicionales no serán considerados en el análisis.

S6. En el conjunto de datos utilizado, cada objeto está identificado exclusivamente por un ID numérico. Por lo tanto, se asume que cada ID representa una clase distinta, independientemente de las similitudes visuales o características que pueda compartir con otros objetos en el conjunto de datos.

2.4. Preguntas de investigación

PI1. ¿Cómo se pueden aplicar y adaptar las técnicas de aprendizaje automático y procesamiento de imágenes para la detección y clasificación de patrones geométricos en vasijas peruanas?

PI2. ¿Qué desafíos y limitaciones se presentan al implementar técnicas de procesamiento de imágenes y aprendizaje automático en el análisis de artefactos culturales?

PI3. ¿Es posible identificar patrones que están parcialmente visibles o que se encuentran con una transformación visual distinta al patrón original?

PI4. ¿El uso de modelos pre-entrenados con imágenes externas al conjunto de datos de vasijas peruanas mejora la precisión del sistema de detección automática con respecto al uso de algoritmos tradicionales?

PI5. ¿Es posible generalizar el sistema de detección automática de patrones geométricos en vasijas peruanas para otros tipos de artefactos culturales?

Capítulo 3

Trabajos relacionados

En el presente capítulo se presentarán las principales investigaciones y trabajos relacionados con el estudio de detección y segmentación de objetos en imágenes, comenzando con la revisión de los trabajos relacionados con la detección y segmentación de objetos en datos arqueológicos, para luego revisar la detección y segmentación con imágenes 2D en otros dominios de datos. Además se presentarán los desafíos presentes en la detección de objetos y la clasificación de problemas que existen según el conjunto de datos que se posee como ground truth.

3.1. Detección y segmentación de objetos en datos arqueológicos

Diferentes investigaciones relacionadas con la detección de motivos y patrones en imágenes de artefactos arqueológicos han mostrado los problemas desde distintos enfoques, especialmente en el enfoque de detección y clasificación de patrones sobre imágenes 3D de objetos arqueológicos. Esto se debe a que los datos relacionados con los artefactos primero son procesados mediante técnicas de fotogrametría para obtener modelos 3D de los artefactos y luego se pueden analizar directamente sus texturas y formas.

En SHREC'18 de Biasotti et al. [12] se investigó el problema donde dado un patrón geométrico se busca segmentar sectores de la superficie del modelo 3D que tuvieran patrones geométricos similares. El estudio mostró las formas en las que es posible resolver el problema, como la caracterización de vértices utilizando multi-escalas [13] y la técnica divide y vencerás donde se busca aplicar técnicas de recuperación de imágenes sobre subregiones del modelo 3D. Sin embargo, el estudio mostró que los resultados obtenidos no fueron satisfactorios debido a las características de los datos y se propone explorar otras soluciones.

En otros estudio relacionados a segmentación de motivos arqueológicos, Lengauer et al. [14] presenta un sistema para la recuperación de motivos en cerámicas griegas utilizando como entrada el motivo de interés de búsqueda y como conjunto de datos la segmentación. Para lograr esto, el sistema utiliza una entrada del usuario remarcando el sector de la imagen que tiene el motivo de interés.

En otro enfoque, Thompson et al. [15] propone investigar la detección de patrones sobre la superficie de modelos 3D de artefactos arqueológicos utilizando el relieve de la textura. En la

investigación se utilizan técnicas tradicionales como el filtro de Sobel [16] para el cálculo de los gradientes de cada vértice y la creación de vectores de características que permitan crear la matriz de disimilitud y poder obtener una métrica de distancia para evaluar la similitud de las regiones. También se mencionan técnicas como caracterización local de secciones de superficie y técnicas que extraen características de la superficie de la malla como Deep Patch Metric Learning y Mesh Local Binary Pattern.

Sin embargo, la solución que entrega mejores resultados en la investigación es el uso de Deep Feature Ensemble e histogramas de orientación, donde se utilizan modelos preentrenados de redes convolucionales como ResNet [17] y DenseNet [18] para extraer vectores de características de imágenes 2D obtenidas desde el modelo 3D. Una vez obtenidos los vectores de características realizan el cálculo de similitud entre los objetos aplicando métricas como la similitud coseno y distancia L1 y L2. Los resultados mostraron que la estrategia obtiene un 83 % de mean Average Precision (revisar Sección 4.5).

Tanto Thompson et al. [15] como Biasotti et al. [12] muestran que entre las mayores dificultades al momento de aplicar técnicas realizadas a aprendizaje en los datasets de cada "track" era la poca cantidad de datos que se tenían para poder entrenar los modelos de aprendizaje supervisado y lograr que estos generalizaran con otros tipos de datos.

Por otro lado, la búsqueda de patrones repetitivos en documentos y en texturas 2D de documentos y objetos de herencia cultural también se ha investigado considerando el enfoque de búsqueda de similitud de patrones. En el trabajo de En et al. [19] se propone utilizar descriptores agregados localmente (VLAD) y vectores de Fisher en la búsqueda de patrones en manuscritos medievales.

En cuanto al uso de redes neuronales, Ubeda et al. [20] propone el uso de redes convolucionales para la extracción de características usando un modelo pre-entrenado con COCO de RetinaNet para la búsqueda de patrones en documentos históricos. En el estudio se muestra que la estrategia presenta mejores resultado que el estado del arte en detección de patrones y requiere menos almacenamiento para indexar las imágenes. Sin embargo, se menciona que falla al detectar patrones con múltiples instancias de la imagen consultada.

De esta manera, si bien los estudios muestran que existen múltiples formas en las que se ha abordado el problema de detección de objetos en datos de herencia cultural, la dificultad en localizar patrones en este tipo de datos es un desafío que aún no ha sido resuelto y que la investigación de nuevas estrategias que utilizan redes neuronales y técnicas de machine learning mejorarían el resultado de las soluciones actuales.

Desafíos en la detección de objetos en datos arqueológicos

Entre los principales desafíos de trabajar con los modelos 3D de los datos presentados en los estudios mencionados anteriormente [15, 12] se encuentran casos comunes entre los distintos datasets:

- Ruido [15]: Los datos de artefactos arqueológicos suelen tener ruido en sus texturas debido al deterioro del pigmento en la textura de los objetos, a piezas rotas, la iluminación durante el proceso de escaneo y la calidad de la cámara utilizada.

- Adquisición de la información [15, 12]: En ambos estudios se destaca la escasez de datos de cada dataset analizado, donde la cantidad de modelos 3D no superaba los 25 objetos (25 en SHREC'18 [12] y 20 en Thompson et al. [15]).
- Tamaño de los datos: Los modelos 3D con los que se suele trabajar suelen ser de alta definición (sobre 6 millones de vertices) [12]. Debido a esto, el costo computacional de procesar los datos es alto y se requiere de un alto poder de procesamiento para poder realizar las tareas de detección de patrones, tanto en el uso de técnicas tradicionales como en el uso de redes neuronales.
- Construcción del conjunto de entrenamiento: [12] Los objetos de Herencia Cultural suelen ser únicos, por lo que las formas y estilos de los objetos no se suelen repetir entre distintos patrones. Esto hace que sea difícil construir un conjunto de entrenamiento que permita generalizar los patrones de los objetos. Por otro lado, existe un problema de etiquetado de los datos, dada la cantidad de patrones que es necesario etiquetar y que se requiere de un experto en el área para poder realizar el etiquetado.

Dado estos inconvenientes, las investigaciones mencionadas han propuesto estudiar la detección de patrones de artefactos arqueológicos reduciendo el problema al reconocimiento de objetos en imágenes, aplanando la textura de los modelos 3D para obtener imágenes 2D de los objetos, ya que esto permite utilizar técnicas de aprendizaje supervisado y no supervisado para la detección de patrones en imágenes 2D.

En Lengauer et al. [1] se aplica la estrategia de estirar la imagen y poder trasformarla desde un modelo 3D a una imagen 2D. Para ello, los autores crearon un software con un flujo de trabajo que permite al investigador trabajar directamente con el modelo 3D y segmentar los patrones de interés para construir el ground truth del conjunto de datos. Una vez los motivos son segmentados es posible obtener una imagen de la textura aplanada con los respectivos datos de los polígonos de los motivos y los datos que permiten transformar la textura a una imagen 3D nuevamente.

3.2. Detección de objetos en imágenes 2D

La detección de objetos en imágenes es un problema que es abordado en distintas investigaciones del área de visión computacional en los últimos años. Dependiendo del conjunto de datos con el cual se trabaja, el problema de detección de objetos puede variar, existiendo distintas maneras de modelar el problema según la composición de los datos.

Los principales tipos de problemas que se estudian dependiendo de los datos son la detección de objetos, segmentación de objetos, detección de saliencia y query-detection. Cada uno de estos enfoques tienen objetivos distintos, pero es común la comparación entre ellos debido a que comparten métricas de evaluación. En las siguientes secciones se presentarán el estado del arte y una breve descripción de la evolución de cada método en cada uno de los problemas.

3.2.1. Detección tradicional de objetos repetidos

En la literatura, las primeras técnicas que se crearon para resolver el problema de detección de patrones repetitivos se basaban en descriptores de características locales de la

imagen como SIFT [21] y SURF [22] cuyo objetivo es localizar sectores de la imagen que tengan características similares con la imagen de interés de entrada (Ver figura 3.1). Este tipo de algoritmos es invariante a la escala, traslación y rotación de las imágenes y además presentaron mejoras en su rendimiento con variantes como BRIEF [23] y ORB [24].

Este tipo de problemas es denominado key-point matching o key-point detection, en el cual, a partir de una imagen proporcionada, se identifican y se correlacionan aquellos puntos distintivos que guardan similitud con las características relevantes del objeto de interés.

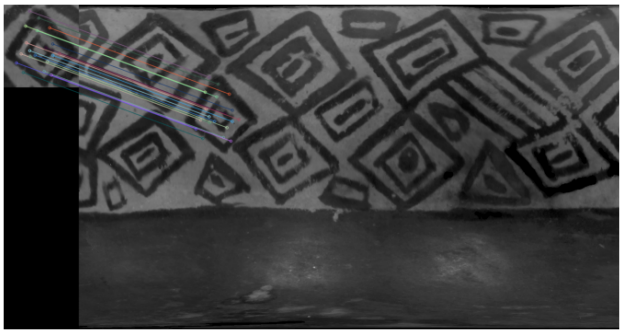


Figura 3.1: Ejemplo del algoritmo de SIFT. La consulta realizada es un rombo obtenido desde la misma imagen. Cada recta representa un key-point detectado por el algoritmo.

El estudio de key-points matching se ha investigado principalmente en imágenes de fachadas de construcciones [25, 26, 27, 28, 29] en donde se han analizado escenas urbanas utilizando variantes de los algoritmos de SIFT y BRIEF logrando mejores métricas de precisión.

Sin embargo, las técnicas de key-point matching tienen limitaciones que impiden generalizar el problema a otros tipos de datos. Principalmente, este tipo de algoritmos dependen significativamente de la calidad y cantidad de los puntos clave detectados y pueden ser susceptibles a errores en presencia de ruido, oclusión o deformaciones significativas del objeto de interés.

Además, la correspondencia basada en descriptores locales puede no ser suficientemente robusta en escenarios donde el objeto de interés tiene pocas características distintivas o cuando hay patrones repetitivos similares en la imagen que pueden conducir a emparejamientos incorrectos.

3.2.2. Detección de objetos con redes neuronales

En la literatura, el problema de detección de objetos utilizando redes neuronales tiene como objetivo identificar y localizar objetos en imágenes. Este proceso implica dos componentes, detectar la ubicación del objeto y clasificar al objeto. La detección del objeto se realiza por medio de una bounding box, los cuales son rectángulos delimitadores que encierran a los objetos de interés. Las bounding boxes definen las coordenadas, la posición y el tamaño del objeto en la imagen.

La clasificación del objeto se realiza por medio de una etiqueta que identifica la clase del objeto. En los modelos de redes neuronales, los modelos son entrenados con datos etique-

tados que contienen las coordenadas de la bounding box y la clase del objeto. Durante el entrenamiento del modelo, se busca que el modelo sea capaz de reconocer patrones y características de los objetos con la misma clase.

Los dos principales grupos que se dividen los modelos de detección de objetos son los modelos de dos etapas [30] y los modelos de una etapa [31, 32, 33]. En los modelos de dos etapas, el primer proceso que realizan es la generación de propuestas de regiones de interés de la imagen, en donde el modelo sugiere posibles ubicaciones del objeto. En la segunda etapa el modelo realiza la clasificación del objeto y refinar la localización.

En los modelos de una etapa, la detección y la clasificación de los objetos se realiza en un solo proceso, sin aplicar una etapa de generación de propuestas de regiones de interés. Esto permite que los modelos de una etapa sean más rápidos que los modelos de dos etapas, sin embargo, los modelos de una etapa han mostrado menor rendimiento en precisión sobre conjuntos de datos como COCO [2] o PASCAL VOC [34].

En los últimos años, el estado del arte en detección de objetos corresponde a arquitecturas que utilizan redes convolucionales. Entre los modelos que han obtenido mejores resultados en métricas estándares se encuentran Faster R-CNN [30], YOLO [31], RetinaNet [32] y SSD [33]. Estos modelos han sido utilizados en distintos dominios de datos como imágenes médicas [35, 36], imágenes satelitales [37, 38] y en imágenes de documentos [39].

Las características de la entrada de los modelos basados en redes convolucionales son un conjunto de imágenes de un tamaño fijo, bounding boxes de los objetos a detectar y las clases de los objetos.

A pesar de que se ha logrado mejorar las métricas de detección de objetos obtenidas por los modelos, todavía existen desafíos en las imágenes que impiden detectar claramente ciertos objetos. Entre ellos se encuentran:

- Detección de objetos pequeños: Se ha mostrado que los modelos de detección de objetos tienen problemas al detectar objetos pequeños [40].
- Variaciones en iluminación y resolución en imágenes: Los modelos de redes convolucionales pueden ser sensibles a las variaciones en la iluminación y las condiciones ambientales de las imágenes de entrada [41].
- Diversidad de contextos: Los modelos de detección de objetos pueden tener problemas al detectar objetos en contextos distintos a los que fueron entrenados [41]. Esto es posible solucionarlo realizando fine-tuning del modelo con datos del contexto de interés o agrandando el conjunto de datos de entrenamiento. Sin embargo, es posible que el modelo no logre generalizar con todos los contextos.

3.2.3. Segmentación de objetos

En este problema se busca segmentar los objetos de interés en una imagen. El objetivo es obtener una máscara binaria que permita identificar los píxeles que pertenecen al objeto de interés y clasificarlo según la clase que posee. Los problemas de segmentación de imagen se

pueden clasificar en segmentación semántica, de instancia y panóptica.

En cada uno de estos problemas se busca segmentar los objetos de interés en una imagen, sin embargo, la diferencia entre ellos es la manera en que se segmentan los objetos. Además, el conjunto de datos necesario para entrenar los modelos de segmentación es distinto en cada uno de los problemas. Las características de los problemas son los siguientes:

1. Segmentación de instancias (Fig. 3.2.c): El objetivo en este problema es identificar y segmentar cada instancia del objeto de interés. Cada objeto es diferenciado, a pesar de que pueden compartir características o formas, por medio de la clase o el color de la máscara. En caso de que los objetos estén sobrepuestos, se debería segmentar a cada objeto por separado.

2. Segmentación semántica (Fig. 3.2.d): En este problema se busca categorizar semánticamente cada pixel de la imagen. A diferencia de segmentación de instancias, objetos distintos pueden recibir la misma etiqueta si pertenecen a la misma categoría, como por ejemplo un conjunto de personas reciben la etiqueta persona en vez de etiquetas distintas. De esta manera, objetos con características similares comparten la misma categoría. Este tipo de problemas es esencial cuando los objetos están sobrepuestos entre sí, ya que pueden ser identificados con el mismo label evitando problemas en la segmentación.

3. Segmentación panóptica (Fig. 3.2.b): Introducido en Kirillov et al. [42] en segmentación panóptica, además de obtener la máscara y etiqueta como en los dos problemas anteriores, se busca combinar la segmentación semántica y de instancia realizando una segmentación de todos los objetos en la imagen. El objetivo es lograr asignarle un valor semántico a todas las instancias de la imagen y además diferenciarlas entre sí.

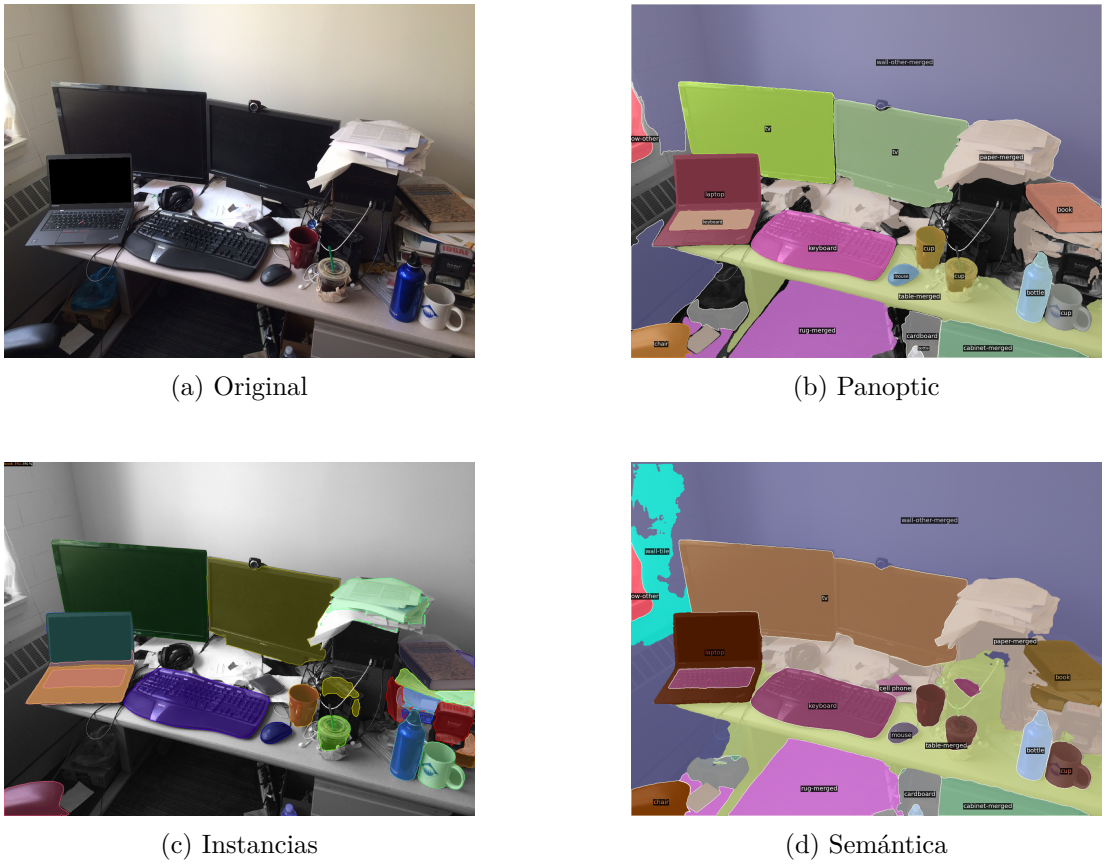
En los problemas de segmentación descritos anteriormente el conjunto de datos necesario para entrenar los modelos puede ser equivalente tanto para realizar segmentación de instancias o semántica, sin embargo, para realizar segmentación panóptica se requiere de un conjunto de datos que posea las máscaras de segmentación de todos los objetos y el significado semántico de cada uno de ellos.

Esto último es un desafío para aquellos conjuntos de datos que no poseen una segmentación semántica completa de la imagen, ya que además de ser necesario completar segmentaciones de objetos que no poseen segmentación es necesario añadir un significado semántico que tenga consenso con otros conjuntos de datos o respecto a la tarea que se quiere realizar.

Algoritmos tradicionales

Los algoritmos tradicionales utilizados para segmentar objetos varían entre técnicas como *region-based* y *edge segmentation* [44, 45], Watershed [46, 47] y técnicas de clustering como k-means [46]. En los algoritmos de *region-based* se busca segmentar la imagen en regiones que tengan características similares, utilizando técnicas como *region growing* [44]. En los algoritmos de *edge segmentation* se busca segmentar la imagen en regiones que tengan bordes distintos, utilizando técnicas como Canny Edge Detection [48] y clustering como k-means [46].

Estas técnicas han sido utilizadas en problemas para la segmentación de objetos en imágenes médicas [46] y en imágenes satelitales [47], mostrando resultados que permiten realizar



(a) Original

(b) Panoptic

(c) Instancias

(d) Semántica

Figura 3.2: Ejemplo de segmentación de instancias, semántica y panóptica realizada con el modelo Mask2Former [43]. En el ejemplo se puede observar que hay objetos que no se segmentan al realizar la tarea de instancias debido a que no son objetos de interés. Mientras que en la segmentación semántica se segmentan elementos que se sobreponen levemente con el mismo label y color, mientras que en panoptic se segmentan con un distinto color y label, haciendo la diferencia que son instancias distintas.

la segmentación con un menor costo computacional y de manera eficaz respecto al tiempo de ejecución. Sin embargo, tareas como la segmentación semántica y panóptica no han sido resueltas con estos algoritmos.

Segmentación con redes neuronales

En los últimos años se ha extendido el estudio en modelos basados en redes neuronales, en donde han sido utilizados para problemas complejos como la segmentación semántica, la segmentación en tiempo real de personas, segmentación 3D, entre otros [49, 41].

Los primeros modelos de segmentación se basaron en el aprendizaje de máscaras binarias de los objetos de interés como U-Net [50] mediante la técnica de encoder-decoder y el uso de redes convolucionales como ResNet [17]. Otro tipo de modelos han permitido extender la segmentación de objetos a segmentación de instancias y panóptica como Mask R-CNN [51], SeMask [52] y Panoptic FPN [42].

Los modelos convolucionales mencionados son supervisados y han sido entrenados y evaluados con conjuntos de datos como COCO [2], Cityscapes [53] o Pascal VOC [34], los cuales poseen una gran cantidad de imágenes y segmentaciones de objetos, además de contener objetos similares entre las imágenes que permiten que el modelo generalice con distintos tipos de formas, colores y texturas.

Además de la arquitectura de redes convolucionales, recientes estudios han mostrado que las redes neuronales basadas en Vision Transformer (ViT) [9] han mostrado tener un mejor rendimiento que las CNN [9] en especial en tareas con nuevos conjuntos de datos y logrando ejecutar tareas de manera *self supervised* o *unsupervised*.

Uno de los modelos que ha mostrado tener un buen rendimiento en segmentación self supervised es DINO [54] el cual utiliza una arquitectura basada en un ViT estudiante y profesor. El ViT estudiante es el encargado de aprender a predecir las características globales en una imagen a partir de parches locales supervisados por cross entropy loss de los embeddings de un momentum ViT profesor, mientras afina sus predicciones para evitar que la red aprenda una función trivial que no es útil.

El modelo de DINO ha mostrado contener información explícita sobre la segmentación semántica de una imagen, a diferencia de los ViTs supervisados y las redes convolucionales, lo cual lo convierte útil en tareas donde es necesario realizar segmentación o extracción de características sobre imágenes que no poseen etiqueta o segmentación y por tanto no pueden ser entrenados por medio de un modelo supervisado convencional (ver Figura 3.3).

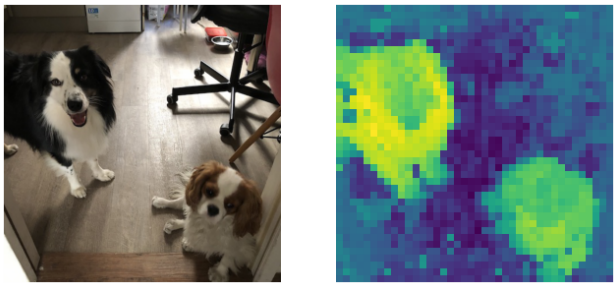


Figura 3.3: Ejemplo de mapeo de características de DINO [54].

Desde DINO han surgido distintos modelos que utilizan la arquitectura para realizar tareas de segmentación no supervisada. En Hamilton et al. [55] proponen el modelo STEGO, el cual utiliza las características extraídas por DINO para realizar segmentación no supervisada mediante el algoritmo de K Nearest Neighbors (KNN). A pesar de que los resultados de los modelos no supervisados aumentan respecto a otras variantes, los resultados son menores a los modelos supervisados.

Modelos fundacionales

En Li et al. [56] se dividen los modelos fundacionales en dos conjuntos, aquellos que utilizan adaptadores de visión y modelamiento de instrucciones (prompting modeling) y aquellos que utilizan aprendizaje de vocabulario abierto (open vocabulary). Estas arquitecturas adaptan modelos fundacionales pre-entrenados con una gran cantidad de datos para tareas de las cuales no fueron entrenados. Cuando un modelo muestra la capacidad de adaptarse a estas

nuevas tareas se dice que el modelo es *zero-shot*.

Uno de los modelos fundacionales más utilizados es CLIP [57], el cual utiliza un modelo ViT preentrenado con una gran cantidad de imágenes y texto para realizar tareas de clasificación de imágenes y texto. El modelo logró un resultado de 76.2 % en el conjunto de datos de ImageNet [58] y ha sido extendido para realizar tareas de segmentación de objetos [59] utilizando la técnica de prompting modeling.

Otro modelo que ha mostrado excelentes resultados en tareas de segmentación *zero-shot* ha sido el modelo de Segment Anything [60]. Este modelo utiliza prompts de puntos, bounding boxes o texto para segmentar objetos en imágenes. El modelo fue entrenado con 1 billón de máscaras y 11 billones de imágenes de distintos lugares del mundo y que poseen múltiples objetos. La versatilidad de este modelo permite que sea utilizado en distintos dominios de datos y que sea capaz de segmentar objetos en imágenes que no poseen etiquetas.

Este tipo de modelos ha presentado ser una alternativa para realizar segmentación de objetos en imágenes que no poseen etiquetas o para complementar segmentaciones realizadas por modelos supervisados. Sin embargo, el tamaño de los modelos pre-entrenados y la cantidad de parámetros que poseen los hacen lentos o muy costosos para el procesamiento masivo de imágenes.

3.2.4. Detección de saliencia

En este problema se busca identificar los sectores de la imagen que son más relevantes para el ojo humano mediante una máscara binaria. En este tipo de problema los objetos sobrelapados se identifican como un solo objeto, donde no es necesario diferenciar los objetos entre sí.

Similar al objetivo de los modelos *zero-shot*, los modelos entrenados para la tarea de saliency detection suelen ser entrenados con un gran conjunto de imágenes para realizar tareas nuevas donde no fueron entrenados y donde las imágenes no poseen segmentación de los objetos. Las tareas en las que se utilizan estos modelos son eliminar fondos de imágenes y destacar objetos en procesamiento de imágenes.

Uno de los modelos del estado del arte es U-2-Net [61] el cual fue entrenado con datasets de imágenes de personas, animales, objetos y paisajes. El modelo posee una arquitectura basada en U-Net [50] donde cada bloque de la red consiste en un residual U-block, una arquitectura de encoder-decoder modificada de U-Net.

3.3. Métricas de evaluación

Las métricas de evaluación en segmentación y detección de objetos varían respecto a las tareas que se estudian y en el último tiempo se han propuesto nuevas métricas a problemas más complejos de segmentación.

En general las métricas más utilizadas son las métricas de intersección sobre la unión (IoU) y el promedio de precisión (AP). Sin embargo, dependiendo de la tarea que se estudia es más conveniente usar una métrica por sobre otra. Por ejemplo, la métrica de IoU es utilizada en

segmentación semántica debido a que el análisis suele enfocarse en las máscaras predichas por el modelo en cada categoría. Mientras que el AP suele ser utilizado en detección de objetos o en segmentación de instancias debido a que el análisis se enfoca en la precisión del modelo respecto al bounding box o a todas las clases del conjunto de datos.

3.3.1. Intersección sobre la unión

La métrica de intersección sobre la unión (IoU) o índice de Jaccard es una métrica utilizada tanto para detección de objetos como para segmentación. La métrica compara la segmentación predicha con la segmentación real y calcula el porcentaje de pixeles que coinciden entre ambas segmentaciones [62].

La media de IoU (mIoU) se calcula como el promedio de IoU de cada objeto en la imagen. En el caso de la detección de objetos, la métrica se calcula como el promedio de IoU de cada bounding box con la segmentación real.

Un valor de IoU cercano a 1 indica que la segmentación predicha se acerca a la segmentación real, mientras que un valor bajo, por consenso menor a 50 %, de IoU indica que la segmentación predicha no es similar a la segmentación real.

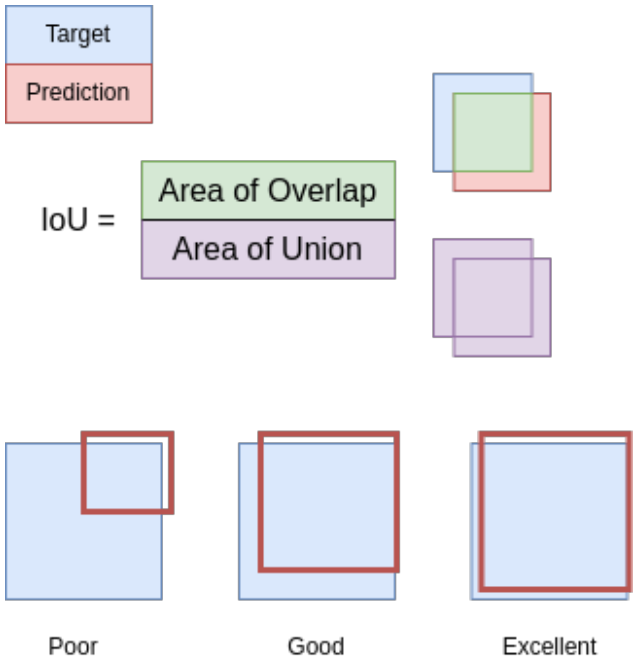


Figura 3.4: Cálculo de IoU entre dos detecciones.

3.3.2. Average Precision

La métrica de Average Precision (AP) es una métrica utilizada en detección de objetos y segmentación de instancias. La métrica calcula la precisión de la segmentación predicha con la segmentación real, donde la precisión se calcula como el número de verdaderos positivos sobre el número de verdaderos positivos más falsos positivos respecto a la IoU de cada objeto.

El mean Average Precision (mAP) se calcula como el promedio de AP de cada objeto en

la imagen. Para calcular mAP se debe calcular el AP para todas las clases de objetos en la imagen y luego promediar todos los AP. En el caso del dataset COCO [2] se utiliza la métrica de AP@IoU=0.5, AP@IoU=0.75 y AP@IoU=[0.5:0.05:0.95] para calcular el mAP.

3.3.3. Dice Coefficient

El Coeficiente de Dice es una métrica utilizada principalmente en segmentación en el área de medicina debido a que es una métrica que es más estricta que la métrica de IoU. La métrica calcula el porcentaje de pixeles que coinciden entre la segmentación predicha y la segmentación real, pero a diferencia de IoU, el coeficiente de Dice penaliza más los falsos positivos y falsos negativos.

El coeficiente de Dice se calcula como:

$$DSC = \frac{2 \times |X \cap Y|}{|X| + |Y|} \tag{3.1}$$

Donde X es la segmentación predicha y Y es la segmentación real.

3.3.4. Panoptic Quality

La métrica de Panoptic Quality (PQ) [42] es una métrica utilizada en segmentación panóptica realizando una combinación entre los objetivos de segmentación semántica y de instancias. La métrica PQ se descompone en dos términos: Calidad de Segmentación (SQ) y Calidad de Reconocimiento (RQ). El SQ mide la calidad media de la segmentación de objetos y regiones correctamente segmentados (verdaderos positivos), utilizando el promedio del IoU (Intersección sobre Unión) de estos segmentos. Por otro lado, el RQ evalúa la capacidad del modelo para detectar y clasificar correctamente los objetos y regiones, utilizando una media armónica de precisión y recall (recordatorio).

El PQ se calcula para cada clase de forma independiente y luego se promedia sobre todas las clases. Esto hace que sea insensible al desequilibrio de clases. Para cada clase, el emparejamiento único divide la segmentación predicha y la segmentación real en tres conjuntos: verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN). La fórmula general para PQ es la siguiente:

$$PQ = \frac{\sum_{(p,g)\in TP} IoU(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{3.2}$$

Donde k es la clase de objetos, p es la segmentación predicha, g es la segmentación real.

3.4. Comentarios

En este capítulo se presentaron los principales trabajos relacionados con la detección de objetos en datos arqueológicos y en imágenes 2D. En el caso de los datos arqueológicos, se presentaron los principales desafíos que existen al momento de realizar detección de patrones en modelos 3D de artefactos arqueológicos, donde se destaca la escasez de datos y la dificultad de construir un conjunto de entrenamiento que permita generalizar los patrones de los objetos.

En cuanto a la detección de objetos en imágenes 2D, se presentaron los principales prob-

lemas que se han investigado en la literatura, donde se destaca la detección de objetos, segmentación de objetos, detección de saliencia y query-detection. Además se presentaron los principales algoritmos que han obtenido mejores resultados en cada uno de los problemas, donde se destaca el uso de redes neuronales en la detección de objetos y segmentación de objetos.

En los modelos de redes neuronales, se hizo una breve revisión de la literatura del estado del arte debido a la gran cantidad de modelos y también a la gran cantidad de variantes que existen.

Debido a que el problema de detección de patrones en datos arqueológicos se puede reducir a un problema de detección de objetos en imágenes 2D, esto invita a explorar esta familia de modelos, considerando también las limitantes de los datos que se poseen, como la pérdida de información, la falta de un etiquetado estandarizado en arqueología y el tamaño de los datos.

Por otro lado, el uso de modelos supervisados ha mostrado tener buenos resultados al ser utilizado con pesos pre-entrenados, lo cual ayuda también en el estudio de conjuntos de datos con poca cantidad de imágenes. Sin embargo, el tamaño de los modelos pre-entrenados y la cantidad de parámetros que poseen son características que pueden influir en la elección de los modelos a utilizar, debido al alto costo computacional de hardware y de tiempo de ejecución.

El enfoque que se utilizará en este trabajo será el uso de modelos pre-entrenados de redes convolucionales y fundacionales para realizar detección de objetos en imágenes 2D, considerando el uso de modelos supervisados y no supervisados. Además, se utilizarán modelos que han sido entrenados con una gran cantidad de datos para realizar tareas nuevas donde no fueron entrenados y donde las imágenes no poseen una etiquetación semántica de los objetos. Los algoritmos tradicionales serán utilizados para comparar resultados con los modelos del estado del arte y para proponer alternativas en la segmentación y detección de objetos en texturas arqueológicas.

Capítulo 4

Metodología

En el presente capítulo se realizará una descripción de los datos a utilizar, las estrategias de entrenamiento y de división de datos para poder evaluar los modelos y algoritmos de detección. Por otro lado, se detallarán los desafíos del problema, cómo se planea abordarlos y cuáles son las consecuencias en las estrategias consideradas para entrenar los modelos. Finalmente, se describirá el preprocesamiento de los datos, las métricas de evaluación utilizadas para medir el rendimiento de los modelos y la selección de modelos y algoritmos a utilizar.

4.1. Descripción de los datos

En Eurographics Symposium on Geometry Processing (EUSGP) 2021, fue publicado el conjunto de datos "*Repetitive Patterns on Textured 3D Surfaces*" [1] (desde ahora referido como RPT3DS), el cual está constituido por 82 imágenes 3D de diferentes objetos de cerámica con motivos pintados sobre su superficie. Los objetos pertenecen al Museo Josefina de Ramos, Lima, Perú, donde fueron digitalizados como parte de un proyecto de investigación FONDECYT [1].

El dataset original corresponde al desafío SHREC 2021 [63] "*SHREC 2021: Retrieval of cultural heritage objects*" donde se comparte un conjunto de imágenes 3D de aproximadamente 1000 objetos de cerámica, de los cuales fueron seleccionados solamente 82 objetos que exhibían un buen estado de conservación y que poseían motivos pintados sobre su superficie. La colección seleccionada corresponde a dos culturas pre-Colombinas, en donde 10 objetos pertenecen a la cultura Chancay y Lurin.

La forma de los 82 objetos 3D seleccionados varían entre lebrillo (basin), maceta (pot), cuencos (bowls), frascos (jar), vasos (vase), cántaros (pitcher), platos (plate) y figurinas (figurine) (Ver Tabla 4.1 y Figura 4.1).

[1] Project 02-2018-FONDECYT-BM-IADT-AV (Concytec-Perú): *Restoration and conservation of archeological pieces using deep learning and convolutional auto-encoder on graphs.*

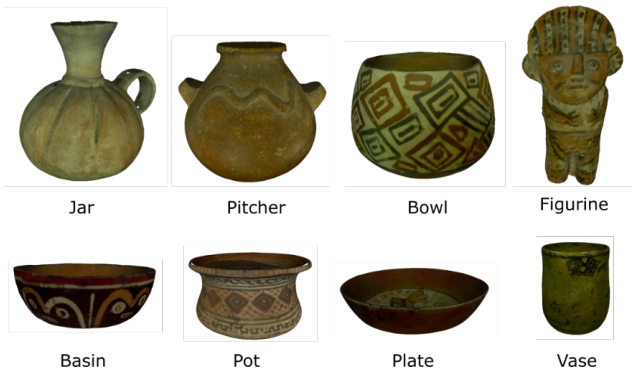


Figura 4.1: Formas de figuras 3D en dataset RPT3DS [63]

Tabla 4.1: Tipos de formas de objetos 3D en dataset RPT3DS.

Forma	**Cantidad**
Basin	23
Pot	20
Bowls	13
Jar	9
Vase	5
Pitcher	5
Plate	4
Figurine	2

Por cada modelo 3D del conjunto de datos se realizó la segmentación manual de las figuras decorativas que formaban patrones repetitivos alrededor del objeto, por medio del uso del software compartido por los autores del trabajo [1]. De los datos obtenidos por la herramienta se logran extraer los puntos poligonales que representan la segmentación de cada patrón y las propiedades de cada textura como la orientación, escala, proporción sobre la superficie y *n-foldness*. Estos últimos atributos permiten reconstruir la imagen al formato 3D sobre la superficie del objeto de estudio.

El dataset RPT3DS está disponible en el sitio web de Graz University of Technology [2] y está licenciado bajo la licencia Common Creative Attribution 4.0 International (CC BY 4.0). En total se logran encontrar 2828 motivos segmentados de los 82 objetos, existiendo un total de 102 clases o entidades distintas.

Cada imagen del conjunto de datos posee una resolución que varía respecto a la altura, pero se mantiene constante al ancho de 5000 píxeles. La variación respecto a la altura es una consecuencia de que los objetos poseen distintos tamaños, mientras que el ancho se estandarizó al repetir la textura en los extremos, técnica utilizada al momento de realizar la anotación para poder obtener las imágenes del dataset.

[2] Pattern-Benchmark 2021 [Accedido 2023-10-15]

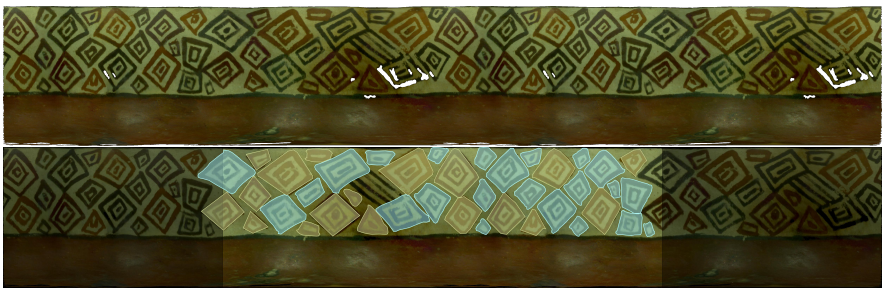


Figura 4.2: Textura aplanada de un objeto 3D del dataset RPT3DS. Imagen superior corresponde a la textura original aplanada, la imagen inferior corresponde a la textura aplanada con la segmentación realizada sobre el objeto 3D antes de aplanar la imagen.

La Figura 4.2 muestra un ejemplo de una textura aplanada con la segmentación de 2 entidades. En el ejemplo, las entidades segmentadas poseen la misma forma, pero las clases se diferencian respecto a su color (rojo y azul). El sector oscuro de la imagen son regiones repetidas o que no poseen segmentación.

En cuanto a las clases de los motivos presentes en las imágenes del conjunto de datos, estas fueron clasificadas al momento de segmentar cada motivo en la textura por parte del arqueólogo experto, por lo que cada motivo que el arqueólogo consideró que era distinto a otro fue clasificado como una clase distinta. Sin embargo, esta clasificación fue realizada de manera local por cada imagen, por lo que pueden existir motivos con forma o características similares en distintas imágenes que tienen clases distintas.

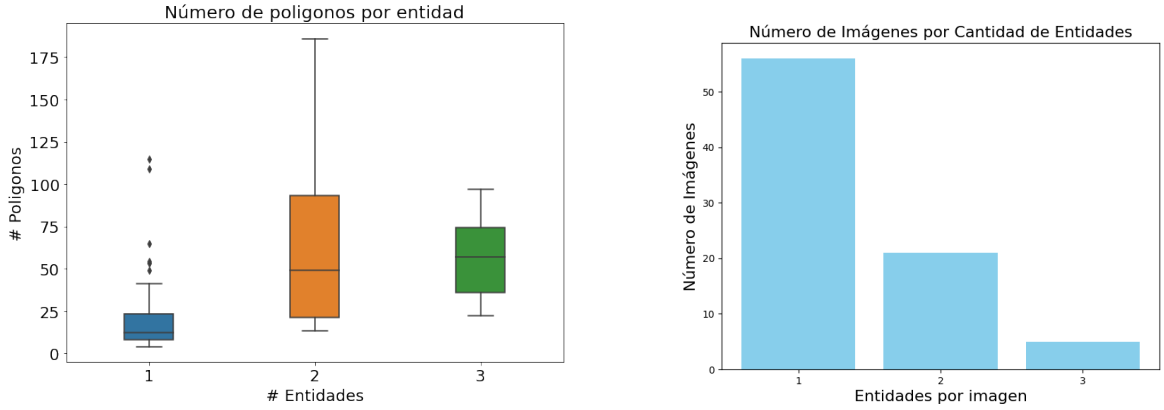


Figura 4.3: El gráfico izquierdo muestra la distribución de polígonos respecto a la cantidad de entidades en una imagen. En el lado derecho, se muestra la cantidad de entidades

La cantidad de entidades en una imagen varía entre una y tres (Ver Figura 4.4), donde sobre el 50 % de las imágenes posee solamente una entidad, como se puede observar en el gráfico derecho de la Figura 4.3. Por otro lado, la cantidad de polígonos por entidad varía entre 4 y 175, donde las imágenes que poseen una entidad poseen menor variabilidad y su mediana es menor a 25 polígonos. Respecto a las que tienen 2 y 3 entidades, tienen una

mayor variación y su mediana es mayor a 50 polígonos, como se puede observar en el gráfico izquierdo de la Figura 4.3.

Figura 4.4: De arriba hacia abajo, texturas con una entidad, dos entidades y tres entidades respectivamente. Cada entidad es diferenciada por un color distinto.

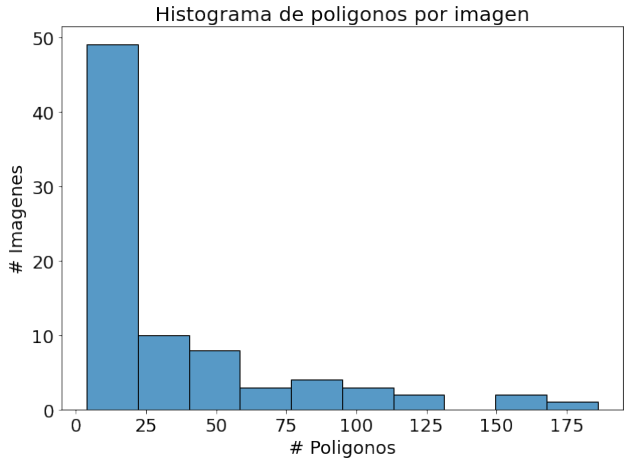


Figura 4.5: Distribución de imágenes respecto al número de polígonos.

La distribución de la cantidad de polígonos por imagen se puede observar en la Figura 4.5, mostrando que sobre el 50 % de las imágenes poseen 25 o menos polígonos, mientras que la cantidad de imágenes que poseen sobre 50 polígonos disminuye considerablemente.

4.2. Desafíos del problema

El conjunto de datos *RPT3DS* tiene un gran valor cultural, al contener una variedad de objetos 3D digitalizados con variadas formas, además de presentar una gran variedad de motivos arqueológicos. Una de sus características principales es la repetición considerable de motivos, donde las entidades del 76 % de las imágenes (63 imágenes) se repiten más de 10 veces.

Sin embargo, existen distintos desafíos que es necesario considerar antes de abordar el problema como un problema de detección de objetos o de segmentación.

4.2.1. Segmentación no etiquetada

Si bien se tiene información de los puntos poligonales que segmentan al objeto de la textura, la información que se posee para diferenciar una entidad de otra es solamente un ID o identificador único. Esta característica genera varios problemas al momento de abordar el problema como una tarea de segmentación semántica, ya que no se tiene información sobre:

- Falta de contexto semántico y cultural: La segmentación no etiquetada proporciona puntos poligonales que delimitan áreas en la textura, pero no se cuenta con información sobre el contexto cultural o el significado de las formas individuales.
- Descripción visual específica: No hay descripciones detalladas que expliquen visualmente qué representa cada entidad. Esto es importante para comparar objetos dentro del mismo conjunto de datos o con conjuntos de datos externos.
- Descripción abreviada que permita comparar un objeto con otro perteneciente a una al mismo conjunto de datos o incluso con un conjunto distinto

Esto puede dificultar enfocar el problema como una tarea de segmentación semántica, ya que no se tiene un contexto semántico sobre lo que se está buscando, como tampoco se puede validar si existe un conjunto de datos externo con el cual sea posible complementar los datos de RPT3DS.

4.2.2. Pérdida de información

Debido a diversos daños presentes en los artefactos como grietas, roturas o desgastes, la información de los motivos puede estar incompleta o incluso puede no existir. A pesar de que los datos de RPT3DS fueron obtenidos de artefactos en buen estado de conservación, existen entidades que se ven afectados por estos daños, como se puede observar en la Figura 4.6.

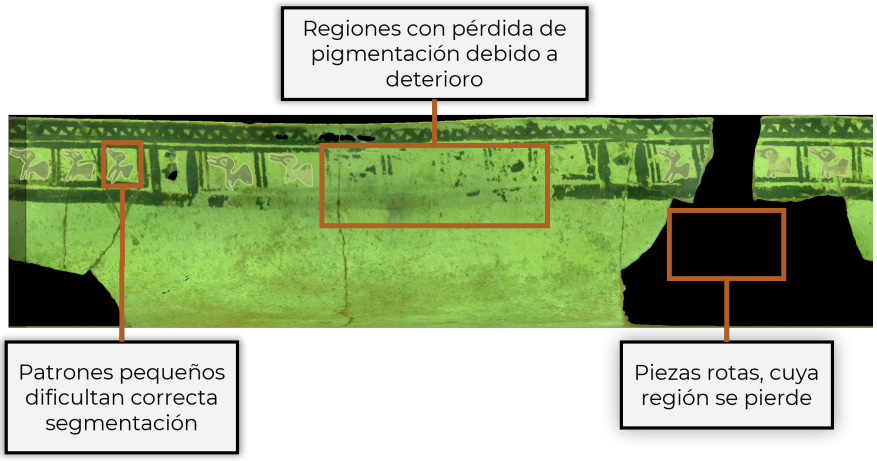
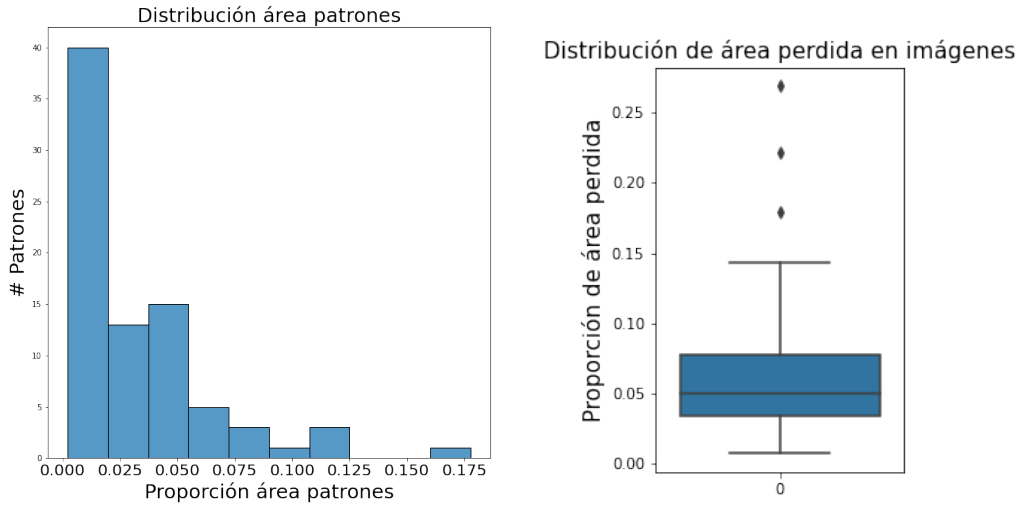


Figura 4.6: Ejemplo de datos faltantes en una imagen del dataset RPT3DS.

La perdida de información también se ve reflejada en el tamaño de la entidad segmentada, donde existen entidades que poseen un tamaño considerablemente menor a otras, como se

observa en la Figura 4.6, y por tanto su calidad de imagen es menor. Esto puede provocar que la entidad no sea detectada por el modelo, o que sea detectada pero con una baja precisión.



(a) Distribución de la proporción del área del patrón respecto al área total de la imagen.

(b) Distribución de la proporción de área perdida respecto al área total de la imagen.

Figura 4.7: Distribución de áreas de patrones y de áreas perdidas.

Por otro lado, al analizar la proporción del área pérdida en las imágenes respecto al área total de la imagen, los datos varían entre 0 y 25 %, como se puede observar en la Figura 4.7.a, existiendo outliers sobre el 15 % de pérdida de pixeles en la imagen. Los datos fueron obtenidos considerando la imagen RGB y la transparencia como el color negro (0,0,0).

4.2.3. Entidades similares

Figura 4.8: La figura izquierda muestra 4 entidades con forma de pájaro presentes en 4 imágenes distintas, en la figura central se muestran 8 entidades con forma de rombo, las cuales son diferenciadas por el ID o por estar presentes en distintas imágenes, mientras que en la figura derecha se muestran 4 entidades con forma de triángulos, presentes en distintas imágenes.

Como se mencionó anteriormente, el conjunto de datos solamente posee una clasificación de las entidades por imagen, por lo que no se tiene información sobre la similitud entre entidades de distintas imágenes. Un ejemplo de esto se puede ver en la Figura 4.8, donde

distintos patrones presentes en distintas imágenes muestran formas o rasgos similares entre ellos. Esto puede provocar el aumento de falsos positivos en imágenes que poseen entidades similares, como también puede provocar la disminución de verdaderos positivos en imágenes que poseen entidades similares, pero que fueron clasificadas como distintas.

Cabe destacar que el etiquetado de las entidades en arqueología es una tarea compleja, especialmente cuando se consideran figuras de objetos poco comunes de observar o cuyo significado es difícil de inferir, donde el contexto de la cultura o el objeto puede ser un factor importante para la etiqueta de la entidad.

Por otro lado, si bien existen trabajos realizados para la multi-etiquetación de motivos de objetos de herencia cultural [64], realizarlo requeriría la validación adicional de un experto que pueda validar la multi-etiquetación de los motivos, lo cual puede ser un proceso costoso en tiempo y recursos.

4.3. Preprocesamiento de datos

En la siguiente Sección se detallan las transformaciones realizadas a los datos para poder utilizarlos en los modelos de redes neuronales y poder evaluar con las métricas de detección y segmentación de objetos. Se explican procesos como la limpieza de datos, la transformación de las anotaciones y de las imágenes, y se describen las principales estrategias que se utilizarán para entrenar y evaluar los modelos.

Las transformaciones realizadas a los datos se pueden replicar y observar en el repositorio de GitHub del proyecto [3]. En el repositorio es posible encontrar los datos con los cuales fueron entrenados los modelos y las librerías utilizadas para realizar las transformaciones.

4.3.1. Limpieza de datos

La primera transformación que se realizó a las imágenes fue eliminar regiones que no poseían puntos polígonales, pero sí existían figuras decorativas, como se observa en la Figura 4.4, donde el sector con figuras segmentadas sí posee puntos polígonales y las regiones grises no poseen. El objetivo de esta transformación es disminuir la cantidad de falsos positivos que se pueden obtener al momento de realizar la detección de objetos.

Para ello, dado el conjunto de puntos poligonales S, se definen los puntos de cortes C_l y C_r como los puntos extremos izquierdo y derecho respectivamente, y H como la altura de la imagen. Se define la distancia δ como el margen que existirá entre el punto más a la izquierda y el punto más a la derecha de la imagen respecto al borde de la imagen en ambos extremos.

[3] Repetitive-Archetypes-Patterns-Dataset [Accedido 19-01-2024]

$$C_l = (\min_{p \in S}\{p\} - \delta, H)$$
$$C_r = (\max_{p \in S}\{p\} + \delta, H)$$

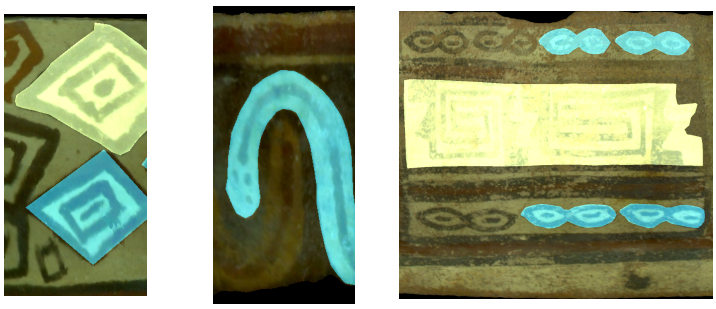


Figura 4.9: Ejemplos de casos bordes en el preprocesamiento de datos.

El valor de δ fue definido como 20 pixeles para evitar que puntos poligonales quedaran al borde de la imagen. En cuanto a la altura de las imágenes, esta se mantuvo debido a que las imágenes no poseen patrones similares al *ground truth* en las partes superiores o inferiores. Sin embargo, existen casos donde la etiqueta de un patrón no se encuentra debido a casos bordes, como se muestra en la Figura 4.9.

En ambos casos los motivos sin segmentar pertenecen al sector sin etiqueta de la imagen original. Se decidió mantener estos casos bordes dada la complejidad que existe en generalizar un algoritmo que permita eliminarlos sin afectar el *ground truth* original.

Figura 4.10: Imagen superior, muestra la imagen original. Imagen inferior muestra las dos entidades en colores amarillo y celeste, donde se observa que los objetos no poseen una correcta segmentación.

En las anotaciones de los polígonos en RPT3DS, se observó que una imagen en particular no poseía una correcta anotación de los polígonos, como se observa en la Figura 4.10. En este caso, las anotaciones de ambas entidades se encuentran en distintas coordenadas con las

figuras de la imagen. La imagen no será considerada en el entrenamiento del modelo, pero sí se estudiará al momento de evaluar a los modelos visualmente.

4.3.2. Transformación de datos

Se realizan transformaciones tanto a las anotaciones de RPT3DS como a las imágenes para estandarizar los datos y poder utilizarlos en los modelos de redes neuronales. En las siguientes secciones se detallan cada una de las transformaciones realizadas, cuáles serán las configuraciones a considerar para el entrenamiento y validación de los modelos, y las librerías utilizadas para realizar las transformaciones.

Transformación anotaciones

Las anotaciones de RPT3DS originalmente poseen información de los puntos poligonales que conforman la segmentación de las figuras de las imágenes. Además posee información sobre la escala, rotación, y la simetría de cada punto poligonal que permite transformar las anotaciones al formato 3D. Sin embargo, como el problema se enfocará en trabajar solamente con la segmentación de los patrones, no son considerados los datos adicionales para la transformación.

Figura 4.11: Ejemplo de etiquetación de entidades. Cada entidad posee como etiqueta un id único dentro de la imagen y el nombre de la imagen como categoría única en el conjunto completo de RPT3DS.

Respecto a la etiqueta que es asignada a cada entidad, se mantiene en el formato inicial del conjunto de datos donde se identifica como el ID `[nombre-imagen][id-numerico-imagen]` (Ver Figura 4.11). De esta manera cada entidad es considerada como una categoría única, a pesar de las similitudes que puedan existir entre entidades de distintas imágenes como se detalló en la Sección 4.2.

Un dato importante al momento de entrenar modelos de detección de objetos es el uso de *bounding boxes*, por lo que se decidió agregar esta información al conjunto de datos. Para ello, dado el conjunto de puntos poligonales S de una entidad, el ancho W y largo H de la imagen en pixeles se define la bounding box absoluta como sigue:

$$
\begin{aligned}
x_{min}, y_{min} &= \min_{p \in S}\{p_x\} \\
x_{max} &= \max_{p \in S}\{p_x\} \\
y_{min} &= \min_{p \in S}\{p_y\} \\
y_{max} &= \max_{p \in S}\{p_y\} \\
\text{bbox} &= (x_{min}, y_{min}, x_{max} - x_{min}, y_{max} - y_{min})
\end{aligned}
$$

El formato en el que se almacenan las etiquetas finales del conjunto de datos RPT3DS se estandariza al formato del dataset de detección de objetos COCO [2], donde el conjunto completo de anotaciones es almacenado en un único archivo JSON.

Estandarizar las anotaciones al formato COCO entrega varias ventajas, una de ellas es lograr comparar métricas de detección de objetos y segmentación de instancias entre las predicciones entregadas por los modelos de redes neuronales. Por otro lado, el formato COCO permite la utilización de librerías especializadas en la lectura y escritura de anotaciones en formato COCO, como lo es la librería `pycocotools` [65].

Normalización

La normalización de imágenes es una técnica que permite estandarizar el rango de valores de los píxeles de una imagen, de manera que los valores de los pixeles se encuentren en un rango de 0 a 1. Esto permite que los modelos de redes neuronales converjan más rápido al momento de entrenarlos, además de evitar problemas de saturación en las capas de activación de las redes neuronales.

Cuando se utilizan modelos pre-entrenados, se recomienda normalizar las imágenes de entrada de acuerdo a los valores de los píxeles que fueron utilizados para entrenar los modelos. En el caso de los modelos YOLOv8 [66], Mask RCNN [51], Fast RCNN [30] y Retina-Net [32] normalizan los datos con los valores RGB de la media y desviación estándar de ImageNet [58]. Por tanto, en este estudio se utiliza la misma práctica y se normalizan los datos con los mismos valores por cada canal, como se muestra en la Ecuación 4.1.

$$
I_{\text{norm}}(c) = \frac{I(c) - \mu(c)}{\sigma(c)} \tag{4.1}
$$

Donde $I_{\text{norm}}(c)$ es el valor normalizado del canal c de la imagen I, $\mu(c)$ es la media del canal c de ImageNet y $\sigma(c)$ es la desviación estándar del canal c de ImageNet.

4.3.3. Recorte especial en RPT3DS

Debido a que el conjunto de imágenes de RPT3DS es acotado, pero la cantidad de segmentaciones que posee es significativa y la dimensión de las imágenes es grande, se decidió realizar un recorte especial de las imágenes para aumentar la cantidad de datos de entrenamiento.

Los objetivos principales del recorte de las imágenes es disminuir la dimensión de las imágenes sin perder información de las entidades y lograr obtener a lo menos 2 conjuntos de imágenes que contengan la misma cantidad de entidades que el conjunto inicial.

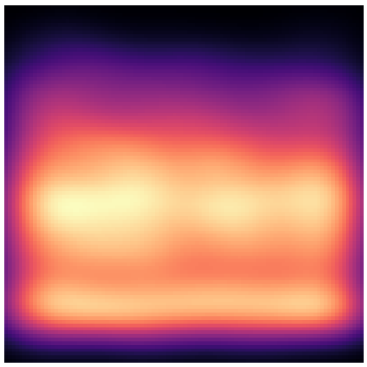


Figura 4.12: Distribución normalizada del centro de las máscaras del conjunto de datos RPT3DS. La distribución se obtiene al calcular el centro de cada máscara y normalizarlo respecto al ancho y alto de la imagen.

Para ello, luego de realizar la limpieza de datos descrita en Subsección 4.3.1, se estudió la distribución de los centros de las bounding box de las imágenes como se muestra en la Figura 4.12, en donde se observa que la densidad de los puntos se concentra horizontalmente en la imagen y disminuye a los bordes.

Debido a esto, la estrategia propuesta fue recortar la imagen en secciones horizontales. Sin embargo, tomando en cuenta que la cantidad de polígonos es menor cuando se presenta 1 sola entidad (Ver Figura 4.3) y en los experimentos se considerarán las entidades distintas entre sí por cada imagen (Sección 2.3, S5), fue necesario estudiar visualmente los casos bordes de los cortes a realizar.

Figura 4.13: Ejemplo de caso borde en el recorte de la imagen. La división en 3 o más partes de la imagen provoca que existan regiones sin entidades o con entidades incompletas.

En general, se observó que los casos bordes se presentan cuando la entidad se encuentra en el borde de la imagen, como se muestra en la Figura 4.13. Por tanto, se decidió realizar el recorte de las imágenes en 2 partes con la misma dimensión.

En cuanto a las máscaras presentes en la mitad de la imagen, se consideraron los casos

bordes donde las máscaras no poseen una visibilidad que permita capturar características relevantes de la entidad. El parámetro de mínima visibilidad (v) que se definió para considerar una máscara como no visible fue la proporción del área de la máscara recortada (A_c) respecto al área de la máscara original (A_i). Se consideró que una máscara es visible si:

$$\frac{A_c}{A_i} > 0.1$$

Otra alternativa a la estrategia de recortar las imágenes a la mitad es realizar recortes cercanos al bounding box de las entidades, sin embargo se decidió no considerar esta estrategia debido a que se deseaba mantener la información del contexto de la imagen y no solo de la entidad.

4.3.4. División del conjunto de datos

Una vez realizado el recorte de las imágenes, se procedió a dividir el conjunto de datos en 2 conjuntos de entrenamiento y validación. Para ello, se aplicaron distintas estrategias para las divisiones de los datos.

4.3.4.1. Estrategia estándar

Se dividió el dataset en 2 conjuntos de entrenamiento y validación sujeto a la condición de que las entidades presentes en el conjunto de validación sean un subconjunto del conjunto de entrenamiento. Para esta estrategia se dividió el dataset original en la proporción 60 % y 40 % para entrenamiento y validación respectivamente. Luego la mitad de las imágenes del conjunto de validación fueron agregadas al conjunto de entrenamiento, quedando en una proporción final de 80 % y 20 % para los conjuntos de entrenamiento y validación respectivamente.

Figura 4.14: Estrategia estándar de división de datos.

4.3.4.2. Estrategia *zero-shot*

En esta estrategia el objetivo es obtener dos conjuntos de datos que no posean entidades en común. Para ello, se dividió el conjunto de datos en 2 conjuntos de entrenamiento y validación de manera aleatoria en la proporción 80 % entrenamiento 20 % validación, donde la única condición es que las entidades del conjunto de validación no estén presentes en el conjunto de entrenamiento.

Figura 4.15: Estrategia *zero-shot* de división de datos.

4.3.5. Data augmentation

Una vez obtenidos los conjuntos de entrenamiento y validación se procede a realizar *data augmentation* sobre el conjunto de entrenamiento. La estrategia de *data augmentation* consiste en aplicar transformaciones sobre las imágenes para aumentar la variabilidad de los datos. Se ha demostrado que con esta estrategia es posible mejorar la generalización de los modelos de redes neuronales [67].

El proceso de *data augmentation* se aplica durante el proceso de entrenamiento de los modelos para evitar aumentar el tamaño del conjunto de datos y además mantener los datos originales. En este estudio se utilizan técnicas comunes en *data augmentation* para imágenes, como son el volteo horizontal, rotación, recorte, ajuste de brillo, contraste saturación, mosaico y copiar-pegar.

Las librerías utilizadas para el framework de Detectron2 [68] son `albumentations` [69] y `torchvision`. En cuanto al framework de YOLOv8 [66], las transformaciones vienen integradas en el framework en donde solamente es necesario configurar las probabilidades de ocurrencia de la transformación en cada batch de entrenamiento.

Volteo

El volteo de una imagen es una transformación que invierte la imagen horizontal o verticalmente. Dada una imagen I, un flip horizontal se puede representar como $I'(x, y) = I(w-x, y)$, donde w es el ancho de la imagen y (x, y) son las coordenadas de los píxeles.

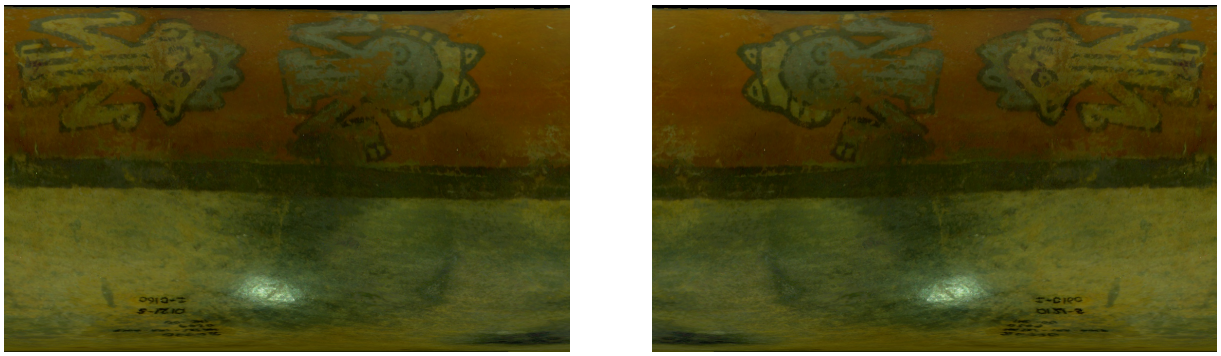


Figura 4.16: Ejemplo de volteo horizontal.

Rotación

La rotación de una imagen implica rotar la imagen por un ángulo θ. Dada una imagen I y un ángulo θ, la imagen rotada I' se calcula como $I'(x, y) = I(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$ para todos los píxeles en I. La rotación de una imagen puede provocar que la imagen resultante tenga un tamaño mayor al original, por lo que se debe considerar el tamaño de la imagen al momento de realizar la rotación. Para esta investigación se consideró realizar solamente rotaciones de 90 y 180, realizando recortes adicionales a la imagen para mantener un tamaño estándar para el conjunto de datos.

Figura 4.17: Ejemplo de rotación.

Mosaico

Esta técnica consiste en combinar N imágenes en 1 sola. Suelen utilizarse entre 4 a 9 imágenes las cuales son concatenadas formando una sola imagen. En este estudio se consideró utilizar 4 imágenes para formar un mosaico.

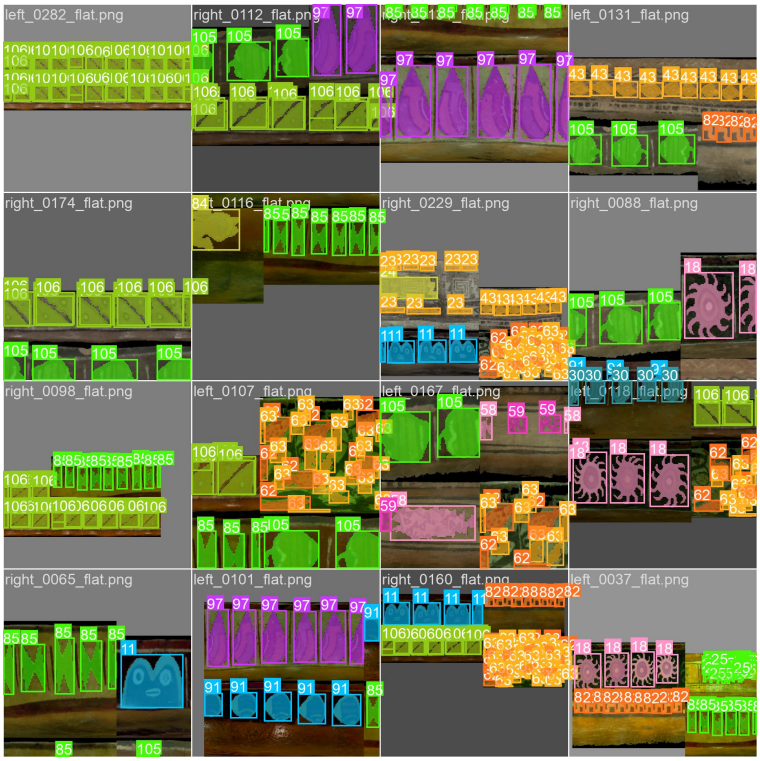


Figura 4.18: Ejemplo de mosaico combinado con otras estrategias

Copiar-pegar

Esta técnica implica copiar un segmento de una imagen y pegarlo en otra parte de la misma imagen o en otra imagen. Dadas dos imágenes I_1 e I_2, un segmento S de I_1 y una posición (x, y) en I_2, la imagen resultante I'_2 se calcula como $I'_2(x, y) = S(x, y)$.

Figura 4.19: Ejemplo de copiar-pegar de Ghiasi et al. [70]

Escala

Tranformación que cambia el tamaño de la imagen por un factor de escala. Dada una imagen I y un factor de escala s, la imagen escalada I' se calcula como $I'(x, y) = I(\frac{x}{s}, \frac{y}{s})$.

Recorte

El recorte implica seleccionar una región de interés de la imagen y descartar el resto. Dado que el conjunto de datos RPT3DS ya posee un recorte especial, esta estrategia se considerará con una menor probabilidad de ocurrencia.

Ajuste de brillo, contraste y saturación

El brillo y el contraste de una imagen se pueden ajustar multiplicando cada píxel por un factor de escala y luego sumando una constante. Dada una imagen I, un factor de escala s y una constante c, la imagen resultante I' se calcula como $I'(x, y) = sI(x, y) + c$. El factor de escala s se puede utilizar para ajustar el contraste de la imagen, mientras que la constante c se puede utilizar para ajustar el brillo de la imagen.

En cuanto a la saturación, dada una imagen I se convierte cada píxel de la imagen al espacio de color HSV. Luego, se multiplica solo el componente de saturación (S) por un factor de escala s. Finalmente, se convierte la imagen de vuelta al espacio de color RGB.

Figura 4.20: Ejemplo de ajuste de brillo.

Figura 4.21: Ejemplo de ajuste de saturación.

Figura 4.22: Ejemplo de ajuste de contraste.

4.4. Técnicas de visión

Para realizar la segmentación de los objetos del dataset de RPT3DS se utilizan técnicas basadas en algoritmos tradicionales de visión computacional y técnicas basadas en redes neuronales. Como se mencionó en la hipótesis, el enfoque de la investigación será el uso de técnicas basadas en redes neuronales, sin embargo, se utilizarán técnicas basadas en algorit-

mos tradicionales para comparar las métricas de los modelos de redes neuronales.

Se propone el uso de las técnicas de detección y segmentación de objetos en el proceso donde la textura de la imagen está aplanada y se posee una imagen 2D completa con los patrones de la imagen. En el proceso se entrenará a una red convolucional con las estrategias mencionadas en la Subsección 4.3.4 para que logre aprender las características de los patrones de los objetos. Luego, se unirán las imágenes que fueron recortadas para reconstruir la imagen segmentada (ver Figura 4.23).

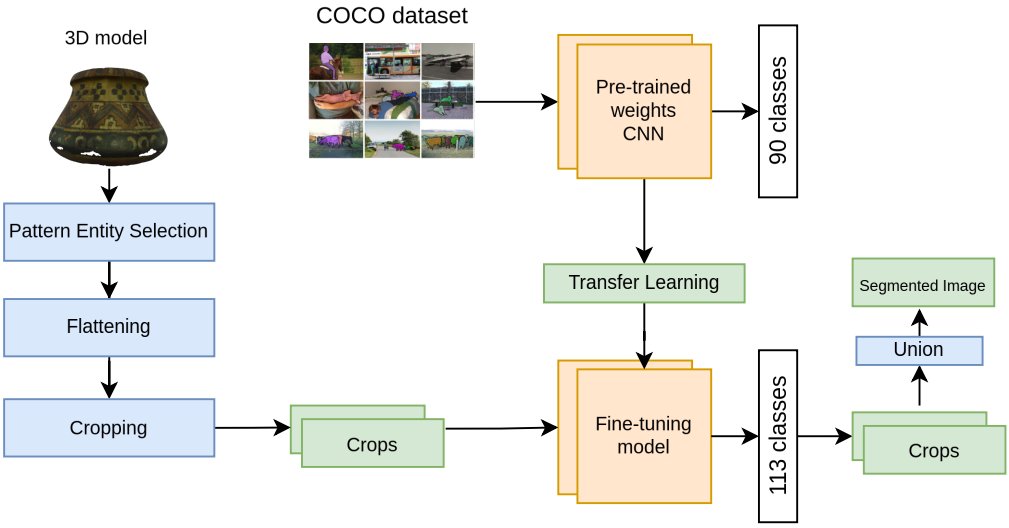


Figura 4.23: Flujo de técnicas a utilizar para realizar segmentación de patrones.

El conjunto de datos de RPT3DS posee segmentaciones y bounding boxes de las entidades, sin embargo no poseen información sobre el significado semántico de cada entidad, por lo que se utilizarán técnicas de detección de objetos y segmentación de instancias. De esta manera se buscará identificar a cada objeto diferenciándolo entre sí a pesar de su similitud o que posean el mismo significado semántico.

4.4.1. Algoritmos tradicionales

Los algoritmos tradicionales a considerar se basan en detección de objetos y segmentación. Debido a que no se posee una segmentación completa de la imagen, solamente se medirán métricas de los algoritmos de detección de objetos, mientras que los algoritmos tradicionales de segmentación se mostrarán imágenes que permitan visualizar la segmentación de los objetos.

El algoritmo de detección o query-detection que se utilizará será Template Matching [71] [72], el cual consiste en comparar una imagen con un template o patrón.

En segmentación de objetos se utilizará el algoritmo de Watershed [46], el cual consiste en realizar una segmentación de la imagen basada en la topografía que posee. En conjunto con este algoritmo se utilizará el algoritmo de detección de bordes Canny [48], el cual permite

detectar los bordes de los objetos de la imagen y utilizarlos como entrada para el algoritmo de Watershed.

4.4.2. Algoritmos basados en redes neuronales

Detección de objetos

En detección de objetos se tomará el enfoque de utilizar redes neuronales convolucionales. Para ello se utilizarán los modelos de YOLOV8 [66] y RetinaNet [32]. En ambos modelos se utilizará el enfoque de aprendizaje supervisado, en donde se configurarán conjuntos de datos para ser entrenados y evaluados mediante el conjunto de validación. Ambos modelos poseen pesos pre-entrenados en el dataset COCO [2], por lo que se utilizarán como punto de partida para el entrenamiento de los modelos.

Segmentación de instancias

En segmentación de instancias se utilizarán los modelos de Mask RCNN [51] y Faster RCNN [30]. Ambos modelos también están basados en redes convolucionales y poseen pesos pre-entrenados en el dataset COCO [2], por lo que se utilizarán como punto de partida para el entrenamiento de los modelos.

Por otro lado, se utilizará el modelo de Segment Anything [60] como modelo base para comparar las métricas de los modelos de segmentación y detección de objetos. En este modelo se utilizará el enfoque *zero-shot* para realizar la detección de objetos.

4.5. Evaluación de Modelos

Para evaluar los resultados obtenidos por los modelos de detección de objetos y segmentación de instancias, se utilizaron métricas estándar propuestas en COCO [2, 30] y PASCAL VOC [34]. Estas métricas corresponden al promedio ponderado de precisión (mAP) y al promedio ponderado de recall (mAR). Las ecuaciones con las que se calculan las métricas son las siguientes:

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN} \tag{4.2}$$

$$AP = \frac{1}{n}\sum_{i=1}^{n} P_i,\ AR = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{4.3}$$

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i,\ mAR = \frac{1}{c}\sum_{i=1}^{c} AR_i \tag{4.4}$$

En donde P y R corresponden a la precisión y recall respectivamente, TP son los verdaderos positivos, FP son los falsos positivos, FN son los falsos negativos, AP es el promedio de precisión, AR es el promedio de recall, n es la cantidad de imágenes y c es la cantidad de clases.

La precisión y el recall se calculan a partir del IoU entre los bounding boxes o máscaras

predichas por el modelo con respecto al ground truth. Para calcular las métricas relativas a la detección de objetos se utiliza el acrónimo (bb), que hace referencia a los bounding boxes, tanto para AP como para AR. De manera similar, el acrónimo (s) se utiliza para la segmentación de objetos.

En las métricas COCO, la medida de AP y AR se calcula con los threshold 50 % (basado en la métrica PASCAL VOC [34]) y 70 % (una medida más estricta) de corte de IoU para determinar los verdaderos positivos. Por otro lado, existe la métrica primaria de COCO la cual se calcula como el promedio entre el 50 % y el 95 % de IoU con un incremento del 5 % (Ec. 4.5).

$$\mathrm{AP}^{IoU=0.50:0.95} = \frac{1}{10}\sum_{r=0}^{9}\mathrm{AP}^{IoU=0.5+r\cdot 0.05} \tag{4.5}$$

- IoU @ 50 %: Este es el umbral más popular debido a que permite una tolerancia a pequeños errores en la localización del objeto. Además, permite realizar un balance entre dificultad y utilidad, especialmente en casos complejos de imágenes con objetos pequeños o con mucha oclusión, donde es más difícil que los modelos o algoritmos de detección obtengan detecciones perfectas. Un verdadero positivo se cuenta cuando la IoU entre la predicción del modelo y el ground truth es igual o superior al 50 %. Esto significa que al menos la mitad del área de la predicción del modelo debe solaparse con el área real del objeto.
- IoU @ 75 %: Este es un umbral más estricto. Un verdadero positivo se cuenta cuando la IoU entre la predicción del modelo y la anotación verdadera es igual o superior al 75 %. Este umbral exige una mayor precisión en la predicción, ya que requiere que tres cuartas partes de la predicción del modelo coincidan con el área real del objeto.

Para la detección de objetos, se utiliza el área del bounding box predicho por el modelo y el bounding box del ground truth. En el caso de la segmentación de instancias, se utiliza el IoU del área de la máscara predicha con respecto a la máscara del ground truth.

Debido a que existen figuras que tienen similitudes entre sí y pertenecen a clases distintas por la definición que se asumió en esta investigación (S6, Sección 2.3), es posible que las predicciones sean muy estrictas respecto al elemento a detectar. Esto puede generar que la precisión de los modelos nunca pueda alcanzar un óptimo. Sin embargo, se espera que los modelos que logren aprender las características de los motivos puedan obtener un recall cercano a uno.

La similitud entre algunas figuras que se desean detectar conlleva a eligir una métrica que no sea tan estricta entre máscaras cercanas entre sí. Por esta razón, la métrica que se considera utilziar para comparar los resultados entre los modelos es la AP_{50}, la cual es la métrica de AP calculada con un IoU de 50 %. Esta métrica es adecuada para objetos de tamaños variados [34] y tienen menor sensibilidad a errores pequeños.

La elección de considerar solamente las métricas de mAP y mAR es debido a que la tarea que se busca resolver es segmentación de instancias, donde el objetivo es lograr detectar la mayor cantidad de instancias posibles con la mayor precisión posible. El uso de mIoU no es

considerado debido a que el problema abordado no es de segmentación semántica ni tampoco se busca enfocar el problema solamente en segmentación, sino que se busca estudiar la precisión del los modelos en la detección de objetos repetidos en la escena completa de la textura.

La métrica del coeficiente de Dice no se considera debido a que en este tipo de objetos, los cuales son arte abstracto usualmente pintado a mano, la exactitud en la segmentación predicha no se considera relevante debido al deterioro que pueden poseer algunas entidades y a la dificultad en construir un ground truth que se ajuste exactamente a la forma de la entidad.

Por otro lado, la métrica de panoptic quality (PQ) [42] no se considera debido a que no se poseen las segmentaciones de toda la textura en RPT3DS, por lo que el objetivo de la segmentación panóptica no se lograría con los datos que se tienen disponible. Aunque existen técnicas que permiten transformar el problema de segmentación de instancias a segmentación panóptica, estas no son consideradas debido a que en las imágenes de RPT3DS el fondo puede contener otros patrones que poseen perdidas de texturas o simplemente no fueron considerados por el arqueólogo.

4.6. Estrategias de etiquetación

Debido a que los datos que se poseen en RPT3DS no poseen etiqueta semántica, pero se observa que existen similitudes en las formas de los objetos, se planifica estudiar dos estrategias de etiquetación para entrenar los modelos:

- One-Class: En esta estrategia se le asigna una única clase a todos los objetos de la imagen. De esta manera, se espera que el modelo sea capaz de detectar y segmentar objetos que posean similitudes en la forma y disminuir la cantidad de falsos positivos. También, se espera que el modelo sea capaz de detectar nuevos objetos que no están presentes en el ground truth como objetos segmentados.
- Multi-Class: En esta estrategia se le asigna una clase distinta a cada objeto de la imagen. A pesar de que existen objetos similares en forma entre distintas imágenes estos serán diferenciados por el ID inicial que poseen en el ground truth. Se espera que el modelo sea capaz de segmentar correctamente los objetos de una misma imagen y además clasificarlos distintos respecto a la etiqueta del ground truth. Sin embargo, el modelo podría contener errores al clasificar a objetos similares de distinta imagen con la misma etiqueta. Esto se podría observar al ver disminuir el AP respecto al AR.

Capítulo 5

Experimentos

Para poder evaluar el desempeño de los modelos propuestos, se realizaron una serie de experimentos variando el conjunto de datos de entrenamiento, los modelos y algoritmos a utilizar y el objetivo de la detección y segmentación de los objetos. En este capítulo se detallan los experimentos realizados, la arquitectura de los modelos, las configuraciones iniciales y resultados preliminares que serán utilizados para la evaluación de los modelos.

5.1. Baselines

Para comparar los resultados que entregan los modelos de redes neuronales supervisados entrenados con los datos de RPT3DS, se utilizaron los resultados obtenidos por dos métodos. El primer método utilizado fue el de *template matching*, un algoritmo tradicional utilizado para la detección de objetos mediante una consulta y el segundo fue el modelo de *Segment Anything* [60] el cual es un modelo fundacional basado en ViT capaz de segmentar objetos en una imagen sin ser entrenado mediante el uso de *prompts* o instrucciones.

5.1.1. Template Matching

El algoritmo de *template matching* [71] es un método de visión por computadora utilizada para encontrar regiones similares o coincidentes dentro de una imagen de destino en función de una imagen de plantilla (patrón) predefinida.

Los pasos del algoritmo son los siguientes:

1. Seleccionar una imagen de referencia y una imagen objetivo, la cual puede o no contener la imagen de referencia.
2. Definir un threshold de coincidencia. Este threshold permitirá determinar si la imagen de referencia posee una coincidencia con la imagen objetivo según la medida de coincidencia que se utilice.
3. Se desliza la imagen de referencia por toda la imagen objetivo. El espacio de deslizamiento es un parámetro del algoritmo, por lo que es necesario definirlo previamente.
4. Se realiza el cálculo de la medida de coincidencia entre la imagen de referencia y la imagen objetivo. La manera en la que se puede calcular es mediante los siguientes métodos:

- Diferencia cuadrada (disimilitud): Calcula la diferencia cuadrada entre los píxeles de la imagen de referencia y la imagen objetivo. Un valor alto indica que las imágenes son diferentes. La versión normalizada de la diferencia cuadrada se calcula dividiendo la diferencia cuadrada por la suma de los cuadrados de los píxeles de la imagen de referencia. La fórmula para calcular la diferencia cuadrada es la siguiente:

$$D(i,j) = \sum_{x,y}[I(x,y) - T(x+i, y+j)]^2 \tag{5.1}$$

$$D_n(i,j) = \frac{D(i,j)}{\sqrt{\sum_{x,y} T(x,y)^2 \cdot \sum_{x,y} I(x+i, y+j)^2}} \tag{5.2}$$

  Donde I es la imagen de referencia, T es la imagen objetivo, x e y son las coordenadas de los píxeles, e i y j son los parámetros de deslizamiento.

- Correlación cruzada (similitud): Calcula la correlación cruzada entre la imagen de referencia y la imagen objetivo. Un valor alto indica que las imágenes son similares. La versión normalizada de la correlación cruzada se calcula dividiendo la correlación cruzada por la raíz cuadrada de la suma de los cuadrados de los píxeles de la imagen de referencia. La fórmula para calcular la correlación cruzada es la siguiente:

$$C(i,j) = \sum_{x,y}[I(x,y) - \bar{I}][T(x+i, y+j) - \bar{T}] \tag{5.3}$$

$$C_n(i,j) = \frac{C(i,j)}{\sqrt{\sum_{x,y} T(x,y)^2 \cdot \sum_{x,y} I(x,y)^2}} \tag{5.4}$$

  Donde I es la imagen de referencia, T es la imagen objetivo, x e y son las coordenadas de los píxeles, e i y j son los parámetros de deslizamiento. $\bar{I}$ y $\bar{T}$ son los promedios de los valores de los píxeles de las imágenes de referencia y objetivo respectivamente.

5. Seleccionar la región si el valor de coincidencia es mayor que el threshold definido.
6. Se repite el proceso hasta que se recorre toda la imagen objetivo.

Una vez se obtienen las regiones de interés es necesario realizar un post-procesamiento para eliminar las detecciones que se superponen debido a la ventana de deslizamiento. Para ello se utiliza el algoritmo de Non Maximum Supression (NMS) [73] el cual permite eliminar las detecciones que se superponen respecto a un threshold de overlap.

Detalles de la implementación

La medida utilizada es la correlación cruzada normalizada debido a que es la medida de similitud que permite definir un threshold entre 0 y 1. El patrón es tomado al azar por cada imagen y se utiliza la misma imagen a la que pertenece el patrón como objetivo.

Para la implementación del algoritmo se utilizó la biblioteca OpenCV, en donde es aplicado un deslizamiento del patrón de 1 píxel y un threshold de 0.5 para la medida de correlación cruzada normalizada. También se configura el algoritmo de NMS con un threshold de 0.5 de IoU para eliminar las detecciones que se superponen, además del threshold ya definido para la correlación cruzada.

El algoritmo NMS a utilizar se define de la siguiente manera:

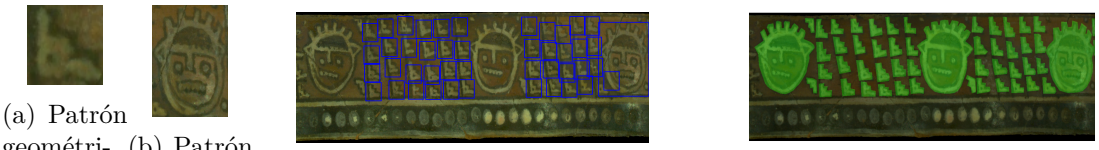
Algorithm 1 Non-Maximum Suppression

```
procedure NMS(Boxes, ScoreThreshold, IoUThreshold)
    selectedBoxes ← ∅
    sortedBoxes ← sort Boxes by score in descending order
    for each box in sortedBoxes do
        if score of box < ScoreThreshold then
            continue
        end if
        isMax ← true
        for each selectedBox in selectedBoxes do
            if IoU(box, selectedBox) > IoUThreshold then
                isMax ← false
                break
            end if
        end for
        if isMax then
            add box to selectedBoxes
        end if
    end for
    return selectedBoxes
end procedure
```

Resultados previos

Los resultados obtenidos con el algoritmo de template matching se pueden ver en la Tabla 5.1. En la Figura 5.1 se puede ver un ejemplo donde el algoritmo de template matching logra detectar correctamente el patrón en la imagen objetivo para un patrón de los dos que se encuentran en la imagen. El patrón (b) de la Figura 5.1 presenta complejidades debido a la representación de una aparente cara humana.

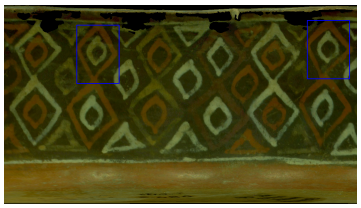
Por otro lado, en la Figura 5.2 se puede ver un ejemplo donde el algoritmo de template matching no logra detectar ningún patrón de la imagen objetivo, a pesar de la forma geométrica del patrón.



(a) Patrón geométrico (b) Patrón cara (c) Predicción TM (d) Ground truth

Figura 5.1: Ejemplo de resultado del algoritmo Template Matching. Imagen con 2 patrones. El primer patrón (a) corresponde a un objeto geométrico que tiene un buen resultado en la detección a pesar de que su tamaño es pequeño y que su forma cambia. El segundo patrón (b) corresponde a una figura compleja debido a los detalles y a que su forma varía levemente.

(a) Rombo



(b) Imagen objetivo

(c) Ground truth

Figura 5.2: Ejemplo de resultado de algoritmo de Template Matching. Imagen con 1 patrón. El patrón (a) corresponde a un objeto geométrico que tiene un mal resultado en la detección en la imagen objetivo (b). Esto se puede deber principalmente a que la forma y el color del objeto varía respecto a cada patrón similar en la imagen.

5.1.2. Segment Anything Model

Segment Anything [60] (SAM) es un modelo de visión que permite la detección y segmentación de objetos utilizando *prompts* o instrucciones, los cuales pueden ser puntos de coordenadas, cajas demilitadoras (bounding boxes), máscaras binarias o texto. Este modelo, basado en Vision Transformers (ViT) [9], permite por medio de los prompts, detectar y segmentar objetos en imágenes sin necesidad de ser entrenado con datos de entrenamiento. Este proceso es conocido como *zero-shot transfer* y proporciona una nueva forma de realizar segmentación sobre conjuntos de datos son segmentar.

El modelo de SAM se compone de tres componentes principales: un prompt encoder, un image encoder y un lightweight mask decoder (ver Figura 5.3). El image encoder utiliza un Mask Autoencoder (MAE) [74] para obtener una representación de la imagen. El MAE fue pre-entrenado con un ViT mínimamente adaptado para inputs de alta resolución. El MAE se compone de un encoder y un decoder, donde el encoder es una red neuronal convolucional que codifica la imagen en un espacio latente y el decoder es una red neuronal convolucional que decodifica el espacio latente en una máscara binaria.

El prompt encoder se divide en dos conjuntos: dispersos (puntos, bounding boxes, texto) y denso (máscaras binarias). Los puntos y cajas son representados mediante codificaciones posicionales, además de embeddings aprendidos por cada tipo de prompt. En cuanto al texto se procesa utilizando un codificador de texto independiente proveniente de CLIP [57]. Los prompts densos se incrustan utilizando convoluciones y se suman con el embedding de la imagen.

Por último, el mask decoder emplea la modificación de un block decoder Transformer [75] el cual utiliza self-attention y cross-attention en dos direcciones (prompt a imagen y viceversa) para actualizar todos los embeddings. Luego, se realiza un upsampling al embedding de la imagen y un multi layer perceptron (MLP) mapea el token de salida a un clasificador lineal dinámico.

Debido a que los prompts pueden ser ambiguos el modelo promedia múltiples máscaras válidas. De esta manera, el modelo predice múltiples máscaras para un prompt dado (ver Figura 5.3). En particular, la cantidad máxima de máscaras predichas son 3 por cada prompt. Por

otro lado, para “rankear” las máscaras, el modelo predice un confidence score (IoU estimado) por cada máscara.

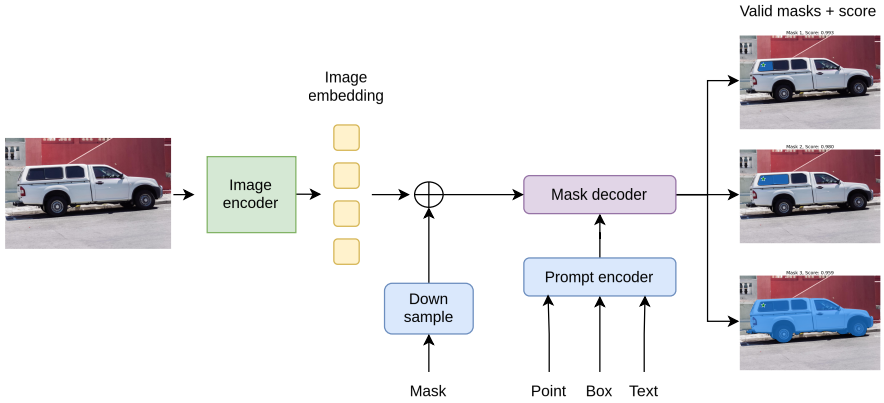


Figura 5.3: Secciones principales de la arquitectura de SAM [60]. El modelo utiliza un embedding de la imagen para ser eficientemente consultado por la variedad de prompts y predecir máscaras.

SAM también fue utilizado para la construcción de un conjunto de imágenes segmentadas por medio de la segmentación automática mediante el uso de *prompts* de puntos aleatorios. Este proceso tuvo el seudónimo de Fully Automatic Stage (FAS). Para ello, el prompt proporcionado al modelo utiliza una grilla de 32×32 puntos en toda la imagen y por cada punto se predice un conjunto de máscaras que podrían corresponder a objetos válidos. Para seleccionar las máscaras que se acerquen a un objeto válido, se filtran las máscaras que poseen mayor confidence score (IoU estimado) y se realiza un post-procesamiento con NMS para eliminar las máscaras que se superponen.

Detalles de la implementación

Para realizar la segmentación automática sobre las imágenes de RPT3DS se utilizaron las imágenes reducidas a la mitad como se mencionó en la Sección 4.3. Para la detección automática se utilizó la estrategia propuesta por los autores y se mantuvieron las configuraciones iniciales del modelo, modificando solamente el tamaño de la grilla de puntos a 64×64. Por otro lado, se utiliza el modelo pre-entrenado ViT-H el cual posee la mayor cantidad de parámetros, pero también el que posee más tiempo de inferencia.

Adicionalmente, una vez se obtiene la máscara *zero-shot*, se realiza un post-procesamiento que calcula la media y desviación estándar de las áreas de las bounding boxes detectadas por el modelo, eliminando los valores extremos definidos como aquellos que superan $\mu \pm \sigma$, los cuales son considerados como outliers.

5.1.3. Resultados previos

La forma en la que se evalúa el modelo de SAM con los datos de RPT3DS es considerando una etiqueta general para todas las instancias segmentadas por SAM. Una breve muestra de los resultados obtenidos al realizar el post-procesamiento a las predicciones de SAM se muestran en la Figura 5.4, en donde se seleccionaron tres ejemplos de las capacidades del

modelo.

En la primera imagen se observa que las máscaras predichas por SAM son muy similares a las máscaras de ground truth, en donde se segmentaron máscaras de objetos circulares que también forman un patrón, pero que no se encuentra en el ground truth. En la segunda imagen si bien se logra segmentar pequeños patrones cuadrados que existen dentro de cada triángulo, no son segmentadas las instancias de interés, por lo que el AP es muy bajo. En la tercera imagen se observa una predicción de la región donde se observan patrones. Esto se puede deber a que los tamaños de los objetos son pequeños y también a la cercanía de los objetos.

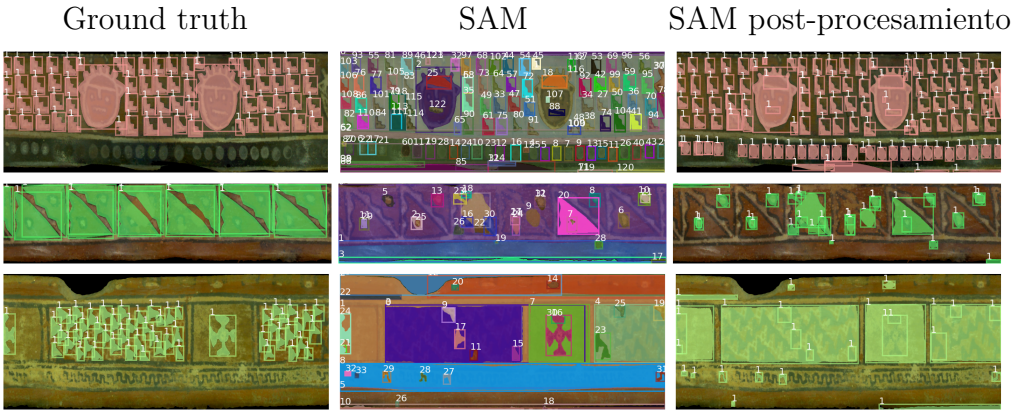


Figura 5.4: Ejemplo de detección realizada por SAM. De izquierda a derecha, ground truth de RPT3DS, resultado de la detección automática realizada por SAM y resultado del post-procesamiento eliminando áreas outliers y asignándoles la misma etiqueta a cada objeto detectado.

Las métricas obtenidas con los baselines se pueden ver en la Tabla 5.1. Ambos métodos fueron evaluados utilizando las métricas de COCO [2], en donde se utilizó una etiqueta general para los objetos predichos por los dos métodos. Según los resultados obtenidos, el método de template matching obtiene mejores resultados en las métricas de precisión en comparación con el modelo de SAM. Esto se puede deber a la gran cantidad de falsos positivos que no fue posible limpiar con el post-procesamiento propuesto en el modelo de SAM. Sin embargo, en recall, se observa que el modelo de SAM obtiene resultados considerablemente mejores que template matching.

En cuanto al proceso de post-procesamiento en SAM, se observa que reduce el recall obtenido inicialmente y aumenta en las métricas de precisión. Aunque las diferencias en ambos casos son mínimas, en los resultados visuales se observa que las máscaras muy pequeñas y aquellas que abarcan regiones largas de la imagen son removidas, cumpliendo el objetivo buscado de eliminar outliers.

5.2. Modelos de CNN

Con los conjuntos de entrenamiento y validación, se evaluaron cuatro redes neuronales convolucionales del estado del arte en el problema de detección y segmentación de objetos:

Tabla 5.1: Métricas obtenidas con los baselines: Template Matching, SAM 64×64 con estrategia de grilla de puntos asignados aleatoriamente, y con post-procesamiento eliminando outliers de áreas. Cada método fue evaluado sobre todo el dataset RPT3DS.

Method	$\mathbf{AP}^{bb}$	$\mathbf{AR}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{s}$	$\mathbf{AR}^{s}$	$\mathbf{AP}^{s}_{50}$	$\mathbf{AP}^{s}_{75}$
Template Matching [71]	**18.6**	22.5	**34.9**	**17**.1	-	-	-	-
SAM (64×64) [60]	12.1	**54.9**	18.3	12.9	10.5	**50.9**	17.55	11.42
SAM post-procesamiento (64×64) [60]	13.1	54.0	19.9	14.0	**11.4**	49.9	**19.02**	**12.32**

Mask R-CNN [51], Faster R-CNN [30], RetinaNet [32], y YOLOv8-s [66]. Para los cuatro modelos, se utilizaron pesos pre-entrenados con el conjunto de datos COCO.

La elección de utilizar los pesos pre-entrenados con el dataset de COCO fue debido a que las formas de los objetos son variadas, sus tamaños son similares en proporción con las imágenes a las del dataset RPT3DS y el número de categorías es cercano, donde COCO tiene 91 y RPT3DS 113 categorías.

5.2.1. Residual Network

Antes de introducir los modelos de detección y segmentación de objetos, es necesario explicar la arquitectura ResNet [17] la cual es utilizada como columna vertebral (backbone) para los modelos de Mask R-CNN, Faster R-CNN y RetinaNet.

ResNet o Residual Network es una arquitectura de red neuronal convolucional compuesta de bloques residuales. Cada bloque residual se compone de conexiones de salto (skip connections) los cuales ayudan a mitigar el problema de desvanecimiento de gradientes (vanishing gradient) y permite entrenar redes neuronales más profundas.

Las ResNet se basan en la arquitectura de VGGNet [76] en donde se utilizan capas convolucionales de 3×3 y capas de pooling de 2×2. La diferencia principal entre ambas arquitecturas es que ResNet utiliza bloques residuales y VGGNet utiliza capas convolucionales de 3×3.

En la Figura 5.5 se puede ver la arquitectura de un bloque residual, en donde se observa qu|e la entrada x es sumada a la salida de la capa convolucional $F(x)$. La suma de la entrada con la salida de la capa convolucional se denomina conexión de salto (skip connection).

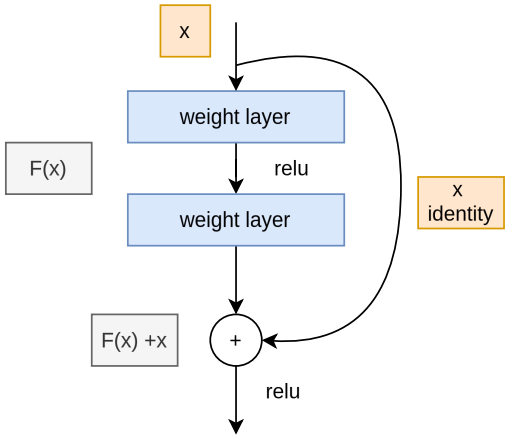


Figura 5.5: Arquitectura de un bloque residual [17].

Un bloque residual o building block se define como:

$$y = F(x, \{W_i\}) + x \tag{5.5}$$

Donde x es la entrada, y es la salida, F es la función residual, y W_i son los pesos de la capa convolucional. La función $F(x, \{W_i\})$ representa el mapeo que se aprende en el bloque residual.

La red residual se compone de dos capas convolucionales de 3×3, una capa de Batch Normalization [77] y una función de activación ReLU [78]. La capa de Batch Normalization normaliza la entrada de una capa de activación anterior, es decir, normaliza la entrada de la función de activación ReLU.

Las arquitecturas de ResNet que suelen ser utilizadas son ResNet-50, ResNet-101 y ResNet-152. Estas arquitecturas se diferencian en la cantidad de bloques residuales que poseen, donde ResNet-50 posee 50 bloques residuales, ResNet-101 posee 101 bloques residuales y ResNet-152 posee 152 bloques residuales.

En cuanto a la entrada de las arquitecturas de ResNet, estas suelen ser imágenes de 224×224, 256×256 o 299×299. En el caso de las arquitecturas utilizadas para los modelos de detección y segmentación de objetos, la entrada es de 224×224. La salida de las arquitecturas de ResNet está compuesta por un vector cuyo tamaño depende de la cantidad de clases que se desean predecir.

5.2.2. Feature Pyramid Network

Feature Pyramid Network (FPN) [79] es una arquitectura de red neuronal convolucional que permite la extracción de características de múltiples escalas. FPN se compone de un bottom-up, un top-down pathway y conexiones laterales. El bottom-up pathway representa el backbone del modelo, el cual puede ser cualquier red convolucional, el cual se encarga de extraer características de la imagen de entrada. El top-down pathway refina las características obtenidas por el bottom-up pathway y las fusiona con las características de la misma escala con las conexiones laterales, las cuales se obtienen mediante una capa convolucional de 1×1.

La componente bottom-up se suele componer de una ResNet, en donde se recibe la imagen de entrada generando el mapa de características en diferentes escalas de la red. En cada etapa del bottom-up el mapa de características se reduce, usualmente a la mitad, pero la semántica de las características aumenta al aumentar la profundidad de la red.

En la componente top-down se realiza un upsampling de las características de la etapa anterior debido a la baja resolución, estimando cómo se verían las características en la escala actual. Luego, se realiza una fusión de las características de la misma escala con las características de bottom-up mediante una suma y a través de la componente lateral.

El proceso de top-down se repite hasta que se llega a la escala más alta de la pirámide de características. Una vez se completa el proceso para todos los niveles, se obtiene un conjunto final de mapas de características de diferentes escalas pero enriquecidos con la información tanto de alta como de baja resolución.

Los mapas de características obtenidos por FPN son utilizados para la detección y segmentación de objetos. En el caso de la detección de objetos, se utiliza un Region Proposal Network (RPN) [30] para generar propuestas de regiones de interés (RoI) y se utiliza un clasificador para clasificar las propuestas de región de interés. Para segmentación de objetos, se utiliza una rama de predicción de máscara para generar una máscara binaria para cada región de interés.

Se utiliza una rama de predicción de máscara para generar una máscara binaria para cada región de interés. La rama de predicción de máscara se compone de una capa convolucional de 28×28, una capa de pooling de 2×2, una capa convolucional de 14×14, una capa de pooling de 2×2, una capa convolucional de 7×7 y una capa de pooling de 2×2.

5.2.3. Retina Net

RetinaNet [32] es una arquitectura de red neuronal convolucional one-stage para la detección de objetos. La arquitectura original de RetinaNet se compone de un backbone ResNet [17] combinado con una FPN [79] y dos subredes: una subred de box regression y una subred de clasificación.

El modelo utiliza anchors (anclas) para generar propuestas de regiones de interés (RoI). Los anchors son bounding boxes de diferentes tamaños y relaciones de aspecto que se deslizan por la imagen de entrada.

Subred de clasificación

La subred de clasificación predice la probabilidad de la presencia de un objeto en cada anchor A y clases K. Esta subred está compuesta por un pequeño Fully Convolutional Network (FCN) [62] que se aplica a cada nivel de la FPN.

Subred de box regression

La subred de box regression está en paralelo con la subred de clasificación y predice las coordenadas de las bounding boxes para cada anchor box hasta un objeto del ground truth,

si existe alguno. La subred de box regression se compone de un FCN que se aplica a cada nivel de la FPN. La estructura de la subred de box regression es similar a la subred de clasificación, pero en lugar de predecir la probabilidad de la presencia de un objeto, predice las coordenadas de las bounding boxes.

Focal loss

El modelo además utiliza una estrategia de focal loss [32] para resolver el problema de desbalance de clases. El problema de desbalance de clases se genera cuando la mayoría de las regiones de interés propuestas por el modelo no contienen objetos de interés. La focal loss se define como:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{5.6}$$

Donde p_t es la probabilidad predicha por el modelo, γ es un parámetro de focalidad que controla la velocidad con la que disminuye la contribución de los ejemplos que el modelo ya puede clasificar con precisión (fáciles) y cuya valor es más alto para que la pérdida se concentre más en los ejemplos difíciles; y α es un factor de equilibrio que pondera la importancia de cada clase.

El valor de γ y α son hiperparámetros que se pueden ajustar. Los valores que han mostrado mejores resultados son $\gamma = 2$ y $\alpha = 0.25$.

En resumen, la focal loss se utiliza para reducir el peso de los ejemplos fáciles y aumentar el peso de los ejemplos difíciles. Esto permite que el modelo detecte mejor los ejemplos difíciles y además soluciona el problema de desbalance de clases.

5.2.4. Faster R-CNN

El modelo Faster R-CNN [30] es una red neuronal convolucional two-stage enfocada en la tarea de detección de objetos. La arquitectura original de Faster R-CNN se compone de un backbone ResNet [17], un Region Proposal Network (RPN) [30] y un clasificador de objetos.

A diferencia de los modelos one-stage, los modelos two-stage se componen de dos etapas: una etapa de generación de propuestas de regiones de interés (RoI) y una etapa de clasificación de objetos.

ResNet es el encargado de extraer características de la imagen completa. Luego, sobre las características extraídas por el ResNet, se aplica un RPN para generar propuestas de regiones de interés (RoI).

Region Proposal Network

El RPN funciona escaneando la imagen con una ventana deslizante y mapeando la imagen a una serie de características. Luego, se utiliza una capa convolucional de 3×3 para generar un conjunto de anclas (anchors) de diferentes tamaños y relaciones de aspecto. Los anchors se clasifican según si tienen un objeto o no y se regresan las coordenadas de las bounding boxes que contienen un objeto.

Clasificador de objetos

Las características extraídas por el ResNet son utilizadas por el clasificador de objetos para clasificar las propuestas de regiones de interés (RoI) generadas por el RPN. El clasificador de objetos se compone de una capa de pooling, una capa de un Fully Connected Network (FCN) y una capa de clasificación Softmax.

La capa de pooling se utiliza para reducir la dimensionalidad de las características extraídas por el ResNet. Luego, la capa de FCN se utiliza para extraer características de las propuestas de regiones de interés (RoI) generadas por el RPN. Por último, la capa de clasificación Softmax se utiliza para clasificar las propuestas de regiones de interés (RoI) en las diferentes categorías.

5.2.5. Mask R-CNN

Mask R-CNN [51] es una arquitectura de red neuronal convolucional que extiende a Faster R-CNN [30] agregando una rama de predicción de máscaras y una capa de ROIAlign. Al igual que Faster R-CNN, Mask R-CNN se compone de un backbone ResNet [17], un Region Proposal Network (RPN) [30] y un clasificador de objetos y mantiene la misma estructura.

Luego de obtener los RoIs por medio del RPN, se predice una máscara binaria para cada RoI. Este proceso se realiza paralelo al proceso de clasificación y detección de objetos. La rama de máscara tiene una salida dimensional de $m \times m \times K$, donde m es el tamaño de la máscara y K es el número de clases. Para predecir la máscara mxm para cada RoI se utiliza una FCN, lo cual permite una mayor precisión y una menor cantidad de parámetros en la predicción de las máscaras en comparación al uso de las Fully Connected Layers.

En cuanto al uso de RoIAlign, este reemplaza al RoIPool [30] en la capa de clasificación y detección de objetos. RoIAlign corrige el problema de RoIPool, el cual no permite que las características de la imagen se alineen con las características de la máscara. RoIAlign utiliza interpolación bilineal para extraer características de la imagen y alinearlas con las características de la máscara.

5.2.6. YOLOv8

YOLOv8 [66] es una arquitectura de red neuronal convolucional one-stage para la detección, segmentación de objetos y detección key-point. La arquitectura original de YOLOv8 se compone de un backbone similar a YOLOv5 [80] y un Head, el cual corresponde a un conjunto de capas convolucionales las cuales se encargan de generar las predicciones de las bounding boxes y las clases de los objetos.

El backbone es una CSPLayer, CSPDarknet53 en su variante para segmentación, la cual agrega un modulo C2f [81] el cual es utilizado como extractor de características por el modelo. El modulo C2f tiene como objetivo reducir la cantidad de parámetros de la red y mejorar la eficiencia computacional. El modulo C2f continua con dos capas convolucionales para la segmentación y de las cuales se aprende la segmentación semántica de la imagen.

Anchor free detection

El modelo utiliza una estrategia de detección sin anchors (anchor free detection) para generar propuestas de regiones de interés (RoI). La estrategia de detección sin anchors se basa en la detección de objetos por medio de puntos de referencia (key-points). El modelo predice directamente el centro de un objeto en vez de predecir las coordenadas de las bounding boxes.

Esta arquitectura permite reducir el número de cajas predichas, lo cual acelera el proceso de NMS y permite que el modelo sea más rápido. Además, la estrategia de detección sin anchors permite que el modelo sea más robusto a la escala y la relación de aspecto de los objetos.

5.2.7. Detalles de Implementación

Para la implementación de las arquitecturas Mask R-CNN, Faster R-CNN y RetinaNet, se utilizó la biblioteca Detectron2 [68] de Meta, y se utilizó una ResNet-101 FPN [17] como backbone para los tres modelos. En cuanto a los parámetros, se redujo el tamaño del batch a 4 debido a limitaciones de memoria que existían al dejarlo en un valor más alto, se configuró la tasa de aprendizaje en 0.001 y el número de iteraciones en 1,000. Los otros parámetros se mantuvieron tal como se construyó el modelo inicialmente.

Además, se estableció en 256 las propuestas de regiones procesadas por imagen en un solo batch. Las técnicas de *data augmentation* utilizadas en cada batch incluyen volteos arriba-abajo, volteos izquierda-derecha, rotación (90º, 180º y 270º), brillo en el rango $[0.8, 1.8]$, contraste $[0.6, 1.3]$ y saturación $[0.8, 1.4]$ [70].

Para cada modelo se utilizaron los pesos pre-entrenados con el dataset COCO y se mantuvo el mismo backbone ResNet-101 FPN debido a sus buenos resultados en la predicción de objetos en el dataset de COCO.

Para el modelo YOLOv8, se utilizó la biblioteca `ultralytics` [66], donde se mantuvieron los parámetros de entrenamiento predeterminados, alterando solo el número de iteraciones a 300. El modelo pre-entrenado utilizado fue el modelo YOLOv8-s, que también fue entrenado con COCO.

Entre todos los modelos pre-entrenados de YOLOv8 (nano, small, medium, large, huge), cada uno varia respecto a la cantidad de datos con el que fue entrenado y por tanto la cantidad de parámetros es menor. Se escogió el segundo más pequeño debido al menor tiempo de ejecución y al rendimiento mostrado sobre el dataset de COCO, el cual no se aleja bastante de los modelos más pesados.

Además, se añadieron parámetros para aumentar la variabilidad de datos y realizar *data augmentation* en los lotes de entrenamiento. Estos incluyeron volteos arriba-abajo, volteos izquierda-derecha, copia y pegado de segmentos [82], escala de imágenes con una probabilidad del 50 %, mosaico con una probabilidad del 100 %, y aumento de imágenes en el espacio de color HSV (Matiz Saturación, Valor).

Capítulo 6

Resultados

En este capítulo se comparan los resultados obtenidos por los modelos de detección y segmentación de objetos entrenados con los datos de RPT3DS. Las métricas utilizadas para comparar los modelos son las mencionadas en la Sección 4.5.

Como se menciona en la Sección 4.2.1 uno de los desafíos presentes en el dataset RPT3DS y en las imágenes de objetos arqueológicos es la dificultad de asignar un contexto semántico a los patrones y encontrar similitudes en formas y significado. El objetivo en los experimentos es observar si los modelos son capaces de predecir regiones de interés donde se encuentren patrones, lo cual se cumplirá como mínimo logrando superar el AP@50 de bounding box o segmentación de los baselines.

En cuanto a los resultados obtenidos por los modelos sin el uso de Data Augmentation (DA), se observa que disminuyen sus

6.1. One-Class

Los experimentos que se utilizaron para entrenar los modelos con una única clase fueron la estrategia estándar y zero-shot (Fig 6.1, 6.2). Se realizó transfer-learning con cada modelo con RPT3DS utilizando los pesos pre-entrenados de COCO. La Tabla 6.1 muestra la comparación de los modelos, donde el modelo con mejor desempeño en AP respecto al bounding box, con el experimento de la estrategia estándar, es YOLOv8. Por otro lado, el modelo con mejor desempeño en bounding box AP en el experimento zero-shot es Retina-Net.
En la Figura 6.1, se observa que el modelo de YOLOv8-s logra segmentar y detectar patrones que incluso no fueron segmentados en el ground truth, como en la primera imagen y en la última de arriba hacia abajo. Por otro lado, se observa que Mask-RCNN obtiene resultados cercanos a los del ground truth en elementos donde no existen objetos adicionales en las imágenes.

Sin embargo, se observa en la cuarta imagen de arriba hacia abajo que el modelo sobre-segmenta regiones donde no existen patrones en el ground truth. Finalmente, el modelo que tanto en las métricas como en la visualización obtiene peores resultados es Faster-RCNN, en donde logra detectar patrones, pero genera una mayor cantidad de falsos positivos.

En el experimento zero-shot visualmente se observa que el modelo Mask-RCNN y Retina

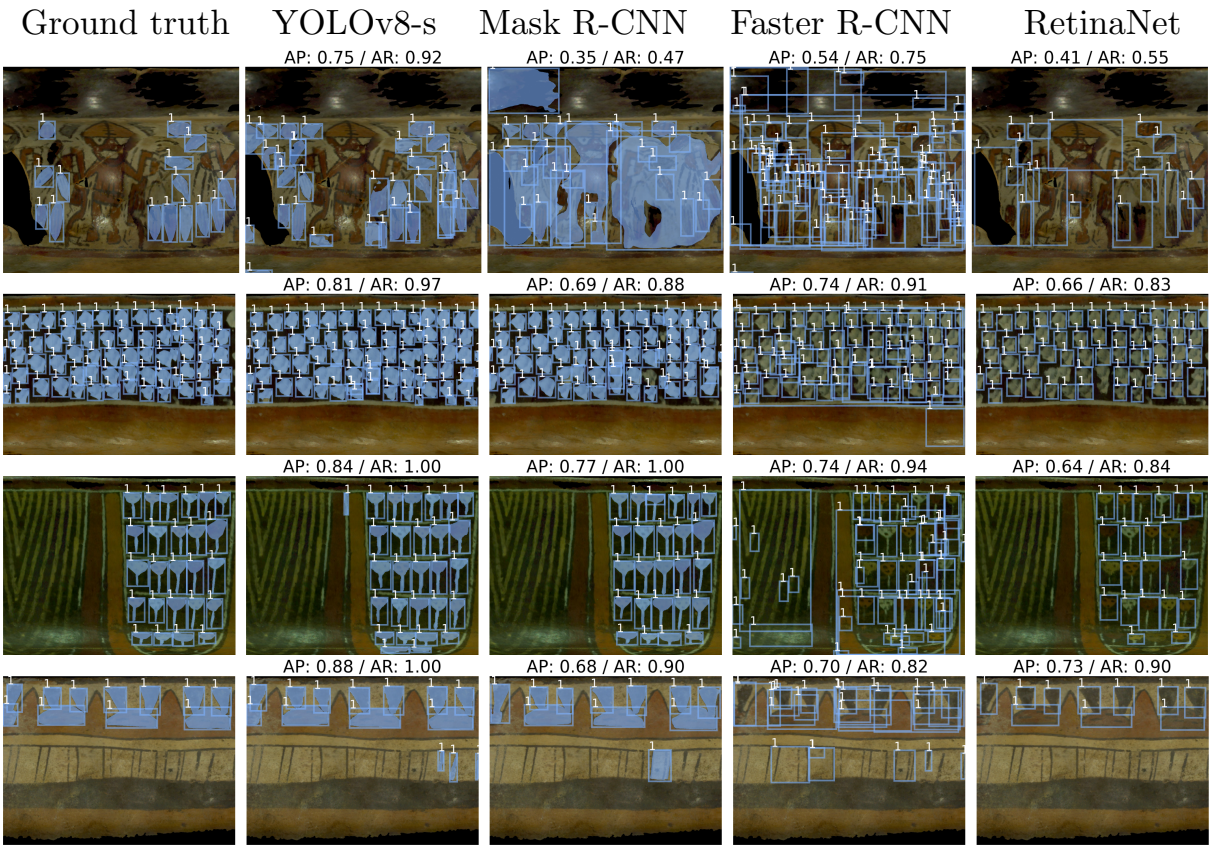


Figura 6.1: Ejemplo de predicciones obtenidas con la **estrategia estándar con una clase**. Las imágenes son un subconjunto del conjunto de validación.

Net son los que obtienen mejores reusltados, acercandose más al ground truth. Por otro lado, los modelos de YOLOv8 y Faster-RCNN obtienen varios falsos positivos, y en el caso de YOLOv8 no logra detectar en varios casos varios patrones existentes en la imagen.

En cuanto a los resultados obtenidos con las métricas de segmentación, se evaluó el modelo Mask R-CNN y YOLOv8, donde el modelo con mejor desempeño utilizando la estrategia estándar fue YOLOv8. La Tabla 6.4 muestra los resultados obtenidos. Sin embargo, al utilizar la estrategia zero-shot, el modelo Mask R-CNN obtiene un mejor desempeño en AP. Con respecto a la métrica AR, ambos modelos obtienen un desempeño similar.

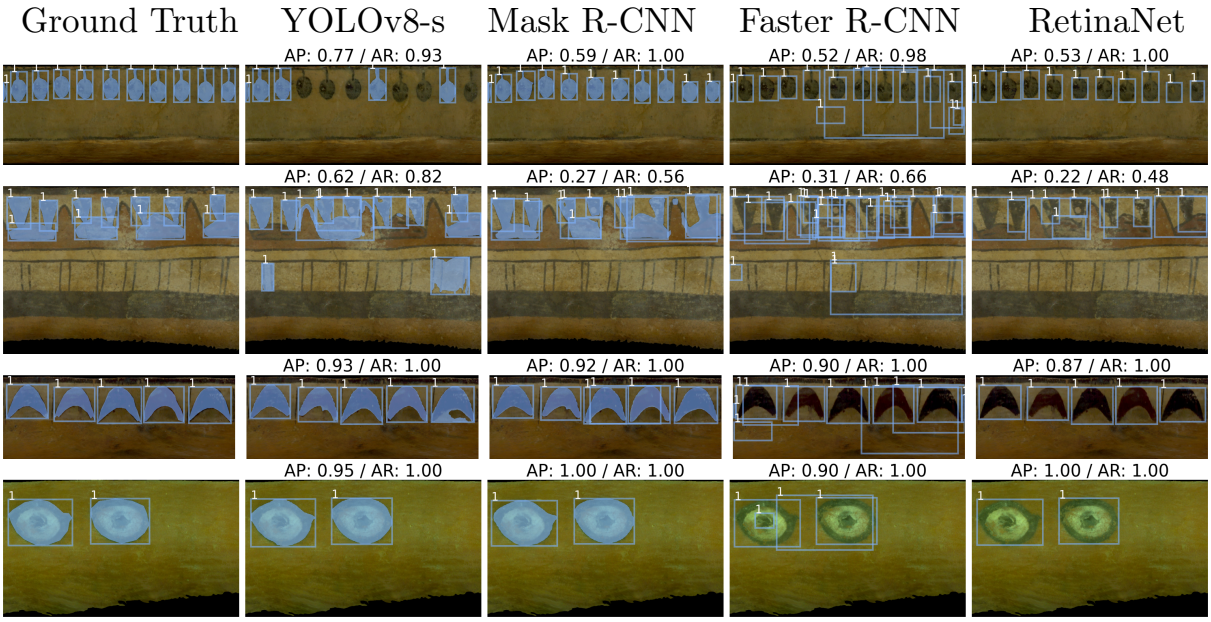


Figura 6.2: Muestra de predicciones obtenidas con la **estrategia zero-shot con una clase**. Las imágenes son un subconjunto del conjunto de validación.

Tabla 6.1: **Comparacion de métricas obtenidas en detección de objetos de una clase (%)**. Se reporta la precisión promedio (AP) y el recall promedio (AR) para la tarea de bounding box (bb) utilizando las estrategias estándar y zero-shot. Las métricas fueron calculadas utilizando el conjunto de validación de cada estrategia.

		Estándar				Zero-shot			
Method	**Backbone**	**AP**^{bb}	**AR**^{bb}	**AP**$^{bb}_{50}$	**AP**$^{bb}_{75}$	**AP**^{bb}	**AR**^{bb}	**AP**$^{bb}_{50}$	**AP**$^{bb}_{75}$
Template Matching	-	16.7	20.3	32.4	13.4	20.6	24.7	48.5	8.8
SAM [60]	-	15.4	54.2	21.3	17.8	21.4	**65.3**	33.0	22.7
One stage									
Retina-Net [32]	ResNet-101-FPN	64.1	71.3	88.6	72.2	**42.4**	54.5	**68.9**	**40.9**
YOLOv8-s [66]	CSPDarknet53	**79.1**	**83.4**	**97.3**	**91.5**	30.9	50.5	48.0	33.9
Two stage									
Mask R-CNN [51]	ResNet-101-FPN	61.1	68.2	85.9	71.8	37.0	43.3	59.6	36.4
Faster R-CNN [30]	ResNet-101-FPN	55.4	60.7	76.0	67.2	30.3	38.7	53.7	28.5

6.2. Multi-clase

En el contexto de la segmentación y detección de objetos de múltiples clases, se lograron resultados utilizando únicamente la estrategia estándar para entrenar los modelos (Fig.6.3). Esta estrategia asegura que el modelo reconozca los objetos del conjunto de datos de validación debido a que la entidad es vista por el modelo durante el entrenamiento y de esta manera se intenta reducir la cantidad de falsos positivos.

En la Figura 6.3 se observa que el modelo YOLOv8-s logra segmentar los objetos del

Tabla 6.2: **Comparacion de métricas obtenidas con segmentación de objetos de una clase (%).** Se reporta la precisión promedio (AP %) y el recall promedio (AR %) obtenido con la tarea de segmentación usando la estrategia zero-shot y estandar.

	Estándar		Zero-shot	
Method	**AP^S**	**AR^S**	**AP^S**	**AR^S**
Mask R-CNN [51]	58.1	65.23	**36.8**	**41.77**
YOLOv8-s [66]	**68.2**	**72.84**	25.4	41.24

ground truth, pero genera falsos positivos con otras clases. Esto es un efecto que se repite aún más en los otros modelos, donde incluso se observa que las clases detectadas no pertenecen al ground truth. Este efecto se puede deber a la cantidad de clases que necesita detectar el modelo, como también a la similitud que existen en las formas de los objetos.

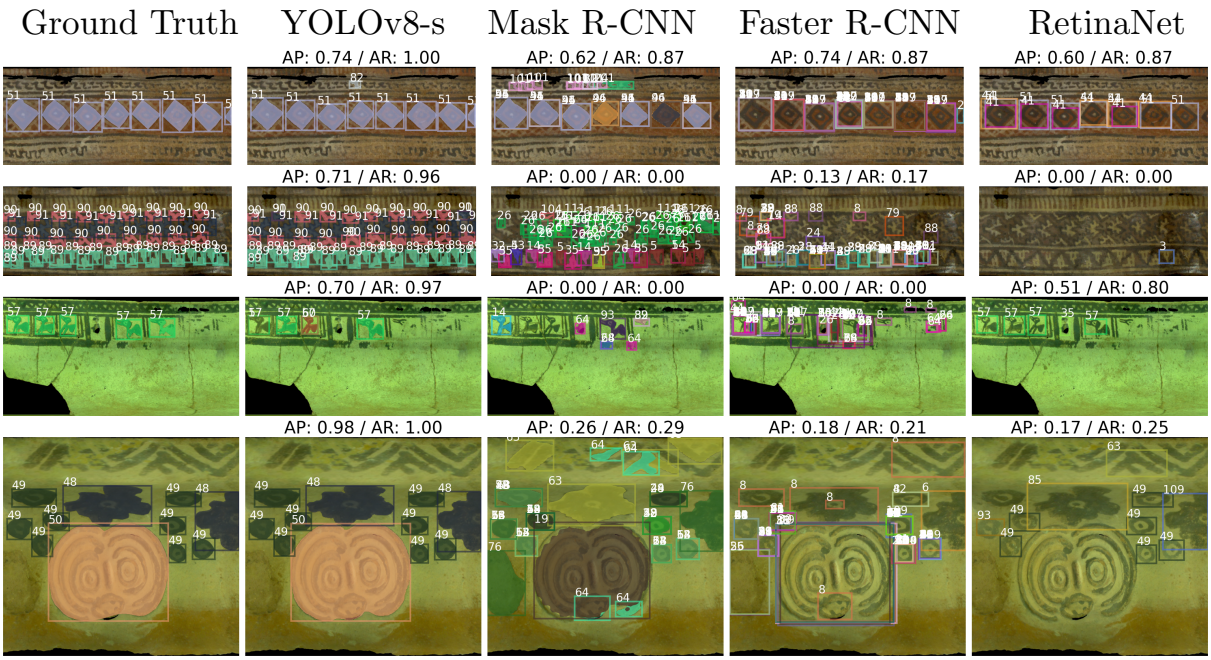


Figura 6.3: Muestra de predicciones obtenidas con la estrategia multi-clase. Las imágenes son el subconjunto del conjunto de validación de la estrategia estándar.

La Tabla 6.3 muestra los resultados de realizar transfer-learning en los modelos utilizando la estrategia estándar y empleando múltiples clases. El modelo que obtuvo los mejores resultados en detección y segmentación en AP y AR en el conjunto de validación fue YOLOv8. Además, la figura muestra los resultados obtenidos en una muestra del conjunto de validación, donde se observa que el modelo YOLO es capaz de capturar objetos que no fueron segmentados en el ground truth.

Tabla 6.3: **Comparacion en RPT3DS con detección y segmentación de objetos de múltiples clases (%)**. Se reporta la precisión promedio (AP) y el recall promedio (AR) para la tarea segmentación de instancias usando la estrategia estándar mencionada en la Sección 4.3.4.1. Las métricas se calcularon utilizando el conjunto de validación de la estrategia estándar.

		Detection				Segmentation			
Method	**Backbone**	$\mathbf{AP}^{bb}$	$\mathbf{AR}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{bb}$	$\mathbf{AR}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$
One stage									
Retina-Net [32]	ResNet-101-FPN	21.7	29.6	25.7	24.3	-	-	-	-
YOLOv8-s [66]	CSPDarknet53	**82.9**	**87.1**	**97.1**	**93.1**	**75.3**	**78.8**	**96.55**	**85.21**
Two stage									
Mask R-CNN [51]	ResNet-101-FPN	12.5	14.2	29.1	7.7	18.4	19.86	27.44	22.72
Faster R-CNN [30]	ResNet-101-FPN	30.0	32.2	50.7	34.8	-	-	-	-

6.2.1. Data Augmentation

Para los resultados mostrados en la Tabla 6.3 y en la Tabla 6.1 se utilizó Data Augmentation (DA) en los modelos de detección y segmentación de objetos. El proceso de Data Augmentation permitió mejorar los resultados principalmente del modelo de YOLOv8, no así en los modelos de Mask R-CNN, Faster R-CNN y Retina-Net.

Tabla 6.4: **Diferencia obtenida al utilizar Data Augmentation en experimento multiclase**. Se reporta la diferencia entre las métricas al aplicar Data Augmentation (D) y sin aplicar Data Augmentation (B) ($\Delta\text{AP} = \text{AP}_D - \text{AP}_B$).

Method	$\mathbf{\Delta AP}^{bb}$	$\mathbf{\Delta AR}^{bb}$
Mask R-CNN [51]	-21.2	-22.2
Faster R-CNN [30]	-24.9	-23.1
Retina Net [32]	-13.5	-18.5
YOLOv8-s [66]	11.4	8.8

En la Tabla 6.4 se observa que el modelo YOLOv8 obtiene una mejora en AP y AR al utilizar Data Augmentation. Por otro lado, los modelos de Mask R-CNN, Faster R-CNN y Retina-Net obtienen una disminución en AP y AR al utilizar Data Augmentation.

Esto se puede deber a que los modelos de Mask R-CNN, Faster R-CNN y Retina-Net no logran generalizar correctamente los datos de entrenamiento al utilizar Data Augmentation. Por otro lado, las técnicas de Data Augmentation utilizadas en el modelo YOLOv8 fueron distintas a las utilizadas en los otro modelos, como agregar la técnica de mosaico y la de copy-paste, por lo que logra generalizar mejor los datos de entrenamiento.

6.2.2. Comparación One-Class vs Multi-Class

En ambos experimentos realizados respecto a la etiquetación de los objetos de RPT3DS, se observa que el modelo YOLOv8 obtiene los mejores resultados en las métricas de detección

y segmentación al utilizar la estrategia estándar. Es por ello que se realiza una comparación entre los resultados obtenidos en los experimentos de una clase y múltiples clases visualmente solamente para el modelo YOLOv8.

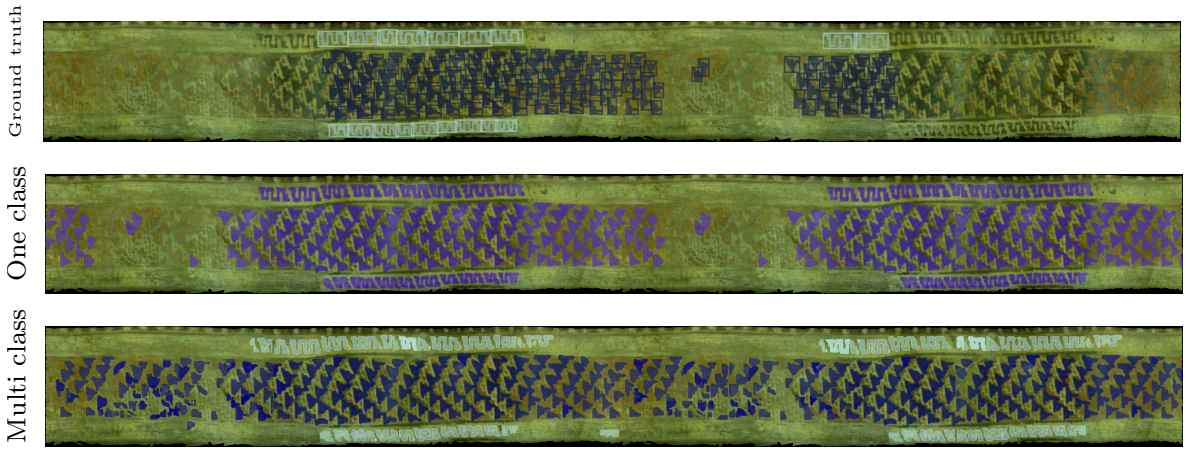


Figura 6.4: Resultados obtenidos con imagen con pérdida de pigmentación, pero cuyos patrones aún podían ser visibles. Se observa que los resultados con el entrenamiento Multi-class son mejores que los obtenidos con el entrenamiento One-class.

En la Figura 6.4 las dos entidades presentan una pérdida de pigmentación en la región central de la textura que impide visualizar claramente los patrones en aquella región. Además, el tamaño y forma de los patrones genera una complejidad en la segmentación debido a que varios patrones triangulares se encuentran cercanos entre sí. En este caso, el modelo entrenado con multi-clases obtiene los resultados ideales para recuperar la mayor cantidad de información en la imagen.

Un caso similar sucede en la Figura 6.5, pero contrario al caso anterior, el método de one-class es el que logra recuperar la mayor cantidad de información. Así mismo, en la Figura 6.6 se observa que el método one-class logra detectar exactamente los patrones que se repiten y que forman un objeto de interés arqueológico. Sin embargo, no logra segmentar todos los objetos y no se logra observar que la forma de la flechas sea captura correctamente.

Respecto a la Figura 6.7 es relevante destacar que el método one-class presenta el problema de no ser útil en casos donde la imagen contiene objetos sobrepuestos, a diferencia del método multi-clase que logra segmentar los objetos de manera correcta, además de diferenciar entre los objetos que son similares entre sí de los que no.

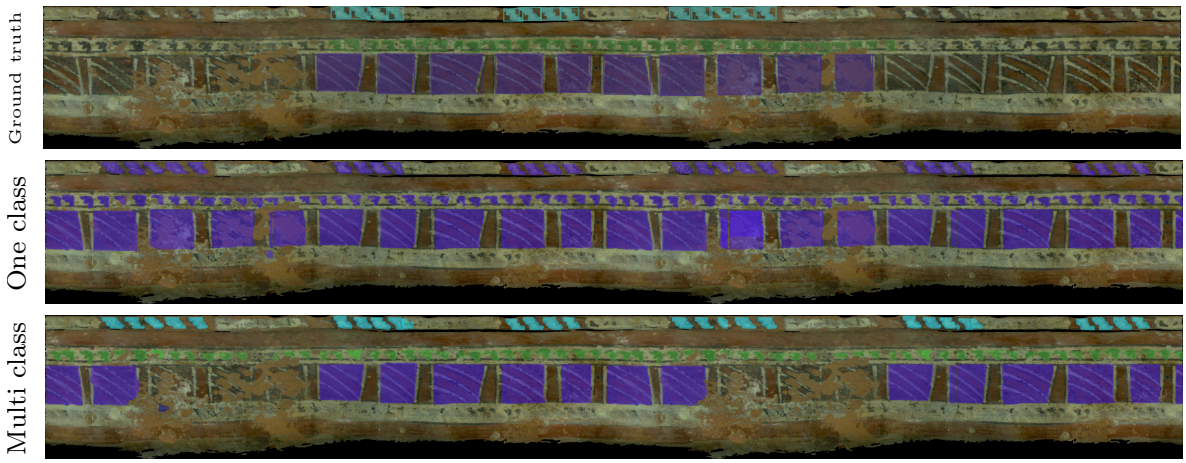


Figura 6.5: Resultados obtenidos con imagen con pérdida de pigmentación, la entidad cuadrada es segmentada en el ground truth. Se observa que los resultados con el entrenamiento One-class son mejores que los obtenidos con el entrenamiento Multi-class.

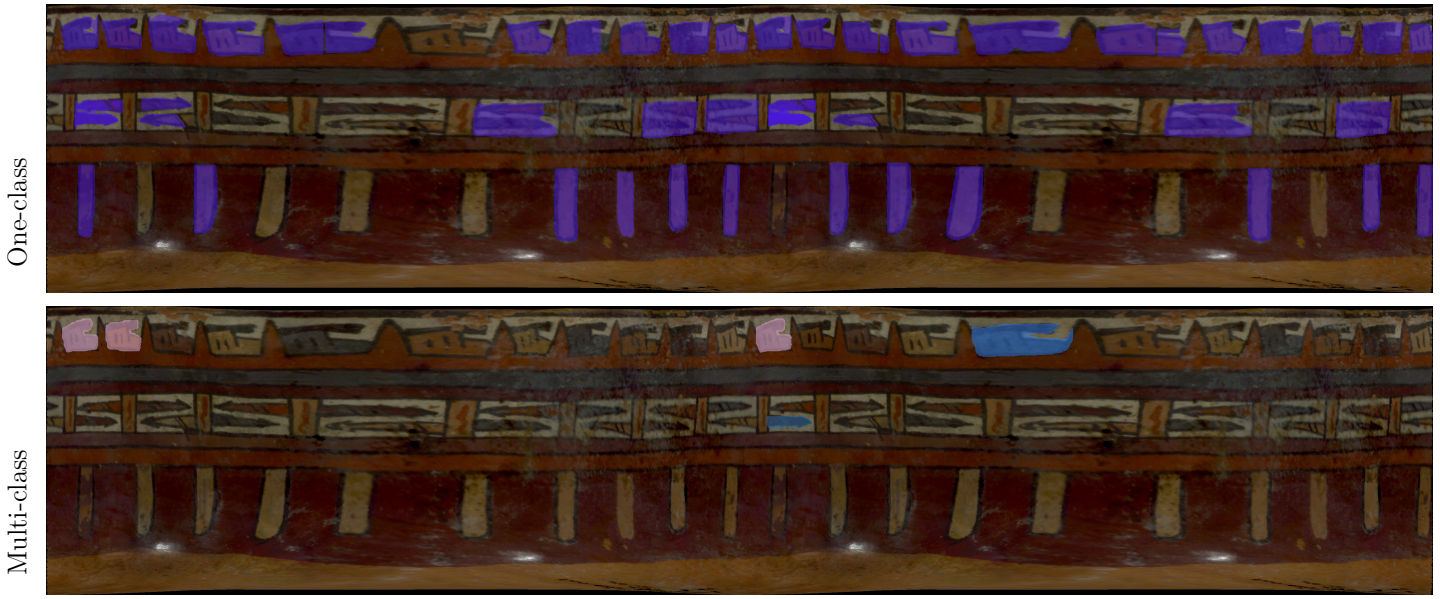


Figura 6.6: Resultados obtenidos con imagen que fue descartada del conjunto de entrenamiento y validación debido a que su segmentación no era correcta. Se observa que el entrenamiento con la estrategia de One-class obtiene mejores resultados que la estrategia Multi-class. Sin embargo se observa que el modelo One-class no logra segmentar todas las figuras.

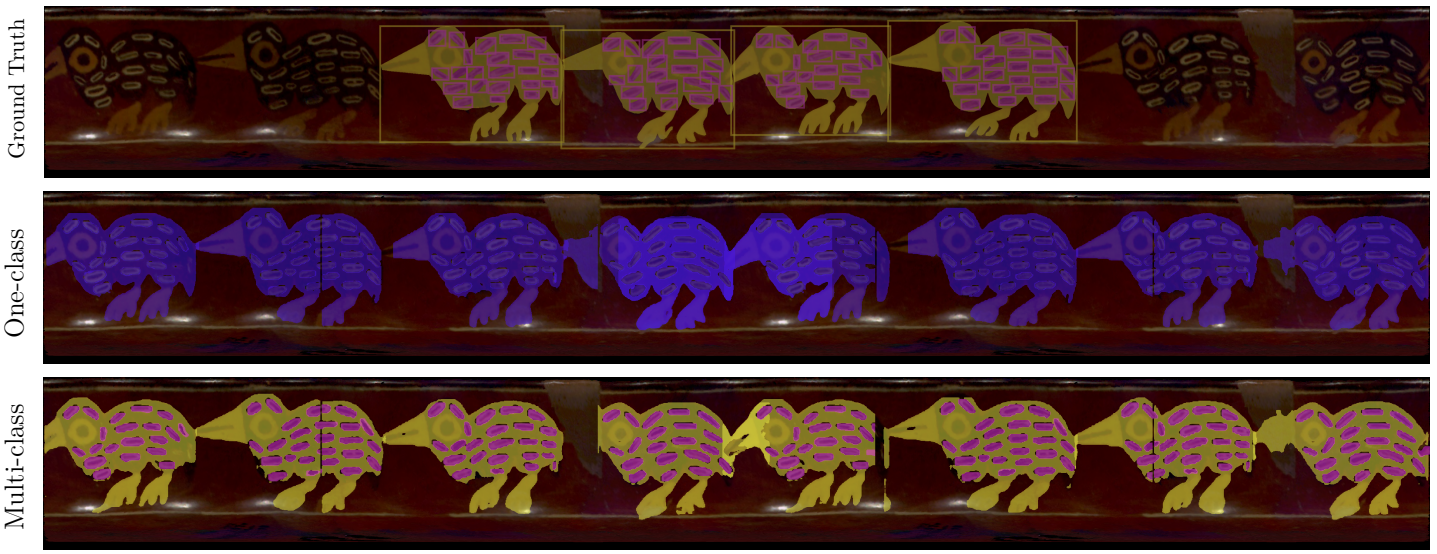


Figura 6.7: Resultados obtenidos con imagen con entidades sobrepuestas. Se observa que los resultados con el entrenamiento Multi-class tienen una mayor utilidad que los obtenidos con el entrenamiento One-class, ya que se logra segmentar todas las entidades requeridas.

Capítulo 7

Discusión

Con el experimento estándar, al entrenar con la estrategia One-Class y Multi-class, observamos que el modelo que obtiene mejores métricas de AP y AR es YOLOv8, con un **97.1 %** de AP_{50} en el problema multi clase y un **97.3 %** de AP_{50} con una sola clase, logrando un mejor rendimiento, en comparación con los otros modelos, en el experimento multi-clase, donde la diferencia porcentual con el segundo mejor modelo fue de aproximadamente el 47 % en AP_{50}.

En las estrategias estándar y zero-shot, observamos que las predicciones obtenidas logran detectar y segmentar entidades que inicialmente no estaban segmentadas, a pesar de las dificultades en la visualización, como la pérdida de pigmentación o pequeñas regiones de una entidad. Esto puede deberse a la implementación con *data augmentation* en el entrenamiento, que permite que el modelo aprenda a detectar patrones bajo diferentes condiciones de iluminación, rotación, escala y otras modificaciones.

En el caso del experimento de *zero-shot*, observamos que tanto para el entrenamiento con una sola clase como para el multi-clase, el modelo YOLO tiene el peor rendimiento en comparación con los otros modelos, donde los modelos con mejor rendimiento son RetinaNet para la detección de objetos y Mask R-CNN para la segmentación.

Sin embargo, las métricas de los mejores modelos en esta estrategia no superan los resultados obtenidos por los modelos entrenados con la estrategia estándar. Si bien la estrategia de *zero-shot* permitiría detectar patrones en un conjunto de imágenes con figuras completamente nuevas para el modelo, los resultados obtenidos muestran que las predicciones están bastante lejos de la detección total de las figuras, lo que puede deberse a que el conjunto de validación tiene formas de entidades diferentes a las entrenadas por el modelo.

Este comportamiento en los resultados puede deberse a la naturaleza de ambas familias de métodos: la familia YOLO como detector de objetos ligero y RetinaNet como un método que aborda el desequilibrio de clases. YOLO, como una arquitectura más ligera, puede aprovechar el proceso de fine-tuning para obtener mejores características sobre cómo se ve un patrón.

Por la misma razón, RetinaNet tiene dificultades para adaptarse, porque no observa tanta información durante el proceso de fine-tuning. Sin embargo, en un escenario de *zero-shot*, la mejor capacidad de generalización de RetinaNet puede deberse a que tiene una mejor re-

spuesta para identificar posibles ubicaciones de nuevos patrones debido a la estrategia focal loss.

A pesar de los buenos resultados obtenidos en general con los modelos en el conjunto de datos RPT3DS, el bajo número de imágenes en el conjunto de datos puede ser una limitación en la generalización de las predicciones de nuevas figuras en otro conjunto similar a RPT3DS.

Es así como se observa en la Figura 6.6, donde YOLOv8 es capaz de segmentar figuras que no fueron segmentadas en el conjunto inicial de datos, sin embargo, existen figuras que siguen sin ser detectadas, posiblemente por la poca cantidad de ejemplos con los que fue entrenado el modelo.

En cuanto a los objetivos iniciales de investigación, el rendimiento superior del modelo YOLOv8 en términos de AP y AR, particularmente con un AP_{50} de 97.1 % y 97.3 % en configuraciones multi-clase y de una sola clase respectivamente, muestra la eficacia de la estrategia de One-Class para objetivos específicos. Este hallazgo está alineado con el Objetivo Específico 1 (OE1), ya que proporciona una evaluación detallada del rendimiento de los modelos de estado del arte en la detección y segmentación de motivos culturales.

El procesamiento de las imágenes y la transformación de los datos de RPT3DS para el entrenamiento de los modelos, como se describe en el Objetivo Específico 2 (OE2), permitió la implementación de un flujo de trabajo estándar para replicar los experimentos con otros conjuntos de datos.

Además, la implementación de técnicas de *data augmentation* para mejorar la detección en diversas condiciones responde directamente al Objetivo Específico 3 (OE3), que se enfoca en la evaluación y comparación de modelos. Sin embargo, es necesario considerar que esta técnica funcionó favorablemente solamente al modelo YOLOv8, por lo que se espera un mayor análisis en futuros trabajos respecto al uso de Data Augmentation sobre estos datos, además de explorar las transformaciones de mosaicos y copy-paste.

Finalmente, los desafíos observados en la detección de objetos en imágenes con motivos culturales lograron ser identificados en la Sección 4.2, además de la limitación de generalización de los modelos a nuevos conjuntos de datos, lo que se relaciona con el Objetivo Específico 4 (OE4).

Al observar que los modelos como RetinaNet y Mask R-CNN tienen un mejor rendimiento en escenarios de *zero-shot*, se aborda la Pregunta de Investigación 2 (PI2), que explora las limitaciones de aplicar técnicas de procesamiento de imágenes en el análisis de artefactos culturales.

La capacidad del modelo YOLOv8 para detectar patrones no segmentados inicialmente, aunque con limitaciones, también se relaciona con la Pregunta de Investigación 3 (PI3) y 5 (PI5), evaluando la posibilidad de identificar patrones parcialmente visibles o transformados. Estos puntos subrayan la importancia de adaptar y aplicar técnicas avanzadas de aprendizaje automático y procesamiento de imágenes, como se investiga en la Pregunta de Investigación 1 (PI1).

El uso de modelos pre-entrenados con imágenes externas al conjunto de datos de vasijas peruanas mostró tener resultados al realizar el proceso de fine-tuning, donde las detecciones y segmentaciones mejoran con respecto al uso de los pesos iniciales de los modelos, lograndode responder a la Pregunta de Investigación 4 (PI4).

Capítulo 8

Conclusión

En este trabajo se mostraron los principales resultados en la evaluación de métodos de detección y segmentación de objetos para la identificación de patrones repetitivos de motivos arqueológicos en imágenes de cerámica antigua peruana. Para ello, se realizaron estrategias de división de datos (estándar y zero-shot) y *data augmentation* que permitieron aumentar la cantidad de datos de entrenamiento y evaluación sin tener que recolectar más imágenes.

Se evaluaron distintos algoritmos y modelos de detección y segmentación de objetos, como: YOLOv8, Retina-Net, Mask-RCNN, Faster-RCNN, Template Matching y Segment Anything. En los resultados el modelo YOLOv8 obtuvo el mejor resultado las métricas de evaluación utilizando la estrategia estándar entre todos los modelos estudiados, logrando sobre un 97 % en AP_{50}. Esto además se vio reflejado en los resultados visuales, en donde se observa que en texturas con pérdidas de pigmentación y con patrones con formas irregulares, el modelo YOLOv8 logra detectar y segmentar correctamente los motivos.

En cuanto a la estrategia zero-shot, el modelo Retina-Net obtuvo el mejor resultado en las métricas, pero logrando solamente un 42 % en AP_{50} y siendo superado por el modelo de Segment Anything en AR por 10 %. Esto muestra que puede haber un margen considerable de mejora en esta estrategia, respecto a la arquitectura del modelo a utilizar y la variedad de datos en el entrenamiento. Así mismo, se mostró una dificultad general de los modelos convolucionales en la detección de motivos que no se encuentran en el conjunto de entrenamiento.

Además, en esta investigación se evidenciaron limitaciones inherentes en el conjunto de datos utilizados, en particular en las imágenes de RPT3DS y en el etiquetado de los motivos. Aunque los modelos mostraron la capacidad de lograr detectar y segmentar los motivos del ground truth, la limitada diversidad y cantidad de motivos podría restringir la generalización del modelo a motivos de otras regiones o culturas.

Por otro lado, la estrategia multi-clase mostró el desafío de lograr segmentar semánticamente los motivos, ya que el etiquetado utilizado no contenía información de la forma de los motivos, y por tanto impidió que los modelos lograran diferenciar entre motivos con formas similares.

En conclusión, los resultados obtenidos en este estudio son un paso importante hacia la automatización y mejora de la detección y segmentación de motivos en texturas de cerámica

antigua. Sin embargo, también destacan la necesidad de una continua experimentación y evaluación, especialmente en lo que respecta a la generalización de modelos a nuevos conjuntos de datos.

Con estos hallazgos, se espera que futuras investigaciones puedan construir sobre esta base para desarrollar nuevos conjuntos de datos con una mayor diversidad de figuras, mejorar la documentación semántica de los motivos modelos aún más robustos y versátiles para una variedad de aplicaciones en visión por computadora.

8.1. Trabajo futuro

La investigación presentada abre diversas avenidas para futuros trabajos en el campo de la detección y segmentación de motivos arqueológicos. A continuación se presentan algunas de las posibles líneas de investigación que se pueden seguir a partir de este trabajo (OE5):

1. Aplicación de los pesos obtenidos tras realizar fine-tuning con el modelo YOLOv8 en nuevas texturas de cerámica antigua. Esto permitiría evaluar la capacidad de generalización del modelo a nuevas texturas, y si es posible detectar y segmentar motivos que no se encuentran en el conjunto de entrenamiento. Por otro lado, con los resultados ya obtenidos se podría complementar los datos de RPT3DS con las segmentaciones que no se encontraban inicialmente en el conjunto de datos.

2. Exploración de nuevas arquitecturas y modelos como aquellos basados en Vision Transformers (ViT) [9]. Estos modelos han mostrado buenos resultados en tareas de detección y segmentación de objetos, y podrían ser una alternativa para mejorar los resultados obtenidos en este trabajo.

3. Enfocar el estudio en escenarios de zero-shot y few-shot learning podría permitir mejorar resultados y ofrecer alternativas al aprendizaje supervisado estudiado en esta investigación como la detección completa correctamente de cada textura en el caso zero-shot o la detección de patrones mediante la consulta de una imagen mediante el uso de few-shot learning.

4. Se recomienda la expansión y diversificación del conjunto de datos, incluyendo la totalidad de la segmentación de cada textura y la agregación de descripciones semánticas generales y detallas de los motivos presentes en el conjunto de datos. Por otro lado, agregar imágenes con una mayor diversidad de iluminación, ángulos, fondos y tipos de objetos permitiría explorar aumentar la capacidad del modelo de YOLOv8.

5. La construcción de una herramienta que permita unificar el flujo completo de la extracción de la textura y la detección automática con el modelo pre-entrenado de YOLOv8 con los datos de RPT3DS.

Bibliografía

[1] S. Lengauer, I. Sipiran, R. Preiner, T. Schreck, and B. Bustos, "A benchmark dataset for repetitive pattern recognition on textured 3d surfaces," *Computer Graphics Forum*, vol. 40, no. 5, p. 1–8, 2021.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[3] R. C. and B. P. G, *Archaeology essentials: theories methods and practice (2nd ed.)*. Thames and Hudson, 2010.

[4] D. Fiore, "La materialidad del arte. modelos económicos, tecnológicos y cognitivo-visuales," *Perspectivas actuales en arqueología argentina*, pp. 123–154, 2009.

[5] E. M. Thompson, S. Biasotti, G. Sorrentino, M. Polig, and S. Hermon, "Towards an Automatic 3D Patterns Classification: the GRAVITATE Use Case," in *Eurographics Workshop on Graphics and Cultural Heritage* (R. Sablatnig and M. Wimmer, eds.), The Eurographics Association, 2018.

[6] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: a review," *Journal of medical systems*, vol. 42, pp. 1–13, 2018.

[7] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, and A. Rubtsov, "Comparison of different convolutional neural network architectures for satellite image segmentation," in *2018 23rd conference of open innovations association (FRUCT)*, pp. 172–179, IEEE, 2018.

[8] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[10] B. Usero and J. Angel del Brío, "Review of the 2009 unesco framework for cultural statistics," *Cultural Trends*, vol. 20, no. 2, pp. 193–197, 2011.

[11] V. Croce, G. Caroti, L. De Luca, A. Piemonte, and P. Véron, "Neural radiance fields (nerf): Review and potential applications to digital cultural heritage," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 453–460, 2023.

[12] S. Biasotti, E. M. Thompson, L. Barthe, S. Berretti, A. Giachetti, T. Lejemble, N. Mella-

do, K. Moustakas, I. Manolas, D. Dimou, *et al.*, "Shrec'18 track: Recognition of geometric patterns over 3d models," in *Eurographics workshop on 3D object retrieval*, 2018.

[13] N. Mellado, G. Guennebaud, P. Barla, P. Reuter, and C. Schlick, "Growing least squares for the analysis of manifolds in scale-space," in *Computer Graphics Forum*, vol. 31, pp. 1691–1701, Wiley Online Library, 2012.

[14] G. Bishop, S.-H. Cha, and C. Tappert, "A greek pottery shape and school identification and classification system using image retrieval techniques," *Proceedings of Student/Faculty Research Day CSIS. Pace University*, vol. 2, 2005.

[15] E. M. Thompson, S. Biasotti, A. Giachetti, C. Tortorici, N. Werghi, A. S. Obeid, S. Berretti, H.-P. Nguyen-Dinh, M.-Q. Le, H.-D. Nguyen, *et al.*, "Shrec 2020: Retrieval of digital surfaces with similar geometric reliefs," *Computers & Graphics*, vol. 91, pp. 199–218, 2020.

[16] N. Kanopoulos, N. Vasanthavada, and R. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[18] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Q. Weinberger, "Convolutional networks with dense connectivity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 8704–8716, 2019.

[19] S. En, C. Petitjean, S. Nicolas, and L. Heutte, "A scalable pattern spotting system for historical documents," *Pattern Recognition*, vol. 54, pp. 149–161, 2016.

[20] I. Úbeda, J. M. Saavedra, S. Nicolas, C. Petitjean, and L. Heutte, "Pattern spotting in historical documents using convolutional models," in *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, HIP '19, (New York, NY, USA), p. 60–65, Association for Computing Machinery, 2019.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 404–417, Springer Berlin Heidelberg, 2006.

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision – ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 778–792, Springer Berlin Heidelberg, 2010.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," *2011 International Conference on Computer Vision*, pp. 2564–2571, 2011.

[25] J. Liu, E. Psarakis, and I. Stamos, "Automatic kronecker product model based detection of repeated patterns in 2d urban images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[26] M. Kushnir and I. Shimshoni, "Epipolar geometry estimation for urban scenes with repetitive structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

vol. 36, no. 12, pp. 2381–2395, 2014.

[27] M. Wolff, R. T. Collins, and Y. Liu, "Regularity-driven facade matching between aerial and street views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] H. Xiao, G. Meng, L. Wang, and C. Pan, "Facade repetition detection in a fronto-parallel view with fiducial lines extraction," *Neurocomputing*, vol. 273, pp. 435–447, 2018.

[29] Y. Lian, X. Shen, and Y. Hu, "Detecting and inferring repetitive elements with accurate locations and shapes from façades," *Vis. Comput.*, vol. 34, p. 491–506, apr 2018.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

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

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016.

[34] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, p. 303–338, jun 2010.

[35] S. Degadwala, D. Vyas, U. Chakraborty, A. R. Dider, and H. Biswas, "Yolo-v4 deep learning model for medical face mask detection," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 209–213, IEEE, 2021.

[36] B. Wu, C. Pang, X. Zeng, and X. Hu, "Me-yolo: Improved yolov5 for detecting medical personal protective equipment," *Applied Sciences*, vol. 12, no. 23, p. 11978, 2022.

[37] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, "Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd," *Sensors*, vol. 20, no. 17, p. 4938, 2020.

[38] M. Segal-Rozenhaimer, A. Li, K. Das, and V. Chirayath, "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn)," *Remote Sensing of Environment*, vol. 237, p. 111446, 2020.

[39] X. Yi, L. Gao, Y. Liao, X. Zhang, R. Liu, and Z. Jiang, "Cnn based page object detection in document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 230–235, IEEE, 2017.

[40] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464–7475, 2023.

[41] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[42] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9404–9413, 2019.

[43] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[44] H. G. Kaganami and Z. Beiji, "Region-based segmentation versus edge detection," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1217–1221, IEEE, 2009.

[45] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," *Advances in neural information processing systems*, vol. 22, 2009.

[46] H. Ng, S. Ong, K. Foong, P.-S. Goh, and W. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *2006 IEEE southwest symposium on image analysis and interpretation*, pp. 61–65, IEEE, 2006.

[47] I. Levner and H. Zhang, "Classification-driven watershed segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1437–1445, 2007.

[48] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[49] Y. Yu, C. Wang, Q. Fu, R. Kou, F. Huang, B. Yang, T. Yang, and M. Gao, "Techniques and challenges of image segmentation: A review," *Electronics*, vol. 12, no. 5, p. 1199, 2023.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[52] J. Jain, A. Singh, N. Orlov, Z. Huang, J. Li, S. Walton, and H. Shi, "Semask: Semantically masked transformers for semantic segmentation," *arXiv preprint arXiv:2112.12782*, 2021.

[53] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[54] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[55] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," *arXiv preprint arXiv:2203.08414*, 2022.

[56] X. Li, H. Ding, W. Zhang, H. Yuan, J. Pang, G. Cheng, K. Chen, Z. Liu, and C. C. Loy, "Transformer-based visual segmentation: A survey," *arXiv preprint arXiv:2304.09854*, 2023.

[57] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell,

P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[59] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7086–7096, 2022.

[60] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[61] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," vol. 106, p. 107404, 2020.

[62] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[63] I. Sipiran, P. Lazo, C. Lopez, M. Jimenez, N. Bagewadi, B. Bustos, H. Dao, S. Gangisetty, M. Hanik, N.-P. Ho-Thi, *et al.*, "Shrec 2021: Retrieval of cultural heritage objects," *Computers & Graphics*, vol. 100, pp. 1–20, 2021.

[64] M. Vergara, B. Bustos, and I. Sipiran, "Aprendizaje multietiqueta de patrones geométricos en objetos de herencia cultural," Master's thesis, University of Chile, Santiago, Chile, 2023.

[65] "Coco api - dataset." https://github.com/cocodataset/cocoapi. Accessed: 20-10-2023.

[66] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023.

[67] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*, pp. 1542–1547, IEEE, 2018.

[68] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." https://github.com/facebookresearch/detectron2, 2019.

[69] I. Buslaev and Kalinin, "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[70] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," *arXiv preprint arXiv:2012.07177*, 2020.

[71] R. Brunelli and T. Poggiot, "Template matching: Matched spatial filters and beyond," *Pattern recognition*, vol. 30, no. 5, pp. 751–768, 1997.

[72] J. P. Lewis, "Fast template matching," in *Vision interface*, vol. 95, pp. 15–19, Quebec City, QC, Canada, 1995.

[73] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th international conference on pattern recognition (ICPR'06)*, vol. 3, pp. 850–855, IEEE, 2006.

[74] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[75] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, $\mathcal{L}$. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[76] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[77] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[78] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[79] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[80] G. Jocher, "YOLOv5 by Ultralytics," May 2020.

[81] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.

[82] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2918–2928, 2021.