Universidad Internacional de Valencia

Segformer.

Segmentación de imágenes aéreas mediante Transformers para determinar la evolución urbanística de una determinada aérea.

Titulación:
Máster Inteligencia Artificial

Curso académico
2022 – 2023

Alumno/a:
Velasco Romero, Álvaro
D.N.I: **71277901G**

Director/a de TFM:
Marco Detchart, Cédric

Convocatoria:
Periodo Extra

Noviembre 2023

De:
Planeta Formación y Universidades

Universidad Internacional de Valencia

Índice

1. Resumen

A lo largo del tiempo, el paisaje urbano cambia y se transforma, evidenciando el desarrollo y crecimiento de la población, las respuestas a necesidades cambiantes y la adaptación al entorno. Esta evolución, a menudo imperceptible a corto plazo, puede ser cuantificada y visualizada con las herramientas adecuadas. En este contexto, las imágenes satelitales son una fuente inestimable de información, ofreciendo una vista aérea que permite una comprensión más clara y objetiva de los cambios acaecidos.

Por otro lado, en el ámbito de la inteligencia artificial, los modelos basados en la arquitectura de Transformers han revolucionado la capacidad de procesamiento y análisis de datos, particularmente en el procesamiento del lenguaje natural y la visión por computadora. Segformer emerge como un innovador framework basado en Transformers para la segmentación semántica, que combina la potencia de estos con decodificadores MLP (Perceptrón Multicapa) optimizando así la eficiencia y la precisión.

En el presente proyecto se describirán los trabajos llevados a cabo para desarrollar un modelo de inteligencia artificial basada en la arquitectura Segformer que nos permita identificar edificios en imágenes satelitales y su posterior aplicación en un caso práctico consistente en determinar la evolución urbanística en un área urbana mediante la detección de nuevas edificaciones gracias a una serie temporal de imágenes.

2. Objetivos

Conocer la evolución urbanística de las ciudades es esencial para comprender el desarrollo histórico, socioeconómico y cultural de una región. El urbanismo, como disciplina encargada de planificar y organizar el espacio urbano, juega un papel fundamental en el diseño de ciudades que respondan a las necesidades cambiantes de sus habitantes, al tiempo que se

integran de manera sostenible con el entorno natural. A medida que las ciudades crecen y se transforman, la planificación urbana adecuada puede mitigar problemas como la congestión, la falta de acceso a servicios básicos y los desafíos medioambientales. Además, mediante el estudio de la evolución urbanística, los planificadores pueden identificar patrones, anticipar tendencias futuras y diseñar intervenciones que fomenten ciudades más habitables, inclusivas y resilientes. En este sentido, herramientas que permitan visualizar y cuantificar estos cambios, como la propuesta en este trabajo, se vuelven esenciales para una gestión urbana efectiva, garantizando que las ciudades del mañana estén mejor preparadas para enfrentar los desafíos del futuro.

1. Objetivo Principal:

Desarrollar un modelo de segmentación semántica capaz de identificar y cuantificar edificaciones en imágenes satelitales.

Objetivos asociados:

- Seleccionar una arquitectura adecuada para la segmentación semántica en imágenes satelitales.
- Entrenar el modelo utilizando conjuntos de datos relevantes.
- Validar la precisión y eficacia del modelo en imágenes de distintos entornos urbanos.

El proyecto que se presenta a continuación tiene como finalidad principal conocer e implementar la arquitectura Segformer no solo para comprobar su efectividad, sino también profundizar en el entendimiento de esta herramienta puntera y su potencial para transformar el análisis urbanístico a través de la teledetección.

El primer lugar compararemos su desempaño con otros modelos que venían siendo muy importantes en el campo de la segmentación semántica. Para ello analizaremos el estudio que realizaron sus creadores y lo pondremos a prueba con dos conjuntos de datos de características muy diferentes.

2. Objetivo Secundario:

Crear una herramienta de usuario amigable que permita cargar imágenes satelitales y obtener análisis sobre la evolución urbanística.

Objetivos asociados:

- Desarrollar un script o interfaz para la carga de imágenes.
- Integrar el modelo entrenado en la herramienta para generar análisis automatizados.
- Proporcionar visualizaciones y métricas claras sobre la evolución identificada.

El siguiente objetivo, una vez comprobadas las ventajas de Segformer, será entrenar un modelo para ponerlo a prueba en una aplicación real. Se facilitará un notebook para usar en Google Colab que permita a los usuarios introducir varias imágenes satelitales de diferentes momentos temporales, obtenidas preferiblemente en plataformas de acceso público como Google Earth. Esta herramienta permitirá, no solo identificar edificios ya existentes, sino también detectar las nuevas edificaciones que se han añadido en el transcurso del tiempo entre las diferentes imágenes. Adicionalmente, el cuaderno ofrecerá un recuento numérico de los edificios identificados, lo que facilita una comprensión cuantitativa del crecimiento urbanístico.

Aunque la propuesta se presenta como innovadora, es importante mencionar que no se basa en trabajos previos realizados por el tutor del alumno. Sin embargo, el campo de la segmentación de imágenes para identificar distintos elementos en entornos urbanos está ampliamente estudiado y existen numerosos trabajos en los que nos podemos apoyar como veremos en el apartado del “Estado del arte”.

Lo que se busca con esta investigación es aprender sobre nuevas herramientas y ponerlas en práctica en una aplicación que permita a cualquier usuario, desde planificadores urbanos hasta estudiantes y curiosos, entender y visualizar la transformación del paisaje urbano en una zona determinada, combinando la potencia de la inteligencia artificial con la accesibilidad de las imágenes satelitales públicas.

3. Introducción

Desde sus inicios, la segmentación de imágenes ha sido fundamental para analizar y comprender las dinámicas urbanas, permitiendo a los planificadores urbanos e investigadores identificar y categorizar elementos clave en el entorno urbano, como carreteras, parques, cuerpos de agua y, especialmente, edificios (Memon et al., 2022). Esta capacidad de reconocer objetos y estructuras en imágenes ha resultado esencial en la toma de decisiones relacionadas con la planificación urbana, incluyendo la asignación de recursos y la identificación de áreas susceptibles a desastres naturales (Wu et al., 2023; G. Yang et al., 2020).

En el campo de la planificación urbana ha experimentado un avance notorio gracias al uso de técnicas de visión artificial y segmentación de imágenes, esenciales para comprender y analizar la evolución de las áreas urbanas. A lo largo del tiempo, diversas arquitecturas de segmentación semántica han desempeñado un papel crucial en esta revolución, permitiendo la identificación precisa de objetos y estructuras en imágenes, incluyendo edificios y otros elementos urbanos. Este proyecto se enfocará en la arquitectura Segformer (Xie et al., s. f.) una propuesta innovadora que integra los Transformers con un perceptrón multicapa, constituyendo así una de las arquitecturas más avanzadas y prometedoras en segmentación semántica. Y pondremos a prueba esta afirmación comparando su rendimiento con respecto a otras arquitecturas ya consolidadas como DeepLabv3+ (Liang-Chieh Chen, 2018).

La segmentación semántica de imágenes satelitales representa un campo desafiante y de gran relevancia en el análisis urbanístico contemporáneo. Modelos como Segformer se enfrentan a una diversidad de obstáculos intrínsecos al procesamiento y análisis de estos datos a gran escala. Entre los principales desafíos destacan la variabilidad en las condiciones de iluminación, que puede alterar significativamente la apariencia de las superficies terrestres, complicando la consistencia en la detección de patrones por parte del modelo. La definición y resolución de las imágenes también varían, lo que puede llevar a una segmentación imprecisa, especialmente en áreas donde la densidad de información es alta, como en las zonas urbanas. Además, la heterogeneidad arquitectónica entre diferentes

regiones geográficas introduce una complejidad adicional, ya que el modelo debe ser capaz de generalizar y adaptarse a una amplia gama de características urbanas. Estos factores, junto con las limitaciones en la cantidad y calidad de los datos etiquetados disponibles para el entrenamiento, presentan una barrera significativa que aún está siendo abordada por investigadores y desarrolladores en el ámbito de la visión por computadora y el aprendizaje automático.

En este trabajo, se explorará detalladamente la evolución de estas arquitecturas, resaltando sus puntos fuertes y limitaciones. Además, se examinarán trabajos previos en la segmentación de imágenes satelitales, con un enfoque especial en el papel de Nvidia, que ha utilizado la inteligencia artificial para identificar edificios afectados por desastres naturales. Este avance no solo ilustra la utilidad de estas técnicas en la planificación urbana, sino que también subraya la versatilidad y el potencial de las soluciones basadas en segmentación de imágenes para abordar desafíos urbanos en constante cambio.

4. Estado del Arte

El campo de la visión artificial, conocido como Computer Vision en inglés, se enfoca en el desarrollo de técnicas y algoritmos que permiten a las máquinas interpretar y comprender el mundo visual que les rodea, similar a cómo lo hacen los seres humanos. En particular, la identificación de objetos es una de las tareas fundamentales en Computer Vision, donde se busca detectar y reconocer elementos específicos dentro de una imagen o un video.

Dentro de la identificación de objetos, existen dos enfoques clave: la segmentación semántica y la segmentación de instancia. La segmentación semántica se centra en la asignación de una etiqueta de clase a cada píxel en una imagen, lo que significa que todos los píxeles pertenecientes al mismo tipo de objeto reciben la misma etiqueta. Por otro lado, la segmentación de instancia va un paso más allá y no solo asigna etiquetas de clase, sino

que también distingue objetos individuales del mismo tipo, asignando una identificación única a cada instancia.

Estos dos enfoques desempeñan un papel crucial en la identificación precisa y detallada de objetos en imágenes y videos, y han sido la base de numerosas investigaciones y avances en la Visión por Computadora, especialmente en aplicaciones como la detección de objetos, la robótica, la conducción autónoma y la interpretación de imágenes médicas, entre otros.
En este apartado nos centraremos en la segmentación semántica por ser el enfoque empleado en el proyecto para identificar los distintos edificios que conforman las imágenes a analizar. En nuestro caso no estamos interesados en identificar cada edificio de forma individual si no el conjunto de todos ellos. En este sentido, las arquitecturas que conforman el estado del arte en la segmentación semántica son:

- FCN (Fully Convolutional Network) – 2014
- SegNet – 2015
- U-Net – 2015
- PSPNet (Pyramid Scene Parsing Network) – 2016
- DeepLabv3+ - 2018
- HRNet – 2019
- Segformer - 2021

Modelos de segmentación semántica

A lo largo de los años, se ha observado una evolución constante en las arquitecturas de los modelos de segmentación semántica. Desde los primeros enfoques, como FCN (Fully Convolutional Network), que introdujeron el uso de convoluciones completas para preservar la información espacial (Vigueras-Guillén et al., 2019), hasta U-Net, que incorporó conexiones de omisión para mantener tanto los detalles contextuales como los locales, cada etapa ha representado un avance en la mejora de la precisión y la eficiencia en la identificación de objetos urbanos (Vasavi et al., 2022).

No obstante, uno de los hitos más notables en esta evolución ha sido la aparición de Mask R-CNN. Esta arquitectura no solo permitió la detección precisa de objetos, sino también la segmentación detallada de máscaras, consolidándose como el estándar en la segmentación

de instancias. Mask R-CNN ha sido ampliamente utilizado en aplicaciones urbanas, desde la identificación de edificios hasta el análisis de cambios urbanos en imágenes de satélite (He et al., 2017).

Recientemente, se ha observado una transición en el campo de la segmentación semántica, donde Segformer ha emergido como una alternativa prometedora a DeepLabv3+. Segformer destaca por su eficiencia y capacidad para abordar tareas de segmentación semántica con éxito al emplear transformers en lugar de convoluciones. Esta innovación ha abierto nuevas perspectivas en la segmentación de imágenes urbanas, ofreciendo un enfoque más ágil y eficiente (Bi et al., 2022; M. Li et al., 2023a).

A pesar de ser una incorporación reciente, no se puede pasar por alto la última adición a esta lista de arquitecturas prometedoras: el Semantic Aggregation Module (SAM) (Behera et al., 2023; Press, s. f.) de Meta. Aunque SAM es relativamente nuevo, promete un potencial sin precedentes en el campo de la segmentación de imágenes al aspirar a segmentar cualquier imagen de manera eficiente y efectiva, lo que podría tener un impacto significativo en aplicaciones urbanas entre otras.

Dado que el trabajo se centra en imágenes satelitales, es importante considerar las características específicas de este tipo de imágenes: resolución variada, cambios en la iluminación, perspectiva, etc. Algunas redes como HRNet y Segformer pueden ser especialmente útiles debido a su capacidad para manejar detalles a diferentes escalas y relaciones a largo alcance, respectivamente (Li et al., 2023b).

A continuación, se presenta una descripción de algunos de los modelos y arquitecturas anteriormente mencionados en la segmentación semántica de imágenes, junto con sus puntos fuertes y puntos débiles.

FCN (Fully Convolutional Network) – 2014:

- Características principales: Fue una de las primeras redes en adaptar redes neuronales convolucionales, que normalmente son utilizadas para clasificación, para segmentación semántica. Propone cambiar las capas totalmente conectadas predominantes hasta ese momento por capas convolucionales para generar mapas de segmentación (Vigueras-Guillén et al., 2019).

- Arquitectura: FCN consta de una fase descendente que extrae características y una fase ascendente que aumenta las dimensiones espaciales para producir un mapa de segmentación del mismo tamaño que la entrada. Utiliza conexiones "skip" para fusionar características de diferentes niveles de resolución (Neumann et al., 2022).
- Ventajas: Introdujo el concepto de aprendizaje end-to-end para la segmentación. Manejo de imágenes de cualquier tamaño (Parra-Mora & da Silva Cruz, 2022).
- Desventajas: Limitaciones en la recuperación de detalles finos debido al uso de pooling.

SegNet – 2015:

- Características principales: Basada en la arquitectura de VGG16, SegNet es muy usada para segmentación en imágenes de carreteras y paisajes urbanos. Se compone de un codificador que proporciona características de alto nivel y un decodificador que asigna estas características a píxeles para obtener una segmentación precisa (Neumann et al., 2022).
- Arquitectura: Codificador basado en VGG16 seguido por un decodificador que mapea características de baja resolución a entradas de alta resolución (Vigueras-Guillén et al., 2019).
- Ventajas: Uso eficiente de memoria. Buen desempeño en escenarios con variabilidad en apariencia.
- Desventajas: Menor capacidad para capturar detalles finos en comparación con U-Net.

U-Net – 2015:

- Características principales: Es quizás la arquitectura más popular para segmentación semántica en imágenes médicas, pero se ha generalizado para muchas otras tareas. Su estructura en forma de "U" consiste en un codificador que captura el contexto espacial y un decodificador que permite una localización precisa. Es conocida por su eficiencia y eficacia, especialmente cuando hay datos limitados (Wang et al., 2020).
- Arquitectura: Simétrica con codificador-decodificador. Las conexiones "skip" entre capas homólogas en el codificador y decodificador ayudan a recuperar detalles (Wang et al., 2021).
- Ventajas: Eficiente y requiere relativamente pocos datos para entrenar. Alta resolución en los mapas de salida (Wang et al., 2020).
- Desventajas: Arquitectura más pesada que otros modelos.

PSPNet (Pyramid Scene Parsing Network) – 2016:

- **Características principales**: Ganó el desafío de segmentación semántica de ImageNet. Utiliza una estructura piramidal para agregar información contextual a múltiples escalas.
- **Arquitectura**: Después de una red de base (p.ej., ResNet), aplica pooling a diferentes regiones de tamaño para capturar contexto a diversas escalas y luego concatena estos mapas de características (Ma et al., 2023).
- **Ventajas**: Captura efectivamente información contextual. Alta precisión en tareas de segmentación (Pan et al., 2021).
- **Desventajas**: Requiere más recursos computacionales que otras arquitecturas (Ma et al., 2023).

DeepLabv3+ - 2018:

- **Características principales**: Es una de las arquitecturas más modernas y ha establecido el estado del arte en segmentación semántica en conjuntos de datos como PASCAL VOC y Cityscapes. Utiliza dilated convolutions (convoluciones dilatadas) y atrous spatial pyramid pooling (ASPP) para capturar información contextual a múltiples escalas (Mo et al., 2022; Wagh et al., 2020).
- **Arquitectura**: Mejora sobre DeepLabv3 incorporando un módulo de decodificador para refinar los resultados de segmentación (Li et al., 2023).
- **Ventajas**: Logra una segmentación semántica de alta resolución y es capaz de capturar objetos a diferentes escalas (Memon et al., 2022).
- **Desventajas**: Complejidad computacional más alta (Wang et al., 2022).

HRNet – 2019:

- **Características principales**: Es una arquitectura diseñada para mantener una alta resolución a través de la red en lugar de reducir y luego aumentar la resolución como se suele hacer tradicionalmente en las capas convolucionales clásicas. Esto puede ser especialmente útil para imágenes satelitales donde los detalles finos, como los contornos de los edificios, son cruciales.
- **Arquitectura**: Conecta múltiples redes en paralelo con diferentes resoluciones y permite conexiones cruzadas entre ellas.
- **Ventajas**: Mantiene detalles en todas las etapas. Logra un desempeño superior en varias tareas de visión.
- **Desventajas**: Arquitectura más compleja y consume más memoria.

Segformer - 2021:

Segformer es un framework de segmentación semántica que implementa los principios de los Transformers, que han supuesto una revolución en el campo del procesamiento de lenguaje natural o en el campo de la generación de imágenes, entre otros. Su característica distintiva radica en su capacidad para realizar segmentación semántica precisa a nivel de píxeles en imágenes (Xie et al., s. f.). Para lograr esto, Segformer introduce una estructura de atención a nivel de píxeles que le permite comprender las relaciones y características detalladas de cada píxel en la imagen, lo que contribuye significativamente a la precisión de la segmentación (Li et al., 2023a). Este enfoque se basa en la premisa fundamental de que cada píxel en una imagen puede estar asociado con una clase de objeto específica, y Segformer se dedica a la tarea de asignar de manera eficiente la clase correcta a cada píxel en la imagen (Xie et al., s. f.).

Uno de los aspectos más destacados de Segformer es su eficiencia. Esta arquitectura aprovecha las ventajas que ofrecen los Transformers, que son conocidos por su capacidad para capturar relaciones globales en los datos, y la aplica al campo de la segmentación semántica sin la complejidad que caracteriza a algunos modelos previos. La simplicidad y eficiencia de diseño de Segformer lo convierten en una opción particularmente atractiva para aplicaciones en tiempo real, donde la velocidad de procesamiento es esencial. Además, su enfoque en la segmentación precisa a nivel de píxeles lo hace valioso en escenarios donde se requiere una identificación detallada de objetos en las imágenes (Xie et al., s. f.).

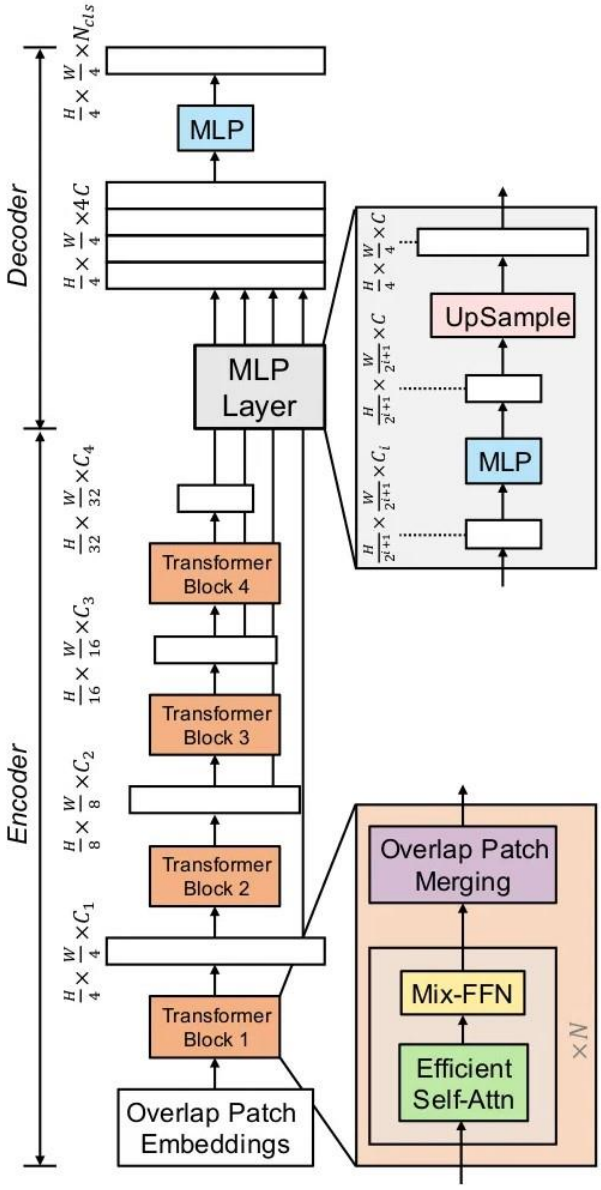
La arquitectura está basada en una estructura Transformer (como BERT o ViT) con cabezas de Codificador-Decodificador, donde el codificador utiliza Atención Propia (Self Attention). La estructura del codificador es jerárquica por naturaleza, la cual produce características a múltiples escalas. No requiere codificación posicional, evitando así la interpolación de códigos posicionales que resulta en una disminución del rendimiento cuando la resolución de prueba difiere de la de entrenamiento (Gao et al., 2023).

Existen dos ideas principales en la estructura que compone un Segformer: el codificador en el modelo produce características a múltiples escalas y el decodificador basado en MLP (red de perceptrones multicapa) agrega esta información de diferentes capas para producir un mapa de segmentación.

En la **Figura 1** se muestra el esquema del que se compone Segformer. La sección inferior, las primeras capas, forman el codificador del modelo y en la superior podemos ver las capas que forman el decodificador.

Arquitectura Segformer

Figura 1. *Arquitectura segformer*



Fuente: (Xie et al., s. f.) *Xie*

En primer lugar, en el codificador una imagen de entrada se divide en parches, tal y como se hace en los transformers de visión, pero mientras que en estos se utiliza un tamaño de

parche de 16x16, en Segformer los autores han usado parches de 4x4 consiguiendo con ello mejores tareas de predicción densa.

Inmediatamente después se encuentra el primer bloque transformador compuesto por tres submódulos:

- Módulo de auto-atención eficiente funciona como la auto-atención multi-cabeza original en los Transformers, pero utiliza una técnica de reducción de secuencia para reducir el costo computacional.
- Bloque Mix Feed-Forward, se utiliza para resolver el problema de resolución fija. En lugar de usar codificación posicional de tamaño fijo, se utilizan capas de convolución y MLP (perceptrón multicapa) para implementar codificación posicional impulsada por datos.
- El bloque de fusión de parches superpuestos, se utiliza para reducir el tamaño del mapa de características. Como se puede ver en la figura 1, el tamaño del mapa de características se reduce a medida que avanza hacia la parte superior de la red.

La parte del decodificador es más simple en comparación con los módulos del codificador. En la parte del codificador vemos que se generan diferentes mapas de características de diferentes tamaños en cada capa. El decodificador es un perceptrón multicapa que toma las características del codificador como entrada y las fusiona. Se compone de cuatro pasos principales:

1. las características de los diferentes niveles del codificador se introducen en la capa de perceptrón multicapa para unificar en la dimensión del canal.
2. Las características se amplían a 1/4 de su tamaño y se concatenan juntas.
3. En tercer lugar, un perceptrón multicapa fusiona las características concatenadas.
4. Finalmente, otro perceptrón multicapa toma la característica fusionada para predecir la máscara de segmentación.

Gracias a esta arquitectura (**Figura 1**), Segformer combina tanto la atención local como la atención global para generar representaciones potentes. Esto confiere una gran ventaja a este modelo por combinar la capacidad de los Transformers para manejar relaciones a largo

alcance y la eficiencia de las redes convolucionales en el procesamiento espacial (Fatty et al., 2023).

A pesar de sus numerosos beneficios, Segformer también presenta algunas limitaciones. Su eficiencia puede verse comprometida cuando se enfrenta a imágenes de alta resolución o conjuntos de datos masivos. Esto se debe a que los modelos basados en Transformers pueden requerir recursos significativos de cómputo para realizar tareas de segmentación detallada en imágenes de gran tamaño. Por lo tanto, si bien Segformer es altamente eficiente en muchas situaciones, puede no ser la mejor opción para aplicaciones que involucran imágenes de alta resolución o grandes volúmenes de datos, donde la capacidad de cómputo puede convertirse en un factor limitante (Xie et al., s. f.).

SAM - 2023

SAM es un modelo de segmentación de imágenes desarrollado por Meta con el propósito de sentar las bases para una herramienta que facilite la identificación y diferenciación de objetos en imágenes y vídeos. El proyecto busca crear un modelo de segmentación altamente preciso para tareas específicas y ha logrado desarrollar SAM, acompañado por el conjunto de datos SA-1B, que es el conjunto de datos segmentados más grande hasta la fecha. En este sentido, SAM es notable por su capacidad para aprender una noción general de lo que son los objetos y generar máscaras para cualquier objeto en cualquier imagen o vídeo, incluso para objetos y tipos de imágenes con los que no ha sido entrenado previamente. Esto lo hace altamente versátil y flexible, ya que puede llevar a cabo tanto segmentación interactiva como segmentación automática (Press, s. f.).

Meta tiene la intención de democratizar el acceso a esta tecnología, y SAM tiene aplicaciones en diversas áreas, desde la edición de vídeos hasta la localización de animales o el seguimiento de eventos naturales en la Tierra. El modelo SAM, junto con el conjunto de datos SA-1B, está disponible para investigadores bajo una licencia abierta permisiva (Apache 2.0), lo que promueve la colaboración y el desarrollo en diversas disciplinas. Esta iniciativa de Meta busca impulsar el avance de la tecnología de segmentación de imágenes y su aplicación en una amplia variedad de campos (Press, s. f.).

DeepLabv3+ vs Segformer

En el campo de la visión artificial, las arquitecturas de segmentación semántica han evolucionado significativamente en los últimos años, buscando optimizar tanto la precisión como la eficiencia computacional. Dos de las arquitecturas más destacadas en este ámbito tal y como hemos visto son DeepLabv3+ y Segformer, cada una con sus propias fortalezas y particularidades. Ambos modelos representan el estado del arte en técnicas de segmentación y, aunque tienen bases conceptuales diferentes, su comparación directa es esencial para entender sus ventajas y limitaciones en aplicaciones prácticas (Memon et al., 2022; Wang et al., 2022).

1. Eficiencia y rendimiento:

La **Figura 2** muestra una comparativa en términos de eficiencia y rendimiento entre diferentes arquitecturas de segmentación semántica en el dataset ADE20K, un conjunto de datos de escenas urbanas. A partir de esta gráfica y del conocimiento general de estas arquitecturas, podemos deducir lo siguiente al comparar DeepLabV3+ y Segformer:

mIoU (Mean Intersection over Union): Es una métrica popular para medir el rendimiento en tareas de segmentación. Un mIoU más alto indica un mejor rendimiento. Segformer-B4 alcanza un mIoU de 50,3 que es significativamente más alto que el DeepLabv3+/R101 con un mIoU de 44,1. Esto indica que Segformer-B4 tiene una capacidad de segmentación superior (Wang et al., 2022).

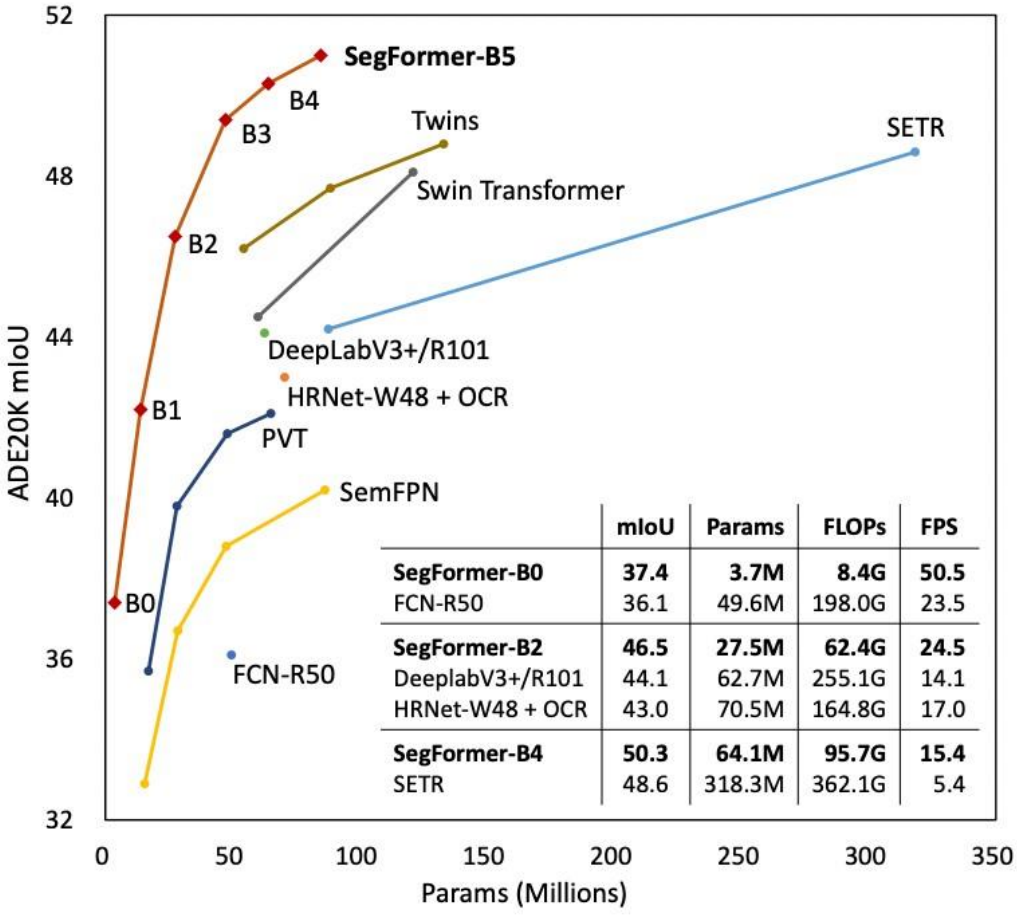
Params (Millones): Indica la cantidad de parámetros que tiene la red. Menos parámetros puede darnos una idea de una arquitectura más ligera y eficiente en términos de almacenamiento. Segformer-B4 tiene 64,1M de parámetros, mientras que DeepLabv3+/R101 tiene 62,7M. Ambas arquitecturas tienen un número similar de parámetros, pero si este dato lo enlazamos con el anterior es revelador que Segformer alcanza un mejor rendimiento con una cantidad similar de parámetros (Liu et al., 2019).

FLOPs: Representa la cantidad de operaciones de coma flotante que la red necesita para procesar una imagen. Es un indicador de la complejidad computacional y el tiempo de inferencia. DeepLabv3+ requiere 255.1G FLOPs, lo que es mucho más alto que los 95.7G

FLOPs de Segformer-B4, lo que sugiere que este último es más eficiente computacionalmente (Li et al., 2023).

FPS (Frames Per Second): Indica cuántas imágenes puede procesar la red en un segundo. Un valor FPS más alto indica un tiempo de inferencia más rápido. Segformer-B4 alcanza 15.4 FPS, que es ligeramente superior al 14.1 FPS de DeepLabv3+/R101. Aunque este dato no es importante para nuestro estudio puesto que es más importante para trabajos con video (Memon et al., 2022).

Figura 2. *Rendimiento versus eficiencia en ADE20K*



Fuente: (Xie et al., s. f.) *Xie*

2. Resultados cualitativos:

Cityscapes es un dataset de imágenes de escenas urbanas de alta calidad desde la perspectiva de un automóvil, es ampliamente usado en el mundo de la conducción autónoma. Los autores de Segformer (Xie et al., s. f.) utilizaron este conjunto de datos para realizar una comparación visual entre SegFormer y DeepLabV3+ y se evidencian diferencias significativas en la precisión de la segmentación. SegFormer demuestra una capacidad superior para discernir y detallar los límites de los objetos, resultando en máscaras de segmentación más finas y precisas. Este avance se refleja particularmente en áreas donde los objetos están próximos entre sí o cuando hay una mezcla de elementos pequeños y complejos en la escena.

DeepLabV3+ tiene tendencia a fusionar elementos que deberían ser segmentados de manera independiente y no es tan detallado en su segmentación. En la **Figura 3** se muestra una comparativa entre ambos modelos:

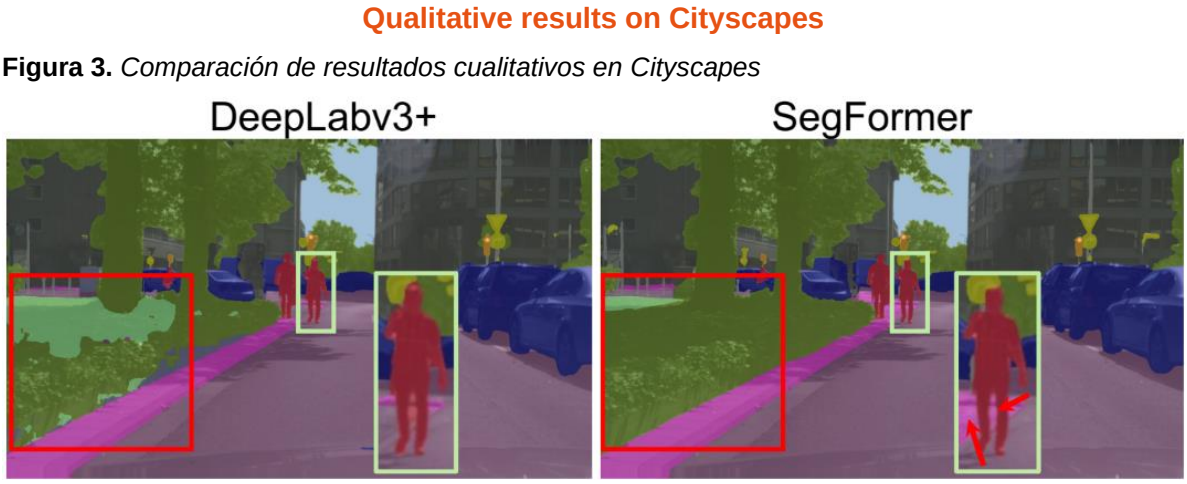


Figura 3. *Comparación de resultados cualitativos en Cityscapes*

Fuente: (Xie et al., s. f.) *Xie*

3. Segformer vs DeepLabV3+, conclusiones:

Las ideas que podemos extraer de estas comparativas son:

- Segformer muestra un mejor rendimiento en términos de mIoU en el dataset ADE20K.
- A pesar de tener un número similar de parámetros, Segformer-B4 tiene menos FLOPs, lo que indica que es más eficiente computacionalmente.

- Segformer-B4 tiene un tiempo de inferencia ligeramente más rápido, lo que podría ser crucial en aplicaciones en tiempo real o cuando se procesa un gran volumen de imágenes.

4. Consideraciones adicionales:

Si bien la gráfica muestra el rendimiento en ADE20K, es relevante considerar cómo se comportan estas arquitecturas en imágenes satelitales, las cuales son objeto de estudio en el presente proyecto. Las imágenes satelitales tienen características distintas, como variabilidad en la resolución, cambios de iluminación y diferencias de escala (Sediqi & Lee, 2021).

De hecho, en entornos del mundo real, especialmente en imágenes satelitales, puede haber un alto grado de ruido o distorsiones debido a las condiciones atmosféricas, cambios estacionales o artefactos de la captura. Es vital evaluar cómo estas arquitecturas manejan tales desafíos y si necesitan pre-procesamiento adicional o ajuste fino para adaptarse a estas variaciones (Wang et al., 2019).

En este sentido, también será vital la disponibilidad de datos. U-Net, por ejemplo, fue diseñado para trabajar con conjuntos de datos más pequeños y podría ser una mejor opción si no se dispone de grandes volúmenes de datos para entrenamiento. Aunque DeepLabV3+ y Segformer son más avanzados, requieren de una buena cantidad de datos etiquetados para lograr un rendimiento óptimo (Vasavi et al., 2022).

Por último, será muy importante determinar si estas arquitecturas pueden beneficiarse de modelos preentrenados en otros conjuntos de datos y adaptarse a nuevos desafíos con un entrenamiento adicional limitado. Esto podría acelerar el tiempo de desarrollo y mejorar el rendimiento en aplicaciones específicas (Xia & Kim, 2023).

En conclusión, la gráfica proporciona una comparación clara entre Segformer y DeepLabV3+ en términos de rendimiento y eficiencia en el dataset ADE20K. Sin embargo, las decisiones sobre qué arquitectura utilizar deben basarse no solo en estas métricas, sino también en las necesidades específicas del proyecto y las características de los datos a procesar (Wang et al., 2022).

Es por este motivo que se realizarán pruebas con ambas arquitecturas sobre los datasets que tenemos disponibles y así tomar una decisión sobre cuál es la arquitectura que mejores resultados nos proporcionará para el caso de estudio de este proyecto (Ibrahem et al., 2022; Wagh et al., 2020).

Trabajos previos en el campo de la segmentación de imágenes aéreas más relevantes

El trabajo de Nvidia (Alarcon, 2018) “AI Helps Detect Disaster Damage From Satellite Imagery”, ha servido de inspiración para el desarrollo de este proyecto y se centra en la detección de daños causados por catástrofes medioambientales a partir de imágenes satelitales. En situaciones de desastre, evaluar manualmente el nivel de daño en áreas afectadas es una tarea desafiante y que consume mucho tiempo. Con el objetivo de proporcionar datos más precisos y rápidos para los equipos de rescate y organizaciones de ayuda, investigadores de Facebook y CrowdAI han desarrollado un algoritmo basado en aprendizaje profundo que puede estimar automáticamente el nivel de daño sufrido por un área. De esta manera, este enfoque innovador tiene como objetivo permitir a los trabajadores de rescate identificar rápidamente las áreas que necesitan ayuda de manera urgente, prescindiendo de conjuntos de datos específicos para desastres que requieren anotaciones manuales. En lugar de eso, este método utiliza conjuntos de datos fácilmente disponibles que contienen características comunes hechas por el hombre en imágenes satelitales, como carreteras y edificios.

Para entrenar su red neuronal convolucional, el equipo de investigación utilizó GPUs NVIDIA Tesla P100 y un marco de aprendizaje profundo acelerado por cuDNN. La red se entrenó con datos de código abierto de Digital Globe y Planet Labs para detectar características hechas por el hombre en imágenes satelitales y calcular cambios relativos entre múltiples instantáneas de datos capturados antes y después de un desastre. Asimismo, la precisión de la red fue probada en imágenes de dos desastres naturales: el huracán Harvey en Texas y el incendio de Santa Rosa en el norte de California. La red logró una precisión del 88,8 por ciento en la identificación de carreteras dañadas durante el huracán Harvey y una precisión del 81,1 por ciento en la detección de edificios dañados en

los incendios de Santa Rosa. En esta línea, además de la detección precisa de daños, el equipo introdujo un nuevo indicador llamado "Índice de Impacto del Desastre" (Disaster Impact Index) que ayuda a comprender mejor el impacto de un desastre. En este sentido, cabe especificar que este índice proporciona información valiosa para las organizaciones de ayuda y suministro, permitiéndoles dirigir recursos hacia las áreas más afectadas de manera rápida y eficiente (Alarcon, 2018).

En el trabajo “An improved algorithm for semantic segmentation of remote sensing images based on deeplabv3+” (Liu et al., 2019) se desarrolla un modelo que incorpora conexiones de omisión y capas de convolución en el decodificador para imágenes de teledetección. Se han realizado extensos experimentos utilizando el conjunto de datos SpaceNet, lo que demuestra que el modelo propuesto supera a los métodos existentes en términos de detección precisa de edificios. Posteriormente se presentó una nueva red de aprendizaje profundo llamada SSNet (Liu et al., 2020), diseñada específicamente para extraer automáticamente edificios de imágenes de teledetección de alta resolución. El modelo SSNet combina las fortalezas de U-Net y ResNet para mejorar la precisión de detección mientras reduce la complejidad de la red. Esto se logra mediante la incorporación de conexiones de omisión y convoluciones dilatadas en la arquitectura de la red, junto con la aplicación de aprendizaje residual profundo para optimizar los parámetros del modelo. Para abordar las variaciones en la distribución de la cobertura terrestre en imágenes multitemporales, en la literatura se ha desarrollado una solución innovadora llamada red Siamesa asimétrica (ASN).

La ASN consta de dos módulos distintos, a saber, la Pirámide Espacial Asimétrica (aSP) y la Pirámide de Representación Asimétrica (aRP). Estos módulos extraen de manera efectiva los píxeles cambiados mediante operaciones de diferencia de características. La aplicación de la ASN aborda los desafíos asociados con la identificación y el análisis de regiones que muestran variaciones en la cobertura terrestre en la superficie de la Tierra (K. Yang et al., 2022). Se ha propuesto una novedosa técnica llamada BGC-Net con el propósito de realizar segmentación semántica de edificios en imágenes aéreas de alta resolución (HRAIs). BGC-Net abarca varios módulos, incluido el módulo de extracción de características (FE), el módulo de pirámide de atención atrópica (AAP) y el módulo de convolución de gráficos duales (DGC). El módulo FE captura de manera efectiva los detalles de las características de los edificios en múltiples niveles, mientras que el módulo AAP construye una estructura

de pirámide utilizando convolución atrópica y un mecanismo de atención para capturar dependencias globales e información contextual.

El módulo DGC se centra en modelar la información contextual en dimensiones espaciales y de canal, mejorando así el aprendizaje de características y la extracción de características de edificios (Zhang et al., 2022). El Mecanismo de Atención Multiescala de Fusión (FMAM-Net), una arquitectura de red introducida para mejorar la precisión de la segmentación de edificios en imágenes de teledetección, aborda específicamente las dificultades asociadas con los límites difusos, la falta de distinción entre clases y la diversidad de edificios en la tarea de segmentación. Para abordar el problema de los límites ambiguos y mejorar las capacidades de reconocimiento, la red incorpora el Módulo de Compensación de Refinamiento de Características (FRCM), que consta del Módulo de Refinamiento de Características (FRM) y el Módulo de Compensación de Características (FCM). Además, para manejar la falta de distinción entre clases, el FMAM-Net integra el Módulo de Atención en Tándem (TAM) y el Módulo de Atención Paralela (PAM), lo que permite filtrar características y guiar la selección de características en función de la información contextual (Ye et al., 2022). Se presenta una arquitectura de red única llamada HCRB-MSAN, que combina bloques residuales conectados horizontalmente con un mecanismo de atención multiescala.

Los bloques residuales conectados horizontalmente facilitan el intercambio de información de características entre diversos grupos de canales, mejorando así el reconocimiento de categorías de edificios y objetivos pequeños. Mientras tanto, la estructura de atención multiescala combina características de diferentes escalas, mejorando en última instancia la precisión de la extracción de edificios. Además, la red emplea un mecanismo de aumento paso a paso durante el proceso de decodificación para lograr una segmentación semántica precisa de edificios (Li et al., 2022). Este estudio implica la evaluación del potencial de energía solar en los techos en Líbano mediante técnicas de segmentación de edificios aplicadas a imágenes aéreas. El enfoque principal se centra en la extracción y el reconocimiento precisos de las huellas de los edificios a partir de imágenes aéreas para estimar el potencial de generación de energía solar en los techos. La metodología emplea una arquitectura similar a UNet para generar máscaras de segmentación semántica dirigidas específicamente a los techos. Para abordar el desafío de distinguir entre edificios cercanos, el estudio también incorpora la predicción de máscaras de bordes de edificios.

Para la parte del codificador de la arquitectura, se utiliza el modelo Efficient-Net-B3 (Nasrallah et al., 2022).

Se ha introducido un nuevo método llamado detección de bordes de múltiples tareas (MTED) con el propósito de realizar vectorización de edificios a partir de imágenes aéreas. El marco MTED consta de tres componentes esenciales: un enfoque eficiente basado en el aprendizaje profundo para la detección de bordes de edificios, una estrategia de aprendizaje de múltiples tareas que combina la segmentación de edificios y un método guiado por geometría para la reconstrucción de polígonos de edificios (Wu et al., 2023).

Se presenta EANet, una arquitectura de red que integra la segmentación de imágenes y las redes de percepción de bordes para mejorar significativamente la extracción automática de información de edificios a partir de imágenes de teledetección. La arquitectura aborda los desafíos en la extracción de edificios y enfatiza la importancia de mejorar la representación de características para un reconocimiento preciso a nivel de píxeles (Yang et al., 2020). Se ha desarrollado un nuevo método de aprendizaje profundo llamado extracción de edificios basada en segmentación de instancias, que combina el aprendizaje de transferencia con la Red Neuronal Convolucional Regional de Máscara (Mask R-CNN) y PointRend. La técnica se enfoca específicamente en aprovechar el aprendizaje de transferencia para mejorar el rendimiento del modelo Mask R-CNN e incorpora PointRend para lograr máscaras de segmentación más precisas y detalladas.

La metodología propuesta abarca múltiples etapas, incluida la recopilación de datos, el redimensionamiento, la segmentación de edificios y la evaluación de precisión, para describir el flujo de la técnica de extracción de edificios (Fatty et al., 2023). Se presenta un marco en forma de MTGCD-Net, un modelo de red de detección de cambios guiado por múltiples tareas, para abordar el problema en cuestión. El modelo MTGCD-Net introduce tres tareas auxiliares para mejorar el rendimiento. La primera tarea implica la clasificación píxel a píxel para predecir techos y fachadas de edificios. La segunda tarea se centra en aprender desplazamientos de techo a huella para corregir desalineaciones, mientras que la tercera tarea involucra el aprendizaje de flujos de coincidencia de techos entre imágenes aéreas bitemporales para manejar problemas de incompatibilidad de techos. Estas tareas auxiliares proporcionan información valiosa de análisis y coincidencia de edificios, que luego

se fusiona con la rama principal del modelo de Detección de Cambios de Edificios (BCD) mediante un módulo de destilación multimodal (Pang et al., 2023).

Se presenta Sci-Net, una arquitectura de red neuronal desarrollada específicamente para la segmentación precisa de edificios en imágenes aéreas de diferentes resoluciones espaciales. Los desafíos en la segmentación de edificios, que incluyen fragmentación, subsegmentación y sobresegmentación, se abordan mediante el modelo Sci-Net propuesto. Sci-Net se destaca como una arquitectura de red neuronal invariante a la escala que utiliza la representación jerárquica de U-Net y el denso agrupamiento de pirámide espacial atrópica para extraer representaciones detalladas de múltiples escalas (Nasrallah et al., 2023). Se ha introducido un marco de aprendizaje profundo para la segmentación de escenas aéreas, utilizando un enfoque de redes neuronales convolucionales basado en superpíxeles. El marco presenta una arquitectura de dos etapas: la primera etapa implica la segmentación a nivel de baja resolución utilizando la técnica de superpíxeles SLIC, mientras que la segunda etapa se centra en la clasificación a nivel de píxel utilizando una arquitectura de redes neuronales convolucionales multinivel (Behera et al., 2023).

Se presenta un conjunto de datos llamado ISAI (Segmentación de Instancias en Imágenes Aéreas) que se enfoca específicamente en los desafíos asociados con la segmentación de instancias en imágenes aéreas. El conjunto de datos tiene como objetivo proporcionar una plataforma para evaluar y avanzar en las técnicas de segmentación de instancias en este dominio. En este sentido, se examinan dos enfoques ampliamente utilizados de segmentación de instancias, Mask R-CNN y PANet, en el contexto de imágenes aéreas (Waqas Zamir et al., s. f.).

Se realiza un análisis comparativo entre dos modelos de aprendizaje profundo, UNet y PSPNet, con el objetivo de la identificación y segmentación de edificios. La investigación se centra en abordar los desafíos que enfrentan las administraciones municipales en países en desarrollo en lo que respecta a la identificación y el mapeo de edificios para la planificación del crecimiento urbano y otros proyectos relacionados (Liu et al., 2020). Se presenta MAN-Net, un modelo de aprendizaje profundo diseñado para la detección de cambios en paisajes urbanos, con un énfasis particular en la detección de cambios en edificios mediante imágenes de satélite. El modelo MAN-Net adopta una arquitectura mejorada de U-Net, aprovechando la segmentación multiescala e incorporando módulos

multiescala para extraer características en diversas escalas e intervalos de tiempo. Además, el modelo incorpora un módulo de atención que mejora selectivamente o suprime mapas de características según su importancia y relevancia (Vasavi et al., 2022).

Se presenta un enfoque novedoso llamado red de auto-mutación (SMN), basado en redes neuronales convolucionales (CNN), capaz de ajustar de forma adaptativa los valores de los parámetros de los filtros convolucionales en respuesta al dominio de la imagen de entrada. Esta adaptabilidad mejora el rendimiento de la segmentación adaptable al dominio. El SMN incorpora una técnica de mutación de parámetros para modificar dinámicamente los parámetros y una técnica de fluctuación de parámetros para introducir variaciones aleatorias en los parámetros. Al utilizar estas técnicas de mutación y fluctuación, el SMN logra el ajuste adaptativo y el ajuste fino de los parámetros para imágenes que provienen de dominios diversos, lo que finalmente conduce a predicciones mejoradas en tareas de segmentación adaptable al dominio (Lee et al., 2021).

También encontramos desarrollos como el novedoso modelo Efficient-UNet++, diseñado para detectar cambios en edificios en imágenes de teledetección de alta resolución. El estudio se enfoca en mejorar la capacidad del modelo para capturar características locales y globales. Esto se logra mediante la integración de Multi-Headed Self-Attention (MHSA) y Depthwise Separable Convolution (DW Conv) en la arquitectura del modelo (Chen et al., s. f.).

5. Desarrollo del proyecto

Definición del problema

La urbanización constante y, en ocasiones, descontrolada, ha hecho que muchas áreas geográficas y barrios sufran cambios significativos en sus paisajes en cortos periodos de tiempo. Estos cambios pueden ser indicativos de diversos fenómenos, como el desarrollo económico, movimientos demográficos o transformaciones en el uso del suelo. Para autoridades, planificadores urbanos, investigadores y la ciudadanía en general, es esencial contar con herramientas precisas que permitan visualizar y comprender estos cambios de manera objetiva para planificar mejor los futuros barrios y anticiparse a los problemas y necesidades que puedan aparecer.

Sin embargo, a pesar de la disponibilidad de imágenes satelitales, la falta de herramientas públicas que segmenten y cuantifiquen las estructuras en estas imágenes ha dificultado este análisis. Este será el problema principal que abordaremos en el presente proyecto: desarrollar un modelo capaz de identificar, segmentar y cuantificar edificaciones en imágenes satelitales para determinar la evolución urbanística de una región determinada.

Solución adoptada

Ante la creciente necesidad de interpretar la rápida transformación urbanística, este proyecto adoptará una solución innovadora a través del desarrollo de un modelo basado en la arquitectura Segformer. Este modelo será entrenado para identificar y segmentar estructuras de edificios en imágenes satelitales. La fortaleza del modelo radica en su capacidad de discernir con claridad la morfología urbana, facilitando así la cuantificación y el seguimiento de la evolución de las construcciones a lo largo del tiempo. Para complementar esta capacidad analítica y extender su empleo, se creará una herramienta de código abierto destinada al uso público. Esta herramienta aprovecha una secuencia temporal de imágenes satelitales para ofrecer una visualización detallada de la evolución

urbanística, posibilitando que autoridades, urbanistas y ciudadanos puedan acceder a una plataforma que les permita planificar y gestionar el crecimiento urbano de manera proactiva y fundamentada.

Estudio previo de arquitecturas y trabajos relacionados

Para hacer frente a este problema, en el punto anterior titulado “Estado del arte” se detalló la investigación llevada a cabo para conocer trabajos previos y las posibles soluciones que se podrían plantear. Además, se conocieron las arquitecturas más relevantes en el ámbito de la segmentación semántica en la actualidad, algunas de ellas han demostrado ser efectivas en tareas similares, pero para este proyecto, se decidió aprender y experimentar con soluciones en la vanguardia de la tecnología. En particular, se decidió explorar las capacidades de Segformer y Deeplabv3+, dos arquitecturas que han demostrado ser particularmente prometedoras en tareas de segmentación avanzada. La elección de estas arquitecturas busca aprovechar los últimos avances en el campo y garantizar resultados precisos y efectivos para el análisis urbanístico propuesto. Además, nos permitirá comparar el rendimiento de dos de las arquitecturas más modernas en este campo.

Hardware y software utilizados.

Para la realización de este proyecto se optó por utilizar Google Colab debido a los problemas de compatibilidad que surgieron al intentar trabajar en mi equipo particular, además de que mi GPU no es tan avanzada como las que Google facilita a los usuarios premium. También se evitan los problemas de compatibilidad de librerías, ya que en mi equipo al instalar una versión de Cuda compatible con mi GPU ciertas librerías esenciales de Python presentaban incompatibilidades con dicha versión de Cuda. Dada esta situación, Google Colab surgió como una solución viable por su acceso a hardware más moderno y la facilidad para gestionar dependencias de software.

Si bien en Colab tampoco estuve exento de problemas ya que si trataba de trabajar con mi dataset en mi cuenta de Google Drive se producían breves desconexiones entre ambas plataformas y provocaban la interrupción del entrenamiento.

Una vez solucionados todos los incidentes, he concluido el desarrollo en un entorno virtual de Google Colab con las siguientes características:

- Memoria RAM: 12,7GB
- CPU: Intel(R) Xeon(R) CPU @ 2.30GHz

En cuanto a la GPU, se eligieron dos modelos específicos dependiendo de la fase del proyecto:

- Pruebas previas: realizadas con la GPU NVIDIA A100. Esta GPU proporciona un buen equilibrio entre rendimiento y costo para la fase de pruebas y ajustes iniciales.

- Entrenamiento del modelo definitivo: cambio a la GPU NVIDIA Tesla V100. Al ser una de las GPUs más avanzadas disponibles en Google Colab, ofrece un rendimiento superior, permitiendo un entrenamiento más rápido y eficiente del modelo basado en la arquitectura Segformer. Además, al tener más memoria te permite aumentar el tamaño del batch en la fase de entrenamiento.

Herramientas y librerías

El desarrollo del proyecto requirió el uso de varias librerías en Python, que proporcionan una gama amplia de herramientas y funcionalidades para el procesamiento, entrenamiento y análisis de imágenes satelitales. A continuación, se destacan las librerías más esenciales para la tarea:

- PyTorch: Esta es la biblioteca principal utilizada para construir, entrenar y evaluar el modelo de segmentación semántica. PyTorch es una librería de aprendizaje automático de código abierto que proporciona un conjunto flexible de herramientas para la investigación en inteligencia artificial. Es especialmente conocida por su interfaz intuitiva y su capacidad para trabajar con cómputo en GPU, lo que la hace ideal para entrenar modelos de deep learning. Pese a que inicialmente se tenía poca experiencia con PyTorch, su comunidad y los recursos disponibles online jugaron un papel crucial en la curva de aprendizaje. Además de su documentación oficial, se accedió a tutoriales, foros y cursos en línea, lo que permitió adquirir una comprensión sólida de la biblioteca y aprovechar al máximo sus características y ventajas para este proyecto.

- Transformers: Una librería que proporciona implementaciones de modelos de aprendizaje automático de vanguardia, incluida la arquitectura Segformer que se utiliza en este proyecto. Especialmente útil para acceder a arquitecturas preentrenadas y herramientas relacionadas.
- Albumentations: Esta biblioteca se utiliza para el aumento de imágenes, lo cual es crucial para mejorar la robustez del modelo al proporcionar variaciones de las imágenes de entrenamiento.
- OpenCV: Una de las bibliotecas más populares para el procesamiento de imágenes y visión por computadora. En este proyecto, se empleó para tareas como la lectura, escritura y manipulación de imágenes satelitales.
- Pandas y Numpy: Estas bibliotecas facilitan la manipulación de datos y operaciones matemáticas, respectivamente. Son esenciales para el tratamiento de metadatos y la manipulación de matrices de imágenes.
- Tensorboard: Proporciona herramientas para visualizar y monitorear el progreso del entrenamiento, lo que es fundamental para entender y mejorar el rendimiento del modelo.

Las demás librerías, aunque también son importantes para tareas específicas, son complementarias a las principales mencionadas anteriormente.

6. Datasets

Lo más importante para empezar a desarrollar un modelo de inteligencia artificial son los datos. Para este proyecto necesitábamos no solo las imágenes satelitales que queremos segmentar sino además las máscaras de dichos objetos para de esta forma enseñar a nuestro modelo a diferenciar entre unos objetos y otros.

Los conjuntos de datos utilizados son públicos y se han facilitado años atrás en eventos o concursos es por ello que, aunque para ambos haya una división en imágenes para

entrenamiento y test, las imágenes de test no nos sirven porque no se facilitan las máscaras o datos necesarios para obtenerlas.

Otra opción podría haber sido la de obtener las imágenes satelitales desde páginas de organismos públicos y crear nuestro propio dataset, pero en este caso no tendríamos máscaras y tendríamos que encarar el problema con un modelo de aprendizaje no supervisado, que por lo general necesitan de grandes cantidades de datos y mucho tiempo para alcanzar unos resultados similares. Además, no siempre es sencillo obtener estas imágenes de forma automática desde los organismos oficiales ni tienen las mismas características.

Descripción de los conjuntos de datos

El análisis y la comprensión urbanística de una región mediante la segmentación semántica requieren la utilización de conjuntos de datos adecuados que proporcionen información relevante y precisa. En este proyecto, se han utilizado dos datasets de renombre para llevar a cabo el entrenamiento y la evaluación del modelo. A continuación, se detalla una descripción de cada uno:

1. INRIA: Aerial Image Labeling Dataset

- **Fuente:** INRIA Aerial Image Labeling[1]
- **Características:** El dataset consta de 190 imágenes con una resolución de 5000x5000 píxeles. La peculiaridad de este conjunto radica en que los edificios no aparecen tan alejados, permitiendo identificar sus detalles con mayor precisión en comparación con Spacenet.
- **Diversidad geográfica:** Las imágenes capturan diversas ciudades, como Austin, Chicago, Viena o Kitsap entre otros. Esta variedad garantiza una visión amplia y diversificada de la arquitectura, abarcando desde grandes metrópolis hasta poblaciones más rurales, tal y como se puede ver en la ***Figura 4***.

[1] INRIA url: https://project.inria.fr/aerialimagelabeling/

Imágenes del dataset INRIA

Figura 4. *Imágenes representativas del dataset INRIA*

Fuente: link

- Máscaras: Cada imagen viene acompañada de su respectiva máscara, facilitando el proceso de entrenamiento y validación.

2. Spacenet

- Fuente: Spacenet en Kaggle[2]

- Características: Este dataset comprende 1000 imágenes de 1024x1024 píxeles. Una característica distintiva es la presencia de un cuarto canal con datos geográficos en las imágenes con extensión ".tif".

- Perspectiva: A diferencia de INRIA, en Spacenet, los edificios suelen aparecer más alejados o más pequeños, lo que en ocasiones dificulta su diferenciación respecto al fondo.

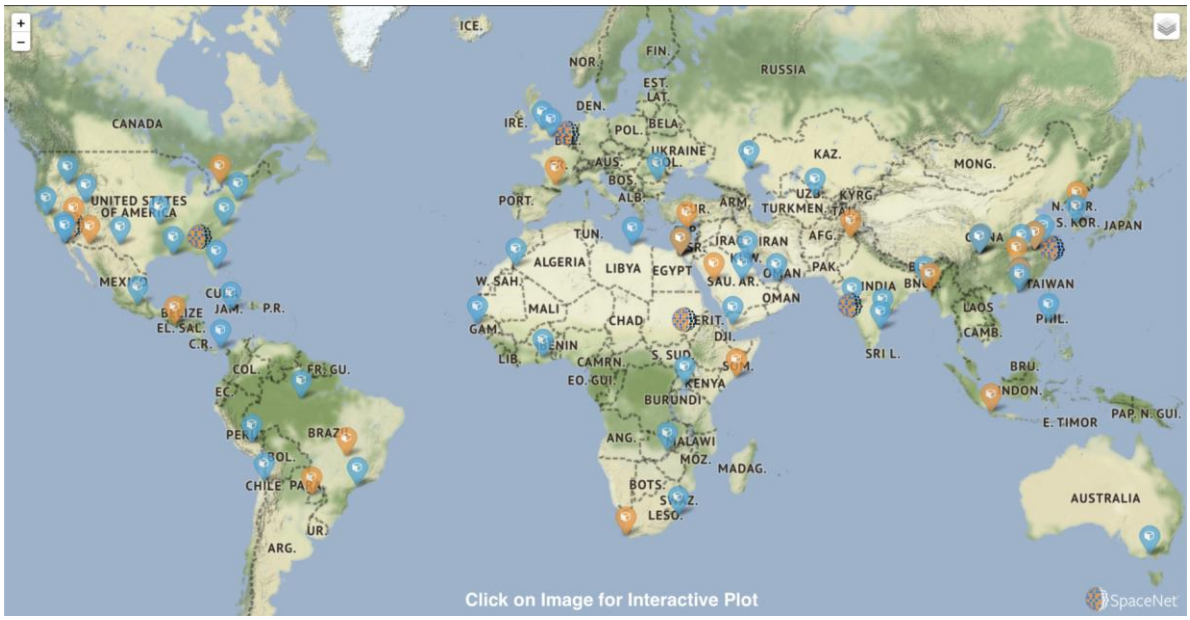
- Diversidad geográfica: Las imágenes se extienden a lo largo y ancho del globo tal y como muestra la ***Figura 5***, ofreciendo un amplio espectro de localizaciones. Además, dentro de cada localidad, se proporcionan imágenes temporales que reflejan el cambio y la evolución de la urbanización a lo largo del tiempo.

- Máscaras: A diferencia de INRIA, Spacenet no viene con máscaras tradicionales. En su lugar, se ofrecen archivos geojson, a partir de los cuales se generan las máscaras.

[2] Spacenet url: https://www.kaggle.com/datasets/amerii/spacenet-7-multitemporal-urban-development

Localizaciones de las imágenes presentes en el dataset Spacenet

Figura 5. *Localizaciones de las que se obtienen las imágenes para el dataset Spacenet*



Fuente: link

Preprocesamiento

Para asegurar que los datos se integran y utilizan eficientemente durante el proceso de entrenamiento, se realizó una etapa de preprocesamiento para cada conjunto de datos por separado por tener necesidades y características diferentes. Se realizaron las siguientes operaciones:

1. Renombrado: Ambos datasets se procesaron para renombrar las imágenes con identificadores numéricos, facilitando su manipulación y acceso.

2. División de imágenes (INRIA): Las imágenes originales de 5000x5000 píxeles se subdividieron en segmentos de 1000x1000 píxeles para un manejo más eficiente y para aumentar el número de ejemplos de entrenamiento, obteniendo finalmente un total de 4500 imágenes.

3. Estandarización: Durante el entrenamiento, las imágenes se estandarizaron para asegurar una distribución uniforme de los píxeles, lo que a menudo ayuda a mejorar la

convergencia de los modelos. Esta estandarización no se aplicó a las imágenes en etapas anteriores para mantener su integridad original.

4. Filtrado (INRIA): Se llevó a cabo un proceso de filtrado para eliminar o reducir la cantidad de segmentos generados que no contenían edificios, optimizando el conjunto de entrenamiento.

5. Ajustes en Spacenet: Se eliminó el cuarto canal con datos geográficos de las imágenes .tif. Además, se crearon máscaras a partir de los archivos geojson proporcionados.

Estos datasets, acompañados de las etapas de preprocesamiento mencionadas, forman la base sobre la cual se entrenará y evaluará el modelo de segmentación semántica para el análisis urbanístico propuesto, si bien en este punto aún nos queda por definir qué conjunto de datos se adapta mejor a nuestro problema o si por el contrario los combinaremos. Esto se decidirá a partir de los resultados que obtengamos en las pruebas que se describen en el siguiente punto.

7. Selección de la arquitectura

Antes de comenzar con el desarrollo y el entrenamiento de nuestro modelo quedan por definir los siguientes puntos:
- La arquitectura que utilizaremos.
- El conjunto de datos que utilizaremos o si utilizaremos una combinación de ambos.

Para aclarar estos aspectos se han completado los siguientes notebooks presentes en el repositorio[3]:

- 4_pruebas_entrenamiento_datasets (realizadas con Segformer)
- 1_pruebas_deeplabv3+

En estos cuadernos se recogen las pruebas realizadas con dos de las arquitecturas Segformer y DeepLabv3+ que conformar el estado del arte actual en el campo de la segmentación semántica. Esta comparativa enriquece el análisis que recoge el paper fundacional de Segformer (Xie et al., s. f.), donde se comparan estas arquitecturas, pero utilizando conjuntos de datos compuestos por imágenes de entornos urbanos. Los resultados que obtuvieron ya fueron comentados en el apartado “Estado del arte”. Sin embargo, dicha tarea difiere significativamente del objetivo de nuestro proyecto actual, que se centrará en el análisis de imágenes satelitales.

Por otra parte, aunque SAM es la arquitectura más moderna y promete segmentar cualquier tipo de objeto en cualquier imagen, se ha descartado su uso en este proyecto por lo exigente que puede ser en términos computacionales y porque es tan reciente que las librerías de Python que lo implementan aún están en fases muy tempranas de desarrollo.

Tests iniciales para elegir la arquitectura óptima

Para la identificación precisa de edificios en imágenes satelitales y su posterior segmentación semántica, es esencial optar por una arquitectura que proporcione una combinación eficiente de precisión, velocidad y economía de recursos. Se llevaron a cabo pruebas preliminares con las arquitecturas Segformer y DeepLabv3+ en las mismas condiciones, sin dar ventaja a una sobre la otra y utilizando los dos conjuntos de datos que tenemos disponibles: INRIA y Spacenet.

[3] Repositorio: https://github.com/AVR185/09MIAR_10_A_2022-23_Trabajo-Fin-de-Master

Los entrenamientos se realizaron en entornos Google Colab con la GPU V100, con configuraciones de 8 epochs y un tamaño de batch de 4 y los siguientes modelos preentrenados:

- Segformer B5[4]
- DeepLabV3+ Resnet 101[5]

Los resultados obtenidos son los siguientes:

Tabla 1. *Rendimiento arquitectura Segformer en el entrenamiento con cada dataset*

Segformer		
	INRIA	**Spacenet**
Pérdida promedio en test	**0.1111**	**0.1471**
Accuracy promedio en test	**0.9581**	**0.9379**
IoU promedio en test	0.7614	0.2234
F1 Score promedio en test	0.8627	0.3566

Tabla 2. *Rendimiento arquitectura DeepLabV3+ en el entrenamiento con cada dataset*

DeepLabV3+		
	INRIA	**Spacenet**
Pérdida promedio en test	**0.1246**	**0.1894**
Accuracy promedio en test	**0.9516**	**0.9343**
IoU promedio en test	0.7272	0.1135
F1 Score promedio en test	0.8396	0.1945

[4] Segformer: https://huggingface.co/nvidia/mit-b5
[5] DeepLabV3+: https://pytorch.org/vision/main/models/generated/torchvision.models.segmentation.deeplabv3_resnet101.html

Como se puede observar las métricas utilizadas durante estas pruebas y que se seguirán usando a lo largo del proyecto son:

- Precisión Pixel-wise (Pixel-wise accuracy): Esta métrica calcula la proporción de píxeles predichos correctamente en todo el conjunto de datos. Es útil para obtener una idea general de cómo se está desempeñando el modelo, pero no proporciona detalles sobre la calidad de la segmentación.

- Pérdida (Loss): Se utilizó una función de pérdida para optimizar el modelo. La pérdida da una indicación de cuán lejos está la predicción del modelo del valor verdadero. Una pérdida más baja indica una mejor predicción.

- IoU (Intersection over Union): Esta métrica mide la superposición entre la predicción y la verdad fundamental. Proporciona una medida más detallada de la calidad de la segmentación, siendo 1 una superposición perfecta y 0 ninguna superposición.

- Puntuación F1 (F1 Score): Es una métrica que combina precisión y exhaustividad para proporcionar una única medida de la calidad de la predicción. Un valor más alto es mejor, siendo 1 la mejor puntuación y 0 la peor.

Selección y justificación de la arquitectura Segformer

Basándonos en los resultados obtenidos y que vienen recogidos en la ***Tabla 1*** y en la ***Tabla 2***, es evidente que, aunque ambas arquitecturas obtienen buenos resultados, Segformer supera a DeepLabv3+ en términos de precisión y otras métricas en ambos conjuntos de datos.

Específicamente, para el dataset INRIA, que se asemeja más a nuestro caso de uso final con imágenes de Google Earth, Segformer muestra un rendimiento superior, especialmente en términos de IoU y F1 Score. Estas métricas son esenciales en la segmentación semántica, ya que indican la calidad y precisión del modelo al segmentar las regiones de interés.

Además, no nos tenemos que olvidar del análisis descrito en el apartado “DeepLabv3+ vs Segformer” con el dataset ADE20K, donde la arquitectura Segformer B5 mostró un mIoU (mean Intersection over Union) significativamente más alto que DeepLabV3+, a pesar de tener una cantidad de parámetros comparable o muy similar. Esto sugiere que Segformer es más eficiente en la utilización de sus parámetros, logrando una mayor precisión con una complejidad similar.

En resumen, la elección de Segformer se justifica por las siguientes razones:
1. Rendimiento Superior: Las métricas demostraron que Segformer supera a DeepLabv3+ en términos de precisión, IoU y F1 Score en los conjuntos de datos probados.
2. Eficiencia de Parámetros: Segformer logra una alta precisión con una cantidad de parámetros comparable a DeepLabV3+.
3. Generalización: Aunque Spacenet tiene un zoom más alejado y, por ende, presenta desafíos distintos, la arquitectura Segformer sigue siendo robusta y muestra resultados superiores en comparación con DeepLabV3+.
4. Adaptabilidad: Dada la diversidad de datasets y la variabilidad en las imágenes, necesitamos una arquitectura que pueda adaptarse y generalizar bien. Segformer ha demostrado ser esa arquitectura.

La elección de una arquitectura es crucial para garantizar la precisión y eficiencia del modelo. En base a los tests y análisis realizados, Segformer ha demostrado ser la mejor opción para nuestra tarea de segmentación semántica en imágenes satelitales, lo que garantizará resultados precisos y confiables en el uso real del modelo.

Pruebas para establecer una estrategia con los datasets

A la vista de los resultados, posteriormente se han realizado ensayos con los diferentes datasets, para comprobar cuál es la mejor estrategia para conseguir que el modelo identifique correctamente los edificios en imágenes de Google Earth. Por lo tanto, en estas pruebas no solo se busca las mejores estadísticas en evaluación, sino ver que dataset le proporciona mejor información a nuestro modelo para generalizar mejor con imágenes de Google Earth.

Los datasets proporcionan la base fundamental para el entrenamiento de cualquier modelo de aprendizaje automático. Es esencial experimentar y analizar varios escenarios de entrenamiento para determinar cuál es el más adecuado para la tarea en cuestión, especialmente cuando el objetivo final no es solo lograr buenas métricas, sino también una generalización robusta en imágenes reales, como las obtenidas de Google Earth.

Los experimentos se realizan con un modelo Segformer B5 preentrenado disponible en Hugging Face a igualdad de epochs, tamaño de batch, learning rate, etc. Para poder comparar los resultados con mayor facilidad. Los resultados obtenidos son los siguientes:

Entrenamiento con el dataset INRIA exclusivamente: Este experimento buscaba determinar el rendimiento del modelo cuando se le expone únicamente a imágenes del dataset INRIA. Estas imágenes tienen una resolución y detalles que se asemejan mucho a lo que esperaríamos de las imágenes de Google Earth.

- Pérdida promedio en test: **0.111**
- Accuracy promedio en test: **0.958**
- IoU promedio en test: 0.761
- F1 Score promedio en test: 0.863

Entrenamiento con el dataset Spacenet exclusivamente: Spacenet, al tener un mayor alejamiento y cubrir una gama más amplia de localidades, presenta un desafío diferente. El objetivo de este experimento era observar cómo se comporta el modelo cuando se le expone a una variedad más amplia, aunque con un nivel de detalle menor.

- Pérdida promedio en test: **0.147**
- Accuracy promedio en test: **0.938**
- IoU promedio en test: 0.223
- F1 Score promedio en test: 0.357

Entrenamiento con un dataset mixto de INRIA y Spacenet: Esta estrategia buscaba combinar lo mejor de ambos mundos. Las imágenes detalladas de INRIA y la diversidad geográfica de Spacenet podrían ofrecer un equilibrio, permitiendo al modelo aprender detalles finos y al mismo tiempo generalizar a través de diferentes tipos de paisajes urbanos.

- Pérdida promedio en test: **0.117**
- Accuracy promedio en test: **0.957**
- IoU promedio en test: 0.682
- F1 Score promedio en test: 0.799

Entrenamiento en sus primeras fases con el dataset INRIA y realizar transfer learning posteriormente con el dataset Spacenet: Aquí, la idea era aprovechar primero el aprendizaje detallado de INRIA y luego adaptar ese aprendizaje a la diversidad que ofrece Spacenet. Sin embargo, parece que esta estrategia no fue tan beneficiosa como se esperaba, reflejándose en las métricas.

- Pérdida promedio en test: **0.310**
- Accuracy promedio en test: **0.897**
- IoU promedio en test: 0.279
- F1 Score promedio en test: 0.422

Las conclusiones se comentan a continuación.

Selección y justificación de un dataset conjunto

Después de considerar cuidadosamente las métricas de evaluación y, más importante aún, los resultados reales al probar con imágenes de Google Earth, se optó por utilizar el dataset mixto de INRIA y Spacenet para el entrenamiento final. La decisión se basó en varios factores cruciales:

1. Rendimiento comparable a INRIA: El dataset mixto presentó métricas que estaban muy cerca de las del dataset INRIA, que en sí mismo proporcionó los mejores resultados. Esto indica que la combinación no comprometió la calidad del aprendizaje, sino que más bien ofreció un equilibrio.

2. Generalización mejorada: Al entrenar con el dataset mixto, se observó que el modelo era capaz de generalizar mejor en las imágenes de Google Earth. Esto es crucial porque

nuestro objetivo no es solo obtener buenos números en términos de métricas con un conjunto de datos, sino asegurarse de que el modelo pueda desempeñarse bien en situaciones del mundo real.

3. Diversidad del entrenamiento: El dataset mixto proporciona una rica diversidad en términos de detalles y geografía. Mientras que INRIA aporta detalles finos y alta resolución, Spacenet aporta variedad geográfica. Esta combinación permite al modelo estar preparado para una gama más amplia de escenarios, lo que mejora su robustez y adaptabilidad.

4. Evitar el sobreajuste: Entrenar solo con un tipo de imagen podría haber llevado a un sobreajuste hacia ese estilo particular de imágenes. Al combinar datasets, se reduce el riesgo de que el modelo se ajuste demasiado a un estilo y no pueda adaptarse bien a otros.

En conclusión, la elección del dataset mixto no solo se basó en métricas de rendimiento, sino también en la capacidad del modelo para adaptarse y generalizar en situaciones del mundo real. Esta combinación garantiza que el modelo esté bien preparado para identificar edificios en una variedad de imágenes satelitales y, más concretamente, en las de Google Earth.

La combinación de estos conjuntos de datos y su procesado previo permitió una formación robusta y adaptativa del modelo a distintos escenarios y condiciones.

8. Código

El código diseñado puede consultarse en el siguiente repositorio de Github[6]. Se trata de una implementación de una arquitectura Segformer para la segmentación semántica de imágenes utilizando PyTorch, Transformers y otros paquetes auxiliares. A continuación, describiremos la estructura y las funciones principales del código.

Estructura del código

El código puede segmentarse en varios bloques, basándonos en su funcionalidad:

1. Importación de bibliotecas: El código comienza con la importación de bibliotecas y paquetes esenciales como “torch”, “transformers”, “albumentations”, “cv2”, entre otros.

2. Clase Dataset: Se define la clase “ImageSegmentationDataset” que hereda de “Dataset” de Pytorch. Esta clase sirve para cargar y procesar las imágenes y sus máscaras correspondientes. En el método get_item de la clase se realiza la normalización de las imágenes.

3. Transformaciones de aumento de datos: Usando la biblioteca “albumentations” se implementa la técnica de Data Augmentation para incrementar el conjunto de datos y mejorar la robustez del modelo mediante la transformación de las imágenes que tenemos. En primer lugar, se introduce la rotación de hasta 360 grados en las imágenes. También, se aplican variaciones en brillo, contraste y saturación, técnicas cruciales en la segmentación de imágenes satelitales debido a la amplia variabilidad de estas características en el campo de las imágenes satelitales. En este caso se ha aplicado zoom puesto que habría que configurarlo de forma distinta para cada conjunto de datos por tener características muy diferentes entre sí.

[6] Repositorio: https://github.com/AVR185/09MIAR_10_A_2022-23_Trabajo-Fin-de-Master

4. Preparación del modelo: Se inicializa el modelo SegformerForSemanticSegmentation de la biblioteca transformers, que se entrenará con imágenes satelitales para identificar edificios o construcciones. El modelo ya está preentrenado.

5. Funciones de entrenamiento y validación: Para un mejor mantenimiento y mayor legibilidad del código, el bucle de entrenamiento y validación se extraen en métodos independientes que a su vez se llamaran desde el método principal de entrenamiento "training".

- train_one_epoch: Entrena el modelo durante una época y devuelve métricas como precisión, pérdida, IoU y F1.

- validate_one_epoch: Valida el modelo en el conjunto de validación y devuelve métricas similares.

- training: Combina las funciones anteriores y realiza el entrenamiento del modelo a lo largo de múltiples épocas (el número se indica por parámetro). Cabe destacar que se ha implementado la técnica de K-Fold o cross-validation para aprovechar al máximo los datos de entrenamiento disponibles y porque nunca lo había desarrollado en Pytorch, de este modo aprendía y experimentaba con esta librería.

Funciones principales y su lógica

1. train_one_epoch:

- Configura el modelo en modo de entrenamiento.
- Itera sobre el conjunto de datos de entrenamiento.
- Realiza una propagación hacia adelante a través del modelo.
- Calcula la pérdida y realiza una propagación hacia atrás para actualizar los pesos del modelo.
- Calcula métricas como precisión, IoU y F1.
- Devuelve un resumen de las métricas.

2. validate_one_epoch():

- Configura el modelo en modo de evaluación.
- Itera sobre el conjunto de datos de validación sin realizar backpropagation.
- Calcula métricas como precisión, IoU y F1.
- Devuelve un resumen de las métricas.

3. training:

- Inicializa registros para TensorBoard.
- Realiza la validación cruzada K-Fold.
- Para cada época y pliegue, entrena y valida el modelo.
- Guarda el modelo si su rendimiento mejora.
- Utiliza un mecanismo de parada temprana para detener el entrenamiento si el modelo no mejora después de un número determinado de épocas.
- Maneja excepciones para gestionar errores que puedan surgir durante el entrenamiento.

En general, este código ofrece una estructura robusta para abordar el problema de segmentación semántica en imágenes. Combina técnicas de aumento de datos, validación cruzada y parada temprana para asegurar un modelo eficiente y resistente al sobreajuste.

Clases

La única clase definida en el código es “ImageSegmentationDataset”, que representa un conjunto de datos de segmentación de imágenes. Esta clase facilita la carga y el procesamiento de las imágenes y sus correspondientes máscaras de segmentación. Además, la clase tiene la capacidad de dividir las imágenes y las máscaras en conjuntos de entrenamiento y validación.

1. ImageSegmentationDataset

- init: Constructor que inicializa el dataset. Toma como entrada la ruta raíz del conjunto de datos, el extractor de características, y una bandera para decidir si cargar los datos de entrenamiento o validación.

- len: Devuelve la cantidad de imágenes en el conjunto de datos.

- getitem: Carga una imagen y su correspondiente máscara, las transforma y las prepara para la entrada al modelo.

En resumen, este código está diseñado para llevar a cabo la tarea de segmentación semántica usando un modelo preentrenado llamado Segformer. Se ha proporcionado una estructura detallada para cargar datos, aplicar aumentos, entrenar y validar el modelo usando validación cruzada K-Fold.

9. Entrenamiento

Metodología

Para abordar la tarea de identificación de edificios en imágenes satelitales mediante segmentación semántica, es esencial utilizar estrategias de optimización que refuercen la capacidad del modelo de generalizar sobre datos no vistos.

- Data Augmentation (Aumento de Datos): Se han utilizado técnicas de data augmentation para aumentar artificialmente el tamaño del conjunto de datos de entrenamiento. Este proceso ayuda a mejorar la generalización del modelo, al exponerlo a variaciones de las imágenes de entrenamiento que no están presentes en el conjunto original. En este proyecto, se ha utilizado la biblioteca Albumentations para llevar a cabo estas augmentaciones. Se ha implementado la rotación de las imágenes hasta 360° y la variación de brillo, contraste y saturación para añadir diversidad y ayudar al modelo a generalizar mejor. No se ha considerado necesario incluir zoom puesto que ya contamos con imágenes con encuadres y especialmente las imágenes de Spacenet si se alejaran aún más los elementos se haría muy complicado para el modelo trabajar con ellas.

- Estandarización de las imágenes: Es esencial que las imágenes que alimentan el modelo tengan un rango y distribución similares para que las características puedan ser aprendidas de manera uniforme. La estandarización ayuda a que la optimización sea más eficiente, ya que permite que el modelo aprenda más rápidamente en el espacio de características normalizado.

- K-fold, Cross Validation: Se ha utilizado el método de validación cruzada K-Fold para mejorar la robustez del modelo. En este enfoque, el conjunto de datos se divide en “K” subconjuntos. El modelo se entrena en “K-1” subconjuntos y se valida en el subconjunto restante. Este proceso se repite “K” veces, asegurando que cada subconjunto sirva como conjunto de validación una vez. Este método es beneficioso porque garantiza que el modelo no dependa de una partición específica de los datos y proporciona una idea más clara sobre cómo el modelo se desempeñaría ante datos desconocidos. Además, es una forma de

aprovechar al máximo todas imágenes de tu dataset de entrenamiento, que en este caso fueron un total de 5453 (el 90% del total).

Configuración

La correcta elección y configuración de hiperparámetros juegan un papel crucial en la eficacia de cualquier modelo de aprendizaje profundo. Durante el entrenamiento, se definieron varios hiperparámetros que influyen directamente en el rendimiento y la convergencia del modelo.:

- Épocas: Se optó por entrenar el modelo durante 6 épocas. Una época es cuando el modelo ha pasado por cada muestra en el conjunto de datos una vez durante el entrenamiento y en este caso hay que tener en cuenta que se realizan 6 épocas por cada subconjunto “K” establecido en el k-fold, en este caso 5, por lo que finalmente las épocas que se completan son 30, lo que es un buen compromiso entre el tiempo de entrenamiento, recursos consumidos y etapas suficientes para que el modelo se ajuste.

- Tamaño de lote (Batch Size): Se seleccionó un tamaño de lote de 4. El tamaño del lote determina la cantidad de muestras que se utilizarán para actualizar los pesos del modelo en cada iteración. Un tamaño de lote más pequeño suele proporcionar una actualización de peso más frecuente, lo que puede llevar a una convergencia más rápida, pero a expensas del tiempo de entrenamiento y de la estabilidad.
En este caso como estamos trabajando con imágenes a color y de grandes dimensiones, estamos limitados por la memoria de la GPU, 8 es el tamaño máximo de batch con el que podíamos trabajar con la tarjeta V100 disponible en el entorno de Google Colab. Sin embargo, tras realizar varias pruebas durante en diversos entrenamientos, obtuve mejores resultados con un tamaño de lote (batch size) de 4, en comparación con los obtenidos al utilizar un lote de tamaño 8.

- K-folds: Como se ha mencionado anteriormente, se usaron 5 particiones para la validación cruzada. Un valor de 5 para K-Fold proporciona una evaluación robusta sin ser excesivamente costoso en términos de tiempo de entrenamiento. Esto asegura que cada muestra en el conjunto de datos se use tanto para entrenamiento como para validación, reduciendo las posibles varianzas en el rendimiento del modelo debido a la división de

datos. El valor se estableció en 5 porque es un estándar en la industria y se ajustaba bien al tamaño de nuestro dataset.

- Paciencia para el Early Stopping: Se estableció una paciencia de 8. Esto significa que el entrenamiento se detendrá si no se observan mejoras en la función de pérdida de validación durante 8 épocas consecutivas. Durante las distintas fases de entrenamiento se observó que era un buen valor ya que es habitual que el modelo durante 2 o 3 épocas consecutivas no mejores sus métricas, pero si no lo hace durante 8 no merece la pena continuar con el entrenamiento y seguir consumiendo recursos. En ese caso es mejor parar y replantear algunos parámetros del entrenamiento.

Resultados

Durante el entrenamiento y la validación, se utilizaron las mismas métricas que usamos para las pruebas: precisión, pérdida, IoU y F1 Score.

Se completaron tres fases de entrenamiento. La última de ellas duró aproximadamente 14 horas en Google Colab. Esta duración es esperada debido a la naturaleza computacionalmente intensiva del entrenamiento de modelos de segmentación semántica, especialmente en conjuntos de datos de imágenes grandes y de alta resolución como las imágenes satelitales.

El mejor rendimiento se obtuvo en la etapa 5 del K-Fold, durante la segunda época. El modelo alcanzó las siguientes estadísticas en validación:

- Precisión: **0,970**
- Pérdida: **0,070**
- IoU: 0,748
- F1 Score: 0,842

Una precisión Pixel-wise de aproximadamente 96.80% en el conjunto de entrenamiento y 97.01% en el conjunto de validación. La pérdida fue de 0.076 en el entrenamiento y 0.070 en la validación. Estos resultados son prometedores y muestran que el modelo ha aprendido

patrones significativos de los datos. La puntuación IoU y F1 también reflejan una buena calidad de segmentación.

En resumen, el modelo entrenado con la arquitectura Segformer ha demostrado ser eficaz en la tarea de segmentar edificios en imágenes satelitales. Los resultados obtenidos indican que el modelo está listo para ser utilizado en la siguiente fase del proyecto: identificar y marcar nuevos edificios en imágenes satelitales a lo largo del tiempo.

10. Evaluación

La evaluación es un proceso fundamental después del entrenamiento de un modelo. No sólo proporciona métricas sobre la eficacia del modelo, sino que también puede indicar áreas donde el modelo puede ser mejorado.

Metodología

El proceso de evaluación aquí detallado fue llevado a cabo utilizando una fracción del 20% del conjunto de datos mixto de INRIA y Spacenet. Esta división se lleva a cabo cuando se inicializa la clase “ImageSegmentationDataset” que deriva de la clase Dataset. Este paso es crucial para asegurar que estamos midiendo el rendimiento del modelo en datos que nunca antes ha visto. Esto proporciona una estimación más realista de cómo se comportará el modelo en escenarios del mundo real.

El proceso de evaluación se desarrolló de la siguiente manera:

1. Preparación: Se cargan las imágenes y sus correspondientes etiquetas desde el conjunto de datos de prueba.

2. Inferencia: Se pasa cada imagen a través del modelo para obtener las salidas (logits). Dado que el modelo fue entrenado para segmentación, estas salidas representan una predicción de máscara para cada clase en cada pixel de la imagen.

3. Upsampling: Las predicciones se ajustan al tamaño original de las imágenes usando interpolación bilineal. Este paso es necesario porque las arquitecturas de segmentación suelen reducir la resolución espacial de las entradas, y es necesario traer las predicciones de vuelta a su resolución original para compararlas con las etiquetas verdaderas.

4. Cálculo de Métricas: Para cada par de predicción y etiqueta verdadera se obtiene:

- Precisión (Accuracy)
- Pérdida (Loss)
- IoU (Intersection over Union)
- Puntuación F1 (F1 Score)

5. Resultados finales: Se acumulan las métricas obtenidas en cada iteración del bucle y se calcula la media de todas las métricas obtenidas para cada imagen en el conjunto de datos de prueba para obtener una visión general del rendimiento del modelo.

Resultados

Con base en el proceso de evaluación anterior, se obtuvieron los siguientes resultados:

- Pérdida promedio en test: **0,076**
- Accuracy promedio en test: **97,03%**
- IoU promedio en test: 75,61%
- F1 Score promedio en test: 84,91%

Estos resultados indican que el modelo tiene una precisión general alta (97,03%) en la predicción de los píxeles de las imágenes, además toman valores muy similares a los de entrenamiento lo que puede ser indicativo de que el modelo es capaz de generalizar bien. Sin embargo, cuando observamos métricas más específicas como IoU y F1, vemos que hay margen para mejorar en términos de la calidad exacta de la segmentación. Un IoU de

75,61% y un F1 Score de 84,91% son indicativos de una buena segmentación, pero hay espacio para refinar aún más el modelo.

11. Resultado final

Tal y como se ha comentado a lo largo del presente proyecto, el objetivo final del es dar una utilidad real al modelo, un valor práctico y tangible, mostrar la evolución urbanística de una determinada zona geográfica a lo largo del tiempo. Para ello, hemos diseñado un cuaderno para Google Colab con una interfaz amigable para que con un par de botones cualquier usuario sea capaz de obtener un gif que muestre la evolución urbanística en una serie temporal de imágenes. En este notebook se realizan las siguientes tareas:

1. Cargar las imágenes: El script carga una serie de imágenes y las almacena en una lista.

2. divide_image: Las imágenes se dividen en porciones de 1000x1000 píxeles para la predicción (al igual que se hizo al preparar los dataset de entrenamiento), esto puede ayudar a procesar imágenes grandes y a obtener predicciones más precisas. Si una imagen no es divisible exactamente por este tamaño.

3. recompose_image: una vez que tenemos las máscaras de las imágenes, recomponemos la máscara siguiendo el mismo orden que se usó para dividirla.

4. generate_gif_overlay: para poder verificar la validez de las predicciones hechas por el modelo se genera un gif en el que se superponen las imágenes y sus respectivas máscaras. De esta forma es más fácil verificar la validez de los resultados obtenidos. El output de este método corresponde con la figura ***Figura 6***.

5. generate_evolution_gif: el objetivo final del proyecto se genera en esta función. Obtenemos un gif como se puede apreciar en la ***Figura 7*** a partir de las máscaras predichas

por el modelo, donde se identifica la evolución que se puede observar en las imágenes facilitadas por el usuario siguiendo el código de colores que se indica a continuación:

- Verde: para marcar los edificios de la primera imagen y los que coinciden de una imagen a la siguiente.
- Azul: se utiliza para marcar los nuevos edificios que se han identificado en una imagen.
- Rojo: para indicar aquellos edificios que han desaparecido con respecto a la anterior imagen.

6. Imprimir el número de edificios en cada imagen: cuando se llama a las funciones para iniciar el proceso, hay una parte del código que realiza una operación de etiquetado para identificar y contar las estructuras detectadas en cada imagen. Luego imprime el número de estructuras detectadas para cada imagen.

En resumen, este notebook procesa imágenes de paisajes urbanos, detecta y cuneta estructuras en ellas, resaltar cambios a lo largo del tiempo y generar un GIF que visualiza estos cambios.

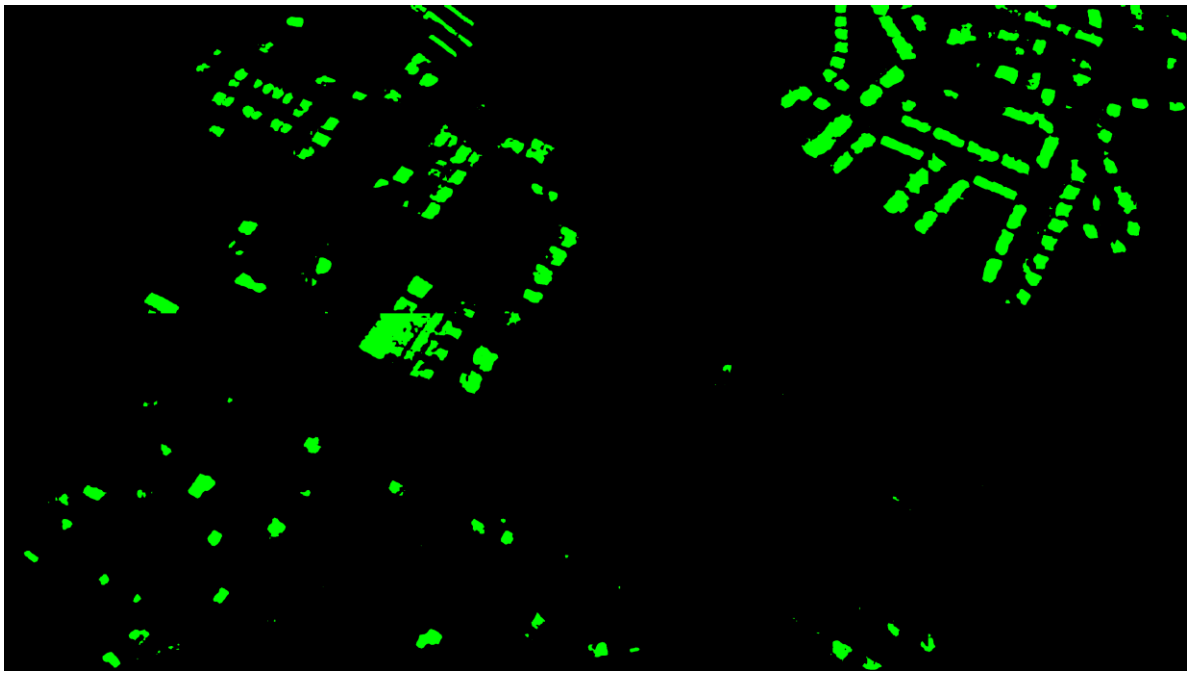
Finalmente, el modelo ha sido puesto a prueba con en este notebook con imágenes obtenidas de Google Earth obteniendo unos resultados muy satisfactorios. A continuación, se muestra un ejemplo y comentamos los puntos más reseñables:

GIF resultado superposición imagen y máscara

Figura 6. *Gif resultado con la superposición de la imagen original y la máscara predicha por el modelo.* **Visualizar:** *Repositorio*

GIF resultado evolución urbanística

Figura 7. *Gif que muestra la evolución urbanística del área urbana.* **Visualizar:** *Repositorio*



Explicación de *Figura 7*:

- Los edificios identificados en la primera imagen del gif se dibujan en verde.
- Cuando se detecta que un edificio desaparece de una imagen a otra se marca en rojo.
- En las sucesivas imágenes que componen el gif, los nuevos edificios se marcan en azul.

Cabe destacar que este resultado tan positivo que se muestra en la ***Figura 6*** ha sido obtenido a partir de imágenes de Google Earth de una localidad española (cerca de Salamanca) con una altura del ojo de 886 metros y con edificios homogéneos, del tipo vivienda unifamiliar. La primera imagen satelital data del año 2003, la segunda de 2011, la tercera de 2018 y la última de 2023, por lo que el gif que hemos obtenido marca perfectamente la evolución urbanística de esta área en los últimos 20 años. El número de edificios detectados han sido:

- Imagen 1: 271 edificios detectados.
- Imagen 2: 404 edificios detectados.
- Imagen 3: 493 edificios detectados.
- Imagen 4: 576 edificios detectados.

Los gifs se pueden visualizar en el repositorio del proyecto en GitHub[7].

Evaluación del cumplimiento de objetivos

1. Objetivo principal: Desarrollar un modelo de segmentación semántica capaz de identificar y cuantificar edificaciones en imágenes satelitales.

La realización de este proyecto refleja un cumplimiento satisfactorio de los objetivos planteados inicialmente. La meta principal ha sido alcanzada con éxito, tras la realización de diversas pruebas nos hemos decantado por la arquitectura SegFormer, adecuada para el análisis semántico satelital, lo cual se evidencia en los resultados obtenidos, que exceden las expectativas iniciales en términos de precisión y eficiencia.

Tras concluir la primera experiencia con SegFormer, no tengo más que opiniones positivas. Este framework ha demostrado ser muy amigable y accesible, simplificando la compleja tarea de programar modelos de segmentación de imágenes. A pesar de estar limitado al

[7] Repositorio: https://github.com/AVR185/09MIAR_10_A_2022-23_Trabajo-Fin-de-Master

uso de par de conjuntos de datos públicos disponibles de concursos para desarrolladores, el modelo ha obtenido muy buenos resultados no solo en la evaluación si no posteriormente en el trabajo en un entorno real con imágenes de Google Earth. Es notable cómo, incluso utilizando imágenes de esta herramienta, para las cuales el modelo no fue específicamente entrenado, SegFormer ha logrado identificar y segmentar con precisión las estructuras urbanas.

Si bien aún hay que realizar pruebas más exhaustivas del modelo para comprobar en qué escenarios el modelo desarrollado flaquea, en este proyecto se puede afirmar que el desempeño de SegFormer con imágenes satelitales es muy bueno. Su capacidad de adaptarse a variaciones de luminosidad y diferenciar tipos de construcciones, incluso con cambios de zoom, habla de su robustez y versatilidad. En términos de programación, la experiencia ha sido intuitiva y directa, lo que representa un gran avance y facilita la inclusión de investigadores y desarrolladores en este campo. Esta capacidad de trabajar eficazmente con diferentes escalas y condiciones de iluminación, sin la necesidad de entrenamiento específico en imágenes de Google Earth, es testimonio del potencial de SegFormer en aplicaciones de teledetección y análisis geoespacial.

2. Objetivo Secundario: Crear una herramienta de usuario amigable que permita cargar imágenes satelitales y obtener análisis sobre la evolución urbanística.
Tal y como se ha comentado, el modelo entrenado se ha implantado en una aplicación de uso real ya que se ha creado una herramienta de usuario intuitiva que facilita la carga y análisis de imágenes satelitales, ofreciendo una ventana al estudio de la evolución urbanística. La integración del modelo entrenado en una interfaz accesible permite generar análisis automatizados y proporciona visualizaciones y métricas claras, cumpliendo así con los objetivos asociados a esta meta. La herramienta, disponible en forma de un notebook para Google Colab, es de fácil manejo para cualquier usuario interesado en la evolución urbanística de una zona específica, y puede obtener las imágenes fácilmente y de forma gratuita en Google Earth.

En este proyecto queda de manifiesto cómo la inteligencia artificial puede ofrecer a la sociedad herramientas gratuitas y de fácil manejo, accesibles para todos, lo que es especialmente notable dada la simplicidad de la solución implementada. La eficacia de esta técnica se refleja claramente en la visualización generada, donde el uso de máscaras para

marcar las construcciones nuevas y aquellas que han desaparecido (en azul y rojo, respectivamente) proporciona una comprensión instantánea de la evolución urbana. Además, el cuaderno que se ha desarrollado también cuantifica el número de edificaciones en las imágenes, ofreciendo una métrica concreta de la transformación urbanística a través del tiempo. Este enfoque no solo simplifica la interpretación de los datos, sino que también facilita un seguimiento objetivo y medible de la expansión urbana.

En definitiva, los resultados actuales son muy positivos, aunque siempre caben mejoras y seguro que hay supuestos en el que el modelo no se comporta tan bien dada la gran variabilidad de imágenes que te puedes encontrar en este campo. A medida que se continúe afinando el modelo incluyendo nuevos conjuntos de datos para entrenar, es probable que se obtenga un sistema aún más robusto y preciso para monitorizar y visualizar la evolución urbanística a partir de imágenes satelitales.

12. Conclusiones

A la conclusión de este proyecto se ha evidenciado con claridad que Segformer, una arquitectura de vanguardia en el campo del machine learning, es una herramienta poderosa para mapear y visualizar la evolución del desarrollo urbanístico de una región a través del tiempo. El uso de Segformer ha resultado en hallazgos significativos, destacando su habilidad para descifrar dinámicas urbanas pasadas y proyectar tendencias futuras. Los resultados obtenidos subrayan su eficacia no solo como un marco teórico sino también como un aplicativo práctico, proporcionando una perspectiva enriquecedora sobre la transformación de las áreas urbanas. La incorporación de este framework como eje central del proyecto refuerza el compromiso con la innovación y la búsqueda de soluciones avanzadas para la comprensión del crecimiento y la planificación urbana.

Las aplicaciones de este proyecto son amplias, desde la planificación urbana y el desarrollo inmobiliario hasta estudios demográficos y medioambientales. Al proporcionar una

herramienta que ofrece una visión visual y analítica de la evolución urbanística, se abre un abanico de oportunidades para diferentes sectores y profesionales.

A nivel social, la capacidad de anticipar y visualizar el crecimiento urbanístico puede influir en las decisiones políticas y públicas, ayudando a crear ciudades más sostenibles y adaptadas a las necesidades futuras de sus habitantes.

A medida que se vaya poniendo a prueba el modelo se irán viendo fortalezas y debilidades en distintas situaciones, pero cabe destacar el comportamiento del modelo teniendo en cuenta que nunca fue entrenado con imágenes sacadas de Google Earth. Pero debido a la gran variedad de situaciones a las que debe enfrentarse sería interesante realizar colaboraciones con organizaciones especializadas en cartografía y estudios urbanos para acceder a datasets más ricos y variados. Incluso otro aspecto a considerar sería utilizar técnicas de aprendizaje semi-supervisado para mejorar la precisión del modelo, especialmente en áreas donde los datos etiquetados son escasos.

En resumen, este proyecto ha establecido los cimientos para desarrollar una herramienta abierta y accesible destinada al público interesado en explorar la evolución urbanística de su barrio o ciudad. Representa una aplicación con el potencial de transformar nuestro enfoque hacia la percepción y planificación del crecimiento urbano. El futuro de este empeño es alentador, brindando numerosas posibilidades para profundizar y diversificar la investigación en este ámbito.

13. Próximos Pasos

Los siguientes pasos que podemos seguir a la finalización de este proyecto se pueden dividir en:

1. Mejoras y optimización del actual modelo predictivo:

- **Entrenamiento continuo:** Dada la rápida evolución de las técnicas de machine learning, es crucial mantener el modelo actualizado. Se propone seguir entrenando el modelo regularmente, especialmente con nuevos datasets que se asemejen a las imágenes obtenidas de Google Earth.
- **Aumento de datos:** tal y como se ha comentado se pueden añadir nuevos conjuntos de datos que complementen los que ya se han utilizado.
- **Aplicar métodos de aprendizaje semi-supervisado** para potenciar la exactitud del modelo y que sea más fácil crear un conjunto de datos propio.
- **Desarrollar una interfaz gráfica** que permita a los usuarios cargar imágenes, visualizar predicciones y proyecciones, y obtener análisis detallados de manera intuitiva y sencilla.

2. Propuestas para continuar o expandir el proyecto:

Realizar proyecciones urbanísticas a Futuro:

- **Modelo Predictivo**: Desarrollar un modelo que pueda anticipar la expansión urbanística basado en tendencias históricas, datos demográficos, políticas urbanas y otros factores relevantes.
- **Datos Cartográficos:** Integrar información cartográfica permitirá al modelo comprender mejor la topografía y geografía de la zona. Esto será fundamental para hacer predicciones realistas sobre cómo podría evolucionar un área.
- **Identificación de Elementos Clave:** El modelo debería ser capaz de identificar y categorizar diferentes elementos, como masas de agua, carreteras, áreas verdes, zonas industriales, y otros elementos urbanos. Esto permitirá hacer proyecciones más precisas y realistas.
- **Integración con Sistemas GIS (Sistema de Información Geográfica):** Al integrarse con plataformas GIS, el proyecto podría beneficiarse de un conjunto más amplio de herramientas y datos geoespaciales, permitiendo análisis más profundos y detallados.

3. Aplicaciones Adicionales:

- **Análisis Ambiental:** Usar el modelo para monitorizar y proyectar cambios en áreas verdes, deforestación o aparición de nuevas zonas verdes en áreas urbanas.
- **Planificación Urbana:** Las autoridades y planificadores urbanos podrían utilizar esta herramienta para visualizar y planificar la expansión de las ciudades, ayudando en la toma de decisiones sobre dónde construir infraestructuras, zonas residenciales, comerciales o industriales.
- **Estudios Demográficos:** Al detectar nuevas estructuras y zonas de desarrollo, se puede inferir el crecimiento poblacional y ayudar en estudios demográficos.

4. Colaboraciones e Integraciones:

- **Establecer colaboraciones** con organizaciones o entidades interesadas en la evolución urbanística. Esto podría ayudar a obtener más datos, mejorar las predicciones y expandir las aplicaciones del proyecto.

En definitiva, las posibilidades para expandir y mejorar este proyecto son vastas y, con la correcta implementación, puede convertirse en una herramienta valiosa para una amplia variedad de aplicaciones en el ámbito de la planificación y análisis urbanístico.

14. Enlaces de interés

Todo el código utilizado para la realización de este trabajo se encuentra en el siguiente enlace. También se puede ver el GIF mostrado como ejemplo en el proyecto: https://github.com/AVR185/09MIAR_10_A_2022-23_Trabajo-Fin-de-Master

Arquitecturas preentrenadas utilizadas:

- Segformer: https://huggingface.co/nvidia/mit-b5
- DeepLabV3+: https://pytorch.org/vision/main/models/generated/torchvision.models.segmentation.deeplabv3_resnet101.html

Conjuntos de datos empleados en el proyecto:

- INRIA: https://project.inria.fr/aerialimagelabeling/
- Spacenet: https://www.kaggle.com/datasets/amerii/spacenet-7-multitemporal-urban-development

15. Bibliografía

Behera, T. K., Bakshi, S., Nappi, M., & Sa, P. K. (2023). Superpixel-based multiscale CNN approach toward multiclass object segmentation from UAV-captured aerial images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 1771-1784.

Bi, X., Hu, J., Xiao, B., Li, W., & Gao, X. (2022). IEMask R-CNN: Information-Enhanced Mask R-CNN. *IEEE Transactions on Big Data*, *9*(2), 688-700.

Chen, H., Zhang, H., Chen, K., Zhou, C., Chen, S., Zou, Z., & Shi, Z. (s. f.). *Remote Sensing Image Change Detection Towards Continuous Bitemporal Resolution Differences*.

Fatty, A., Li, A. J., & Yao, C. Y. (2023). Instance segmentation based building extraction in a dense urban area using multispectral aerial imagery data. *Multimedia Tools and Applications*, 1-16. https://doi.org/10.1007/S11042-023-15905-W/TABLES/4

Gao, Y., Li, Y., Jiang, R., Zhan, X., Lu, H., Guo, W., Yang, W., Ding, Y., & Liu, S. (2023). Enhancing Green Fraction Estimation in Rice and Wheat Crops: A Self-Supervised Deep Learning Semantic Segmentation Approach. *Plant phenomics (Washington, D.C.)*, *5*, 0064. https://doi.org/10.34133/plantphenomics.0064

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 2980-2988. https://doi.org/10.1109/ICCV.2017.322

Ibrahem, H., Salem, A., & Kang, H. S. (2022). Exploration of Semantic Label Decomposition and Dataset Size in Semantic Indoor Scenes Synthesis via Optimized Residual Generative Adversarial Networks. *Sensors*, *22*(21). https://doi.org/10.3390/s22218306

Lee, K., Lee, H., & Hwang, J. Y. (2021). Self-mutating network for domain adaptive segmentation in aerial images. *Proceedings of the IEEE/CVF international conference on computer vision*, 7068-7077.

Li, M., Rui, J., Yang, S., Liu, Z., Ren, L., Ma, L., Li, Q., Su, X., & Zuo, X. (2023a). Method of Building Detection in Optical Remote Sensing Images Based on SegFormer. *Sensors*, *23*(3). https://doi.org/10.3390/s23031258

Li, M., Rui, J., Yang, S., Liu, Z., Ren, L., Ma, L., Li, Q., Su, X., & Zuo, X. (2023b). Method of Building Detection in Optical Remote Sensing Images Based on SegFormer. *Sensors (Basel, Switzerland)*, *23*(3). https://doi.org/10.3390/S23031258

Li, X., Li, Y., Ai, J., Shu, Z., Xia, J., & Xia, Y. (2023). Semantic segmentation of UAV remote sensing images based on edge feature fusing and multi-level upsampling integrated with Deeplabv3+. *PLoS ONE*, *18*(1 January). https://doi.org/10.1371/journal.pone.0279097

Li, Z., Zhang, Z., Chen, D., Zhang, L., Zhu, L., Wang, Q., Chen, S., & Peng, X. (2022). HCRB-MSAN: Horizontally Connected Residual Blocks-Based Multiscale Attention Network for Semantic Segmentation of Buildings in HSR Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 5534-5544. https://doi.org/10.1109/JSTARS.2022.3188515

Liu, J., Wang, S., Hou, X., & Song, W. (2020). A deep residual learning serial segmentation network for extracting buildings from remote sensing imagery. *International Journal of Remote Sensing*, *41*(14), 5573-5587. https://doi.org/10.1080/01431161.2020.1734251

Liu, J., Wang, Z., & Cheng, K. (2019). An improved algorithm for semantic segmentation of remote sensing images based on deeplabv3+. *ACM International Conference Proceeding Series*, 124-128. https://doi.org/10.1145/3369985.3370027

Ma, K., Meng, X., Hao, M., Huang, G., Hu, Q., & He, P. (2023). Research on the Efficiency of Bridge Crack Detection by Coupling Deep Learning Frameworks with Convolutional Neural Networks. *Sensors*, *23*(16). https://doi.org/10.3390/s23167272

Memon, M. M., Hashmani, M. A., Junejo, A. Z., Rizvi, S. S., & Raza, K. (2022). Unified DeepLabV3+ for Semi-Dark Image Semantic Segmentation. *Sensors*, *22*(14). https://doi.org/10.3390/s22145312

Mo, L., Fan, Y., Wang, G., Yi, X., Wu, X., & Wu, P. (2022). DeepMDSCBA: An Improved Semantic Segmentation Model Based on DeepLabV3+ for Apple Images. *Foods (Basel, Switzerland)*, *11*(24). https://doi.org/10.3390/foods11243999

Nasrallah, H., Samhat, A. E., Shi, Y., Zhu, X. X., Faour, G., & Ghandour, A. J. (2022). Lebanon Solar Rooftop Potential Assessment Using Buildings Segmentation From Aerial Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 4909-4918. https://doi.org/10.1109/JSTARS.2022.3181446

Nasrallah, H., Shukor, M., & Ghandour, A. J. (2023). Sci-Net: scale-invariant model for buildings segmentation from aerial imagery. *Signal, Image and Video Processing*, 1-9.

Neumann, D., Reddy, A. S. N., & Ben-Hur, A. (2022). RODAN: a fully convolutional architecture for basecalling nanopore RNA sequencing data. *BMC Bioinformatics*, *23*(1). https://doi.org/10.1186/s12859-022-04686-y

Pan, Q., Gao, M., Wu, P., Yan, J., & Li, S. (2021). A deep-learning-based approach for wheat yellow rust disease recognition from unmanned aerial vehicle images. *Sensors*, *21*(19). https://doi.org/10.3390/s21196540

Pang, C., Wu, J., Ding, J., Song, C., & Xia, G.-S. (2023). Detecting building changes with off-nadir aerial images. *Science China Information Sciences*, *66*(4), 140306.

Parra-Mora, E., & da Silva Cruz, L. A. (2022). LOCTseg: A lightweight fully convolutional network for end-to-end optical coherence tomography segmentation. *Computers in biology and medicine*, *150*, 106174. https://doi.org/10.1016/j.compbiomed.2022.106174

Press, E. (s. f.). *SAM es el modelo de segmentación de imagen con el que Meta pretende democratizar esta tecnología*. Recuperado 5 de octubre de 2023, de https://www.europapress.es/portaltic/sector/noticia-sam-modelo-segmentacion-imagen-meta-pretende-democratizar-tecnologia-20230405171749.html

Sediqi, K. M., & Lee, H. J. (2021). A novel upsampling and context convolution for image semantic segmentation. *Sensors*, *21*(6), 1-16. https://doi.org/10.3390/s21062170

Vasavi, S., Prasad, M. B., Sai, P. J., & Rao, K. V. G. (2022). Change Detection of Urban GIS Maps Using Multi-scale U-Net-Based Attention Neural Network Architecture. *SN Computer Science*, *4*(1), 90.

Vigueras-Guillén, J. P., Sari, B., Goes, S. F., Lemij, H. G., van Rooij, J., Vermeer, K. A., & van Vliet, L. J. (2019). Fully convolutional architecture vs sliding-window CNN for corneal endothelium cell segmentation. *BMC biomedical engineering*, *1*(1), 4. https://doi.org/10.1186/s42490-019-0003-2

Wagh, A., Jain, S., Mukherjee, A., Agu, E., Pedersen, P., Strong, D., Tulu, B., Lindsay, C., & Liu, Z. (2020). Semantic Segmentation of Smartphone Wound Images: Comparative Analysis of AHRF and CNN-Based Approaches. *IEEE access : practical innovations, open solutions*, *8*, 181590-181604. https://doi.org/10.1109/access.2020.3014175

Wang, R., Liu, Y., Lu, Y., Yuan, Y., Zhang, J., Liu, P., & Yao, Y. (2019). The linkage between the perception of neighbourhood and physical activity in Guangzhou, China: Using street view imagery with deep learning techniques. *International Journal of Health Geographics*, *18*(1). https://doi.org/10.1186/s12942-019-0182-z

Wang, Y., Wang, C., Wu, H., & Chen, P. (2022). An improved Deeplabv3+ semantic segmentation algorithm with multiple loss constraints. *PLoS ONE*, *17*(1 January). https://doi.org/10.1371/journal.pone.0261582

Wang, Y. Z., Wu, W., & Birch, D. G. (2021). A Hybrid Model Composed of Two Convolutional Neural Networks (CNNs) for Automatic Retinal Layer Segmentation of OCT Images in Retinitis Pigmentosa (RP). *Translational vision science & technology*, *10*(13). https://doi.org/10.1167/TVST.10.13.9

Wang, Y.-Z., Galles, D., Klein, M., Locke, K. G., & Birch, D. G. (2020). Application of a Deep Machine Learning Model for Automatic Measurement of EZ Width in SD-OCT Images of RP. *Translational vision science & technology*, *9*(2), 15. https://doi.org/10.1167/tvst.9.2.15

Wang, Z., He, X., Li, Y., & Chuai, Q. (2022). EmbedFormer: Embedded Depth-Wise Convolution Layer for Token Mixing. *Sensors (Basel, Switzerland)*, *22*(24). https://doi.org/10.3390/s22249854

Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., Zhu, F., Shao, L., Xia, G.-S., & Bai, X. (s. f.). *iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images*.

Wu, Y., Xu, L., Wang, L., Chen, Q., Chen, Y., & Clausi, D. A. (2023). Multi-Task Edge Detection for Building Vectorization From Aerial Images. *IEEE Geoscience and Remote Sensing Letters*, *20*, 1-5.

Xia, Z., & Kim, J. (2023). Enhancing Mask Transformer with Auxiliary Convolution Layers for Semantic Segmentation. *Sensors*, *23*(2). https://doi.org/10.3390/s23020581

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (s. f.). *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*.

Yang, G., Zhang, Q., & Zhang, G. (2020). EANet: Edge-Aware Network for the Extraction of Buildings from Aerial Images. *Remote Sensing 2020, Vol. 12, Page 2161*, *12*(13), 2161. https://doi.org/10.3390/RS12132161

Yang, K., Xia, G. S., Liu, Z., Du, B., Yang, W., Pelillo, M., & Zhang, L. (2022). Asymmetric Siamese Networks for Semantic Change Detection in Aerial Images. *IEEE Transactions on Geoscience and Remote Sensing*, *60*. https://doi.org/10.1109/TGRS.2021.3113912

Ye, H., Zhou, R., Wang, J., & Huang, Z. (2022). FMAM-Net: Fusion Multi-Scale Attention Mechanism Network for Building Segmentation in Remote Sensing Images. *IEEE Access*, *10*, 134241-134251. https://doi.org/10.1109/ACCESS.2022.3231362

Zhang, W. ;, Yu, M. ;, Chen, X. ;, Zhou, F. ;, Ren, J. ;, Xu, H. ;, Xu, S., Zhang, W., Yu, M., Chen, X., Zhou, F., Ren, J., Xu, H., & Xu, S. (2022). Combining Deep Fully Convolutional Network and Graph Convolutional Neural Network for the Extraction of Buildings from Aerial Images. *Buildings 2022, Vol. 12, Page 2233*, *12*(12), 2233. https://doi.org/10.3390/BUILDINGS12122233

Alarcon, N. (2018, 12 de diciembre). AI Helps Detect Disaster Damage From Satellite Imagery. *Nvidia.com.* Recuperado el 5 de octubre de 2023, de https://developer.nvidia.com/blog/ai-helps-detect-disaster-damage-from-satellite-imagery/.