Evaluación de redes profundas convolucionales para la segmentación semántica de imágenes aéreas

Evaluation of deep convolutional networks for semantic segmentation of aerial images

Gizela Andrea Guzmán [1],
Artículo recibido 26 de septiembre de 2025

Resumen

La generación de cartografía constituye un insumo estratégico para la toma de decisiones informadas a nivel global. No obstante, los enfoques tradicionales de teledetección presentan limitaciones de automatización y escalabilidad. En este marco, el aprendizaje profundo y las redes neuronales convolucionales han emergido como una alternativa poderosa para extraer, clasificar y actualizar información geográfica. Por ende, este estudio evalúa el desempeño de una red neuronal basada en la arquitectura SegNet, para la segmentación de clases urbanas del conjunto ISPRS Vaihingen e identifica sus principales patrones de error. En la validación sobre el conjunto de datos de prueba se obtuvo una exactitud de 90.18%, con alto poder discriminativo en edificios (f1 = 0.9542) y carreteras (f1 = 0.9118); los errores más relevantes se concentraron en la confusión vegetación baja y árboles (f1 = 0.7935, f1 = 0.9013) y en la segmentación de objetos pequeños como carros (f1 = 0.8152); mientras que la clase residual desorden no fue aprendida. Los resultados obtenidos validan el buen desempeño de las redes para la automatización en la generación de cartografía y señalan líneas claras de mejora, con el fin de resolver ambigüedades en límites vegetales y reforzar la detección de objetos minoritarios.

Palabras clave: Segmentación, redes neuronales, SegNet

Código disponible en:

https://github.com/gizeandre13/PRA_Informe_1_Gizela_Guzman

Abstract

Mapping is a strategic input for informed decision-making at the global level. However, traditional remote sensing approaches have limitations in terms of automation and scalability. In this context, deep learning and convolutional neural networks have emerged as a powerful alternative for extracting, classifying, and updating geographic information. Therefore, this study evaluates the performance of a neural network based on the SegNet architecture for the segmentation of urban classes in the ISPRS Vaihingen dataset and identifies its main error patterns. Validation on the test dataset yielded an accuracy of 90.18%, with high discriminatory power for buildings (f1 = 0.9542) and roads (f1 = 0.9118); the most relevant errors were concentrated in the confusion between low vegetation and trees (f1 = 0.7935, f1 = 0.9013) and in the segmentation of small objects such as cars (f1 = 0.8152), while the residual disorder class was not learned. These results validate the good performance of networks for automation in cartography generation and point to clear areas for improvement, with the aim of resolving ambiguities in vegetation boundaries and strengthening the detection of minority objects.

Keywords: Segmentation, neural networks, SegNet

Percepción Remota Avanzada- Maestría en Geomática - Universidad Nacional de Colombia

1 Introducción

El conocimiento preciso y actualizado del territorio resulta indispensable para la planificación territorial y la toma de decisiones informadas, pues permite comprender fenómenos cruciales como la dinámica del crecimiento urbano, la identificación de áreas de vulnerabilidad ambiental y el impacto de factores asociados al cambio climático. Sin embargo, a pesar de esta relevancia, la generación de cartografía de alta resolución, insumo importante para la gestión territorial, ha dependido históricamente de enfoques tradicionales de teledetección. Estos métodos presentan limitaciones significativas en términos de automatización, escalabilidad y eficiencia en la actualización, lo que dificulta mantener un registro dinámico y oportuno de la realidad territorial.

Dentro de este panorama, la automatización de la cartografía vectorial se ha enfrentado a retos importantes, lo que ha motivado el desarrollo de investigaciones orientadas a reducir la dependencia de la intervención manual y a optimizar los procesos de producción cartográfica (Feng et al., 2019).

A inicios de los años 2000, el avance de los Sistemas de Información Geográfica (SIG) y el procesamiento digital de imágenes impulsó enfoques como el Análisis Basado en Objetos Geográficos (GEOBIA), el cual introdujo técnicas de detección de bordes, extracción de características y clasificación, que se han utilizado en el análisis de imágenes de teledetección durante décadas (Blaschke, 2010). Posteriormente con el surgimiento de aprendizaje automático se desarrollaron métodos que buscaban imitar las decisiones de los cartógrafos, aprendiendo reglas o secuencias de operaciones a partir de ejemplos; sin embargo, estos se mantuvieron como pruebas conceptuales (Feng et al., 2019).

Con el desarrollo del aprendizaje profundo y el acceso a grandes volúmenes de datos junto con una mayor capacidad computacional, las redes neuronales convolucionales (CNN) se han consolidado como herramientas fundamentales en el ámbito de la cartografía digital. En particular, su aplicación en tareas de segmentación permite dividir imágenes en regiones significativas, facilitando la identificación de elementos geográficos como cuerpos de agua, edificaciones, bosques, entre otros (Ekundayo & Ezugwu, 2025).

Esta capacidad resulta útil para la generación automatizada de cartografía vectorial, ya que mejora tanto la exactitud como la eficiencia en el análisis espacial a gran escala, contribuyendo a una representación más detallada y actualizada del territorio(Feng et al., 2019, Forero Zapata, 2023).

A medida que ha aumentado la capacidad de cómputo y se ha profundizado en el estudio de las redes convolucionales, ha sido posible diseñar arquitecturas más complejas y profundas, lo que ha facilitado el desarrollo de tareas avanzadas como la segmentación semántica. Entre estas arquitecturas destacan las redes totalmente convolucionales (FCN, por sus siglas en inglés), que superan los enfoques tradicionales basados en capas totalmente conectadas, ya que permiten procesar imágenes de cualquier tamaño y generar segmentaciones exactas y detalladas. La principal innovación de las FCN consiste en modificar la estructura de las CNN de clasificación tradicional, para que la salida no sea un vector de probabilidad, sino un mapa de probabilidad, obteniendo un mapa de calor para cada clase (Audebert et al., 2016).

A partir de los avances en redes neuronales convolucionales, se han desarrollado numerosas investigaciones orientadas al diseño de arquitecturas de segmentación semántica, capaces de extraer y clasificar con exactitud elementos geográficos a partir de imágenes aéreas. Estas arquitecturas han demostrado ser altamente eficaces en distintos conjuntos de datos de referencia. Por ejemplo, (Audebert et al., 2016) exploró el uso de redes completamente convolucionales profundas (DFCN) para el etiquetado denso de escenas, demostrando que arquitecturas codificador-decodificador como SegNet, originalmente diseñadas para imágenes convencionales y entrenadas con pesos de ImageNet pueden adaptarse con éxito a datos de teledetección.

En este contexto, el presente trabajo realiza una implementación en PyTorch de una red totalmente convolucional para segmentación semántica de imágenes aéreas, para ello se replica y adapta la arquitectura segNet, para la segmentación de imágenes aéreas del conjunto de datos ISPRS Vaihingen.

Se implementa un flujo que abarca el entrenamiento y la inferencia a escala de mosaico, utilizando una ventana deslizante con promedio de puntajes en solapes. El desempeño se evalúa mediante métricas estándar (matriz de confusión, exactitud global, F1 por clase y coeficiente Kappa). Con ello se busca no solo corroborar la eficacia de estas redes en escenarios urbanos, sino también caracterizar sus limitaciones típicas (delimitación de bordes, confusión entre clases y objetos pequeños) y, aportar pautas prácticas orientadas a su mejora.

2 Materiales y métodos

2.1 Conjunto de datos y área de estudio

Para el desarrollo del presente estudio se empleó el conjunto de datos ISPRS Vaihingen dataset, capturado sobre la ciudad de Vaihingen an der Enz, Alemania. El cual corresponde a un subconjunto de datos adquiridos durante una prueba de cámaras aéreas digitales organizada por la Asociación Alemana de Fotogrametría y Teledetección (DGPF).

El conjunto de datos está compuesto por tres áreas de prueba principales, para las cuales se dispone de datos de referencia correspondientes a distintas clases de objetos, además de un sitio adicional destinado a la extracción de vías urbanas.

- ✓ Área 1: Centro urbano. Se caracteriza por un desarrollo denso de edificios históricos con formas arquitectónicas complejas, acompañado de vegetación arbórea dispersa.
- ✓ Área 2: Rascacielos. Corresponde a una zona dominada por edificios residenciales de gran altura, rodeados de una alta densidad de árboles.
- ✓ Área 3: Zona residencial. Área de uso netamente habitacional, conformada por viviendas unifamiliares de pequeña escala.
- ✓ Carreteras: Conjunto que abarca todos los sitios anteriores y está destinado a evaluar técnicas de extracción de red vial urbana.

En la **Figura 1** se puede evidenciar la localización del conjunto de datos y de las áreas de prueba.

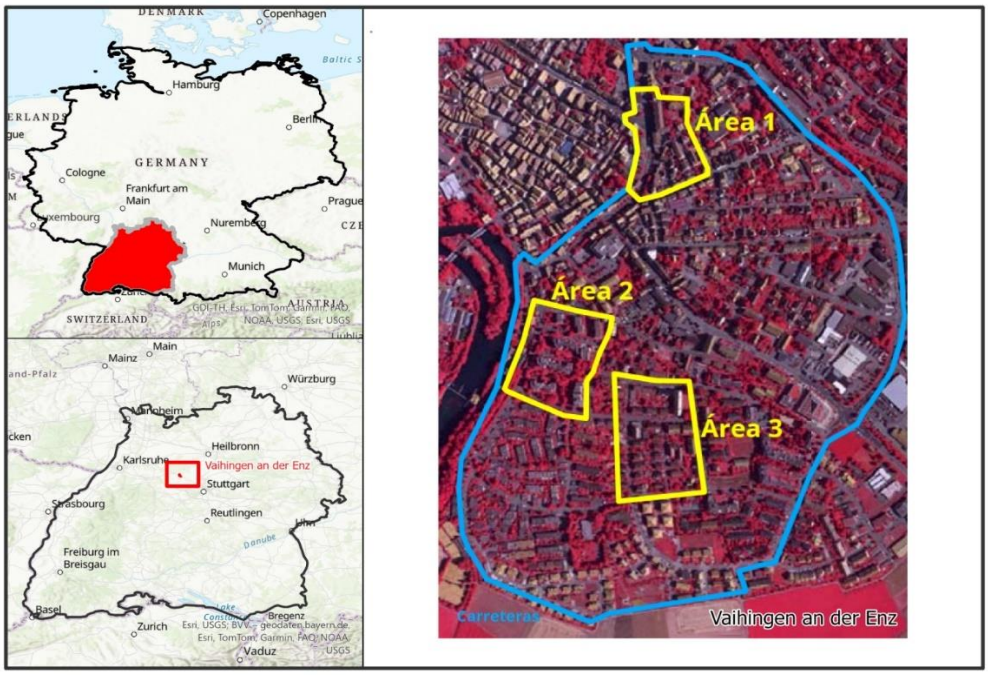

***Figura 1**. Localización de la ciudad de Vaihingen an der Enz y áreas de prueba del conjunto de datos*
***Fuente:** Propia - ISPRS, 2013*

Los datos presentan una resolución espacial de 9 cm/píxel, organizados en mosaicos de 2493 × 2063 píxeles. Donde se incluyen un total de 33 imágenes multiespectrales en bandas infrarrojo, rojo y verde (IRRG), de las cuales 16 cuentan con datos de referencia, y contienen etiquetas asociadas a seis clases semánticas (carreteras, edificios, vegetación baja, arboles, carros y desorden) (ISPRS, 2013).

2.2 Cargue y preprocesamiento de datos

El conjunto de datos de Vaihingen tiene un tamaño de 2493 × 2063 píxeles, el cual es demasiado grande para ser procesado directamente por la mayoría de las redes convolucionales (CNN), que suelen estar diseñadas para trabajar con imágenes de 256 × 256 píxeles. Además, las limitaciones de memoria de la GPU hacen inviable cargar imágenes tan grandes de una sola vez.

Por tanto, se utiliza un enfoque de ventana deslizante que divide cada mosaico en fragmentos más pequeños llamados parches (en este caso de 128 × 128 píxeles). La ventana se desplaza por toda la imagen y va extrayendo parches que luego son usados en el entrenamiento. De esta forma, es posible procesar imágenes de cualquier tamaño de manera eficiente y lineal.

En el código, esta lógica está implementada en la clase ISPRS_dataset, que se encarga de:

- ✓ Leer los archivos de datos y etiquetas.
- ✓ Extraer posiciones aleatorias de parches en cada iteración.
- ✓ Normalizar los valores de las imágenes al rango [0, 1].
- ✓ Convertir las etiquetas de colores a valores numéricos por clase.

Adicionalmente, se aplica un proceso de aumento de datos, que incluye operaciones simples como volteos horizontales o verticales de los parches. Estas transformaciones generan más variabilidad en los ejemplos de entrenamiento, ayudando a que la red sea más robusta y no dependa de la orientación original de los objetos.

2.3 Definición del modelo

Para este estudio se seleccionó la arquitectura SegNet, dado que ofrece un buen equilibrio entre exactitud y costo computacional para la segmentación semántica de imágenes. Su diseño se basa en una estructura simétrica de codificador–decodificador, la primera parte (codificador) comprime la información de la imagen, mientras que la segunda parte (decodificador) la reconstruye para producir el mapa segmentado. (Ver **Figura 2**)

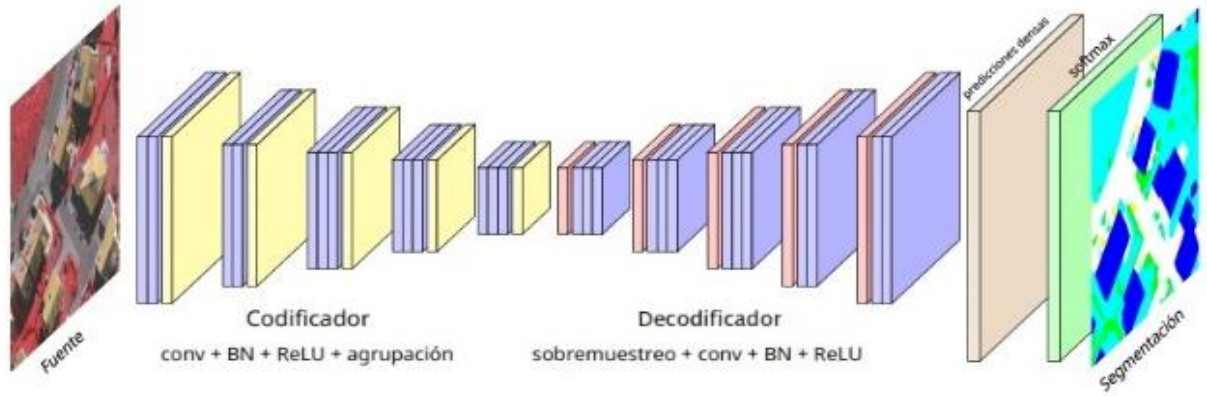

***Figura 2**. Estructura arquitectura SegNet*
Fuente: *(Audebert et al., 2017)*

SegNet está basado en la red VGG-16(Visual Geometry Group), una de las arquitecturas clásicas de visión por computador. El codificador está compuesto por cinco bloques de capas convolucionales de tamaño 3×3, con un relleno (padding) de 1 para conservar la resolución espacial (Audebert et al., 2017).

Cada convolución es seguida por dos operaciones clave:

- ✓ Batch Normalization (BN): normaliza las salidas de cada capa para mantener valores estables y acelerar el entrenamiento, evitando que la red deje de aprender.
- ✓ ReLU (Rectified Linear Unit): introduce no linealidad transformando los valores negativos en ceros y manteniendo los positivos, lo que facilita el aprendizaje de patrones complejos sin aumentar demasiado el costo computacional.

Cada bloque termina con una capa de max-pooling (agrupación máxima) de tamaño 2×2, que reduce a la mitad la resolución de los mapas de características, resumiendo la información de cada región y conservando solo los valores más representativos. De esta manera, al final del codificador, la imagen queda transformada en un conjunto de mapas de características mucho más pequeños, pero que contienen la información esencial.

El decodificador realiza el camino inverso, reconstruye progresivamente la imagen hasta alcanzar su tamaño original. En lugar de pooling, utiliza una técnica llamada unpooling o desagrupamiento, que reubica los valores en las posiciones exactas donde estaban los píxeles más importantes registrados en el codificador. Esto permite recuperar con mayor precisión las formas y bordes de los objetos.

En la **Figura 3** se puede ver el funcionamiento de las operaciones de maxpooling y unpooling. El proceso inicia con un mapa de características de 4×4 que es reducido mediante maxpooling a un mapa de activaciones de 2×2 al seleccionar el valor máximo de cada subregión; simultáneamente, se registran los índices o posiciones originales de estos valores máximos. Posteriormente, el proceso de unpooling utiliza esos índices guardados para desagrupar y restaurar las activaciones a sus ubicaciones originales en un mapa de 4×4, llenando el resto de las celdas con ceros, preservando la correspondencia espacial de los máximos, y garantizando que no se introduzca ruido en las posiciones no activadas.

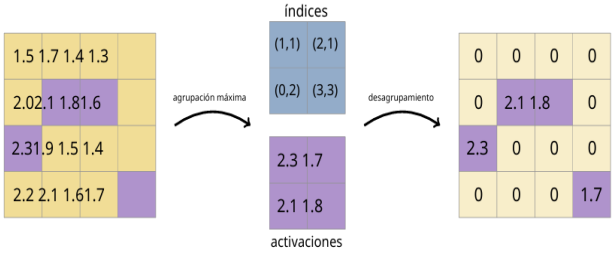

***Figura 3**. Funcionamiento operaciones maxpooling y unpooling (mapa de características 4*4)*
***Fuente**: (Audebert et al., 2017)*

Posteriormente, nuevas capas convolucionales densifican estos mapas hasta obtener un resultado final en forma de segmentación semántica con la misma resolución de la imagen de entrada; generando un mapa de salida de clases por pixel, representado en forma de *logits* (puntajes por clase).

En la etapa final, a partir de la salida de la red en forma de logits, se genera un mapa de probabilidades aplicando la función softmax a lo largo del conjunto de clases. Así, cada píxel obtiene una distribución normalizada (valores entre 0 y 1, que suman 1), indicando la probabilidad de que cada clase sea la correcta en ese punto específico, lo que permite obtener la clasificación final de cada píxel en imagen.

2.4 Entrenamiento del modelo

Con el fin de mejorar el desempeño del modelo y aprovechar conocimiento previo, el codificador de SegNet se inicializó con pesos preentrenados de la red VGG-16, disponibles en PyTorch a partir del entrenamiento con el conjunto de datos ImageNet (imágenes de alta resolución que abarcan miles de categorías de objetos). Estos pesos fueron asignados capa por capa a la parte del codificador de SegNet, lo que permite comenzar con filtros ya optimizados para detectar bordes, texturas y patrones básicos.

De esta forma, se acelera el entrenamiento y se incrementa la capacidad de generalización del modelo sobre el conjunto ISPRS Vaihingen, mientras que las capas del decodificador se inicializan aleatoriamente y se ajustan durante el entrenamiento.

Antes de entrenar, se instancia SegNet con 3 canales de entrada (IRRG) y 6 clases de salida (catálogo Vaihingen), y se configura la ejecución en GPU. Posteriormente, se establece una partición fija de tiles en datos de entrenamiento y datos de testeo.

Consecutivamente se configuró el bucle de entrenamiento, el cual se realizó por épocas, en cada iteración se procesaron parches de 128×128 píxeles normalizados, extraídos aleatoriamente, lo cual evita que la red se acostumbre a ver siempre las mismas posiciones y mejora su capacidad de generalización. La optimización se realizó minimizando una pérdida de segmentación por píxel mediante retropropagación y el optimizador Descenso Estocástico del Gradiente (SGD), con una tasa de aprendizaje inicial de 0,01, un momento de 0,9 y una penalización de peso de 5×10^{-4}.

Además, se aplicó un plan escalonado de reducción de la tasa de aprendizaje en las épocas 25, 35 y 45 (factor 0,1) para estabilizar la fase final del ajuste. Se activó precisión mixta para reducir el uso de memoria y acelerar el cálculo. La pérdida empleada fue entropía cruzada por píxel, con pesos por clase opcionales cuando existió desbalance, evitando que las clases minoritarias quedaran subentrenadas.

De forma paralela se registró la pérdida media móvil y la generación periódica de visualizaciones de control de calidad, comparando la imagen de entrada con el mapa de referencia (verdad del terreno) y la imagen de predicción del modelo, lo que permitió una supervisión efectiva del desempeño cualitativo.

El modelo se validó al final de cada época mediante una ventana deslizante que recorre cada imagen con el mismo tamaño de parche empleado, y, en las zonas solapadas, se promedian los puntajes para suavizar las transiciones en los bordes entre parches y evitar que aparezcan líneas o discontinuidades en el resultado final. Para cuantificar el desempeño, se reportaron métricas clave como la matriz de confusión, la precisión global, la puntuación F1 por clase y el coeficiente Kappa.

2.5 Evaluación del modelo

La evaluación final se realiza cargando el modelo entrenado y activando el modo de inferencia (eval()), para deshabilitar las operaciones exclusivas del entrenamiento, sobre el conjunto de testeo, se aplica una ventana deslizante con tamaño de parche fijo y un desplazamiento(stride)definido. Para cada posición de la ventana, se extraen los parches de la imagen y se procesan para obtener puntajes por clase (logits). En las zonas donde los parches se solapan, los logits se acumulan y promedian antes de seleccionar la clase final, suavizando las transiciones y asegurando la coherencia en los bordes de la predicción final.

Finalmente, la clase definitiva para cada píxel se selecciona mediante la operación argmax, que devuelve la clase con el puntaje promedio más alto.

El procedimiento de evaluación recorre todas las teselas del conjunto de testeo, generando la máscara de predicción para cada una. Paralelamente, se preparan las etiquetas de referencia y se evalúa respecto a ellas para obtener métricas más estables. Las salidas del modelo y las referencian se almacenan en listas y luego se concatenan, para calcular métricas globales, como la matriz de confusión, la exactitud global, F1 por clase, y el coeficiente Kappa.

3 Resultados

3.1 Entrenamiento del modelo

En la **Figura 4** se presentan los resultados del entrenamiento del modelo en dos momentos (época 1 y 50), en la época 1 (a), la curva de pérdida inicia en valores altos (aproximadamente 1.8), con una exactitud de 51.58 % y muestra una tendencia descendente a medida que avanzan las iteraciones. En esta etapa, la predicción aún presenta discrepancias evidentes con respecto a los datos etiquetados (ground truth), especialmente en los bordes entre clases.

En contraste, en la época 50 (b), la pérdida ha disminuido considerablemente (cercana a 0.1) y la exactitud reportada alcanza valores cercanos al 96 %. La comparación visual (RGB,

referencia y predicción) muestra que las predicciones en la época 50 se ajustan mejor a las clases de referencia, con una segmentación más coherente y definida. Esto se traduce en una notable mejoría en la definición de bordes y la separabilidad de clases, resultando en mapas de salida más limpios y fieles a la realidad.

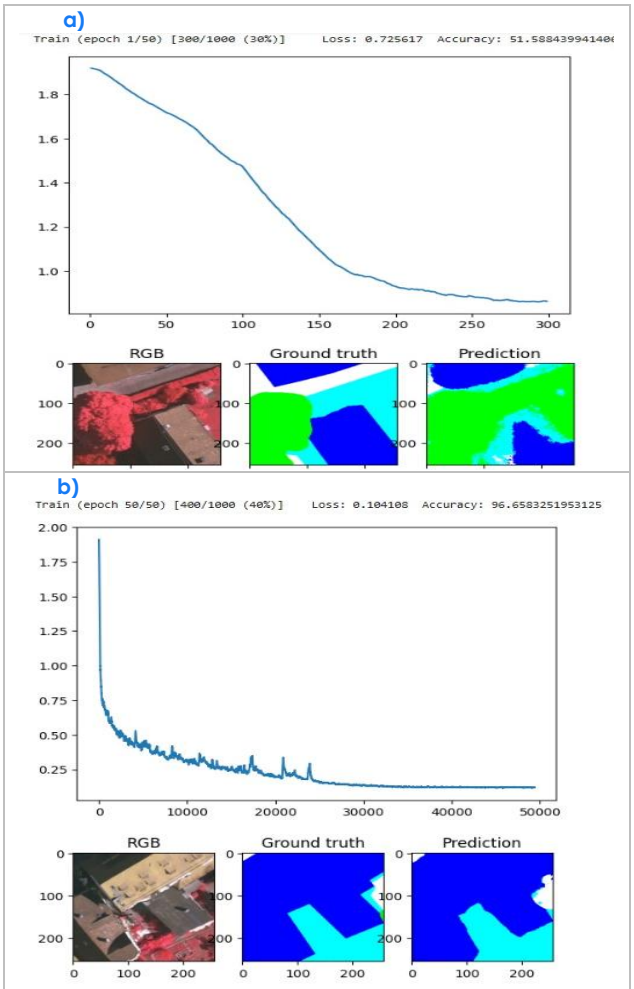

***Figura 4**. Comparación de la Pérdida y la Predicción Visual entre la Época Inicial y Final.*
a) Época inicial (1)
b) Época final (50)

Fuente: Propia

3.2 Testeo del modelo

Una vez completado el entrenamiento, se procedió a validar el modelo sobre el conjunto de datos de testeo, esta validación se centró en la comparación directa de las predicciones obtenidas frente a las etiquetas de referencia. Para cuantificar el desempeño, se reportaron las métricas estándar de evaluación, cuyos resultados detallados serán analizados en capítulos posteriores.

A continuación, se ilustra visualmente el resultado para un mosaico de prueba; a la izquierda se muestra la imagen IRRG original, en el centro la predicción del modelo y a la derecha la referencia (verdad de terreno).

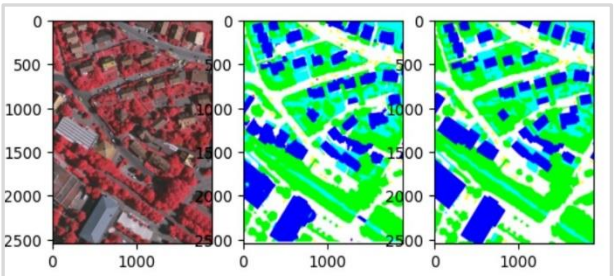

***Figura 5**. Comparación visual: predicción vs. Referencia*
***Fuente**: Propia*

En la representación temática de la **Figura 5**, los edificios aparecen en azul, las carreteras en blanco, los árboles en verde, la vegetación baja en cian, los autos en amarillo y la clase desorden en rojo. Se puede apreciar cómo el modelo reproduce la distribución espacial de las clases presentes en la escena, representando una segmentación coherente de la estructura urbana, edificios alineados, carreteras continuas, y zonas arbóreas bien localizadas. Este resultado visual valida la capacidad del modelo para generar mapas de salida limpios y espacialmente consistentes.

3.3 Matriz de Confusión

Una matriz de confusión es una herramienta que permite visualizar el desempeño del modelo creado, cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real, resume por clase, cuántos píxeles fueron correctamente clasificados (diagonal) y cuántos se confundieron con otras clases (fuera de la diagonal) (Google Developers, 2025b).

En la tabla 1 y 2, se reporta la matriz de confusión por píxel usando el orden de clases del dataset ISPRS:(carreteras, edificios, vegetación baja, arboles, carros y desorden), las filas corresponden a la verdad terreno (GT)

y las columnas a la predicción del modelo (Pred). Para entender la matriz de confusión, es fundamental conocer sus cuatro componentes básicos:

- ✓ TP (True Positive): Son los valores que el modelo clasifica como positivos y que realmente son positivos.
- ✓ TN (True Negative): Son valores que el modelo clasifica como negativos y que realmente son negativos.
- ✓ FP (False Positive): Falsos positivos, valores que el modelo clasifica como positivos cuando realmente son negativos.
- ✓ FN (False Negative): Falsos negativos, valores que el modelo clasifica como negativos cuando realmente son positivos.

Tabla 1
Matriz de confusión datos de entrenamiento

	Carreteras (Pred)	Edificios (Pred)	Veg. baja (Pred)	Árboles (Pred)	Carros (Pred)	Desorden (Pred)
Carreteras (GT)	4.351.485	188.371	173.675	62.772	24.203	3.859
Edificios (GT)	220.449	5.116.824	63.571	14.947	2.208	4.948
Veg. Baja (GT)	195.800	67.281	2.228.467	405.111	1.014	5.037
Árboles (GT)	52.944	10.130	381.390	3.948.601	207	0
Autos (GT)	22.151	6.913	1.043	638	117.314	618
Desorden (GT)	331	323	5.561	0	1	0

Fuente: *Propia*

Tabla 2
Matriz de confusión datos de testeo

	Carreteras (Pred)	Edificios (Pred)	Veg. baja (Pred)	Árboles (Pred)	Carros (Pred)	Desorden (Pred)
Carreteras (GT)	4.396.566	169.505	156.553	62.765	17.678	1.298
Edificios (GT)	191.104	5.169.751	46.792	12.632	1.455	1.213
Veg. Baja (GT)	179.473	58.733	2.284.438	376.540	550	2.976
Árboles (GT)	46.622	9.306	361.609	3.975.106	113	4
Autos (GT)	24.928	5.942	629	737	115.920	521
Desorden (GT)	819	0	5.397	0	0	0

Fuente: *Propia*

3.4 Exactitud del modelo

Accuracy o la exactitud global es la proporción de píxeles correctamente clasificados ya sean positivos o negativos, sobre el total evaluado. (Google Developers, 2025a).

$$Accuaracy = \frac{Clasificaciones\ correctas}{Total\ clasificaciones} = \frac{TP + TN}{TP + TN + FP + FN}$$

En términos prácticos, corresponde a la suma de la diagonal de la matriz de confusión (aciertos por clase) dividida entre todos los píxeles. Es una métrica sencilla y estable para resumir el rendimiento global del sistema de segmentación a nivel de píxel.

El modelo obtuvo una exactitud de **89.16%** durante la evaluación de entrenamiento y de **90.18%** en el testeo final con stride=32 y promedio de logits en zonas de solape.

3.5 F1 Score

Utilizando los componentes de la matriz de confusión, se pueden definir diferentes métricas, por ejemplo, el F1-score por clase es la media armónica entre precision y recall, el cual resume el equilibrio entre falsos positivos y falsos negativos (Google Developers, 2025a).

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

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

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

En la tabla 3 se resume el valor obtenido de F1 por clase, comparando entrenamiento y testeo. Se observan valores elevados en edificios, carreteras y árboles, por parte de la vegetación baja presenta un valor de F1 intermedio y la clase carros presenta un valor más bajo, coherente con su menor tamaño y frecuencia. La clase desorden es residual y no aporta a la métrica.

Tabla 3
Resultados F1 Score por clase

Clase	F1 Entrenamiento	F1 Testeo
Carreteras	0.9021	0.9118
Edificios	0.9464	0.9542
Vegetación baja	0.7743	0.7935
Árboles	0.8948	0.9013
Carros	0.7991	0.8152
Desorden	0.0000	0.0000

Fuente: *Propia*

En términos generales, cuando precisión y recall presentan valores similares, la métrica F1 tiende a aproximarse a ese nivel de desempeño. En cambio, cuando existe una gran diferencia entre ambas, el F1 se alinea más con la métrica que tenga el valor más bajo, reflejando el peor desempeño de las dos.

3.6 Kappa de cohen (K)

El coeficiente Kappa de Cohen (κ) evalúa el acuerdo entre las predicciones y la verdad de terreno corrigiendo el acuerdo esperado por azar, según la prevalencia de cada clase (Mohanan, 2024). A diferencia de la exactitud global, kappa descuenta la parte de coincidencias que podrían ocurrir solo por la distribución de clases (desbalance), la cual se calcula como:

$$K = \frac{\mathrm{Po} - \mathrm{Pe}}{1 - \mathrm{Pe}}$$

Donde:

Po: acuerdo observado

Pe: acuerdo esperado por azar (según las distribuciones de clases).

Kappa indica que tan bien clasifica el modelo más allá de que haya muchas muestras de una clase que eleven la exactitud. Por eso se suele reportar en conjunto con la exactitud y con métricas más específicas como el F1 por clase. Valores de κappa cercanos a 0 reflejan que el acuerdo del modelo con la referencia no es mejor que el azar, mientras que valores cercanos a 1 indican un modelo con un alto nivel de concordancia real. En la práctica, se considera que un κappa superior a 0.80 refleja un acuerdo sustancial, entre 0.60–0.80 un acuerdo moderado, y por debajo de 0.40 un nivel pobre de clasificación.

El modelo entrenado obtuvo un coeficiente kappa de **0.854** durante la evaluación en entrenamiento y **0.868** en el testeo final.

4 Discusión

4.1 Entrenamiento del modelo

Durante el entrenamiento (Ver **Figura 4**) se observó una disminución sostenida de la pérdida y un aumento de la exactitud por lote, evidenciando una tendencia estable del modelo. Esta tendencia indica que la red aprende patrones espaciales y espectrales relevantes y que el esquema de entrenamiento es adecuado.

El contraste entre la época 1 y la época 50 permite evidencia como el modelo en las primeras iteraciones, apenas comienza a aprender patrones, lo que se refleja en errores de clasificación y fronteras difusas en las predicciones. Sin embargo, con el avance del entrenamiento, la pérdida se reduce de manera significativa y la exactitud se incrementa, confirmando un mejor ajuste con los datos de referencia. Visualmente, las predicciones en la época 50 muestran una mayor fidelidad en la delimitación de objetos y clases, lo que indica que el modelo logra captar relaciones espaciales y espectrales más robustas.

4.2 Testeo del modelo

De acuerdo con la **Figura 5** el modelo demuestra un buen desempeño al reproducir correctamente la geometría urbana, confirmando la captura de patrones relevantes desde las imágenes de entrada. Este desempeño se evidencia en la consistencia de los contornos de edificios y la continuidad de las carreteras, incluso con intersecciones y sombras, lo cual valida el aporte del promediado en solapes para suavizar bordes.

Sin embargo, en algunas zonas de edificios se evidencia infra-segmentación, el modelo fusiona objetos distintos en uno solo u omite detalles finos, también se evidencia que persisten confusiones puntuales entre vegetación baja y árboles, sobre todo en límites irregulares; y además se observa omisiones de carros, que son absorbidos por la clase carreteras, debido a su pequeña escala.

En conjunto, se ilustra la capacidad del modelo para generalizar diferentes elementos geográficos en contextos urbanos, aunque también se evidencia la necesidad de ajustes en el manejo de clases minoritarias y en la delimitación de objetos pequeños.

4.3 Matriz de Confusión

De acuerdo a la **Tabla 1** y **Tabla 2**, las matrices de confusión evidencian una diagonal dominante (verdaderos positivos) en edificios y carreteras, con confusiones residuales entre ambas clases principalmente en los bordes de las clases. La mayor ambigüedad se observa entre vegetación baja y árboles, con confusiones bilaterales de magnitud similar en entrenamiento y testeo, lo que sugiere un patrón por la similitud espectral y estructural de ambas clases. La clase de carros continúa siendo la más difícil de segmentar, ya que gran parte de esta clase se confunde con carreteras, esto puede deberse a su pequeña escala y desbalance en la clase.

Y por último se evidencia que la clase desorden, no fue predicha como clase correcta (diagonal en 0) y se reparte en su mayoría como vegetación baja, esto debido a su poca presencia y definición ambigua.

4.4 Exactitud del modelo

La exactitud final del modelo fue de 89.16%, lo que indica que casi 9 de cada 10 píxeles fueron etiquetados correctamente. Esta cifra refleja el buen desempeño en clases dominantes como edificios y carreteras. Pero cabe resaltar que está métrica se ve muy influenciada por el desbalance de clases, el peso de clases extensas puede elevar la exactitud, mientras que puede ocultar las debilidades de clases minoritarias como la vegetación baja y carros.

Por ello, es necesario completar con otras métricas como con F1 por clase que aporta una lectura más fina del comportamiento por clase y con Kappa que corrige el azar.

4.5 F1 Score

Los resultados indican un modelo robusto en clases dominantes y de morfología continua como los edificios, carreteras y árboles, manteniendo el patrón con la matriz de confusión; lo que sugiere que el modelo tiene buena discriminación de objetos grandes y bien definidos.

Por otro lado, la clase de vegetación baja obtiene un F1 moderado, coherente con su mayor variabilidad espectral y geométrica, que tiende a presentar confusiones con árboles o áreas mixtas. En cuanto a la clase de carros presenta el F1 más bajo entre las clases efectivas, valor típico de objetos pequeños, con oclusiones, sombras, o desbalance de muestras. Y la clase desorden permanece en 0.00, indicando que el modelo no aprendió esta clase, lo cual puede deberse a pocas muestras, alta heterogeneidad o ruido en las etiquetas.

En conjunto, las métricas por clase complementan la lectura de la exactitud global, destacando dónde el sistema es fuerte y dónde requiere ajustes específicos

La cercanía entre el F1 de entrenamiento y el de testeo en todas las clases (con ligeras mejoras en testeo) sugiere buena generalización y ausencia de sobreajuste, posiblemente favorecida por el promedio en zonas solapadas.

En conjunto, el modelo es confiable para clases dominantes, pero requiere acciones dirigidas como re-etiquetado, limpieza para la clase desorden, balanceo de clases (pesos inversos o muestreo estratificado), aumento de resolución contextual o funciones de pérdidas más robustas (focal o ponderada), y así poder mejorar en clases como carros y vegetación baja.

4.6 Kappa de Cohen (K)

De acuerdo a los resultados obtenidos del modelo durante la evaluación del entrenamiento y del testeo final, se obtuvieron valores de kappa superiores a 0.80, los cuales se encuentran dentro de la escala habitual correspondiente a un acuerdo sustancial, más allá del azar, lo que respalda la calidad de la segmentación pese al predominio de clases grandes.

Este valor respalda que el desempeño no solo es alto en términos de exactitud (89.16%), sino que permanece sólido al corregir por azar y desbalance. La ligera brecha entre exactitud y kappa refleja que el conjunto contiene clases dominantes (edificios, carretera) que empujan la exactitud hacia arriba, mientras que persisten confusiones en clases minoritarias (vegetación baja vs. árboles; y carros).

5 Conclusiones

i. Las redes neuronales profundas convoluciones demuestran una alta capacidad para la segmentación semántica urbana, logrando una exactitud de 90.18 % y un alto acuerdo más allá del azar (κ=0.868), lo que respalda la calidad global de la segmentación pese al desbalance de clases.

ii. La comparación visual entre la predicción del modelo y los datos de referencia confirma un alto nivel de correspondencia en la clasificación, lo que valida que el modelo ha capturado con éxito los patrones espectrales y espaciales relevantes. No obstante, se aprecian diferencias consistentes en los bordes y zonas de transición, donde la segmentación tiende a simplificar las formas o a confundir elementos cercanos, indicando una limitación en la precisión espacial a nivel de píxel.

iii. Se evidencia una segmentación consistente, especialmente en clases dominantes como edificaciones carreteras, y árboles, lo que se refleja en los valores de F1 altos y en la diagonal dominante de la matriz de confusión, gracias a su continuidad espacial y señal espectral distintiva.

iv. Se presentan limitaciones en la separabilidad de clases similares, como vegetación baja y árboles, atribuible a la similitud espectral y estructural de ambas clases en los datos IRRG (Infrarrojo, Rojo, Verde). Esta confusión bilateral se mantiene tanto con los datos de entrenamiento como con los de testeo.

v. La clase carros resultó ser la más compleja de segmentar, registrando el F1 más bajo entre las clases funcionales. Su pequeña escala dificulta su detección, por lo que tienden a ser absorbidos por la clase carreteras, y además la baja prevalencia de muestras generan un alto desbalance que favorece la omisión y la confusión en la clasificación.

vi. El modelo no logró aprender la clase desorden (F1=0.00), lo que subraya la debilidad del sistema para clasificar categorías con poca representación, alta heterogeneidad o definición ambigua en el etiquetado de referencia.

6 Recomendaciones

i. Integrar información altimétrica y espectral para reducir la confusión entre vegetación baja y árboles, y así mejorar la delimitación de ambas clases, se recomienda incorporar información como Modelo digital de Superficie (MDS), Modelo Digital de Superficie normalizado(nDSM) o el Índice de Vegetación de Diferencia Normalizada (NDVI). Esto puede implementarse mediante enfoques de fusión temprana o fusión tardía.

ii. Aplicar estrategias dirigidas para clases minoritarias, como rebalanceo de clases, utilizando pesos por clase (ponderación inversa a la frecuencia) o muestreo estratificado para aumentar la relevancia de los píxeles de carros durante el entrenamiento; o la incorporación de componentes multi-escala en la arquitectura para captar mejor los patrones de objetos de pequeña extensión.

iii. Considerar la implementación de funciones de pérdida más robustas (ej. Focal Loss o Entropía Cruzada ponderada) para manejar eficazmente el desbalance inherente al conjunto de datos urbanos.

iv. Dado que la clase desorden no fue aprendida, se recomienda una revisión exhaustiva del conjunto de datos para esta clase, lo que puede incluir el re-etiquetado para asegurar una definición clara o la limpieza y eliminación de la clase si se confirma que actúa como ruido en el entrenamiento.

v. Se recomienda implementar un post-procesamiento ligero para refinar la segmentación, aplicando técnicas para alinear las etiquetas con los contornos reales, y suavizar las

transiciones; de esta manera se podrían corregir las inconsistencias por bordes irregulares.

7 Agradecimientos

Agradezco a los autores Nicolas Audebert, Bertrand Le Saux y Sébastien Lefèvre por el desarrollo de la línea base presentada en "*Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks*", la cual sirvió como fundamento metodológico para esta investigación. Y extiendo mi reconocimiento a la iniciativa en github "DeepNetsForEO", por disponer del código abierto y cuadernos reproducibles que facilitaron la implementación del modelo SegNet y la experimentación de segmentación profunda aplicada a imágenes áreas en entornos urbanos.

De igual manera, agradezco al ISPRS Vaihingen dataset, que proporcionó ortofotos IRRG de muy alta resolución (9 cm GSD) y anotaciones de referencia, esenciales para el entrenamiento y la validación del modelo.

8 Referencias

[1] Audebert, N., Saux, B. Le, & Lefèvre, S. (2016). Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks. https://doi.org/10.48550/arXiv.1609.06846

[2] Audebert, N., Le Saux, B., & Lefèvre, S. (2017). Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks. https://doi.org/10.48550/arXiv.1711.08681

[3] Blaschke, T. (2010). Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 65(1), 2–16. https://doi.org/10.1016/J.ISPRSJPRS.2009.06.004

[4] Ekundayo, O. S., & Ezugwu, A. E. (2025). Deep learning: Historical overview from inception to actualization, models, applications and future trends. Applied Soft Computing, 181, 113378. https://doi.org/10.1016/J.ASOC.2025.113378

[5] Feng, Y., Thiemann, F., & Sester, M. (2019). Learning cartographic building generalization with deep convolutional neural networks. ISPRS International Journal of Geo-Information, 8(6). https://doi.org/10.3390/ijgi8060258

[6] Forero Zapata, S. (2023). Evaluación de Redes Convolucionales para la Segmentación de Objetos Geográficos: Un Insumo para la Cartografía Básica a Escala 1:2000 basado en el Catálogo del IGAC.

[7] Google Developers. (2025a, 25 de agosto). Classification: Accuracy, recall, precision, and related metrics. Google. Recuperado de https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

[8] Google Developers. (2025b, 25 de agosto). *Thresholds and the confusion matrix*. Google. Recuperado de: https://developers.google.com/machine-learning/crash-course/classification/thresholding

[9] ISPRS. (2013). ISPRS Test Project on Urban Classification and 3D Building Reconstruction. Commission III, Working Group III/4 – 3D Scene Analysis. Recuperado de: https://www.isprs.org/resources/datasets/benchmarks/UrbanSemLab/PDF/complexscenes_revision_v4.pdf

[10] Mohanan, J. (2024, 2 de mayo). How to use Cohen's Kappa Statistic for ML Model verification. Medium. Recuperado de https://medium.com/@jayamohanmohanan/how-to-use-cohens-kappa-statistic-for-ml-model-verification-6c66258b4ae9