# Evaluación del impacto del balanceo de clases en la segmentación semántica urbana con SegNet en imágenes aéreas de alta resolución

# Evaluation of the impact of class balancing on urban semantic segmentation with SegNet in high-resolution aerial images

Gizela Andrea Guzmán [1],
Artículo recibido 25 de noviembre de 2025

**Resumen**

Este trabajo presenta la implementación de una arquitectura SegNet para la segmentación semántica de imágenes aéreas, aplicada a la zona urbana del municipio de Santa Rosa de Osos, Antioquia, y la evaluación del comportamiento del modelo frente al desbalance existente entre las clases urbanas. Para ello, se diseñaron cuatro configuraciones experimentales: (i) un entrenamiento base sin pesos, (ii) un entrenamiento con pesos por frecuencia inversa, (iii) un entrenamiento con pesos balanceados y normalizados, y (iv) un entrenamiento extendido con pesos normalizados durante 50 épocas. Estas estrategias buscan mitigar el sesgo natural del modelo hacia clases dominantes y reforzar su sensibilidad ante clases minoritarias y pequeñas como muros. Los resultados muestran que la Configuración 1 (sin pesos) presenta un buen desempeño en clases dominantes (construcciones, árboles, vías), pero prácticamente no identifica muros (F1 ≈ 0; IoU ≈ 0). La Configuración 2 mejora la detección de clases minoritarias (F1 = 0.10), mientras que la Configuración 3 incrementa aún más dicho rendimiento (F1 = 0.121; IoU = 0.064) sin afectar de forma significativa las clases mayoritarias. Finalmente, la Configuración 4, que combina pesos normalizados y un mayor número de épocas (50), logra el mejor resultado global, alcanzando mIoU = 0.657, Kappa = 0.857, y una mejora sustancial para muros (F1 = 0.240; IoU = 0.137).

**Palabras clave:** Segmentación, redes neuronales, SegNet, balance clases

Código disponible en: https://github.com/gizeandre13/PRA_Informe_1_Gizela_Guzman

**Abstract**

This paper presents the implementation of a SegNet architecture for the semantic segmentation of aerial images, applied to the urban area of the municipality of Santa Rosa de Osos, Antioquia, and the evaluation of the model's performance in the face of the existing imbalance between urban classes. To this end, four experimental configurations were designed: (i) base training without weights, (ii) training with inverse frequency weights, (iii) training with balanced and normalized weights, and (iv) extended training with normalized weights for 50 epochs. These strategies seek to mitigate the model's natural bias toward dominant classes and reinforce its sensitivity to minority and small classes such as walls. The results show that Configuration 1 (without weights) performs well in dominant classes (buildings, trees, roads), but practically does not identify walls (F1 ≈ 0; IoU ≈ 0). Configuration 2 improves the detection of minority classes (F1 = 0.10), while Configuration 3 further increases this performance (F1 = 0.121; IoU = 0.064) without significantly affecting majority classes. Finally, Configuration 4, which combines normalized weights and a higher number of epochs (50), achieves the best overall result, reaching mIoU = 0.657, Kappa = 0.857, and a substantial improvement for walls (F1 = 0.240; IoU = 0.137).

**Keywords:** Segmentation, neural networks, SegNet, class balance

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

## 1 Introducción

El conocimiento 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, el acceso a grandes volúmenes de datos y mayor capacidad computacional, consolidaron a las redes neuronales convolucionales (CNN) 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, construcciones, 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 Colombia, se ha promovido el uso de inteligencia artificial para automatizar y reducir

los costos asociados a la generación y actualización de información geográfica. Un ejemplo destacado es el proyecto piloto de actualización del Catastro Multipropósito, liderado por el Departamento Nacional de Planeación (DNP) con el apoyo del PNUD y USAID, en el cual se emplearon algoritmos de IA para identificar automáticamente elementos cartográficos vectoriales a partir de ortoimágenes; los resultados obtenidos muestran una exactitud de segmentación entre el 70% y el 94%, lo que demuestra la eficacia del enfoque para generar insumos catastrales detallados sin necesidad de intervención manual. De forma complementaria, el proyecto IA vías terciarias, también impulsado por el DNP, ha evidenciado el potencial de modelos de aprendizaje profundo para la detección y caracterización automática de redes viales rurales por medio de imágenes satelitales (DNP, 2023).

Estos avances evidencian el potencial de la inteligencia artificial para transformar la producción de cartografía básica mediante técnicas de segmentación semántica que integran de forma efectiva datos espectrales, altimétricos y geoespaciales, promoviendo procesos más automatizados, exactos y adaptables a distintos contextos geográficos del territorio colombiano.

Si bien la segmentación semántica ha logrado avances significativos, los elementos pequeños, delgados o con geometrías lineales siguen representando un desafío. Objetos como cercas, postes, muros o vías angostas tienden a ser difíciles de identificar con precisión. Esto sucede porque, durante el proceso de convolución y agrupamiento, la red extrae características de mayor nivel, pero pierde información espacial fina, especialmente aquella relacionada con los bordes y las estructuras de pequeño tamaño (Sang et al., 2023).Como resultado, estos objetos se diluyen, se confunden con clases vecinas o simplemente desaparecen del mapa de predicción.

Además, estas clases suelen estar subrepresentadas en el conjunto de entrenamiento, lo que agrava el problema: el modelo aprende a priorizar las clases dominantes (como vegetación y edificaciones), ignorando las minoritarias. Por ello, es necesario incorporar estrategias como el balanceo de clases mediante pesos en la función de pérdida, para penalizar los errores en categorías pequeñas y mejorar su representación en la salida final (Dong et al., 2019.De esta manera, no solo se busca mejorar la exactitud global del modelo, sino también su capacidad para representar con fidelidad espacial y semántica los objetos pequeños.

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 de la zona urbana del municipio de Santa Rosa de Osos.

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, permitiendo realizar inferencias a escala completa sin afectar la continuidad espacial de los objetos segmentados. Además, se incorporaron estrategias de balanceo mediante pesos por clase en la función de pérdida, con el fin de penalizar más fuertemente los errores en clases minoritarias. Esta adaptación busca mejorar la sensibilidad del modelo hacia objetos poco representados.

El desempeño del modelo se evalúa mediante métricas comunes en segmentación semántica, como la exactitud global, F1-score por clase, IoU, matriz de confusión y coeficiente Kappa. Esta evaluación permite analizar no solo el rendimiento global del modelo, sino también su sensibilidad frente a clases pequeñas, delgadas o de baja representación espacial. Para ello, se incorporan estrategias de balanceo de clases, con el fin de mitigar el sesgo del modelo hacia clases mayoritarias y mejorar la detección de objetos minoritarios.

Con ello, el estudio busca no solo validar la eficacia de SegNet en escenarios urbanos, sino también evidenciar sus limitaciones en la recuperación de detalles finos y proponer ajustes prácticos orientados a mejorar la segmentación de elementos urbanos relevantes para aplicaciones catastrales y cartográficas en el contexto colombiano.

# 2 Materiales y métodos

## 2.1 Área de estudio

El área de estudio corresponde a la zona urbana del municipio de Santa Rosa de Osos (Antioquia, Colombia), localizada en una región de topografía ondulada y en transición urbano–rural. Presenta una morfología urbana representativa de los municipios intermedios andinos, caracterizada por un núcleo central consolidado, con construcciones de baja y media altura (uno a tres niveles), techos heterogéneos y tejido compacto, y una expansión hacia sectores semiurbanos y rurales con construcciones aisladas, lotes abiertos, y cerramientos prediales, donde se mezclan usos residenciales, productivos y agrícolas.

Esta complejidad geométrica propia de los municipios colombianos con transición urbano–rural justifica la necesidad de implementar estrategias como el uso de pesos por clase, que permitan mejorar la detección de elementos minoritarios y de alta relevancia cartográfica, fortaleciendo la representación semántica y espacial del territorio.

En la **Figura 1** se puede evidenciar la localización del área de estudio y del conjunto de datos.

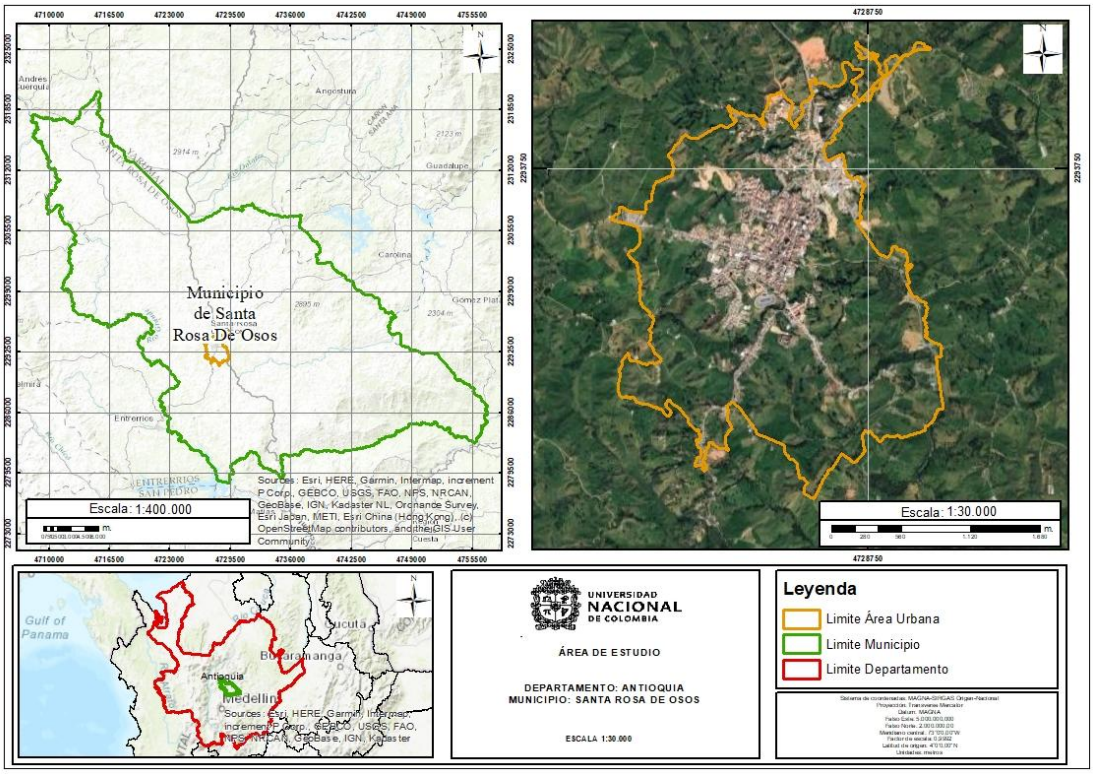


***Figura 1**. Localización conjunto de datos*
***Fuente:** Propia*

## 2.2 Diseño experimental para la evaluación del balanceo de clases

Con el fin de analizar cómo el desbalance de clases afecta el desempeño del modelo SegNet en la segmentación semántica urbana, se diseñó una evaluación experimental basada en cuatro configuraciones progresivas de entrenamiento, implementadas por medio de notebooks independientes. Cada configuración permitió observar el comportamiento del modelo bajo diferentes estrategias de ponderación de clases, tanto en escenarios sin ajuste como con pesos calculados y normalizados. Las ejecuciones se realizaron con los mismos datos, arquitectura y parámetros base, variando únicamente el tratamiento del balanceo y el número total de épocas.

### 2.2.1 Entrenamiento base sin pesos (10 épocas)

Esta primera configuración corresponde a la configuración original del modelo, sin aplicar ningún tipo de balanceo de clases. Todas las clases tienen la misma importancia en la función de pérdida.

### 2.2.2 Entrenamiento con pesos por frecuencia inversa (10 épocas)

En esta configuración se calcula un histograma de frecuencias de cada clase en los datos de entrenamiento, y se asignan pesos inversamente proporcionales a la presencia de cada clase, de modo que las clases menos frecuentes reciben una penalización mayor cuando se produce un error.

### 2.2.3 Entrenamiento con pesos balanceados y normalizados (10 épocas)

En esta etapa se aplicó una estrategia de balanceo más refinada, basada en pesos calculados a partir de la frecuencia de cada clase, pero ajustados mediante un proceso de normalización para mantener relaciones proporcionales y coherentes entre ellas. Este enfoque permite conservar la importancia relativa de cada categoría, pero evita que las clases mayoritarias dominen completamente el aprendizaje.

### 2.2.4 Entrenamiento final (50 épocas)

Con base en los resultados obtenidos, se seleccionó la configuración más estable (pesos balanceados y normalizados) y se llevó a cabo un entrenamiento extendido a 50 épocas, con la finalidad de potenciar la capacidad del modelo para aprender patrones espaciales más complejos, mejorar la coherencia semántica y optimizar el rendimiento por clase.

**Tabla 1.** *Descripción configuraciones evaluadas*

| Configuración | Pesos | Épocas | Nombre |
|---|---|---|---|
| 1 | Sin pesos | 10 | SegNet_Base_NoWeights_Epoch10 |
| 2 | Pesos con frecuencia inversa | 10 | SegNet_ClassWeights_InverseFreq_Epoch10 |
| 3 | Pesos con frecuencia inversa normalizados | 10 | SegNet_ClassWeight_Normalized_Epoch10 |
| 4 | Pesos con frecuencia inversa normalizados | 50 | SegNet_Final_ClassWeight_Normalized_Epoch50 |

***Fuente:*** *Propia*

## 2.3 Características de los datos

Se emplearon productos cartográficos generados en el marco de procesos de actualización cartográfica, a continuación, se relacionan las principales características:

Ortoimágenes verdaderas con resolución espacial de 10 cm/píxel, adquiridas en 2024, en formato GeoTIFF y con composición espectral RGBI (Rojo, Verde, Azul e Infrarrojo cercano), adecuadas para la identificación de elementos urbanos y/o cobertura vegetal.

Base de datos vectorial generada mediante fotointerpretación 3D con restitución fotogramétrica y estructurada conforme al Catálogo de Objetos Geográficos del IGAC utilizada como insumo de referencia para la generación de datos etiquetados.

Estos productos cumplen con los parámetros técnicos establecidos en las especificaciones técnicas del IGAC, garantizando una estandarización conceptual, geométrica y temática que respalda la interoperabilidad y consistencia de la información geográfica en Colombia (IGAC, 2023).

## 2.4 Preprocesamiento y etiquetado de datos

Las ortoimágenes originales fueron divididas en mosaicos de 2000 × 2000 píxeles, preservando su sistema de referencia espacial y su georreferenciación. Este proceso generó un total de 46 mosaicos, los cuales fueron preparados para su procesamiento con el modelo de segmentación semántica.

Adicionalmente, las imágenes fueron reconfiguradas espectralmente para ser estandarizadas a tres bandas, seleccionando las bandas Infrarrojo cercano, Rojo y Verde, lo que permite mejorar la diferenciación entre vegetación, construcciones y superficies artificiales.

Por otro lado, el etiquetado de datos se realizó a partir de la base de datos, los vectores correspondientes a las clases de interés fueron filtrados, estandarizados y codificados mediante un identificador numérico (ID de clase). Posteriormente, cada capa vectorial fue rasterizada, asegurando su correcta alineación espacial con las ortoimágenes y

conservando la resolución original (10 cm/píxel).

Con el fin de adaptarlos al formato de entrada requerido por el modelo, las máscaras raster fueron unificadas en un solo archivo por escena, y se les asignó una codificación RGB, donde cada clase se representó mediante un color específico. Este proceso permitió generar 46 máscaras semánticas etiquetadas en formato GeoTIFF, compatibles con los esquemas de entrenamiento de redes convolucionales en PyTorch.

Las máscaras semánticas generadas contienen etiquetas asociadas a cinco clases cartográficas de interés, seleccionadas por su relevancia en la representación geométrica y semántica del entorno urbano colombiano: indefinido, construcciones, árboles, vías y muros, incluyendo elementos estructuralmente complejos y de baja representatividad espacial como cerramientos.

***Tabla 2.*** *Clases para entrenamiento*

| ID | Clase semántica | Descripción IGAC |
|---|---|---|
| 0 | Indefinido | Áreas no clasificadas |
| 1 | Construcciones | Infraestructuras edificadas de forma permanente y destinadas a distintos usos o funciones |
| 2 | Árboles | Áreas extensas cubiertas por árboles, con altura mínima de 2 metros. |
| 3 | Vías | Estructura diseñada para permitir el tránsito de vehículos, personas y animales, conectando entre sí poblaciones, zonas urbanas y lugares de interés. |
| 4 | Muros | Infraestructuras edificadas de forma permanente y destinadas a distintos usos o funciones |

***Fuente:*** *Propia*

## 2.5 Cargue y procesamiento de datos

El conjunto de datos tiene un tamaño de 2000 × 2000 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 "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.

Por otro lado, para garantizar un entrenamiento adecuado y una evaluación objetiva del modelo, se construyeron los subconjuntos de entrenamiento (train) y prueba (test) a partir de las imágenes y etiquetas generadas para este proyecto. A diferencia del código original, que dependía de listas manuales, se implementó un método, basado en la búsqueda y emparejamiento entre imagen y máscara mediante operaciones de conjuntos. Confirmada la correspondencia, los datos se dividieron aleatoriamente en una proporción 70/30, asegurando que el conjunto de prueba incluya áreas no vistas durante el entrenamiento y permita evaluar la capacidad de generalización del modelo en distintas configuraciones urbanas.

## 2.6 Arquitectura 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**).

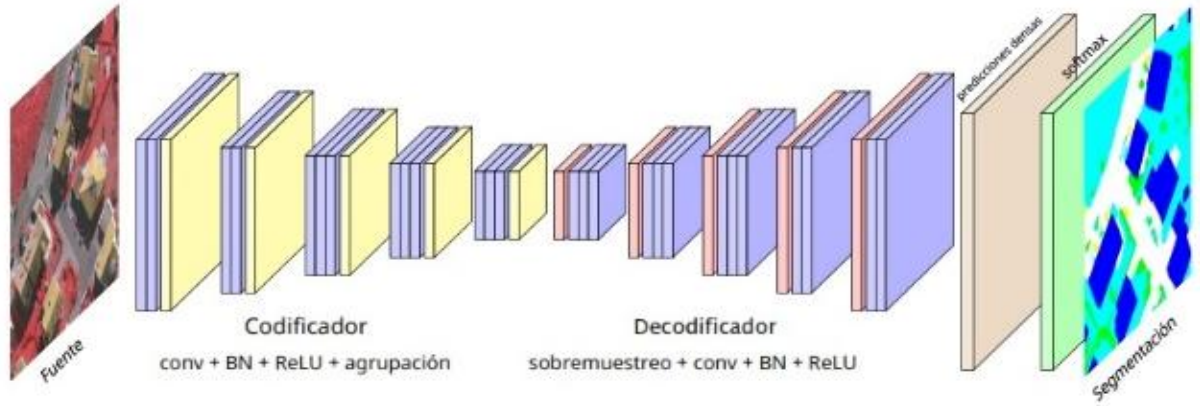


***Figura 2***. *Estructura arquitectura SegNet*
***Fuente***: (Audebert et al., 2017)

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.

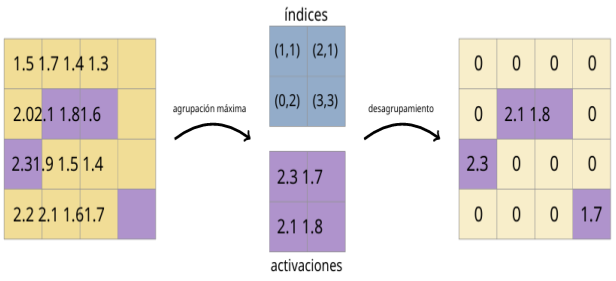


***Figura 3***. *Funcionamiento operaciones maxpooling y unpooling (mapa de características 4*4)*
***Fuente***: (Audebert et al., 2017)

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.

## 2.7 Configuración del entrenamiento

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 de datos, mientras que las capas del decodificador se inicializan aleatoriamente y se ajustan durante el entrenamiento.

Antes de entrenar, se instancia el modelo SegNet con 3 canales de entrada y 5 clases de salida correspondientes a las categorías cartográficas definidas, y se utiliza para el entrenamiento el conjunto de datos "train" previamente dividido, el cual representa el 70 % de la totalidad de los datos. Igualmente se configura la ejecución en GPU con el fin de optimizar los tiempos de cómputo.

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 $5\times10^{-4}$; el cual permite controlar el sobreajuste, favoreciendo la capacidad de generalización del modelo.

Dado que el codificador del modelo se inicializa con pesos preentrenados en ImageNet y el decodificador se entrena desde cero, se implementó una estrategia de aprendizaje con tasas diferenciadas (differential learning rates). El codificador utiliza una tasa de aprendizaje reducida (lr = 0.005) para preservar el conocimiento previo, mientras que el decodificador aprende con una tasa mayor (lr = 0.01), ajustándose más rápidamente a los patrones específicos de los datos geoespaciales. Adicionalmente, para refinar el aprendizaje en etapas avanzadas, se aplicó un esquema de reducción progresiva del learning rate (StepLR), disminuyendo la tasa en un factor de 0.1 en las épocas 25, 35 y 45.

## 2.8 Función de perdida y balanceo de clases

Uno de los principales retos en la segmentación urbana es el desbalance entre clases, ya que elementos como construcciones y árboles suelen estar sobrerrepresentados, mientras que clases pequeñas y fragmentadas, como muros, aparecen con poca frecuencia.

Para mitigar este problema, y mejorar el aprendizaje del modelo, se implementaron tres enfoques progresivos en la función de pérdida. En una primera configuración, se utilizó la función CrossEntropyLoss sin aplicar ningún tipo de ponderación, asignando el mismo peso a todas las clases, estableciendo un punto de referencia. Posteriormente, se incorporó una estrategia de balanceo mediante pesos calculados a partir de la frecuencia inversa de cada clase; en este enfoque, las clases menos frecuentes reciben una mayor penalización en el cálculo del error, lo que incentiva al modelo a prestarles más atención durante el entrenamiento. Finalmente, con el propósito

de mantener la proporción entre clases sin generar diferencias excesivas, los pesos fueron ajustados mediante un proceso de normalización, manteniéndolos dentro de un rango controlado (0.2 a 1.0).

### 2.9 Entrenamiento del modelo

El entrenamiento se ejecutó inicialmente durante 10 épocas en las tres primeras configuraciones, con el objetivo de analizar el comportamiento del modelo bajo distintas estrategias de balanceo de clases. Posteriormente, la configuración con mejores resultados se entrenó durante 50 épocas, permitiendo un aprendizaje más profundo y una optimización progresiva de los patrones espaciales. En todos los casos, las imágenes fueron previamente divididas en parches y normalizadas, facilitando su procesamiento eficiente y estandarizado por la red.

En cada iteración, el modelo recibe un lote de imágenes y genera una predicción para cada píxel; esta predicción se compara con las etiquetas reales mediante la función de pérdida, y el error resultante se utiliza para actualizar los pesos del modelo vía retropropagación.

Durante el proceso, se implementa un sistema de monitoreo visual que permite observar la evolución del aprendizaje. Periódicamente, se registra 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 permite una supervisión efectiva del desempeño cualitativo y verificar la mejora progresiva en la delimitación espacial de las clases.

El modelo se valida 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 para la evaluación del desempeño del modelo, incluyendo la matriz de confusión, la precisión global (Overall Accuracy), la puntuación F1 por clase, el índice IoU (Intersection over Union), el mIoU (IoU promedio entre clases) y el coeficiente Kappa, permitiendo un análisis tanto global como detallado del comportamiento del modelo por clase.

### 2.10 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, índice IoU, el mIoU y el coeficiente Kappa.

## 3 Resultados

Una vez completado el entrenamiento de cada configuración, 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.

## 3.1 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 3Tabla 4Tabla 5Tabla** 6 se reportan las matrices de confusión por píxel de las cuatro configuraciones, usando el orden de clases establecido:(indefinido, construcciones, arboles, vías y muros), 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 3.*** Matriz de confusión configuración 1

| | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| Indefinido (GT) | 31.454.452 | 873.467 | 656.930 | 346.013 | 0 |
| Construcciones (GT) | 390.205 | 14.882.799 | 18.239 | 18.492 | 0 |
| Árboles (GT) | 467.472 | 8121 | 2.415.894 | 304 | 0 |
| Vías (GT) | 1.203.294 | 74.602 | 24.261 | 2.941.027 | 0 |
| Muros (GT) | 181.962 | 25.371 | 11.875 | 5216 | 4 |

***Fuente:*** *Propia*

***Tabla 4.*** Matriz de confusión configuración 2

| | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| Indefinido (GT) | 28.054.473 | 981.449 | 494.091 | 724.313 | 1.345.590 |
| Construcciones (GT) | 603.322 | 15.300.782 | 25.128 | 23.479 | 280.030 |
| Árboles (GT) | 1.331.510 | 2026 | 2.175.151 | 114 | 42.386 |
| Vías (GT) | 907.360 | 107.240 | 13.420 | 3.142.968 | 246.085 |
| Muros (GT) | 62.678 | 14.510 | 7700 | 2618 | 111.577 |

***Fuente:*** *Propia*

***Tabla 5.*** Matriz de confusión configuración 3

| | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| Indefinido (GT) | 30.397.919 | 613.897 | 936.673 | 980.558 | 115.999 |
| Construcciones (GT) | 1.104.107 | 13.875.270 | 24.491 | 170.303 | 20.219 |
| Árboles (GT) | 319.999 | 3533 | 2.580.359 | 165 | 493 |
| Vías (GT) | 836.226 | 61.124 | 25.181 | 3.661.020 | 5016 |
| Muros (GT) | 215.664 | 9851 | 8240 | 7454 | 26.239 |

***Fuente:*** *Propia*

***Tabla 6.*** Matriz de confusión configuración 4

| | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| Indefinido (GT) | 31.418.767 | 592.705 | 679.508 | 607.307 | 155.025 |
| Construcciones (GT) | 530.974 | 13.709.756 | 28.855 | 6945 | 12.447 |
| Árboles (GT) | 538.246 | 2472 | 3.146.372 | 986 | 3338 |
| Vías (GT) | 1.105.008 | 14.136 | 27485 | 3.156.090 | 8491 |
| Muros (GT) | 176.454 | 9873 | 5878 | 3584 | 59.298 |

***Fuente:*** *Propia*

## 3.2 Exactitud del modelo (OA)

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.

En la **Tabla 9** se relacionan los valores de exactitud global por cada configuración.

## 3.3 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 7** se presentan los valores del F1-Score por clase obtenidos en las cuatro configuraciones evaluadas. En términos generales, se observa que las clases Construcciones, Árboles y Vías mantienen valores consistentemente altos en todas las configuraciones, lo que refleja un buen desempeño del modelo en clases con presencia espacial continua y amplia representación.

Para la clase Muros, se evidencian valores considerablemente inferiores en comparación con el resto de categorías, lo que coincide con su menor presencia espacial y su condición de elemento pequeño y lineal. No obstante, se registra una ligera mejora progresiva entre las configuraciones 2 a 4. Finalmente, la clase Indefinido presenta valores elevados debido a la agrupación de píxeles no pertenecientes a clases específicas; sin embargo, no es relevante para la interpretación semántica del modelo.

***Tabla 7.*** Resultados F1 Score por clase

| Configuración | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| 1 | 0.939 | 0.955 | 0.803 | 0.779 | 0.00004 |
| 2 | 0.897 | 0.938 | 0.694 | 0.756 | 0.100 |
| 3 | 0.922 | 0.933 | 0.796 | 0.778 | 0.121 |
| 4 | 0.935 | 0.958 | 0.830 | 0.781 | 0.240 |

***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.

## 3.4 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.

En la **Tabla 9** se relacionan los valores de Kappa obtenidos por cada configuración.

## 3.5 Intersección sobre unión (IoU)

La métrica cuantifica la superposición espacial de la segmentación predicha y la etiqueta real (verdad terreno), respecto al área total que abarca la unión de ambas. Esta puede calcularse individualmente para cada clase mediante el IoU, o de forma global utilizando el mIoU, que corresponde al promedio de los IoU de todas las clases e indica el rendimiento general del modelo. Su valor óptimo es 1, que representa una coincidencia perfecta entre predicción y realidad, mientras que el valor 0 indica una ausencia total de superposición. En los casos en que una clase no se encuentra presente ni en la predicción ni en la verdad

terreno, la métrica retorna un valor de −1 (Lightning-AI., 2025)

$$IoU = \frac{Área\ de\ intersección}{Área\ de\ Unión}$$

***Tabla 8.*** Resultados IoU por clase

| Configuración | Indefinido (Pred) | Construcciones (Pred) | Árboles (Pred) | Vías (Pred) | Muros (Pred) |
|---|---|---|---|---|---|
| 1 | 0.884 | 0.914 | 0.671 | 0.638 | 0.00002 |
| 2 | 0.813 | 0.883 | 0.532 | 0.608 | 0.053 |
| 3 | 0.856 | 0.874 | 0.662 | 0.637 | 0.064 |
| 4 | 0.878 | 0.920 | 0.710 | 0.640 | 0.137 |

***Fuente:*** *Propia*

En la Tabla 4 se evidencian altos valores de IoU en las clases Construcciones, Árboles y Vías, especialmente en la Configuración 4, que presenta el desempeño más equilibrado. Construcciones es la clase mejor segmentada en todas las configuraciones, seguida de Vías y Árboles. En contraste, la clase Muros registra los valores más bajos debido a su baja representatividad, geometría delgada y dificultad de detección; sin embargo, muestra una mejora gradual hasta alcanzar su mejor valor en la Configuración 4 (0.137).

En la **Tabla 9** se presentan los resultados globales obtenidos para cada configuración. Se observa que los valores de OA, F1 Global, Kappa y mIoU varían entre configuraciones, mostrando diferencias en el desempeño general del modelo. La Configuración 1 alcanza valores altos de OA (92.31 %) y Kappa (0.862), mientras que la Configuración 4 registra el mayor F1 Global (0.748) y el mIoU más alto (0.657), evidenciando un mejor comportamiento general en esta métrica.

***Tabla 9.*** *Resultados globales por configuración*

| Configuración | OA (%) | F1 Global | Kappa | mIoU |
|---|---|---|---|---|
| 1 | 92.311 | 0.695 | 0.862 | 0.621 |
| 2 | 87.116 | 0.677 | 0.783 | 0.578 |
| 3 | 90.251 | 0.71 | 0.830 | 0.618 |
| 4 | 91.947 | 0.748 | 0.857 | 0.657 |

***Fuente:*** *Propia*

## 3.6 Entrenamiento del modelo Final

En la **¡Error! No se encuentra el origen de la referencia.** se presentan los resultados del entrenamiento del modelo en la configuración 4 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 70.2 % 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.

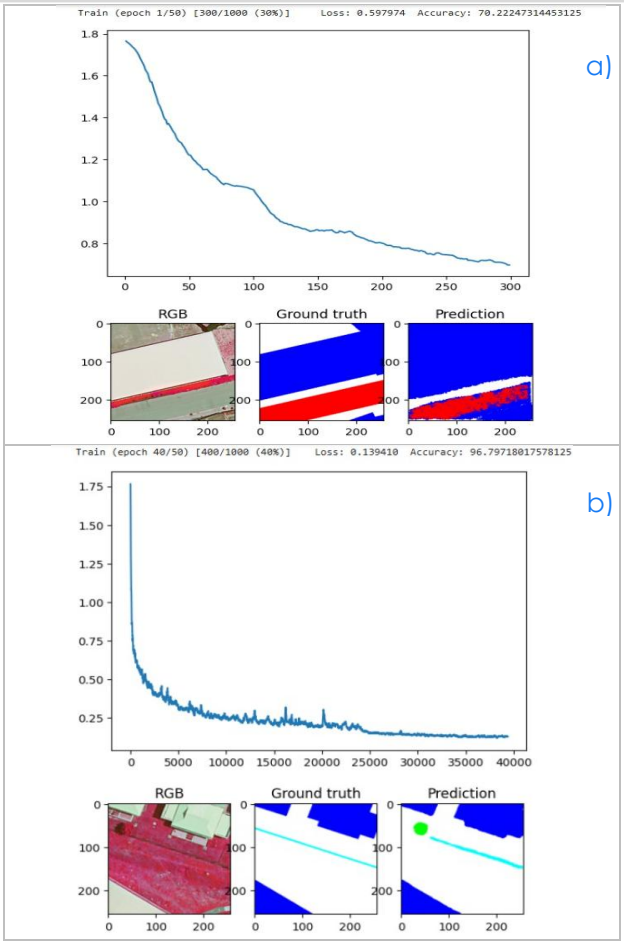


***Figura 4.*** *Comparación de la Pérdida y la predicción Visual entre la Época Inicial y Final*
*a) Época inicial (1) – b) Época final (50)*
***Fuente***: Propia

### 3.7 Evaluación cualitativa del modelo

En la siguiente figura se presenta la comparación visual entre la verdad terreno y los resultados de segmentación obtenidos con las cuatro configuraciones evaluadas. Se observa la interpretación espacial del modelo sobre un mismo mosaico de prueba, permitiendo apreciar diferencias en la delimitación de construcciones, vías, vegetación y muros, así como la variación en la continuidad, profundidad y precisión de las predicciones entre configuraciones.

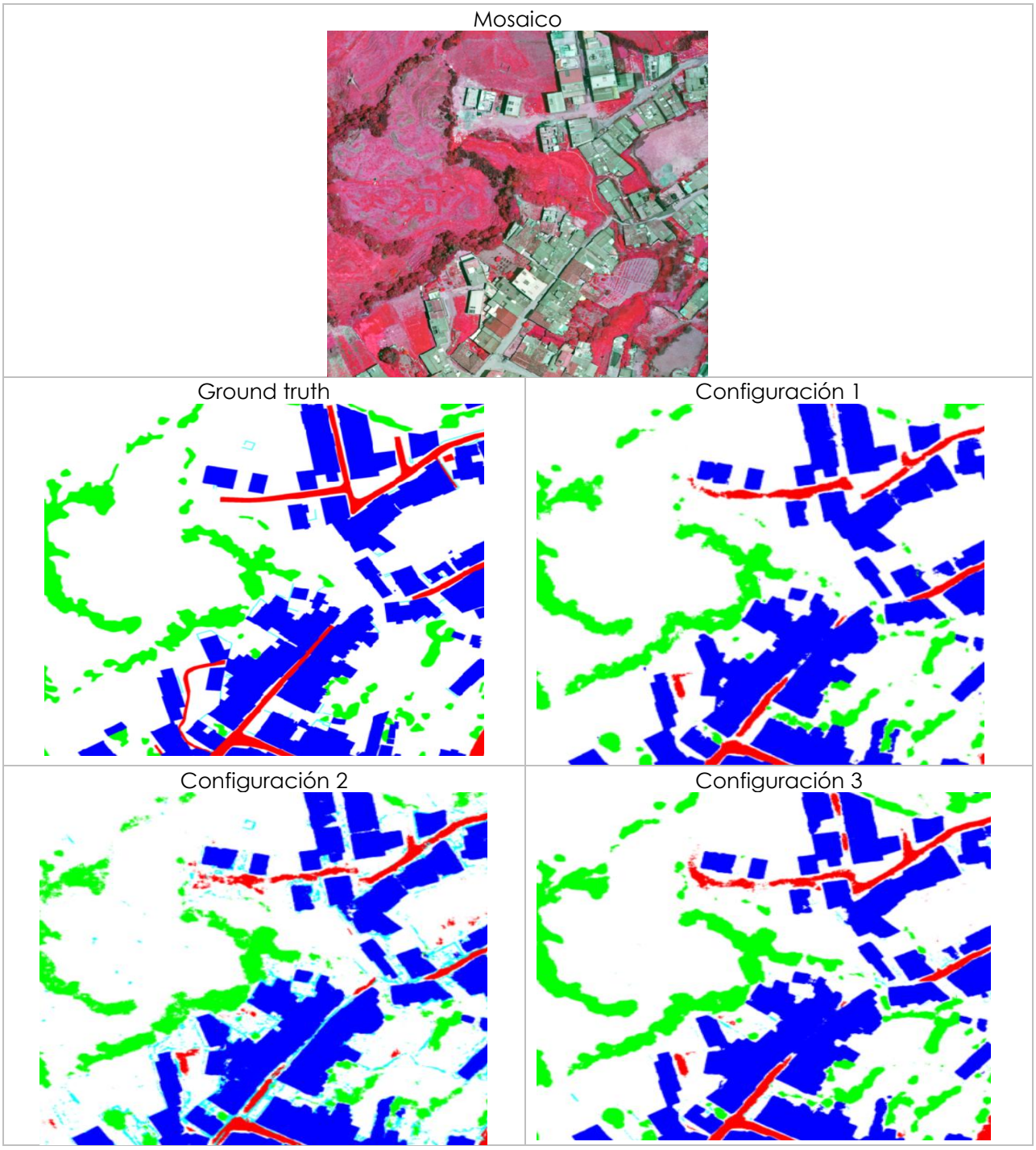


***Figura 5**. Comparación visual de la inferencia de por cada configuración*
***Fuente**: Propia*

La figura muestra la comparación visual de los resultados de segmentación obtenidos con la misma configuración de pesos balanceados normalizados, pero con diferente número de épocas de entrenamiento. En la Configuración 3 (10 épocas), el modelo logra identificar correctamente las clases principales y su distribución espacial, aunque se observan áreas con bordes difusos, presencia de pequeñas omisiones y ciertos errores en zonas edificadas y lineales.

En la Configuración 4 (50 épocas), se evidencia una mayor precisión en la segmentación, con bordes más definidos, mejor continuidad espacial en vías y superficies construidas, así como una representación más clara de las clases minoritarias.

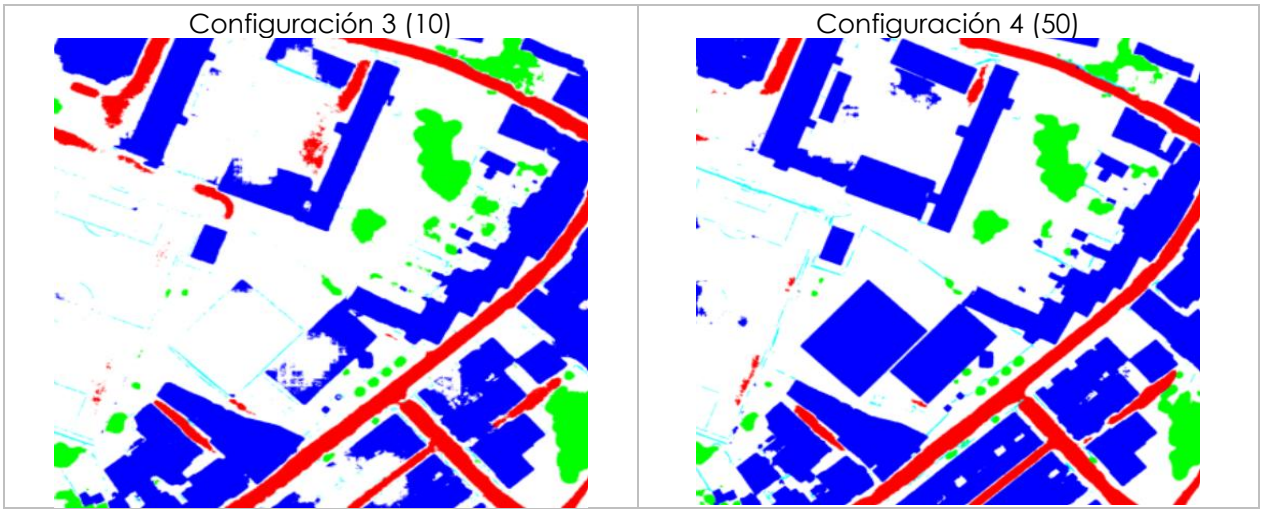


***Figura 6**. Comparación visual por Nª de épocas*
***Fuente**: Propia*

# 4 Discusión

## 4.1 Comportamiento general del modelo

Los resultados globales (**Tabla 9**) muestran que las cuatro configuraciones obtienen un desempeño alto en la segmentación, con valores de OA entre 87 % y 92 %, y mIoU entre 0.578 y 0.657. Sin embargo, la Configuración 1 (sin pesos) alcanza el mayor OA (92.31%), pero no logra capturar la clase de muros (IoU ≈ 0).

Este comportamiento coincide con lo reportado en la literatura, donde se ha documentado que la métrica OA puede resultar engañosa en escenarios con fuerte desbalance de clases, dado que su cálculo no penaliza la omisión sistemática de clases minoritarias, y resume el acierto total sobre todos los píxeles sin considerar explícitamente la proporción de cada clase, por ello se recomienda el uso de métricas como IoU y F1-score para evaluar el desempeño en escenarios desbalanceados, las cuales se calculan por clase y penalizan tanto los falsos positivos como los falsos negativos (Ghanaei & Rouhani, 2025).

Por otro lado, la Configuración 4 (pesos normalizados + 50 épocas) obtiene el mejor mIoU (0.657) y el F1 global más alto (0.748), lo que indica una segmentación más equilibrada entre clases dominantes y minoritarias. El uso de pesos en la función de pérdida es una estrategia validada para mejorar la representación de clases poco frecuentes, permitiendo que el modelo aprenda características relevantes de estas clases y reduzca el sesgo hacia las mayoritarias. Métodos similares han demostrado mejoras significativas en la precisión y el IoU de clases minoritarias sin sacrificar el desempeño global (Mahmoudi et al., 2025)

Estos resultados sugieren que, aunque el entrenamiento sin pesos favorece la precisión global, el uso de pesos

normalizados permite una representación más justa de todas las clases, especialmente las menos frecuentes.

## 4.2 Análisis matrices de Confusión

El análisis de las matrices de confusión de las cuatro configuraciones permite comprender en profundidad cómo evoluciona el desempeño de SegNet al incorporar estrategias de balanceo de clases y al aumentar la cantidad de épocas. Las matrices permiten identificar no solo cuántos píxeles se clasifican correctamente, sino también hacia qué clases se desvían las predicciones incorrectas.

En la Configuración 1, se evidencia una buena clasificación de las clases dominantes, pero ignora completamente la clase muros, con solo 4 píxeles de muros correctamente clasificados, los cuales se confunden con todas las demás clases, lo cual puede deberse a la particularidad de su forma pequeña, lineal y delgada, como lo reportan Sang et al., (2023)quienes destacan que las operaciones convolucionales y de agrupamiento reducen la resolución, lo que a menudo conduce a la pérdida de información detallada de objetos pequeños o delgados.

En la Configuración 2 se observa una mejora significativa en la identificación de la clase muros, el modelo logra clasificar correctamente 111.577 píxeles, en contraste con los 4 identificados en la Configuración 1. Este cambio confirma que el uso de pesos balanceados favorece el aprendizaje de clases minoritarias. Sin embargo, la matriz de confusión también revela ciertos efectos indeseados. Por un lado, aumentan las confusiones en las clases indefinido y construcciones, mientras que árboles y vías muestran un comportamiento menos estable respecto al modelo base. Además, aunque los muros empiezan a ser reconocidos, también surgen numerosos falsos positivos, con 24.159 píxeles clasificados como vías y 11.114 como árboles.

Estos resultados coinciden con lo señalado por Tian et al., (2022)quienes indican que las pérdidas ponderadas estáticas basadas en frecuencia tienden a sobrerreforzar las clases minoritarias, generando un aumento de falsos positivos, y además asumen erróneamente que estas clases son rígidas, pese a su variabilidad morfológica y espacial. Esto provoca una disminución real de la precisión, incluso cuando parece haber aprendizaje de la clase minoritaria. Como alternativa, los autores proponen una pérdida con minería dinámica de clases, que ajusta los pesos según el rendimiento de recuperación en tiempo de entrenamiento. Este enfoque evita sobreestimación de clases poco representadas y logra un equilibrio más estable entre detección y precisión, manteniendo un mIoU competitivo.

En la configuración 3 se observa un comportamiento intermedio, la detección de la clase muros mejora nuevamente, alcanzando 26.229 píxeles correctamente clasificados, aunque en menor medida que en la configuración 2. Sin embargo, esta reducción viene acompañada de la disminución de falsos positivos, lo que indica un aprendizaje más controlado de la clase minoritaria. Estudios como el realizado por Alcover-Couso et al., (2024)demuestran que estrategias de ponderación moderada, como el ajuste dinámico de pesos o la ponderación basada en el gradiente, logran un mejor equilibrio entre la mejora de clases minoritarias y la estabilidad general del modelo.

Por último, la Configuración 4, que combina pesos normalizados con un entrenamiento de 50 épocas, presenta el desempeño más robusto y equilibrado entre todas las evaluadas. En esta configuración, la clase muros alcanza su mejor resultado, con 59.298 píxeles correctamente clasificados, más del doble de lo obtenido en la configuración 3. Al mismo tiempo, se observa una reducción significativa de los errores extremos entre clases, y la distribución de confusiones se vuelve más uniforme y coherente.

Al mismo tiempo, las clases dominantes recuperan parte de la estabilidad perdida anteriormente, construcciones, árboles y vías presentan valores más altos en la diagonal de la matriz de confusión, y se reduce la cantidad de confusiones cruzadas, evidenciando un comportamiento más coherente del modelo. Demostrando que el entrenamiento prolongado, junto con la

normalización de pesos, logra un balance adecuado entre el aprendizaje de clases minoritarias y la preservación del rendimiento en las clases mayoritarias.

Diversos estudios han demostrado que, en general, aumentar el número de épocas durante el entrenamiento de modelos de aprendizaje profundo mejora el rendimiento y la precisión, al menos hasta alcanzar un punto óptimo antes de que aparezca el sobreajuste. Por ejemplo, Saputra et al., (2024)destacan que incrementar la cantidad de épocas es fundamental para mejorar la exactitud de los modelos de redes neuronales convolucionales (CNN), ya que permite que el modelo aprenda patrones más complejos y obtenga mejores resultados en tareas de reconocimiento de imágenes.

### 4.3 Análisis clases mayoritarias

En conjunto, los resultados muestran que las clases mayoritarias (Indefinido, Construcciones, Árboles y Vías) mantienen un desempeño estable y elevado en todas las configuraciones evaluadas, aun cuando se introducen estrategias de balanceo y modificaciones en la función de pérdida. Esto evidencia que el modelo conserva su capacidad para aprender patrones espaciales dominantes incluso cuando se incrementa la penalización relativa sobre clases minoritarias como muros.

Lo que contrasta con diversos estudios que han demostrado que, incluso al introducir estrategias de balanceo y modificaciones en la función de pérdida, las clases mayoritarias tienden a mantener un desempeño elevado y estable, reflejando la capacidad de los modelos para aprender patrones dominantes y estructuras espaciales continuas. Por ejemplo, Chen et al., (2025)señalan que la función de pérdida de entropía cruzada "presenta un sesgo natural hacia la optimización de las clases mayoritarias debido a su mayor representación en los datos de entrenamiento", lo que permite que estas clases mantengan altos valores de IoU y F1, incluso cuando se aplican funciones de pérdida compuestas para mejorar el rendimiento de las clases minoritarias.

En particular, construcciones y árboles presentan de forma consistente valores altos de F1 e IoU, lo que refleja que SegNet identifica adecuadamente superficies continuas y homogénea, esto se relaciona con su morfología geométrica definida, su alta representatividad en el conjunto de entrenamiento y su consistencia visual, lo cual facilita el aprendizaje del modelo. La clase vías, pese a su geometría lineal y variabilidad tonal, también mantiene valores estables, lo que indica que el modelo logra capturar adecuadamente su estructura espacial. Esta estabilidad se refleja también en las métricas globales, donde OA se mantiene entre el 87 % y el 92 %, mientras que el coeficiente Kappa que descuenta el acuerdo por azar se preserva en rangos altos, señalando un desempeño consistente.

La Configuración 4, que se destaca por ser la más equilibrada y robusta, en ella, las clases dominantes alcanzan simultáneamente sus mejores valores, evidenciando mejoras tanto en F1 como en IoU sin sacrificar la estabilidad general del modelo. Esto se aprecia también en el incremento del mIoU, que alcanza su mejor valor (0.657), reflejando un avance significativo en la coherencia espacial del modelo en todas las clases. El aumento del Kappa (0.857) confirma además que, bajo esta configuración, el modelo logra una segmentación más fiable y menos afectada por desequilibrios en la distribución de clases.

### 4.4 Análisis clases minoritarias

La clase muros fue la más desafiante para el modelo debido a los factores como la baja representación en el conjunto de entrenamiento, y su geometría delgada y fragmentada, lo que dificulta que la red capture su continuidad espacial. Los resultados obtenidos a lo largo de las cuatro configuraciones muestran una progresión clara y coherente del comportamiento del modelo frente a esta clase minoritaria, así como su impacto en las métricas globales.

En la configuración 1, sin ningún mecanismo de balanceo de clases, el modelo prácticamente ignora a los muros. Esto se refleja en métricas casi nulas: F1 ≈ 0.00004 e IoU ≈ 0, evidenciando que el modelo se sesga de manera natural hacia las clases dominantes, aun así, en este estado inicial,

las métricas globales como OA (92.31 %) y Kappa (0.862) son relativamente altas, lo que confirma que el desempeño en clases mayoritarias enmascara las deficiencias en clases poco frecuentes.

Con la configuración 2, donde se introduce el balanceo por frecuencia inversa, los resultados mejoran de manera notable para la clase muros. El modelo logra, reconocerlos parcialmente, alcanzando F1 = 0.100 e IoU = 0.053. Sin embargo, el aumento de los pesos genera un impacto adverso en el desempeño global, el OA disminuye a 87.12 % y el Kappa a 0.783, acompañado de una disminución del mIoU. Esto sugiere que el balanceo agresivo introduce ruido en la estabilidad del aprendizaje.

La configuración 3 muestra un efecto intermedio donde la clase muros continúa mejorando (F1 = 0.121), y el modelo recupera estabilidad en métricas globales (OA = 90.25 %, Kappa = 0.830, mIoU = 0.618). Esto indica que un entrenamiento con pesos, pero menos extremo, permite un equilibrio más adecuado entre clases frecuentes y minoritarias.

Finalmente, la configuración 4, que incorpora pesos normalizados y aumenta el número de épocas, presenta la mejora más consistente. La clase muros alcanza F1 = 0.240 e IoU = 0.137, lo que representa un incremento significativo respecto al modelo sin balanceo. Además, esta configuración no solo mejora la clase minoritaria, sino que también conserva un desempeño sobresaliente en clases mayoritarias y recupera los valores globales, OA = 91.95 %, Kappa = 0.857 y mIoU = 0.657.

Estos resultados confirman que las clases minoritarias requieren estrategias de ponderación especializadas y, además, un mayor número de épocas para consolidar su aprendizaje. La normalización de los pesos demostró ser clave para evitar efectos adversos sobre clases dominantes, permitiendo un entrenamiento más estable y equilibrado.

En conjunto, la Configuración 4 logra el mejor compromiso entre precisión global y recuperación de objetos de baja frecuencia, evidenciando la importancia del balanceo controlado en escenarios urbanos con fuerte desbalance de clases, como lo sugieren diversos autores como Audebert et al., (2017). En sus estudios, explican que el uso de frecuencias de clase inversas permite mejorar la precisión promedio de clase, aunque advierten que su aplicación debe realizarse con criterio, especialmente cuando existen clases extremadamente raras o mal definidas como la clase "indefinido", a las que asignan un peso controlado para evitar una penalización desproporcionada. Esta reflexión es coherente con los resultados obtenidos, donde la normalización de pesos permite mejorar la recuperación de clases minoritarias, como muros, sin generar efectos adversos de sobreponderación.

Aunque la Configuración 4 y las estrategias de balanceo de clases han mejorado la segmentación de categorías minoritarias, la clase muros continúa siendo especialmente difícil de segmentar, una dificultad ampliamente reportada por otros autores en el campo. Diversos estudios han identificado que, incluso con arquitecturas avanzadas y técnicas de balanceo, las clases como muros presentan bajos valores de IoU y precisión en comparación con otras clases urbanas, la segmentación de muros se ve limitada por factores como la variabilidad morfológica, la presencia de objetos adyacentes y la calidad de los datos (Pan et al., 2023).

### 4.5 Evaluación cualitativa del modelo

De acuerdo a lo observado en la Figura 5 entre las cuatro configuraciones se evidencia diferencias importantes en la calidad de la segmentación. En la Configuración 1 (sin pesos, 10 épocas), el modelo logra identificar correctamente las clases dominantes construcciones, vías y vegetación (azul, rojo y verde) pero presenta bordes imprecisos y prácticamente no detecta la clase muros, coincidiendo con su F1 e IoU cercanos a cero.

Con la Configuración 2 (pesos inversos, 10 épocas) se observa que la penalización fuerte hacia las clases minoritarias introdujo ruido y ciertas inconsistencias en las clases principales, evidenciando muchas zonas en color cian que representa los muros donde no corresponden.

La Configuración 3 (pesos normalizados, 10 épocas) muestra resultados más equilibrados, se reduce el ruido, se mantienen las clases dominantes con buena precisión y aparece mayor continuidad en muros y vías, aunque aún con fragmentación.

Confirmando que la mejor configuración es la nª 3 con balanceo de clases normalizados, por ende, se procedió a replicar esta misma configuración, pero ahora con mas épocas. En la Figura 6 se muestra el efecto del número de épocas sobre la calidad de la segmentación utilizando la misma configuración. En la Configuración 3 (10 épocas), el modelo es capaz de captar la estructura general de las clases principales, pero aún presenta bordes irregulares, discontinuidades en las vías y una representación limitada de las clases minoritarias, especialmente muros. En contraste, en la Configuración 4 (50 épocas) se observa una mejora visual notable, las construcciones presentan contornos más definidos, las vías muestran una continuidad más homogénea y se reduce significativamente el ruido en zonas de vegetación. Además, la clase muros aparece con mayor estabilidad y menor fragmentación, reflejando que el aumento de épocas permite consolidar patrones espaciales más finos.

Lo cual es consistente con lo evidenciado en la **Figura 4** que muestra la evolución del proceso de entrenamiento al comparar la época inicial (1) y la época final (50). Esta comparación permite observar tanto la reducción progresiva de la función de pérdida como la mejora visual en la calidad de las predicciones. Tras completar 50 épocas, la curva de pérdida presenta una disminución marcada y sostenida, alcanzando valores notablemente más bajos y estables. Esto indica que el modelo ha logrado una convergencia adecuada, optimizando sus parámetros y reduciendo la incertidumbre en la segmentación. La mejora en la precisión espacial y la coherencia de las clases segmentadas confirma que un mayor número de épocas favorece la consolidación del aprendizaje.

En síntesis, la evaluación cualitativa confirma que el balanceo de clases y el aumento del tiempo de entrenamiento mejoran de forma sustancial las clases minoritarias, mientras que la normalización de pesos evita degradar el desempeño en las clases dominantes, logrando resultados visualmente más consistentes en la Configuración 4.

## 5 Conclusiones

i. SegNet demostró un buen desempeño general en la segmentación de escenas urbanas, alcanzando valores altos de exactitud global (OA > 90 % en tres configuraciones) y métricas de consistencia como el coeficiente Kappa (> 0.83 en las configuraciones 1, 3 y 4, confirmando su capacidad para extraer patrones espaciales relevantes en imágenes aéreas de muy alta resolución.
ii. Las clases dominantes (Construcciones, Árboles, Vías e Indefinido) mantienen un rendimiento estable y elevado en todas las configuraciones, con F1-scores superiores al 0.75 en la mayoría de los casos, lo que evidencia que el modelo preserva su capacidad de aprendizaje sobre estructuras con alta frecuencia espacial, aun cuando se introducen pesos diferenciales en la función de pérdida.
iii. La clase Muros constituye el mayor desafío del modelo, debido a su baja representación y a sus geometrías delgadas y fragmentadas, características que dificultan su aprendizaje dentro de arquitecturas convolucionales. En el entrenamiento sin balanceo, el modelo prácticamente no logra identificar esta clase, obteniendo valores de F1 cercanos a cero. La incorporación de pesos por frecuencia inversa introduce una mejora sustancial, evidenciando que el modelo requiere un refuerzo explícito para atender objetos poco frecuentes.
iv. El uso de pesos sin control produjo una sobrerrepresentación artificial de las clases minoritarias, haciendo que

el modelo intentara detectar muros donde no los había, fragmentando la segmentación e introduciendo errores en clases mayoritarias como construcciones y vegetación. Este comportamiento demuestra que, aunque el balanceo de clases es fundamental, su aplicación sin mecanismos de moderación puede desestabilizar el aprendizaje.

v. El uso de pesos normalizados evita la inestabilidad observada en entrenamientos con pesos por frecuencia inversa sin control. La Configuración 3 demuestra que la normalización dentro de un rango acotado permite preservar la estabilidad del entrenamiento y mejorar el equilibrio entre clases dominantes y minoritarias.

vi. Las métricas que evidenciaron mayor sensibilidad al balanceo fueron el IoU y el F1-score, especialmente en clases minoritarias. El mIoU mostro incrementos sustanciales, reflejando una mejora real en la discriminación de todas las clases, no solo de las mayoritarias. El uso de métricas balanceadas es fundamental, ya que la OA puede ser engañosa en escenarios desbalanceados.

## 6 Recomendaciones

i. Se recomienda implementar mecanismos de parada temprana (early stopping) en futuros entrenamientos con el fin de identificar el número óptimo de épocas y evitar el sobreajuste. Si bien en este estudio el entrenamiento extendido a 50 épocas produjo mejoras claras en la segmentación de clases minoritarias, puede que existan puntos intermedios donde el modelo alcance su mejor balance entre generalización y estabilidad, o que un mayor número de épocas continúe mejorando la capacidad del modelo para aprender patrones de clases delgadas y fragmentadas, como muros.

ii. Si bien las estrategias de balanceo de clases y el incremento en el número de épocas mejoraron la segmentación de muros, los valores obtenidos siguen siendo considerablemente menores en comparación con las clases dominantes. Esto evidencia que, aun con ajustes en la función de pérdida, la segmentación de objetos estrechos y de baja representación espacial aún está lejos de alcanzar un desempeño satisfactorio. Por ello, se requiere profundizar en la investigación e incorporar métodos más avanzados como arquitecturas orientadas a objetos pequeños, mecanismos de atención, fusión multimodal o pérdidas específicas para bordes, con el fin de mejorar la recuperación geométrica y semántica de este tipo de elementos urbanos.

iii. Evaluar estrategias de segmentación post-procesada, como CRFs (Conditional Random Fields), filtros morfológicos o refinamiento basado en bordes, que podrían ayudar a mejorar la continuidad y la calidad geométrica de los muros y vías estrechas.

## 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.

## 8 Referencias

Alcover-Couso, R., Escudero-Viñolo, M., SanMiguel, J. C., & Bescós, J. (2024). *Gradient-based Class Weighting for Unsupervised Domain Adaptation in Dense Prediction Visual Tasks*. http://arxiv.org/abs/2407.01327

Audebert, N., Saux, B. Le, & Lefèvre, S. (2016). *Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks*. http://arxiv.org/abs/1609.06846

Audebert, N., Saux, B. Le, & Lefèvre, S. (2017). *Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks*. http://arxiv.org/abs/1711.08681

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

Chen, P., Liu, Y., Ren, Y., Zhang, B., & Zhao, Y. (2025). A Deep Learning-Based Solution to the Class Imbalance Problem in High-Resolution Land Cover Classification. *Remote Sensing*, *17*(11), 1845. https://doi.org/10.3390/rs17111845

DNP. (2023). *Con Inteligencia Artificial, el DNP apoya el proceso de actualización del Catastro Multipropósito*.

Dong, R., Pan, X., & Li, F. (2019). DenseU-Net-Based Semantic Segmentation of Small Objects in Urban Remote Sensing Images. *IEEE Access*, *7*, 65347–65356. https://doi.org/10.1109/ACCESS.2019.2917952

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

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

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*.

Ghanaei, Z., & Rouhani, M. (2025). A context aware multiclass loss function for semantic segmentation with a focus on intricate areas and class imbalances. *Scientific Reports*, *15*(1), 26279. https://doi.org/10.1038/s41598-025-08234-5

Google Developers. (2025a). *Classification: Accuracy, recall, precision, and related metrics*. Https://Developers.Google.Com/Machine-Learning/Crash-Course/Classification/Accuracy-Precision-Recall.

Google Developers. (2025b). *Thresholds and the confusion matrix*. Https://Developers.Google.Com/Machine-Learning/Crash-Course/Classification/Thresholding.

IGAC. (2023). *Catálogo de Objetos Geográficos de la Cartografía Básica Oficial de la República de Colombia. Instituto Geográfico Agustín Codazzi, Subdirección Cartográfica y Geodésica*.

Lightning-AI. (2025). *Mean Intersection over Union (mIoU)*. Https://Lightning.Ai/Docs/Torchmetrics/Stable/Segmentation/Mean_iou.Html.

Mahmoudi, S., Asghari, O., & Boisvert, J. (2025). Addressing class imbalance in micro-CT image segmentation: A modified U-Net model with pixel-level class weighting. *Computers & Geosciences*, *196*, 105853. https://doi.org/10.1016/j.cageo.2025.105853

Mohanan, J. (2024). *How to use Cohen's Kappa Statistic for ML Model verification. Medium*. Https://Medium.Com/@jayamohanmohanan/How-to-Use-Cohens-Kappa-

Statistic-for-Ml-Model-Verification-6c66258b4ae9.

Pan, Y., Xie, F., & Zhao, H. (2023). Understanding the Challenges When 3D Semantic Segmentation Faces Class Imbalanced and OOD Data. *IEEE Transactions on Intelligent Transportation Systems*, *24*(7), 6955–6970. https://doi.org/10.1109/TITS.2023.3256442

Sang, S., Zhou, Y., Islam, M. T., & Xing, L. (2023). Small-Object Sensitive Segmentation Using Across Feature Map Attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(5), 6289–6306. https://doi.org/10.1109/TPAMI.2022.3211171

Saputra, A. K., Erlangga, T. , Tanjung, T., Ariani, F., Aprilinda, Y., & Endra, R. Y. (2024). Review of deep learning using convolutional neural network model. *Engineering Headway*.

Tian, J., Mithun, N. C., Seymour, Z., Chiu, H.-P., & Kira, Z. (2022). Striking the Right Balance: Recall Loss for Semantic Segmentation. *2022 International Conference on Robotics and Automation (ICRA)*, 5063–5069. https://doi.org/10.1109/ICRA46639.2022.9811702