

**Sistema inteligente de detección de asentamientos humanos
informales en el Municipio de Neiva Huila empleando
Aprendizaje Profundo.**

**MAGÍSTER EN MATEMÁTICAS APLICADAS Y
CIENCIAS DE LA COMPUTACIÓN**

Jorge Esneider Henao González

Héctor Leandro Rojas Serrano

Dirección:

Cesar Augusto Salazar Centeno

Universidad del Rosario

Escuela de Ingeniería, Ciencia y Tecnología

Maestría en Matemática Aplicada y Ciencias de la Computación

DEDICATORIA

Héctor Rojas,

A mi hijo y mi esposa, quienes me motivan cada día a crecer y amar.

Jorge Henao,

A mi esposa, por su incondicional amor y apoyo, a mi madre por sus oraciones, a esas dos mujeres quienes me motivan cada día a continuar con las metas.

AGRADECIMIENTOS

Agradecemos de forma especial a nuestro tutor, Cesar Augusto Salazar Centeno, por su gran apoyo en este proceso de investigación, su experiencia y calidad fueron determinantes en cada etapa.

Damos también un extenso agradecimiento a la Universidad del Rosario por todo el proceso de formación en esta maestría, a los docentes que nos acompañaron y compartieron su conocimiento y a todo el equipo académico de la Escuela de Ingeniería, Ciencia y Tecnología.

RESUMEN

Los asentamientos informales en Colombia son una problemática latente que requiere de continuo control y verificación por parte de los entes territoriales, en esta investigación, enfocada en el municipio de Neiva Huila, dicho proceso lleva décadas manifestándose de diferentes formas y su dinámica obedece a múltiples factores como el político, social y ambiental[1]. A pesar de los esfuerzos legislativos, como lo define la ley 388 de 1997 que busca proveer a los municipios de mecanismos apropiados para la correcta administración y gestión del territorio, la realidad muestra que aún existen muchos aspectos que intervenir. Los procesos de reconocimiento pueden llegar a representar desafíos en la administración pública, desde sus orígenes en la modernidad colombiana, los asentamientos se caracterizan por albergar población vulnerable, donde, la labor del Estado es insuficiente [2]. Sumado a ello, los problemas socioeconómicos y ambientales se ciernen sobre estas poblaciones representando un proceso complejo que requiere de atención especializada[3].

Esta investigación presenta en primera medida una descripción del estado actual de los asentamientos informales en Colombia. En el marco teórico, se hará una revisión de la literatura en cuanto a la evolución de los procesos y metodologías de clasificación de imágenes, así como la aplicación de casos alrededor del mundo en la detección de asentamiento informales. Además, se incluirá algunos trabajos relacionados a la clasificación y detección de zonas geográficas y trabajos de investigación en otras problemáticas que fueron útiles para el desarrollo de la investigación.

Finalmente, este trabajo presenta un sistema inteligente para la detección y clasificación de asentamiento informales para el municipio de Neiva, Huila, utilizando técnicas de aprendizaje por transferencia o (transfer learning), donde este recurso puede convertirse en un recurso valioso para las entidades dedicadas a esta problemática, ofreciendo un método ágil y eficaz para la identificación de dichos territorios.

Informal settlements in Colombia are a latent problem that requires continuous control and verification by territorial entities. In this research, focused on the municipality of Neiva Huila, this process has been manifesting itself in different ways for decades and its dynamics obey multiple factors, such as political, social, and environmental [1]. Despite the legislative efforts, as defined by Law 388 of 1997, which aims to provide municipalities with appropriate mechanisms for the proper administration and management of the territory, reality shows that there are still many aspects to address. The process of recognition can represent challenges in public administration, and from their origins in Colombian modernity, settlements are characterized by housing vulnerable populations, where the government and its mechanism are insufficient [2]. In addition to this, socioeconomic and environmental problems loom over these populations, representing a complex process that requires specialized attention [3].

This research first presents a description of the current state of informal settlements in Colombia. In the theoretical framework, a review of the literature about the evolution of image classification processes and methodologies will be described, as well as the application of cases around the world in the detection of informal settlements. Additionally, some works related to the classification and detection of geographical areas and research work on other problems that were useful for the development of the research will be included.

Finally, this work presents an intelligent system for the detection and classification of informal settlements for the municipality of Neiva, Huila, using transfer learning techniques, where this resource can become a valuable resource for entities dedicated to this problem, offering an agile and effective method for the identification of these territories.

Keywords

Deep Learning – Informal Settlement Detection – Transfer Learning – Neiva - Colombia

TABLA DE CONTENIDO

vi

Capítulo 1 JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA.....	1
Capítulo 2 OBJETIVOS	4
1.1 Objetivo general.....	4
1.2 Objetivos específicos	4
Capítulo 3 MARCO TEÓRICO Y ESTADO DEL ARTE.....	5
Capítulo 4 METODOLOGÍA Y MÉTODOS	18
1.1 Generación de la base de datos:	19
2.1 Obtención de imágenes por medio de Google Earth Pro.....	23
3.1 Preprocesamiento.....	24
4.1 Etiquetado usando Labeling.....	25
5.1 Creación archivo (.YAML).....	28
6.1 Implementación de YOLOV8 usando Python	28
7.1 Mejor modelo con archivos .PT usando Pytorch.....	30
8.1 Construcción Aplicativo Web con Django	30
9.1 Métricas de Desempeño.....	34
Matriz de Confusión	34
Curva F1 Confidence o Puntuación F1	35
Precisión.....	35
Sensibilidad o Recall.....	35
Curva Precision Recall.....	36
mAP50-95.....	37
Graficas de Perdida (Box Loss)	37
Graficas de Perdida de clase (Cls Loss).....	37
Perdida de Distribución Focal (DFL Loss).....	37
Capítulo 5 RESULTADOS Y DISCUSIÓN	38
10.1 Evaluación del modelo.....	40
Matriz de Confusión	40
Curva F1 Confidence Modelo.....	41
Curva Precision-Recall Modelo.....	42
Evaluación de pérdidas y rendimiento del Modelo Uno.....	43
Evaluación Modelo entorno web	45
Capítulo 6 CONCLUSIONES Y TRABAJO FUTURO	48
REFERENCIAS.....	51
APÉNDICE.....	56
11.1 Diseño Experimental – Resultados	56
12.1 Aplicativo Web	61

LISTA DE FIGURAS

Figura 1: Arquitectura de una red neuronal convolucional LeNet-5 [6]	6
Figura 2 : Arquitectura de la red AlexNet [7].....	8
Figura 3: Arquitectura YOLOV8 [33].....	16
Figura 4: Diagrama Metodología y Métodos.....	19
Figura 5: Clasificación del Suelo Neiva POT [39].....	21
Figura 6: Plano General Asentamiento Neiva [40].....	22
Figura 7: Obtención de imágenes usando Google Earth Pro	23
Figura 8: Imagen preprocesada – ejemplo para ilustración.	25
Figura 9: Etiquetado usando Labelimg	26
Figura 10: construcción de los conjuntos de datos para el modelo.....	27
Figura 11: estructura del archivo txt producto del etiquetado	27
Figura 12: Creación del archivo (.YAML).	28
Figura 13: Modelos YOLO disponibles para implementación [44]	29
Figura 14: Entrenamiento del Modelo YOLOV8X en Python	29
Figura 15: Carpeta raíz del aplicativo web	31
Figura 16: Home.html.....	32
Figura 17: Vista General Aplicativo Web.	33
Figura 18: Función Home	33
Figura 19: Matriz de confusión para clasificación binaria[48].....	34
Figura 20: Ecuación curva F1 [49]	35
Figura 21: Precisión [49]	35
Figura 22: Sensibilidad (Recall) [27].....	36
Figura 23: Curva PR de ejemplo [50].....	36
Figura 24: Argumentos en YOLOV8	38
Figura 25: Matriz de Confusión.....	40
Figura 26: Curva F1 Confidence Modelo	41
Figura 27: Curva Precision-Recall Modelo	42
Figura 28: Gráficas de perdida y métricas de rendimiento modelo uno.	43
Figura 29: Entorno Web creado en Django por los autores.....	45
Figura 30: Imagen de prueba Cartagena, Colombia.	46
Figura 31: Imagen de prueba Medellín, Colombia.	46
Figura 32: Salida de entrenamiento Modelo Uno	56
Figura 33: Precision Recall Modelo Dos.....	57
Figura 34: Evaluación de pérdidas y rendimiento del Modelo Dos.....	57
Figura 35: Matriz de confusión Modelo Dos.....	58
Figura 36: Salida de entrenamiento Modelo Dos	58
Figura 37: Evaluación de pérdidas y rendimiento del Modelo Tres	59
Figura 38: Salida de entrenamiento Modelo Tres.....	59
Figura 39: Precision Recall Modelo Cuatro.....	60
Figura 40: Evaluación de pérdidas y rendimiento del Modelo Cuatro	60
Figura 41: Salida de entrenamiento Modelo Cuatro	61
Figura 42: Caso de Uso del Aplicativo en Django	61

LISTA DE TABLAS

Tabla 1: Modelos – Diseño Experimental YOLOV8	39
Tabla 2: Caso de prueba Aplicativo en Django.	62
Tabla 3: Test de Usabilidad basada en Heurística de Jakob Nielsen [54]	63

Capítulo 1

JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

En Colombia, la falta de actualización de los procesos de ordenamiento territorial son una problemática que permite el crecimiento de los territorios de forma desordenada y sin parámetros de uso de ocupación que eviten el desarrollo de urbanismos en lugares que no son aptos para la vida de las personas, tal es la situación que de acuerdo con el Instituto Geográfico Agustín Codazzi – IGAC el 80% [5] de los municipios tienen desactualizado su Plan de Ordenamiento Territorial, siendo este el instrumento facilitador para la expansión de la sociedad buscando la reducción de los daños o perjuicios generados por este mismo (DNP, 2014) [4] por tanto, es relevante y de vital importancia su continua actualización e implementación para así mitigar los impactos en las regiones de Colombia. Sin embargo, lo que agrava la aplicabilidad de estos planes y su real estado de conformidad con los objetivos de los departamentos y municipios frente al plan de desarrollo; razón por la cual el legislador tuvo la necesidad de sancionar la Ley 388 de 1997, misma que promueve la eficiencia en el ordenamiento del territorio y el aprovechamiento de los recursos naturales y urbanos de una manera óptima facilitando su desarrollo del entorno.

En este sentido, una de las problemáticas que se desprenden de los procesos desactualizados del ordenamiento del territorio, son los asentamientos humanos informales, debido a que día a día crecen de manera potencial en los diferentes territorios, haciendo ocupaciones indebidas en los mismos que ponen en riesgo la vida de quienes residen en dichos lugares, así como también generan desequilibrios ambientales, según lo indica Departamento Administrativo de Estadística (DANE), a través del indicador ODS [7] estima para un conjunto de principales ciudades la proporción de asentamientos informales, encontrando que el 11.4% de la población en Colombia habita estas zonas de riesgo, situaciones que han generado la necesidad de desarrollar acciones que permitan la aplicabilidad de políticas y programas que ayudan a mitigar este flagelo, sin embargo,

pueden verse ralentizados por la misma naturaleza de los asentamientos dadas sus condiciones de acceso a los instrumentos o ayudas estatales,

Lo anteriormente expuesto es identificado y analizado luego de investigar los primeros casos documentados de los asentamientos informales, [2] analizando los orígenes de los mismos en Colombia, centrando su análisis en la ciudad de Cali, siendo esta ciudad uno de los primeros lugares en Colombia donde se documentan estos procesos, para luego continuar con estudios que abordan esta problemática y que analizan los mismos comportamientos en diferentes latitudes. Dentro de estos, es importante mencionar el caso de [6] Ituango, Antioquia, lugar donde se identifica un déficit importante en los programas gubernamentales de ayuda de la población afectada que vive en la informalidad, reforzado en el aumento de la población migrante y unas demandas cada vez más complejas hacia el gobierno, y en contraparte su crecimiento va ligado a un incremento de las acciones delictivas reforzando el actuar de las organizaciones al margen de la ley.

Asimismo, dentro de los efectos de este fenómeno [3] se encontró para el caso del municipio de Soledad, Atlántico, la pobreza monetaria y las desigualdades socioculturales como los principales detonantes en el crecimiento de las zonas informales, además, se observa que, los daños en zonas naturales protegidas y deterioro ambiental como factor crítico.

Motivado por la magnitud de este problema en Colombia, en particular, al municipio de Neiva, Huila, este estudio se alinea con las problemáticas planteadas en el Plan de Ordenamiento Territorial del territorio, específicamente el Acuerdo 026 del 2009 [8], el cual está en proceso de revisión general y se reconoce como una problemática a abordar y controlar, fundamentada en la política de distribución y uso de suelo donde la ubicación y reubicación de personas que habitan asentamientos informales hacia zonas de expansión legalizadas, o, si el asentamiento no representa riesgo y no se encuentra en zona protegida, pueden existir acciones enfocadas a mejorar de manera integral la habitabilidad de estas zonas o una reubicación a zonas cercanas. Estos propósitos y actividades dentro

del acuerdos son parte vital para la correcta ejecución y gestión de los recursos destinados a esta problemática. Esta investigación se propone recolectar imágenes satelitales del municipio mencionado y diseñar, desarrollar e implementar un sistema inteligente de detección de asentamientos humanos informales. Dicho sistema estará basado en la clasificación expuesta en la Ley 388 de 1997 Capítulo 4, Artículo 30 al 35 (suelo urbano, suelo rural y suelo de protección).

Este esfuerzo se enmarca en un contexto donde la aplicación de aprendizaje profundo y el aprendizaje por transferencia (transfer learning) en diferentes países utilizando diferentes metodologías y aproximaciones, ha demostrado ser prometedora para el tratamiento de imágenes satelitales, la clasificación de suelos, la detección y clasificación de asentamientos humanos informales, entre otros trabajos de investigación [9]–[31].

Capítulo 2

OBJETIVOS

1.1 Objetivo general

Diseñar, desarrollar e implementar un sistema inteligente de detección escalable de asentamientos humanos informales en el Municipio de Neiva Huila desarrollando una aplicación basada en Aprendizaje Profundo, como herramienta de soporte al Plan de Ordenamiento Territorial.

1.2 Objetivos específicos

Generar un banco de imágenes diversificado y adecuado como base de datos que permita el entrenamiento, la validación y la evaluación del sistema de detección inteligente.

Implementar un modelo de reconocimiento de objetos para mejorar la eficiencia en la identificación de asentamientos informales.

Evaluar los resultados obtenidos mediante métricas que logren contemplar su capacidad de detección y clasificación.

Implementar un entorno web amigable y accesible para el sistema inteligente de detección y clasificación de uso en las entidades territoriales y que pueda ser un insumo en la evaluación de la problemática de los asentamientos humanos informales.

Capítulo 3

MARCO TEÓRICO Y ESTADO DEL ARTE

En el Departamento del Huila, históricamente las cuencas de los ríos y sus alrededores han sido objeto del surgimiento de asentamientos informales desde hace décadas y surgen como una respuesta a los fenómenos demográficos que han sido reforzados por eventos como el desplazamiento interno forzado, la migración, entre otros. El estado tiene un papel relevante en este asunto, ya que su falta de objetividad en asuntos fuera del marco legal lo hacen un actor sin gestión dado que su actuar se basa en el derecho y el concepto de ilegalidad restringe su actuación tal como lo plantea Trujillo Quintero (2022) [1]. En Este contexto de complejidad, la aplicación de las redes neuronales convolucionales es un tipo de red neuronal artificial utilizada en aprendizaje profundo para analizar diferentes problemas y ofrece una solución que gracias a las capacidades de procesamiento en el mundo actual son un recurso óptimo y efectivo frente a distintos escenarios donde los datos pueden ser enormes, es en este sentido y problema donde la medición y clasificación de estas variables puede ser soportada por modelos de aprendizaje profundo que eficientemente pueden generar resultados más efectivos para el desarrollo de las políticas regionales.

En cuanto a los orígenes de las redes neuronales convolucionales, es importante hablar del trabajo de Fukushima en 1980 [4] es uno de los primeros precursores del uso de estas para el procesamiento de imágenes, su enfoque, parte de emular el reconocimiento de patrones en los seres humanos y como este comportamiento puede ser aplicado e implementado en una red neuronal. Un elemento interesante es la aplicación de resultados obtenidos en pruebas por Hubel y Wiesel mencionado por Gross [5] con chimpancés y como el reconocimiento de imágenes no se veía afectado por la posición de esta, su respuesta era similar, y teniendo este concepto validado con pruebas hechas en animales su aplicación en la red neuronal que reconozca el patrón sin importar su posición agregando además la posibilidad de identificarlo así este contenga distorsiones que afecten su

integridad. Su trabajo introduce el término “Neocognitron” siendo como el mismo autor lo menciona una extensión a trabajos previos. Respecto a la implementación de la red y su construcción se implementó para reconocer números en diferentes posiciones, su estructura se basó en el sistema nervioso donde las células “S” o células simples y células complejas “C” de tal forma que su configuración y posición ayudan a que el reconocimiento de patrones enfoque e identifique la forma sin importar la posición o ángulo de este.

Luego de estos avances aparecen nuevas arquitecturas y propuestas que buscan emular el funcionamiento del cerebro humano. En el año 1998 apareció una nueva red neuronal convolucional que impacto a la comunidad, esta red llamada LeNet-5 [6] la cual acuñó el término de “Red Neuronal Convolucional” como respuesta a las nuevas necesidades del mundo actual y el acompañamiento y ventajas ofrecidas por las nuevas tecnologías que permiten mayor procesamiento. La arquitectura LeNet-5 propuesta (Figura 1) consta de 7 capas que reciben parámetros de entrenamiento, en esta arquitectura que contiene 3 capas convolucionales, 2 capas “Max Pooling” o de submuestras, seguido de dos capas totalmente conectadas terminando en una séptima capa de salida el procesamiento de píxeles blancos y negros se hace asignando valores normalizados donde el blanco corresponde a -0.1 y el negro será 1.175, esto ayuda a que el procesamiento sea más veloz, ya que su varianza llegará o estará cerca de 1.

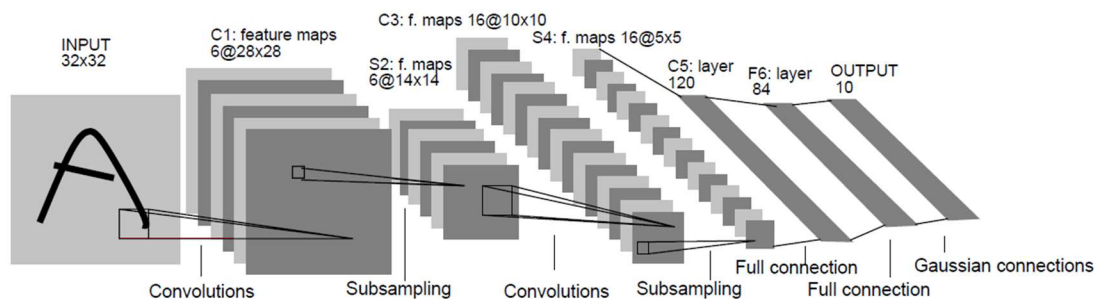


Figura 1: Arquitectura de una red neuronal convolucional LeNet-5 [6]

Respecto a los resultados obtenidos, se utilizó la base de datos de números escritos a mano NIST por sus siglas en inglés, sin embargo, al ser modificada y potenciada por los autores, recibe el nombre de NIST modificada o MNIST, y los resultados obtenidos mejoraron notoriamente al aumentar el número de datos y adicionalmente un punto importante en esta investigación es la comparación frente a otros métodos de forma específica midiendo su tasa de error entre ellos (K vecinos cercanos, lineal, máquina de vectores soporte, análisis de componentes principales (PCA), entre otros). Adicionalmente, los autores revisan escritos o palabras que carecen de simetría y evalúan la red LeNet-5 con un componente que usa heurísticas llamadas red neuronal de desplazamiento espacial SDNN, donde este ofrece o muestra ser un optimizador clave para su uso en las redes neuronales convolucionales.

Hablando del siglo XXI y sus avances en esta materia, en el año 2012 [7] como solución a un reto llamado ImageNet apareció la arquitectura AlexNet (Figura 2), dichos autores desarrollaron una nueva arquitectura para procesamiento de imágenes donde lograron obtener un desempeño considerable empleando abandono o conocido normalmente como “Dropout” con una arquitectura en una red de 8 capas, 5 capas convolucionales y 3 capas totalmente conectadas, dentro de la discusión de los resultados obtenidos y su metodología es importante ver que este tipo de arquitecturas utilizan alargamiento de imágenes y abandono de ciertas zonas de la red para lograr una precisión elevada, no obstante, como lo discuten los autores el solo remover una capa de dicha arquitectura representa una caída en la precisión del modelo, por tanto, es importante comprender la utilidad de cada uno de los elementos en el entrenamiento de redes neuronales convolucionales.

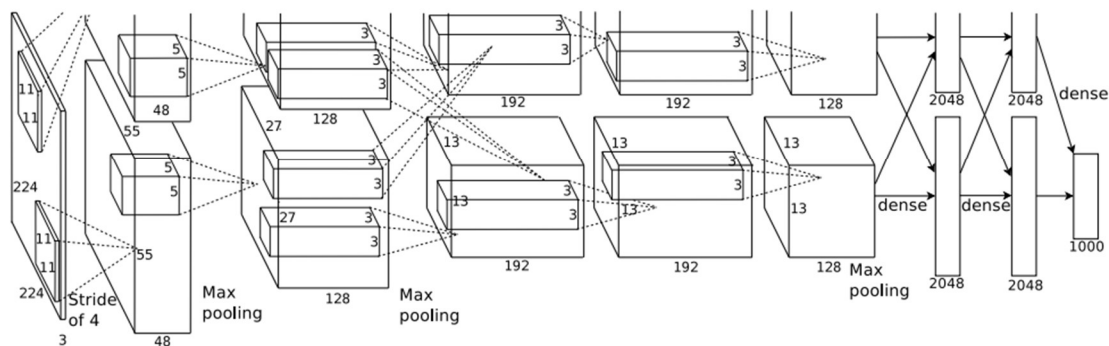


Figura 2 : Arquitectura de la red AlexNet [7]

En el mundo existen estudios relacionados con la utilización de técnicas no solo de redes convolucionales sino de análisis de imágenes basado en objetos (OBIA), técnicas de análisis orientado a objetos (OOA), redes neuronales convolucionales (CNN) con algunos trabajos donde se realizan modificaciones y ajustes de las arquitecturas buscando potenciar los resultados.

Desde un enfoque ontológico, Kohli et al. (2012) [8] analizaron métodos diferentes al aprendizaje profundo utilizando técnicas de análisis orientado a objetos (OOAD), parten de la definición de un asentamiento ilegal analizando sus características principales y evalúan distintos autores y sus hallazgos, su análisis ontológico es una gran herramienta para la correcta clasificación de asentamientos informales y realizan una transformación a los atributos encontrados y lo convierten en un recurso importante para el despliegue de un ambiente orientado a objetos (OOAD). Claramente, su alcance en la actualidad es superado por las redes convolucionales (CNN) y otros modelos de aprendizaje profundo, no obstante, su aproximación a las condiciones y características de los asentamientos informales son de gran valor para la observación y parametrización de estudios dedicados a evaluar dicha problemática, dentro de estos conceptos clave se tiene que la extensión de los asentamientos informales tiende a ser más pequeña que en los asentamientos formales. Respecto a su forma y orientación es asimétrica y poco ordenada, carecen de redes viales

apropiadas, y dentro de sus principales patrones esta que siempre tienden a estar cerca de carreteras principales o puntos de desagüe y alcantarillado, su densidad es más elevada que sitios formales y en cuanto a su ubicación lo usual es encontrarse en lugares no predilectos o que de facto no tiene las condiciones para contener o desarrollar construcciones regulares.

En regiones como Asia, se observa a varios autores y su enfoque sobre los asentamientos informales en diferentes ciudades tal es el caso del estudio en Yakarta Indonesia [9] que sigue líneas alternativas de análisis para la detección de asentamiento informales, en este trabajo es utilizado el análisis de imágenes basado en objetos (OBIA) siendo comparado con clasificación manual de imágenes realizada por expertos, sirviendo de recurso para este estudio se usaron las bases ontológicas para la correcta y apropiada clasificación de asentamientos informales, en este caso se analiza la ciudad de Yakarta Indonesia (distrito de Tebet) y se busca evaluar como el plan de gobierno que argumenta una total cobertura en saneamiento básico en esta ciudad puede ser medido en su desempeño. En este sentido, los autores no encontraron datos totalmente concluyentes y esto se debe a que el manejo de datos manualmente revisados puede tener un sesgo importante, aumentando el número de falsos positivos en sus resultados. En cuanto al uso de (OBIA) y su comparación frente a las redes neuronales convolucionales es importante mencionar que de acuerdo a lo encontrado por [10] el rendimiento y flexibilidad de las redes convolucionales CNN frente a (OBIA) establece una brecha inicial en cuanto a sus ventajas, las redes convolucionales CNN no requieren de tanta observación humana y son más adaptables frente a otros problemas de aplicación lo cual en caso de usar (OBIA) no puede darse con la misma facilidad.

En India [11] abordando esta misma problemática en cuanto a los asentamientos informales en diferentes locaciones se emplean técnicas multi-clase de maquina soporte de vector (SVM) con un componente de bolsa de palabras visuales (BoVW) encontrando para un conjunto de imágenes de zonas urbanas en ciudades de la India que en este modelo dada su metodología las imágenes de menor dimensión se obtiene un mejor desempeño que aquellas zonas donde su área superaba los 100 metros aproximadamente. Otros estudios

que discuten el asunto de los asentamientos informales versus los planes de gobierno se dan como se mencionó anteriormente en Indonesia, ya que son claramente territorios donde el desplazamiento y las construcciones informales tienen un alto impacto. En dicha investigación [12] se utilizaron algoritmos de máquina soporte de vector (SVM) y bosque aleatorio o (random forest) con algunos complementos que refuerzan su desempeño logrando una precisión del 85%, como parte importante de esta investigación se encuentra el componente comparativo de nuevas soluciones basadas en aprendizaje profundo frente a métodos tradicionales de mapeo de asentamientos informales que son más costosos en cuanto al capital humano frente a un modelo de aprendizaje profundo, donde este último, sin dejar a un lado sus características puede tener costos computacionales elevados. Esta comparativa proporciona una buena fuente de comparación de los modelos tradicionales frente a estos nuevos modelos que logran ser de impacto y en términos prácticos pueden ser la solución efectiva para reducir los tiempos de ejecución y obtención de datos en dichos procesos.

En diversas investigaciones centradas en la detección de asentamientos informales, se han empleado conjuntos de imágenes reducidos con resultados destacados. Un ejemplo relevante es el estudio propuesto por Ajami et al. [13] en la ciudad de Bangalore, India. En este caso, una base de entrenamiento de tamaño modesto, compuesta por imágenes verificadas de asentamientos informales, fue suficiente para implementar una red neuronal convolucional (CNN). A pesar de contar únicamente con 1461 imágenes de asentamientos informales en la base de entrenamiento, el modelo logró localizar con buenos resultados los asentamientos en una muestra de tan solo 121 imágenes, permitiendo así la construcción de un índice de privación específico para estas áreas. Estos resultados refuerzan la eficacia de las redes neuronales convolucionales como herramienta útil en estos procesos, incluso cuando se trabaja con conjuntos de datos pequeños.

En los últimos años, asentamientos informales y redes neuronales convolucionales han sido objeto de diversas aproximaciones, como se evidencia en el trabajo de Wurm et al. [14] En este estudio, se aborda la segmentación de barrios de asentamientos informales

en diferentes zonas geográficas mediante el aprendizaje por transferencia en redes neuronales convolucionales (CNN). Se destacan las bondades de utilizar una red preentrenada; no obstante, se subraya que elementos clave como la calidad de la imagen y el tamaño de las bases de imágenes son muy importantes y pueden impactar el desempeño del modelo que se pretende evaluar.

La utilización de redes neuronales convolucionales ha tenido múltiples aplicaciones en el tratamiento de imágenes satelitales, en ocasiones resolviendo cuestionamientos diferentes al objeto de estudio de esta investigación, en ese sentido, en una investigación sobre el uso de algoritmos para clasificar uso y cobertura del suelo [15], los autores cuestionan la efectividad y utilidad de las redes neuronales convolucionales, ya que en su argumento en tareas de clasificación de zonas satelitales su desempeño puede ser menor, para este ejercicio utilizan un algoritmo modificado del algoritmo de optimización de ballenas conocido como (WOA) y en cuanto al proceso de clasificación efectúan esta por medio de una memoria bidireccional a largo plazo o corto plazo (Bi-LSTM) para un conjunto de datos de uso de tierra conocido como (UCM), un conjunto de imágenes aéreas llamado (AID) y el conjunto de imágenes de Google (NWPU). Los resultados obtenidos claramente son concluyentes y se observa que frente a redes neuronales convolucionales tradicionales (CNN) o el mismo algoritmo de optimización de ballenas (WOA), redes neuronales recurrentes (RNN), redes neuronales antagónicas (GAN), entre otras logrando así una precisión por encima del 95%.

Bajo esta misma línea de utilización de modelos de aprendizaje profundo para evaluar o clasificar uso y cobertura del suelo existen aportes interesantes como la propuesta de redes neuronales convolucionales (CNN) general comparada con modelos de K vecinos más cercanos (KNN), bosque aleatorio (random forest) y máquina soporte de vector (SVM) realizado por García et al. En 2019. Para llevar a cabo este análisis los investigadores utilizaron imágenes de hiperespectro (Indian Pines, Pavia University y Salinas) e imágenes de radar JPL Air Star (San Francisco and Flevoland) [16] que comprenden diferentes ambientes tanto rurales como urbanos, en cuanto a los datos de entrenamiento es

interesante observar el método de rotación por partes para las imágenes para permitir al modelo aprender diferentes distribuciones espaciales dentro de los modelos, esto como se argumenta por los autores reduce el sobreajuste del modelo. En cuanto al modelo de la red neuronal convolucional, ellos proponen una red que combina capas convolucionales seguidas de capas (ReLU) y (Max Pooling) transmitiendo dicha configuración a una capa totalmente conectada que termina en una capa (Softmax) que asigna la probabilidad a cada clase. Con estos parámetros, los resultados obtenidos por los autores demuestran la efectividad de las redes neuronales convolucionales (CNN) frente a otros modelos sugeridos de K vecinos más cercanos (KNN), bosque aleatorio (random forest) y máquina soporte de vector (SVM). En este caso aplicado se demostró la estabilidad de las redes neuronales convolucionales (CNN) para el procesamiento de imágenes con diferentes características, logrando obtener resultados sobresalientes de clasificación de tierra.

Los postulados y modelos usando diferentes arquitecturas de aprendizaje profundo para resolver problemas o aportar a la investigación como modelos apropiados abordan por lo general la eficiencia y rapidez de estos, no obstante, también el componente de seguridad juega un papel fundamental en el despliegue de arquitecturas de redes neuronales convolucionales y no debe dejarse a un lado este flagelo que puede impactar el desempeño y comprometer la seguridad de la información. En 2021, [17] esta investigación aborda los riesgos de trabajar modelos de aprendizaje Profundo con imágenes satelitales en la nube y sugieren una mejora de encriptación para el modelo mejorando y reduciendo la exposición de datos sensibles, planteando que, a pesar de que los modelos de aprendizaje profundo para procesamiento de imágenes a gran escala deben ser flexibles no se debe dejar de un lado la seguridad, en donde esta metodología llamada preservación de la privacidad para aprendizaje profundo o PPDLP por sus siglas en inglés es un referente en temas de seguridad para el despliegue y uso de redes neuronales convolucionales. Para lo anterior se utilizó una técnica de cifrado homomórfico (esquema de Paillier) con una arquitectura de tres capas convolucionales, tres capas (Max Pooling), una capa de (Dropout), una capa de aplanamiento o (Flatten), dos capas totalmente conectadas y una función de activación (ReLU, Softmax). Las imágenes que se trabajaron por estos autores provienen del satélite

SPOT con distinta clasificación, entre ellas (urbana, vegetación, suelo y carreteras) buscando un ejercicio de clasificación con el componente adicional de seguridad y encriptado, en ese sentido los resultados para la arquitectura de redes neuronales convolucionales (CNN) encontraron que su rendimiento depende en alta medida de la cantidad de imágenes procesadas, en pequeños conjuntos de datos encontraron desempeños bajos. Respecto a la técnica de cifrado homomórfico no se encontraron deficiencias en el desempeño dado el proceso de encriptación. Lo anterior refuerza el sentido de incorporar componentes de seguridad en dichas arquitecturas cuando se pretenda abordar temas sensibles que requieran de componentes de protección en la información procesada.

Los aportes y aplicaciones de las redes neuronales convolucionales en la evaluación para la clasificación del uso y cobertura del suelo se tienen los ejercicios realizados por Seffrin et al. utilizando combinaciones de arquitecturas memoria a largo plazo o corto plazo (LSTM) junto con redes neuronales convolucionales (CNN) [18] seleccionando zonas geográficas para medir la efectividad de las arquitecturas propuestas, algunos autores también han optado por analizar uso y cobertura de suelo utilizando bandas en imágenes obtenidas por satélite utilizando técnicas de optimización logrando resultados de hasta el 98% de precisión [19].

En Colombia, al igual que el trabajo de Duque, Patiño y Betancourt en 2017 [20] existen algunas aproximaciones de uso del aprendizaje profundo en diferentes problemas. Entre ellos existen, el uso de redes neuronales convolucionales (CNN) en parques nacionales de Colombia [21] para clasificar capas de terreno y poder ser un elemento clave en la evaluación y prevención de estas zonas protegidas. Así mismo, este tipo de modelo ha sido usado para medir la deforestación en zonas del bajo Cauca [22] comparando un modelo de regresión logística con una red neuronal convolucional (CNN) encontrando que esta última tiene un desempeño superior al modelo logístico y siendo un elemento clave para entender que la deforestación está asociada a la cercanía de las vías con dichas zonas ya que es más frecuente encontrar este tipo de eventos en las cercanías, en temas urbanos, puntualmente para la ciudad de Bogotá [23] y con el objetivo de predecir los crecimientos

en la población también se implementó este tipo de red para lograr obtener una modelación de las mallas de población. En la Guajira [24] se ha implementado las redes neuronales convolucionales (CNN) y modelos preentrenados VGG16 para detectar el número de viviendas sin servicio de Luz encontrando para dichos problemas de investigación un alto nivel de precisión, sin embargo, una característica de estos estudios es que se encontraron bajos desempeños de las arquitecturas cuando dentro de las capas se agregaban capas de (Max pooling) pues reduce el tamaño de la imagen lo que va en detrimento de algunas características claves al momento de clasificar.

Continuamente los problemas de investigación y la aplicación de la inteligencia artificial con los modelos de aprendizaje profundo (deep learning) han revolucionado actualmente la forma de abordar estos, los modelos mencionados anteriormente para el procesamiento de imágenes han evolucionado y la mejor manera para sortear los objetivos propuestos es aprovechar sus ventajas y aplicaciones en distintos entornos y como su rendimiento ha sido destacado y utilizados en disciplinas diversas, el aprendizaje transferido o (transfer learning) en su origen es netamente un proceso humano, donde por ejemplo, la forma de reconocer un objeto se puede usar para reconocer otro de forma o textura similar tal como lo plantea [25] este proceso es inherente al ser humano y la implementación en las ciencias de la computación no es ajeno, ya en años anteriores ha recibido varios nombres, entre ellos: “aprendiendo a aprender”, “transferencia de conocimiento” “transferencia inductiva”, entre otros. Su propósito es aminorar la carga y procesos de recolección, etiquetado, entrenamiento que sin lugar a duda sin una arquitectura definida para un investigador abordar dicha tarea puede ser enorme y estos modelos preentrenados se ha visto como su manera de resolver los problemas planteados es eficiente y resuelven en muchos casos el problema de obtener bases de entrenamiento robustas. Tal como es planteado por algunos investigadores, el camino que ofrece el aprendizaje profundo (deep learning) amplía las oportunidades de evaluar diferentes problemas y reduce las barreras al momento de trabajar con la información [26].

La utilización de aprendizaje transferido o (transfer learning) para responder a problemas en la industria ha tenido unos primeros avances en Colombia tal como lo plantea (Arias, 2023) [27] en un despliegue de YOLOV8 para la detección de cascos de seguridad en la industria que arrojo unos resultados satisfactorios frente a otras arquitecturas como EfficientDet y busca su implementación en ambientes de producción para ofrecer una solución efectiva en la detección de los elementos de seguridad para así brindar mejoras en los procesos industriales, estos ejemplos también permiten revisar la adopción de aprendizaje transferido o (transfer learning) y como la aplicación de metodologías de aprendizaje automático pueden ser abordados de forma dinámica sin necesidad de despliegues robustos. Dentro de los últimos trabajos que utilizan esta arquitectura se pueden nombrar un sin número de trabajos, sin embargo, para dejar nota de sus contribuciones en la labor de detección de distintos problemas se puede mencionar a (F. M. Talaat and H. ZainEldin, 2023) [28] que usan estos modelos para la detección de incendios en ciudades inteligentes o abordar problemas de tráfico en ciudades tal como lo plantean [29], análisis de corrientes de resaca en las playas [30] siendo los trabajos de investigación mencionados aplicaciones cercanas a problemas de índole espacial y geográfica.

La arquitectura YOLOV8 [31] es actualmente la herramienta más versátil y dinámica en la detección de objetos, que, hoy en día puede ser usada en soluciones en tiempo real que ofrece alto rendimiento sin afectar el desempeño con un costo de despliegue muy inferior a procesos anteriores como él (OOAD), la construcción de redes neuronales convolucionales, entre otros. Para la consecución de los objetivos propuestos se hará uso de esta arquitectura YOLOV8 utilizando la técnica de aprendizaje transferido o (transfer learning). Un aspecto importante por destacar dentro de esta arquitectura [32] es su uso para optimización y refinación usando nuevas técnicas de entrenamiento lo cual lo dota de múltiples enfoques mejorando su rendimiento, YOLOV8 es actualmente un modelo usado en distintas disciplinas y su evolución no ha dejado de mostrar grandes progresos en la detección de objetos, donde, su flexibilidad ha sido uno de sus factores más relevantes al momento de la adaptabilidad en los problemas de investigación.

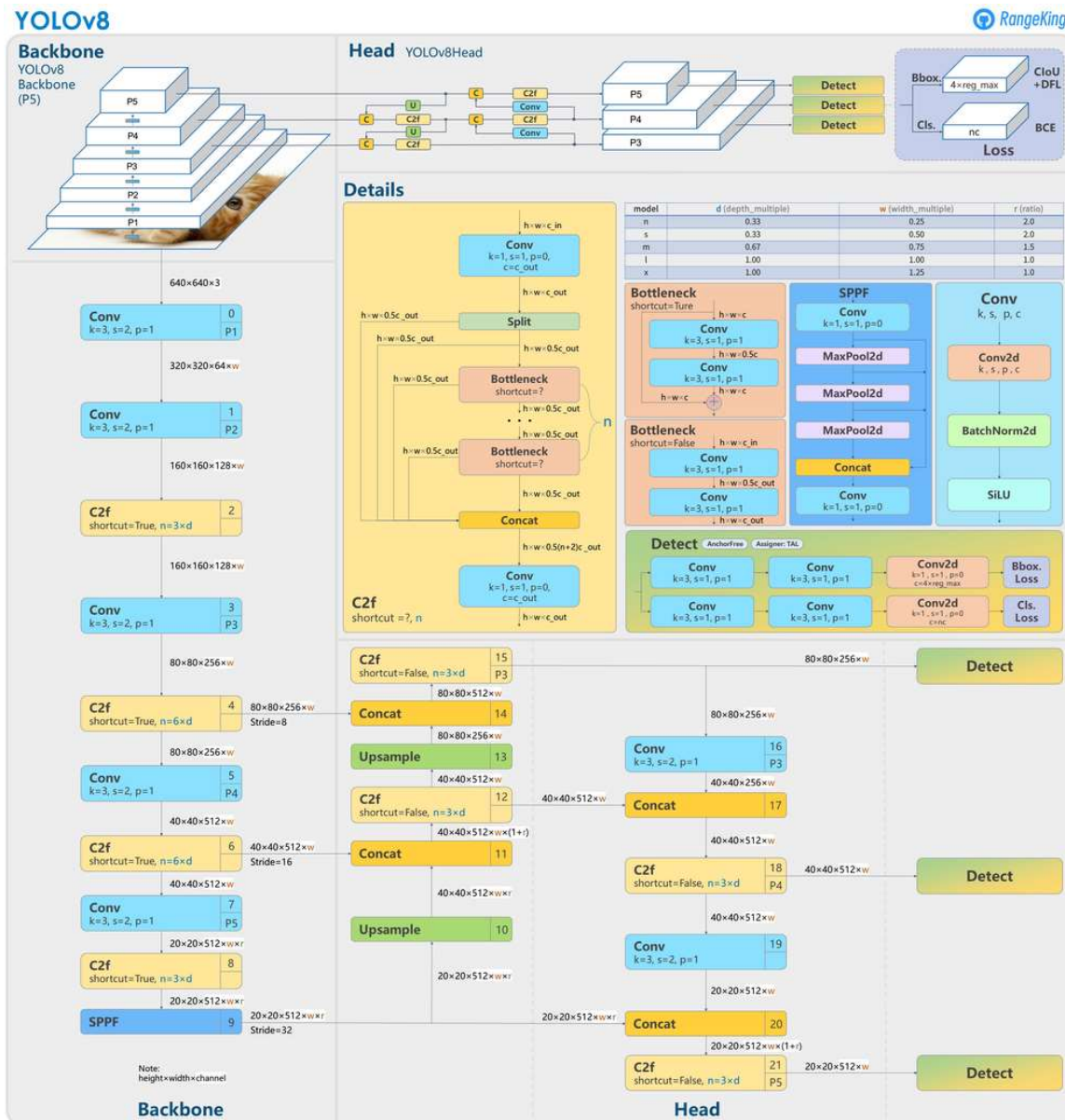


Figura 3: Arquitectura YOLOV8 [33]

La revisión de la literatura a lo largo de esta investigación brinda diferentes puntos de vista y como se mencionó anteriormente, algunos trabajos abordaron problemas de investigación distintos, pero que han sido fundamentales para entender y complementar la

solución al objetivo planteado por los autores, se realizó una revisión sobre los orígenes de las redes neuronales convolucionales y como en la actualidad estos procesos han tenido una evolución exponencial, es importante mencionar que en el transcurso de la investigación se han incluido nuevas referencias que han revolucionado la actualidad en lo que respecta a los sistemas inteligentes de detección de imágenes, donde, actualmente el aprendizaje por transferencia o (transfer learning) ofrecen flexibilidad y aplicación de conocimiento de forma simple para soluciones más rápidas que en términos económicos y de esfuerzo en la construcción de bases de datos [25] ha flexibilizado y potenciado las implementaciones de herramientas que utilizan el aprendizaje por transferencia dejando atrás procesos extensos y prolongados de levantamiento de información para entrenamiento de redes neuronales “caseras” o de modelos tradicionales como las ya mencionadas[26], [34].

Los asentamientos informales en Colombia tienen un origen profundo en el estado y su incapacidad de abordar los temas humanitarios más graves [2] y que en la actualidad se suma el factor migratorio, siendo Colombia el país que más recibe migrantes en la región y donde los organismo del estado tienen mayor necesidad de atención a estas comunidades [35] la literatura sobre la aplicación de redes neuronal convolucional (CNN) en la detección es casi nula, sin embargo, el trabajo realizado por [20] en la ciudad de Medellín son una aproximación a la identificación del problema que evalúa el uso de métodos de bajo costo para la detección de asentamiento informales, respecto a otras investigaciones no existen referentes o guías para la utilización de inteligencia artificial en la detección de asentamiento informales en Colombia , esta investigación, en el municipio de Neiva, Huila, busca proponer un sistema inteligente en la detección y clasificación de asentamiento informales que sea pionero en el país y sirva de insumo en los programas de atención municipal.

Capítulo 4

METODOLOGÍA Y MÉTODOS

Teniendo en cuenta el objeto de estudio planteado en el presente trabajo investigativo, se considera que el enfoque epistemológico más eficiente, pertinente y conducente para desarrollar la presente investigación es el de carácter cualitativa-aplicada, debido a que busca a través del análisis e interpretación de datos obtenidos, buscar una estrategia que permita poner su objetivo en práctica para favorecer a las personas.

Tratándose de una investigación cualitativa, se deben incluir dos aspectos inherentes a esta: la recolección de los datos se fundamenta en la medición de variables, dicha recolección o medición se lleva a cabo al utilizar procedimientos estandarizados y debido a que los datos son producto de mediciones, se representan mediante números que se deben analizar a través de métodos estadísticos. [36]

En este orden de ideas, el componente cuantitativo del estudio integra un número considerable de datos que son analizados a través de mediciones que se transforman en valores numéricos (datos cuantificables), que finalmente se analizan por medio de la estadística.

El componente de la investigación aplicada o tecnológica estaría dado porque con base a la investigación básica, pura o fundamental en las ciencias fácticas o formales se formulan problemas o hipótesis de trabajo para resolver los problemas de la vida productiva de la sociedad, pero su producto no es un conocimiento puro, sino tecnológico. [37]

Para lograr lo anterior se definirán el tipo de datos que se deben recolectar, la herramienta que me permitirá obtenerlos, el método de recolección de la información y su procedimiento de evaluación, así como el posterior producto obtenido, mediante los siguientes pasos:

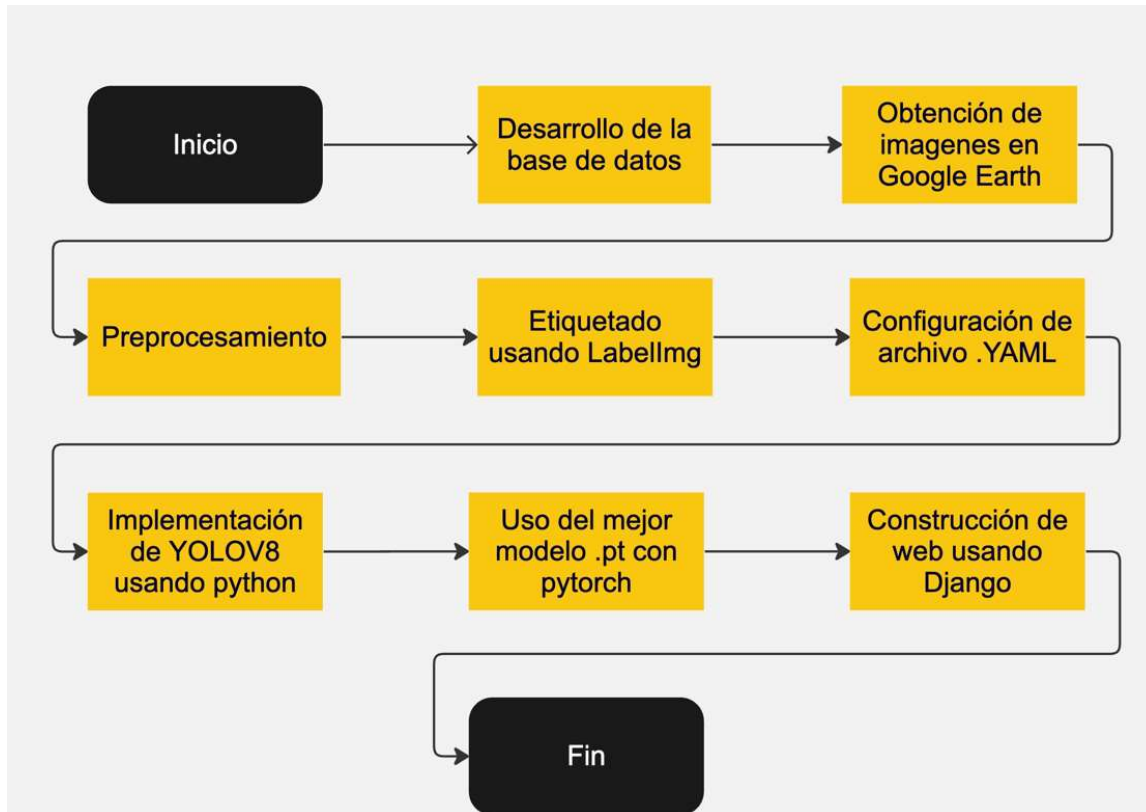


Figura 4: Diagrama Metodología y Métodos

1.1 Generación de la base de datos:

En primera instancia, para generar la base de datos se expondrá los fundamentos legales y metodológicos para la obtención del banco de imágenes y su posterior preprocesamiento para su implementación en el modelo usando aprendizaje por transferencia o (transfer learning).

Contexto Normativo

Basados en la ley 388 de 1997, capítulo IV, que establece la clasificación del suelo en Colombia, y para esta investigación para el uso del suelo del municipio de Neiva Huila, permite comprender la clasificación a usar en el modelo a desarrollar, se soportará la recolección de datos con la información de la dirección de legalización de asentamientos

de la Alcaldía Municipal, que ofrece una fuente cartográfica para comprender a fondo la problemática existente.

En el desarrollo del modelo de clasificación del suelo para el municipio de Neiva, Huila, se ha optado por centrarse en tres categorías claramente definidas por la Ley 388 de 1997: suelo urbano, suelo rural y suelo de protección. Esta decisión se fundamenta en la necesidad de contar con parámetros bien delimitados que permitan al modelo operar con la mayor precisión posible.

El suelo suburbano, según la definición legal, presenta una naturaleza híbrida que combina características del suelo urbano y rural. Esta mezcla de atributos puede inducir a ambigüedades en la clasificación automática, dado que el modelo podría requerir criterios adicionales y más complejos para diferenciar con exactitud entre usos mixtos, lo cual excede las capacidades actuales de nuestro sistema. En este sentido, se encontró disparidad y posibles limitaciones para implementar dichas condiciones en el modelo planteado gracias a la presencia en muchos casos de voluntades políticas que no hacen parte de esta investigación [38].

Por otro lado, el suelo de expansión urbana implica una proyección de crecimiento y desarrollo urbanístico que está sujeta a cambios y condiciones futuras, lo que introduce un grado de incertidumbre y variabilidad que nuestro modelo actual no está diseñado para resolver. Al concentrarse en suelos con regulaciones más establecidas y uniformes, se asegura que el sistema mantenga altos estándares de fiabilidad y aplicabilidad.

Por estas razones, la presente investigación se enfoca en la clasificación de suelo urbano, rural y de protección. Se reconoce la relevancia del suelo suburbano y de expansión urbana en la planificación territorial y se considera su inclusión en futuras mejoras del modelo, una vez que se disponga de la metodología y tecnología necesarias para abordar la complejidad que estos representan.

Establecida la cartografía existente ya para las revisiones del Plan de Ordenamiento Territorial, para la ciudad de Neiva Huila se identificaron las áreas establecidas como suelo urbano, suelo rural y suelo de protección, las cuales nos permitirá referenciar la ubicación de los asentamientos existentes.

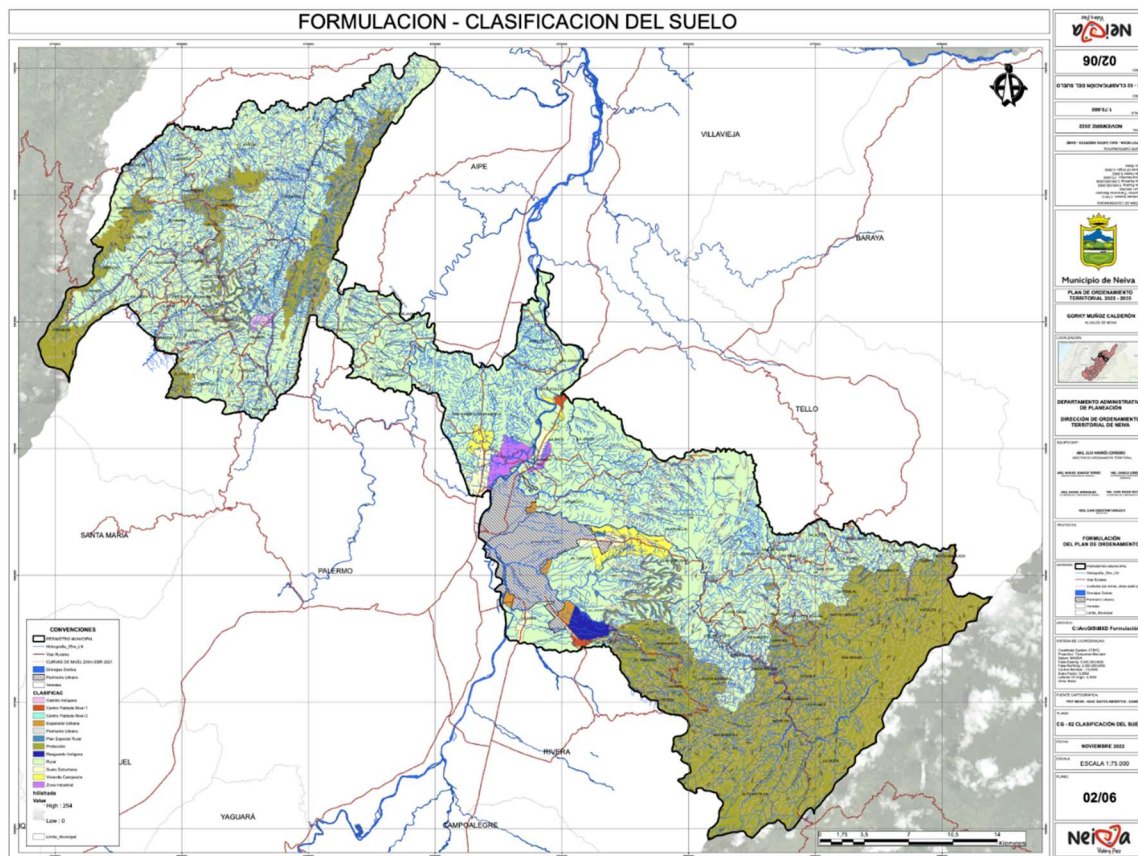


Figura 5: Clasificación del Suelo Neiva POT [39]

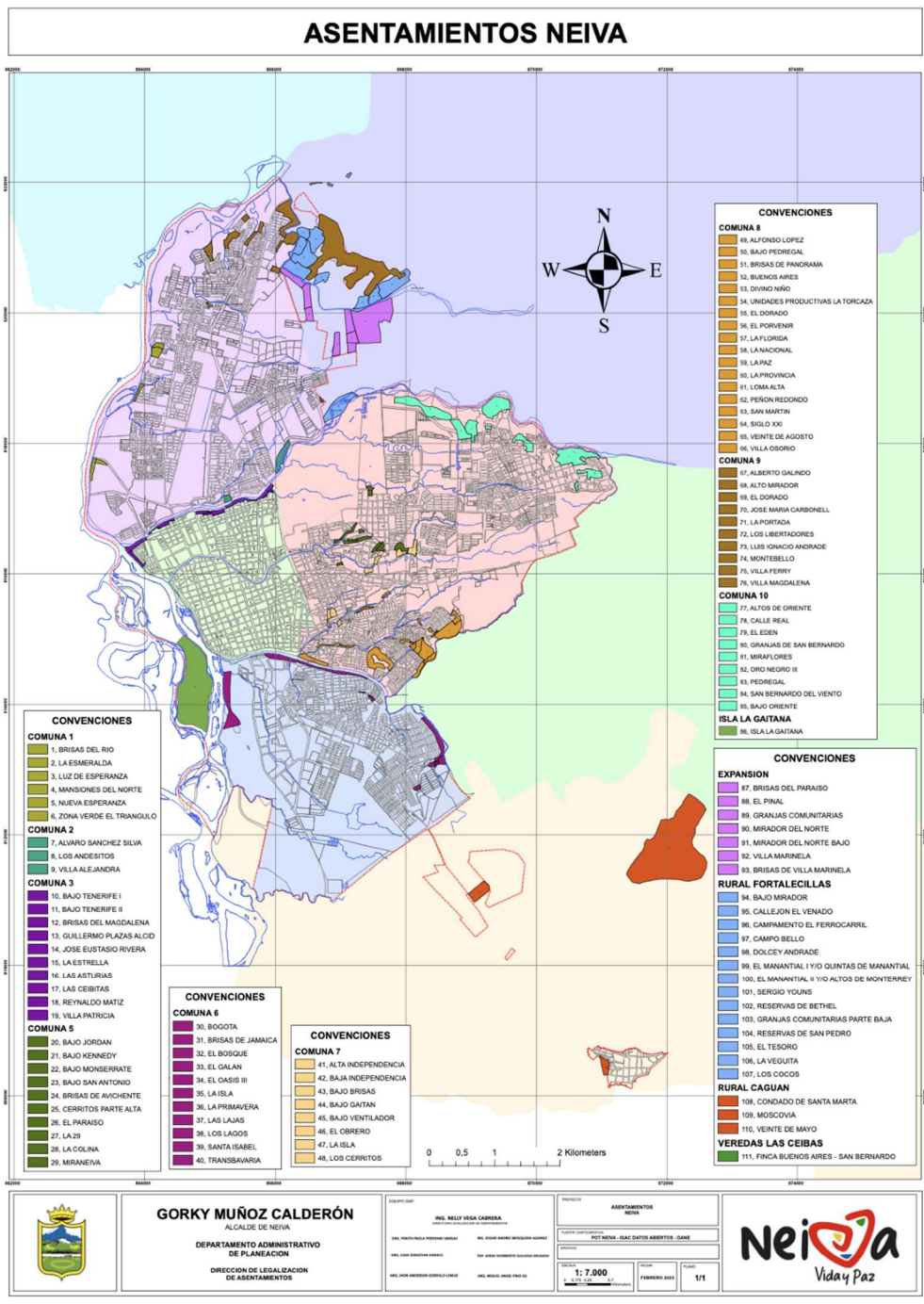


Figura 6: Plano General Asentamiento Neiva [40]

2.1 Obtención de imágenes por medio de Google Earth Pro

Mediante un proceso de selección de polígonos de áreas, se capturan imágenes de los satélites de la compañía Maxar Technologies y Airbus quien nos ofrecen unas imágenes satelitales de muy alta calidad, todo esto mediante el sistema de información geográfica de Google Earth Pro que posee datos geoespaciales disponibles de manera pública incluyendo diferentes datos de satélites alrededor del mundo como los anteriormente enunciados.

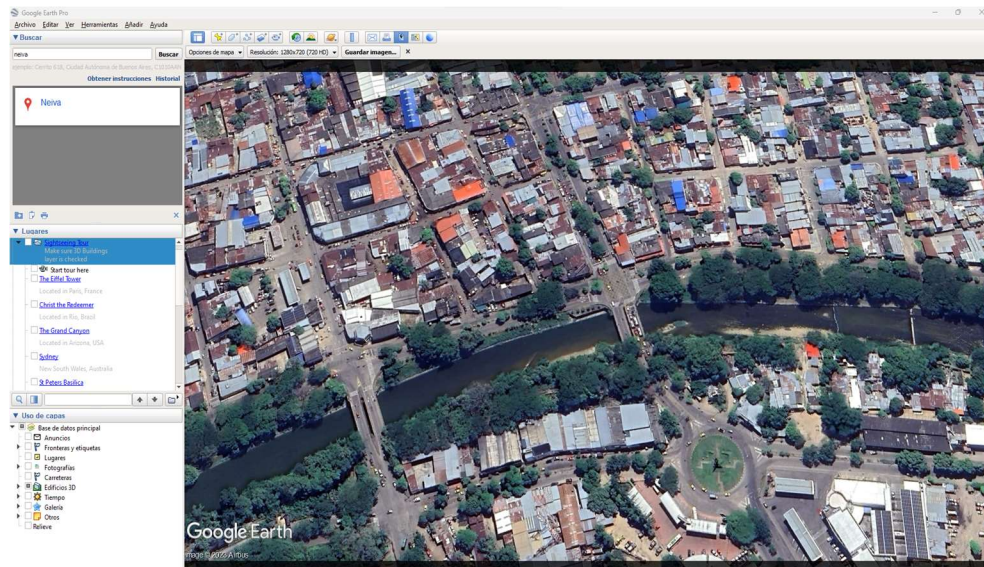


Figura 7: Obtención de imágenes usando Google Earth Pro

Las características de las imágenes obtenidas de esta herramienta son las siguientes:

1. Dimensiones de 1280 x 720 píxeles.
2. 96 dpi.
3. 24 bit.
4. Altura ojo: 500 metros aproximadamente.

Se obtuvieron un total de 144 imágenes para la construcción de las bases de datos para su posterior etiquetado utilizando Labelimg.

Las 144 imágenes se dividieron de la siguiente forma, en la carpeta de entrenamiento se guardaron 120 imágenes y en la carpeta de validación 20 imágenes es importante recalcar que las imágenes que se obtuvieron de Neiva en su mayor proporción fueron de asentamientos en zona urbana y de protección debido a que este el flagelo que más padece Neiva y menos proporción asentamiento en zona rural, fueron etiquetados en imágenes de zona rural urbana 153 etiquetas, zona de protección 154, y zona rural 103.

3.1 Preprocesamiento

En el proceso de utilización de imágenes que debido a sus características iniciales pueden llegar a no satisfacer las condiciones óptimas para su paso por el modelo de aprendizaje por transferencia (transfer learning) usando YoloV8, se requiere una transformación en el tamaño de las imágenes para lo cual utilizando Python y su librería PIL especializada en procesamiento de imágenes se procedió a definir imágenes de 640 x 640 píxeles que son los aceptados de por YoloV8 y se hizo adicionalmente un recorte definiendo la zona a extraer desde el centro de la imagen inicial y no desde alguno de sus bordes.



Figura 8: Imagen preprocesada – ejemplo para ilustración.

4.1 Etiquetado usando Labelimg

Luego de recortar las imágenes se etiquetan utilizando la herramienta Labelimg que es un programa de código abierto que permite etiquetar imágenes a través de gráficos para su posterior uso en modelos de aprendizaje automático [41], para esta investigación se utilizó para etiquetar las zonas de clasificación de conformidad a la ley 388 de 1997, en total se etiquetaron 144 imágenes en este programa.

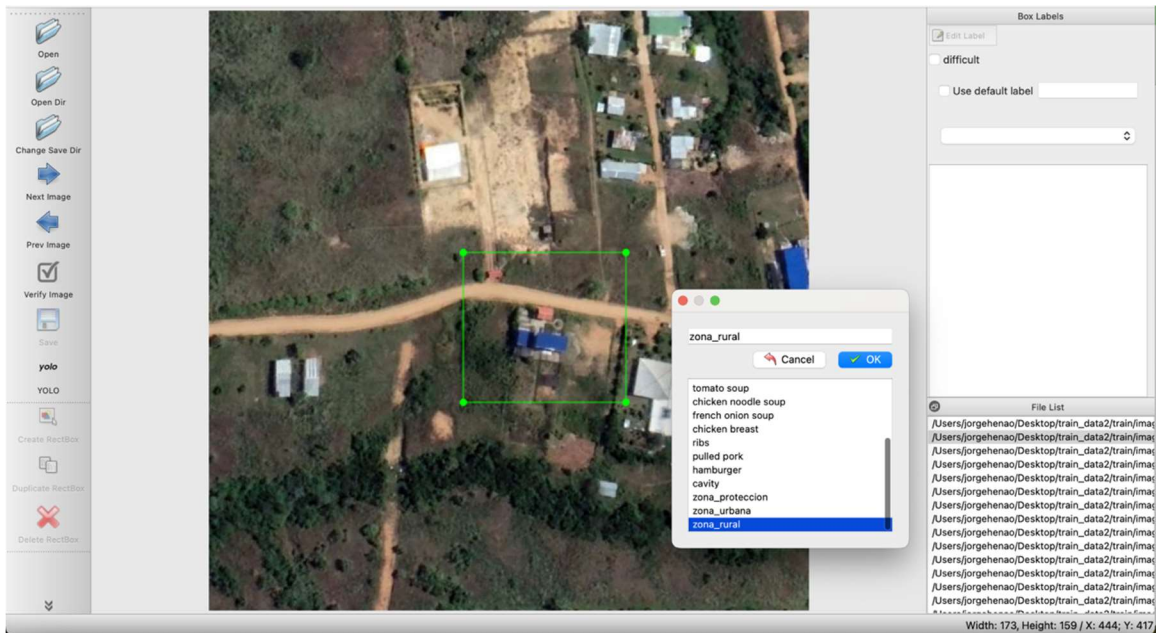


Figura 9: Etiquetado usando Labelimg

Este proceso se lleva a cabo creando carpetas de entrenamiento y validación llamadas “valid”, “train” y “test” en donde ubican las imágenes de las 3 clases etiquetadas para su etiquetado usando LabelImg, al guardar la imagen en la figura 8 la herramienta crea un archivo txt de etiquetado que tiene las siguientes características generales.

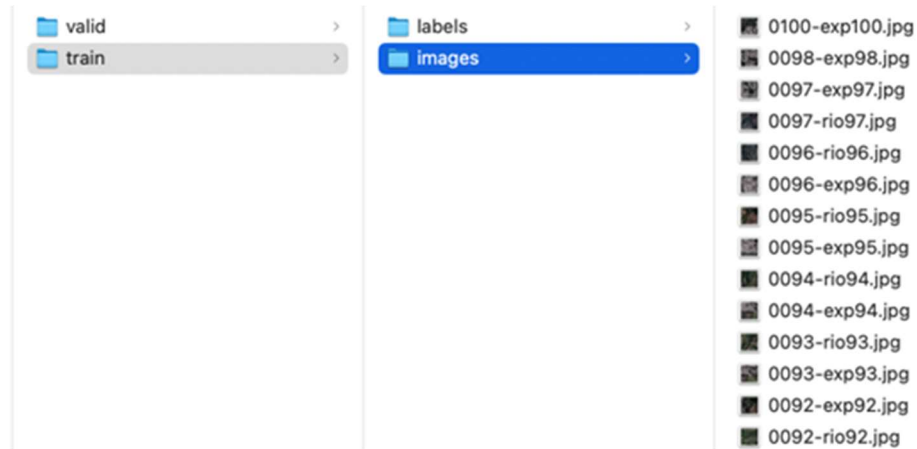


Figura 10: construcción de los conjuntos de datos para el modelo.



Figura 11: estructura del archivo txt producto del etiquetado

En este archivo como ejemplo se pueden observar las características de los polígonos etiquetados y su correspondiente clase y localización dentro de la imagen de entrenamiento en donde se tiene que la primera columna tomará los valores entre: 0 para a zona de protección, 1 para zona de asentamiento urbano, 2 igual zona de asentamiento rural, seguidamente están las coordenadas del polígono creado para cada etiquetado.

5.1 Creación archivo (.YAML)

YAML es un lenguaje de serialización de datos utilizado para configurar archivos de estructura para múltiples lenguajes de programación [42].

```
1 # Train/val/test sets as
2 ## 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
3
4
5 path: /content/drive/MyDrive/Colab Notebooks/train_data2 # dataset root dir
6 train: /content/drive/MyDrive/Colab Notebooks/train_data2/train/images # train images (relative to
7 val: /content/drive/MyDrive/Colab Notebooks/train_data2/valid/images # val images (relative to 'path
8 # test: # test images (optional)
9
10 # Classes
11 nc: 3 # number of classes
12 names: ['zona_proteccion', 'zona_urbana', 'zona_rural'] # class names
13
```

Figura 12: Creación del archivo (.YAML).

6.1 Implementación de YOLOV8 usando Python

Para esta investigación los distintos modelos fueron implementados en Python, usando Google Colab como plataforma de implementación, se optó por usar la licencia Pro, debido a que los tiempos de procesamiento son elevados y usando una licencia gratuita se limitaba las posibilidades de procesamiento, gracias a los beneficios de Google Colab Pro se pudo evaluar y probar distintas configuraciones de una manera rápida lo cual ha beneficiado el desarrollo de los objetivos planteados por los autores[43].

La implementación de YOLOV8 en Python es sencilla, primero, se instala la librería “Ultralytics” [44] en el ambiente Python con lo cual se nos da acceso a YOLOV8 con todas sus versiones y funcionalidades, es importante mencionar que dependiendo de

las necesidades a suplir y el problema de investigación YOLO ofrece distintitos modelos que contienen diferentes capacidades y parámetros.

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figura 13: Modelos YOLO disponibles para implementación [44]

Para la ejecución de nuestras bases de datos se utilizó YOLOV8x como se muestra en la siguiente línea de código en Colab usando Python:

```
!yolo train model=yolov8x.pt data=/content/custom.yaml epochs=200
imgsz=640
```

Figura 14: Entrenamiento del Modelo YOLOV8X en Python

Como se observa en la figura 13 se está usando el archivo (.yaml) creado por los autores donde se encuentran contenidas las imágenes etiquetadas, la elección de épocas fue un proceso de validación continuo donde se encontró mejor desempeño en este nivel, este análisis será expuesto en la fase de resultados con mayor detalle.

7.1 Mejor modelo con archivos .PT usando Pytorch

Pytorch es una librería disponible en Python para aprendizaje profundo que ofrece múltiples beneficios, uno de estos es la posibilidad de crear archivos llamados .PT para una fácil implementación en aplicativos sin necesidad de repetir procesos de entrenamiento en distintos ambientes si se requiere [45]. Dicho proceso es fundamental para uno de los objetivos de creación de un ambiente web para la utilización del modelo de detección y clasificación.

Respecto a los modelos evaluados los autores realizaron un diseño experimental para múltiples modelos realizando modificación a las bases de datos construidas, modificación en los hiperparamnetros y variación de las tasas de aprendizaje y numero de épocas, como se comentó en la sección 2.1, para este diseño en algunos modelos se dividió la base de datos de imágenes en 100 imágenes para entrenamiento, 22 imágenes para validación y 22 imágenes para prueba. Adicionalmente, como propuesta de los autores se desarrolló una base de datos de 122 imágenes de entrenamiento y 22 imágenes de validación, los resultados obtenidos ofrecen una perspectiva interesante respecto a las mejores prácticas en términos de metodología como se plantea la división de bases por los autores que se ha evidenciado en algunos documentos de investigación como [29], [30] usando YOLOV8, lo anterior da sustento en su implementación para el objetivo de investigación planteado. Con el propósito de brindar facilidad en la discusión de los resultados en la próxima sección se mostrará un resumen de los modelos evaluados con sus principales características.

8.1 Construcción Aplicativo Web con Django

Una vez obtenido el archivo .PT con el mejor modelo encontrado, se realizó un entorno web para la utilización del modelo permitiendo la carga de imágenes para su

procesamiento inmediato con resultados de clasificación de las zonas de conformidad con lo definido por los autores (Zona Rural, Zona Urbana y Zona de protección).

Django es una estructura de desarrollo web en Python que permite el desarrollo de interfaces limpias de rápida implementación [46]. Siguiendo la metodología del caso de uso apéndice 12.1. se realizó el proceso de configuración del proyecto de Django efectuando la instalación y posteriormente creando el proyecto e inicializando el servidor de desarrollo. A continuación, se observa la carpeta de instalación donde se agregan templates.

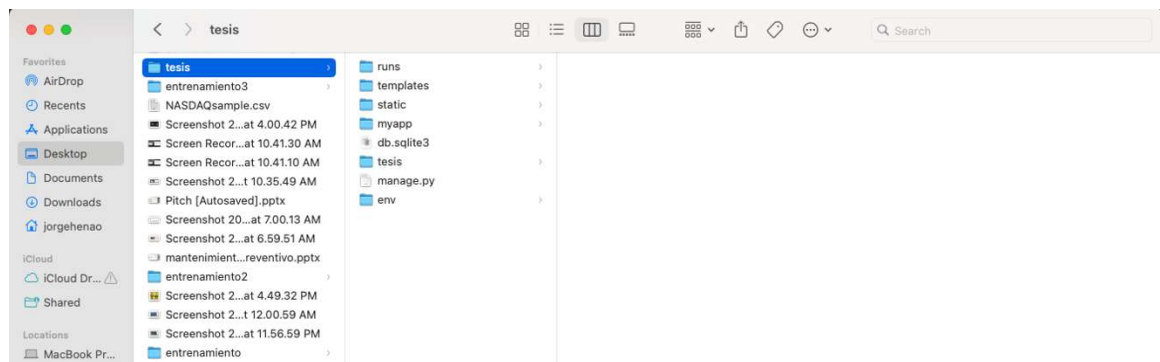


Figura 15: Carpeta raíz del aplicativo web

Se toma la decisión de usar Bootstrap como framework, el cual permite crear interfaces web en donde el contenido es responsivo lo cual facilita la adaptabilidad en múltiples dispositivos,[47] por su agilidad y sus excelentes prestaciones en el diseño responsive, para la construcción del ambiente de usuario se descarga la carpeta de la página oficial y se guarda sus archivos en static el cual trae consigo sus archivos JS, CSS que se usarán para el proceso del diseño.

La carpeta templates contiene la página principal home.html que en este caso será el interfaz principal que observará el actor que usará el aplicativo quien como primera medida invoca la carpeta de CSS para el diseño del Front-End.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="{% static './css/bootstrap.css'%}">
  <title>Predicciones de Asentamientos</title>
</head>
<body>
  <div class="container pt-3">
    <div class="row">
      <div class="col-md-12">
        <div class="card">
          <div class="card-header bg-success text-white">Predicción</div>
          <div class="card-body">
            <div class="row">
              <div class="col-md-12 text-center">
                
              <div class="col-md-12">
                <span class="text-info">Puedes dar click en la imagen para des
              </div>
              <div class="col-md-6 text-center" style="border: 1px solid #d3d3d3">
                Modelo</b> | <b>Optimizador</b> | <b>épocas</b> | <b>mAP50</b> | <b>mAP50-95</b> | <b>LR0</b> | <b>LR1</b> | <b>R</b> |
|---------------|--------------------|---------------|--------------|-----------------|------------|------------|----------|
| 1             | AdamW              | 200           | 0.921        | 0.642           | 0.01       | 0.01       | 0.80     |
| 2             | SGD                | 100           | 0.894        | 0.814           | 0.001      | 0.001      | 0.98     |
| 3             | RMSProp            | 100           | 0.00983      | 0.0034          | 0.001      | 0.001      | 0.18     |
| 4             | AdamW              | 195           | 0.371        | 0.151           | 0.01       | 0.01       | 0.47     |

Tabla 1: Modelos – Diseño Experimental YOLOV8

Tal como se planteó en la metodología en la sección 7.1 se presenta la tabla 1 con un resumen general de los modelos evaluados y sus principales métricas de desempeño, en la tabla se presenta, el modelo evaluado numerados en orden descendente, el nombre del optimizador, el número de épocas, la métrica mAP50, la métrica mAP50-95, la tasa de aprendizaje inicial, la tasa de aprendizaje final y, el “Recall” o sensibilidad que evalúa la calidad del modelo propuesto.

Luego de realizar la experimentación de los modelos, se encontraron los siguientes hallazgos, el modelo número 2, a pesar de tener una métrica de Recall cercano a 1 presento un sobreajuste tal como se puede revisar en la sección de apéndices el modelo mencionado, por su parte el modelo 3 presento sobreajuste que fue detectado directamente por Google Colab en su fase de despliegue por medio de la característica de “early stopping” sugiriendo una reducción en el número de épocas.

El modelo 4 se inició con 300 épocas, sin embargo, al igual que en el modelo 3, Google Colab sugirió a través de “early stopping” un mejor modelo en la época 195, razón por la cual se utilizó este número para dicho modelo. En general y luego de esta fase de experimentación el modelo 1 demostró ser el más prometedor mostrando mejores métricas de desempeño. A continuación, se describirán los resultados obtenidos luego de su selección.

## 10.1 Evaluación del modelo

Una vez seleccionado el modelo se procedió a validar su desempeño con las métricas estándar de evaluación que entrega YOLO, entre ellas se tiene la Curva F1-Confidence, la curva Precision-Recall, la Matriz de Confusión Normalizada, las gráficas de perdida (Box Loss, Cis Loss, DFL Loss) y las métricas de rendimiento.

### Matriz de Confusión

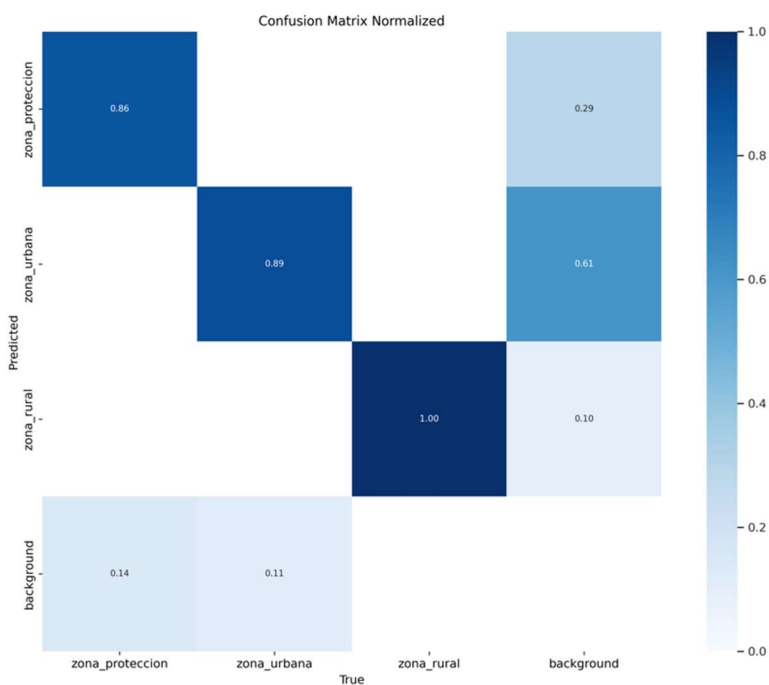


Figura 25: Matriz de Confusión

### Curva F1 Confidence Modelo

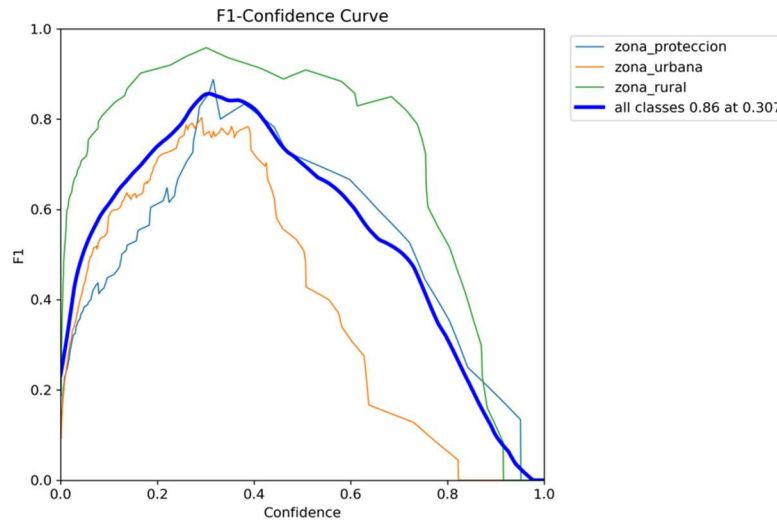


Figura 26: Curva F1 Confidence Modelo

Como se puede observar en la figura 25, el modelo ha predicho la zona rural de manera precisa con un valor de 1. Para las zonas de protección el modelo predijo correctamente el 0.86 de las veces. En cuanto a la zona urbana lo predijo el 0.89 de las veces, no obstante, al revisar el fondo o background FP y FN se observa que para la zona urbana el fondo FP es de 0.61. Esto quiere decir que en algunas zonas el modelo identificó como zona rural cuando esta no tenía clase de fondo. Para las demás clases el fondo FP y FN no es elevado.

De acuerdo con lo obtenido en la figura 26, las curvas de cada clase tienen en general un comportamiento que en un nivel de confianza del 0.30 aproximadamente logran su punto más elevado, esto mismo sucede con la curva para todas las clases o F1 Score que tiene un promedio de 0.85 para todas las clases en el umbral de confianza sobre 0.35. la clase con mejor desempeño en el F1 score es la zona rural.

### Curva Precision-Recall Modelo

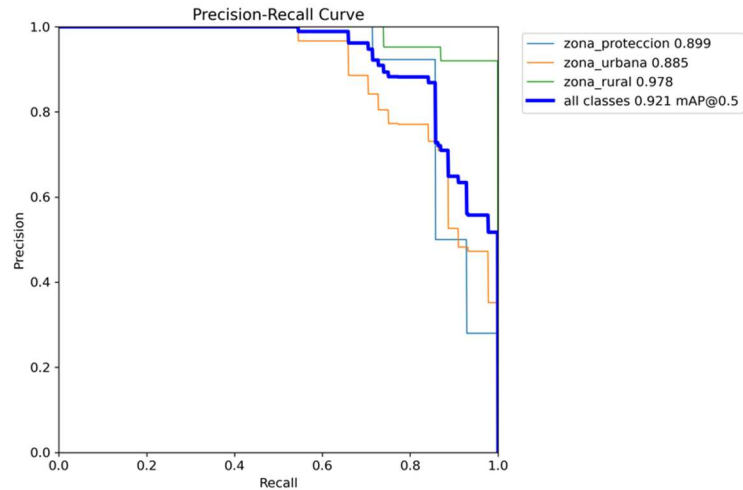


Figura 27: Curva Precision-Recall Modelo

Para las curvas Precision-Recall de la figura 27, se observa un área bajo la curva AUC elevada en todas las clases, resalta la clase de zona rural como ya se observaba en la curva de la figura 26 o curva F1 Confidence y en la matriz de confusión en la figura 25 con un alto desempeño, las clases de zona urbana y zona de protección presentan un rendimiento inferior de forma ligera a la zona rural, sin embargo, esta diferencia no es representativa. En general todas las clases presentan un umbral de mAP mayor a 0.5 que es indicativo positivo en la precisión.

### Evaluación de pérdidas y rendimiento del Modelo Uno

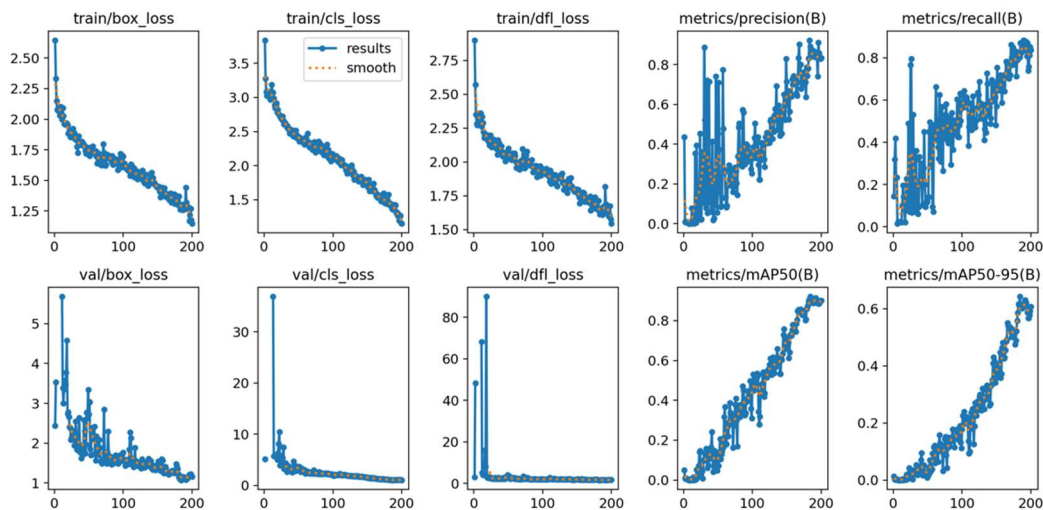


Figura 28: Gráficas de pérdida y métricas de rendimiento modelo uno.

En términos generales la evaluación de pérdidas y rendimiento del modelo número uno para las gráficas de Metrics/Precision(B) y metrics/recall(B) mejoran a lo largo de las épocas sugiriendo una mejor precisión. Así mismo, se observa esta tendencia en las gráficas de Metrics/Map50(B) y Metrics/Map50-95(B), así como en la mayoría de las métricas, no obstante, en la sección de validación la pérdida tiene un ajuste descendente pronunciado iniciando las épocas lo que podría sugerir un sobreajuste que requiera una configuración en los hiperparámetros o como ya se comentó anteriormente una expansión de la base de datos.

Los resultados expuestos evidencian comportamientos interesantes, para el mejor modelo encontrado, no obstante, se puede inferir un sobreajuste dado los elevados resultados en las métricas teniendo en cuenta que la base de datos de entrenamiento no es robusta, también, en el proceso de detección se ha encontrado que para algunas zonas luego de una revisión exhaustiva se puede dar el hallazgo de falsos positivos de zonas que morfológicamente es un asentamiento informal pero luego de una revisión exhaustiva se encontró para casos puntuales que eran zonas que a pesar de su asimetría no son asentamientos informales sino zonas habitadas fuera de la ilegalidad, casos similares se observan en la literatura [11]. Adicionalmente, el incluir zonas de expansión o zonas rurales y urbanas mixtas en la clasificación también arrojó un número de predicciones que se alejan de lo que es un asentamiento informal, como caso puntual el estudio en Indonesia encontró estos mismos retos [9], también, y frente a la evaluación de resultados la ley 388 de 1997 fue de gran apoyo para determinar con mayor grado de confianza las zonas detectadas, esta labor, ha demostrado que el factor de identificación de estas zonas es enorme y no se debe dejar a un lado el factor humano, otros trabajos de investigación en Indonesia también usaron la validación de zonas detectadas por expertos luego del procesamiento en sus modelos [12].

## Evaluación Modelo entorno web

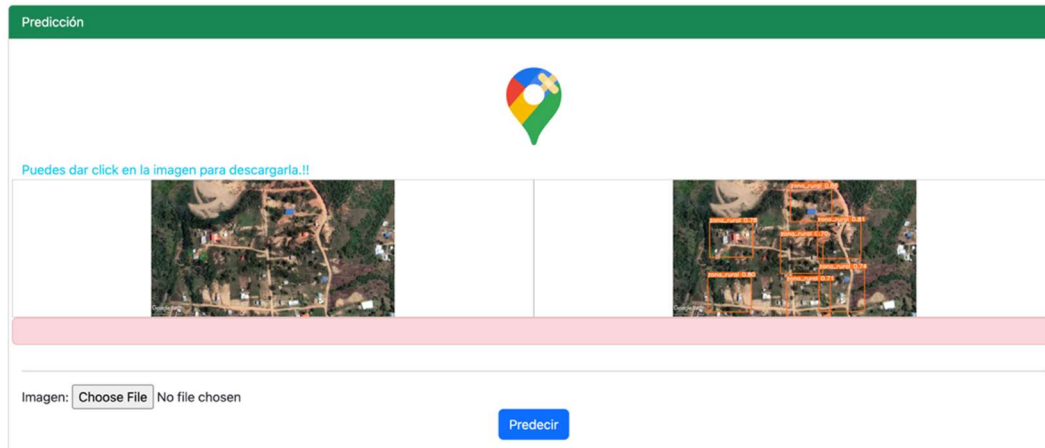


Figura 29: Entorno Web creado en Django por los autores.

Con el propósito de validar el mejor desempeño del modelo número uno, se hicieron pruebas incorporando imágenes tomadas de Google Earth de zonas geográficas distintas a el municipio de Neiva Huila, encontrando una detección bastante ajustada a las morfologías de los asentamientos informales según [8], se hicieron validaciones de conformidad con el caso de pruebas construido por los autores en el apéndice Tabla 2, con zonas del área circundante a las zonas urbanas de la ciudad de Medellín y Cartagena en Colombia. Así mismo se realizaron pruebas de usabilidad siguiendo la heurística de Jakob Nielsen, dichos resultados se pueden validar en la tabla 3.



Figura 30: Imagen de prueba Cartagena, Colombia.



Figura 31: Imagen de prueba Medellín, Colombia.

Los resultados de detección del modelo número uno tienen un desempeño favorable, se observa la detección de zonas cercanas a los ríos en el caso de la figura 31, así mismo, para la imagen de la ciudad de Cartagena la identificación de zonas urbanas es

muy aproximada a las condiciones básicas de un asentamiento informal, sin embargo, en este ejercicio se evidenció que los cambios de color y contraste en la parte derecha de la figura 30 si pueden llegar a reducir la efectividad en la detección, se observa que en esta zona también existen zonas habitadas que no fueron detectadas por el modelo.

## **Capítulo 6**

### **CONCLUSIONES Y TRABAJO FUTURO**

En esta investigación, se logró implementar el aprendizaje por transferencia o (transfer learning) utilizando la arquitectura YOLOV8 en la detección y clasificación de asentamiento informales en el municipio de Neiva, Huila. El propósito fue crear un modelo a partir de una base de datos generada por los autores para su uso y prueba en zonas geográficas dentro del territorio y en un aplicativo web de consulta inmediata.

Durante la investigación, se presentaron desafíos significativos relacionados con la implementación de las normas de clasificación de suelos en Colombia. En particular, se identificó una cuarta clase que quedó fuera del alcance del modelo propuesto: las zonas de expansión urbana. La complejidad inherente a estas áreas y la necesidad de emplear métodos avanzados para recopilar coordenadas, utilizando técnicas cartográficas y delimitaciones precisas, impidieron su evaluación en el marco de la presente investigación.

Este aspecto específico se perfila como un área clave para futuras investigaciones. Para abordar este vacío en este modelo, se propone expandir la base de datos para incluir información más detallada y específica sobre las zonas de expansión urbana. Esto podría implicar la obtención de planos y datos morfológicos especializados que permitan una clasificación más precisa y detallada de estas áreas.

Asimismo, en la planificación de trabajos futuros, se considera la posibilidad de ofrecer este aplicativo como una herramienta práctica para organismos territoriales, gobernaciones o alcaldías. La visión es proporcionar una solución accesible y eficiente que permita a dichas entidades utilizar el aplicativo web mediante una suscripción. Esta suscripción posibilitaría cargar imágenes relevantes para la región en cuestión, permitiendo que el aplicativo realice automáticamente la clasificación correspondiente. Esta iniciativa no solo contribuiría a mejorar la eficiencia en la gestión territorial, sino que también podría

generar recursos para la sostenibilidad y mejora continua del aplicativo, garantizando su utilidad a largo plazo.

La arquitectura YOLOV8 demostró ser una herramienta potente de detección y clasificación en el procesamiento de imágenes satelitales, siendo actualmente la última tendencia en esta materia y su implementación resulta muy flexible y económica para los entornos de investigación, las aplicaciones de esta arquitectura son extensas pasando de la clasificación de objetos para ejercicios académicos, la detección de olas o corrientes de resaca [30] que claramente es un tema ambiental donde su aplicabilidad demostró ser sobresaliente, también, en el campo de la medicina y uno de los pioneros en el desarrollo de aplicativos para la detección de tumores [26] se utiliza YOLO como herramienta de detección, en la prevención de incendios, se implementó [28] con desempeño notable lo cual lo hace una herramienta flexible que ha demostrado ser eficaz en la detección y clasificación de imágenes en general, por lo anterior, los autores han decidido elegir esta arquitectura y aprovechar sus beneficios para así implementarlo en la detección de asentamientos informales en Neiva, Huila. Este tipo de despliegues y sistemas inteligentes también demostró ser de bajo costo en entornos de investigación, y, que en caso de un despliegue en alguna entidad puede ser un elemento clave al momento de elegirlo.

Respecto a la metodología propuesta, se evidenció que en esta disciplina existen variedad de criterios alineados a cada problema de investigación, y, para este caso puntual con el propósito de seguir buenas prácticas metodológicas, se propusieron distintos modelos y bases de entrenamiento, el primero, con mejor desempeño, demostró ser el más robusto y detectar los asentamiento basados en su morfología como lo describe Kohli [8], formas irregulares, cercanas a ríos, con vías asimétricas y dispersos alrededor de zonas urbanizadas. Los hallazgos de esta implementación revelaron, sin embargo, la necesidad de ampliar la base de datos de imágenes satelitales en la zona de Neiva. Se observaron señales de sobreajuste en el mejor modelo propuesto que pueden llevar a no ser suficiente en términos de una pobre generalización con nuevos datos, tal como se demostró con zonas externas al área de investigación, dichos resultados no fueron sobresalientes.

Esta herramienta, como un insumo para el desarrollo y evaluación del Plan de Ordenamiento Territorial en Colombia, puede convertirse en un elemento clave y de bajo costo en la detección y clasificación de los asentamientos informales, en este sentido, la búsqueda de aplicativos que mejoren la obtención de información relevante respecto a los asentamientos informales tiene un interés global [53], no es una necesidad del municipio de Neiva únicamente, emprender en procesos que ayuden en la pronta ejecución de los planes de acción en conformidad con las normas vigentes en Colombia enfocadas en la acción temprana para mitigar riesgos diversos con dicha población puede cambiar con el uso de este tipo de herramientas disruptivas.

## REFERENCIAS

- [1] “Ilegalidad urbana y producción del espacio en el borde del río Las Ceibas en la ciudad de Neiva, Huila\*”, Accessed: Jun. 10, 2023. [Online]. Available: [https://revistas.javeriana.edu.co/files-articulos/CVU/15%20\(2022\)/629771821013/index.html](https://revistas.javeriana.edu.co/files-articulos/CVU/15%20(2022)/629771821013/index.html)
- [2] H. Uribe Castro, “Los asentamientos ilegales en Colombia: las contradicciones de la economía-mundo capitalista en la sociedad global,” *Latinoamérica. Revista de estudios Latinoamericanos*, no. 53, pp. 169–200, Dec. 2011.
- [3] K. Therán-Nieto, R. Pérez-Arévalo, and D. García-Estrada, “Asentamientos informales en la periferia urbana de áreas metropolitanas. El caso de Soledad, Colombia,” *urbe, Rev. Bras. Gest. Urbana*, vol. 14, p. e20210275, Dec. 2022, doi: 10.1590/2175-3369.014.e20210275.
- [4] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.
- [5] C. G. Gross, C. E. Rocha-Miranda, and D. B. Bender, “Visual properties of neurons in inferotemporal cortex of the Macaque.,” *Journal of Neurophysiology*, vol. 35, no. 1, pp. 96–111, Jan. 1972, doi: 10.1152/jn.1972.35.1.96.
- [6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Jun. 05, 2023. [Online]. Available: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- [8] D. Kohli, R. Sliuzas, N. Kerle, and A. Stein, “An ontology of slums for image-based classification,” *Computers, Environment and Urban Systems*, vol. 36, no. 2, pp. 154–163, Mar. 2012, doi: 10.1016/j.compenvurbsys.2011.11.001.
- [9] J. Pratomo, M. Kuffer, J. Martinez, and D. Kohli, “Coupling Uncertainties with Accuracy Assessment in Object-Based Slum Detections, Case Study: Jakarta, Indonesia,” *Remote Sensing*, vol. 9, no. 11, Art. no. 11, Nov. 2017, doi: 10.3390/rs9111164.
- [10] E. Guirado, S. Tabik, D. Alcaraz-Segura, J. Cabello, and F. Herrera, “Deep-learning Versus OBIA for Scattered Shrub Detection with Google Earth Imagery: *Ziziphus lotus* as Case Study,” *Remote Sensing*, vol. 9, no. 12, Art. no. 12, Dec. 2017, doi: 10.3390/rs9121220.
- [11] E. Rangelova, B. Weel, D. Roy, M. Kuffer, K. Pfeffer, and M. Lees, “Image based classification of slums, built-up and non-built-up areas in Kalyan and Bangalore, India,” *European Journal of Remote Sensing*, vol. 52, no. sup1, pp. 40–61, Mar. 2019, doi: 10.1080/22797254.2018.1535838.
- [12] G. Leonita, M. Kuffer, R. Sliuzas, and C. Persello, “Machine Learning-Based Slum Mapping in Support of Slum Upgrading Programs: The Case of Bandung City,

- Indonesia,” *Remote Sensing*, vol. 10, no. 10, Art. no. 10, Oct. 2018, doi: 10.3390/rs10101522.
- [13] A. Ajami, M. Kuffer, C. Persello, and K. Pfeffer, “Identifying a Slums’ Degree of Deprivation from VHR Images Using Convolutional Neural Networks,” *Remote Sensing*, vol. 11, no. 11, Art. no. 11, Jan. 2019, doi: 10.3390/rs11111282.
- [14] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, “Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 59–69, Apr. 2019, doi: 10.1016/j.isprsjprs.2019.02.006.
- [15] V. N. Vinaykumar, J. A. Babu, and J. Frnda, “Optimal guidance whale optimization algorithm and hybrid deep learning networks for land use land cover classification,” *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 1, p. 13, Jan. 2023, doi: 10.1186/s13634-023-00980-w.
- [16] M. Carranza-García, J. García-Gutiérrez, and J. C. Riquelme, “A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks,” *Remote Sensing*, vol. 11, no. 3, Art. no. 3, Jan. 2019, doi: 10.3390/rs11030274.
- [17] M. Alkhalawi, W. Boulila, J. Ahmad, A. Koubaa, and M. Driss, “An Efficient Approach Based on Privacy-Preserving Deep Learning for Satellite Image Classification,” *Remote Sensing*, vol. 13, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/rs13112221.
- [18] O. Sefrin, F. M. Riese, and S. Keller, “Deep Learning for Land Cover Change Detection,” *Remote Sensing*, vol. 13, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/rs13010078.
- [19] P. Helber, B. Bischke, A. Dengel, and D. Borth, “EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification,” Aug. 2017, doi: 10.1109/JSTARS.2019.2918242.
- [20] J. C. Duque, J. E. Patino, and A. Betancourt, “Exploring the Potential of Machine Learning for Automatic Slum Identification from VHR Imagery,” *Remote Sensing*, vol. 9, no. 9, Art. no. 9, Sep. 2017, doi: 10.3390/rs9090895.
- [21] A. S. S. L. Suárez L, A. F. Jiménez L, M. Castro-Franco, and A. Cruz-Roa, “Clasificación y mapeo automático de coberturas del suelo en imágenes satelitales utilizando Redes Neuronales Convolucionales.” Accessed: Mar. 20, 2023. [Online]. Available: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0121-37092017000300064](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-37092017000300064)
- [22] L. F. Gómez Ossa, “Aplicación de redes neuronales artificiales en la modelación de la deforestación asociada a nuevos proyectos de infraestructura vial en las regiones del Nordeste y Bajo Cauca del departamento de Antioquia,” Oct. 2014, Accessed: Mar. 20, 2023. [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/52176>
- [23] A. Montero Leguizamón, “Aplicación de una red neuronal convolucional para la predicción de mallas de población en el área metropolitana de Bogotá,” Trabajo de grado - Maestría, Universidad Nacional de Colombia, 2021. Accessed: Mar. 20, 2023. [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/80363>

- [24] L. S. Valdés Ávila and J. M. Baquero Vanegas, “Deep Learning aplicado a imágenes satelitales como herramienta de detección de viviendas sin servicio de energía en el caserío Media Luna-Uribia-Guajira,” 2019, Accessed: Mar. 20, 2023. [Online]. Available: <http://repository.udistrital.edu.co/handle/11349/16030>
- [25] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [26] F. J. Montalbo, “A Computer-Aided Diagnosis of Brain Tumors Using a Fine-Tuned YOLO-based Model with Transfer Learning,” *KSII Transactions on Internet and Information Systems*, vol. 14, pp. 4816–4834, Jan. 2021, doi: 10.3837/tiis.2020.12.011.
- [27] A. B. Arregui Arias, “Detección de cascos de seguridad en tiempo real mediante modelos de Deep Learning,” *Real-time hard hat detection by Deep Learning models*, May 2023, Accessed: Oct. 01, 2023. [Online]. Available: <https://repositorio.upct.es/handle/10317/12477>
- [28] F. M. Talaat and H. ZainEldin, “An improved fire detection approach based on YOLO-v8 for smart cities,” *Neural Comput & Applic*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.
- [29] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, “Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8,” presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5349–5357. Accessed: Oct. 09, 2023. [Online]. Available: [https://openaccess.thecvf.com/content/CVPR2023W/AICity/html/Aboah\\_Real-Time\\_Multi-Class\\_Helmet\\_Violation\\_Detection\\_Using\\_Few-Shot\\_Data\\_Sampling\\_Technique\\_CVPRW\\_2023\\_paper.html](https://openaccess.thecvf.com/content/CVPR2023W/AICity/html/Aboah_Real-Time_Multi-Class_Helmet_Violation_Detection_Using_Few-Shot_Data_Sampling_Technique_CVPRW_2023_paper.html)
- [30] A. Dumitriu, F. Tatui, F. Miron, R. T. Ionescu, and R. Timofte, “Rip Current Segmentation: A Novel Benchmark and YOLOv8 Baseline Results,” presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1261–1271. Accessed: Oct. 09, 2023. [Online]. Available: [https://openaccess.thecvf.com/content/CVPR2023W/NTIRE/html/Dumitriu\\_Rip\\_Current\\_Segmentation\\_A\\_Novel\\_Benchmark\\_and\\_YOLOv8\\_Baseline\\_Results\\_CVPRW\\_2023\\_paper.html](https://openaccess.thecvf.com/content/CVPR2023W/NTIRE/html/Dumitriu_Rip_Current_Segmentation_A_Novel_Benchmark_and_YOLOv8_Baseline_Results_CVPRW_2023_paper.html)
- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [32] J. Terven and D. Cordova-Esparza, “A Comprehensive Review of YOLO: From YOLOv1 and Beyond.” arXiv, Oct. 07, 2023. doi: 10.48550/arXiv.2304.00501.
- [33] J. Solawetz, F. JAN 11, and 2023 10 Min Read, “What is YOLOv8? The Ultimate Guide.,” Roboflow Blog. Accessed: Nov. 07, 2023. [Online]. Available: <https://blog.roboflow.com/whats-new-in-yolov8/>
- [34] M. E. Taylor and P. Stone, “Transfer Learning for Reinforcement Learning Domains: A Survey,” *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, Dec. 2009.
- [35] F. Ramos Pismataro and R. F. Rodríguez, *Colombia de cara a los desafíos y oportunidades que representa la migración venezolana*. Fundación Konrad Adenauer,

2019. Accessed: Nov. 19, 2023. [Online]. Available: <https://repository.urosario.edu.co/handle/10336/30089>
- [36] C. F. Collado and P. B. Lucio, “METODOLOGÍA DE LA INVESTIGACIÓN”.
- [37] N. Esteban Nieto, “Tipos de Investigación,” *Universidad Santo Domingo de Guzmán*, Jun. 2018, Accessed: Jun. 10, 2023. [Online]. Available: <http://repositorio.usdg.edu.pe/handle/USDG/34>
- [38] A. Restrepo Ruiz, “Poder y gestión del suelo estudio de planes parciales en la ciudad de Medellín,” Dec. 2012, Accessed: Nov. 09, 2023. [Online]. Available: <http://repositorio.esumer.edu.co/jspui/handle/esumer/286>
- [39] “Avances Revisión General.” Accessed: Nov. 07, 2023. [Online]. Available: <https://www.alcaldianeiva.gov.co/POT/Paginas/Avances.aspx>
- [40] “Plano General Asentamiento Neiva.” Accessed: Nov. 09, 2023. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/<https://www.alcaldianeiva.gov.co/Gestion/FICHA%20PRELIMINAR%20DE%20ASENTAMIENTOS/PLANO%20GENERAL%20ASENTAMIENTOS%20-%202023.pdf>
- [41] “GitHub - HumanSignal/labelImg at pside6,” GitHub. Accessed: Nov. 09, 2023. [Online]. Available: <https://github.com/HumanSignal/labelImg>
- [42] “The Official YAML Web Site.” Accessed: Nov. 12, 2023. [Online]. Available: <https://yaml.org/>
- [43] “Colab Subscription Pricing.” Accessed: Nov. 12, 2023. [Online]. Available: <https://colab.research.google.com/signup>
- [44] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics.” Jan. 2023. Accessed: Nov. 12, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [45] “PT File - What is a .pt file and how do I open it?” Accessed: Nov. 12, 2023. [Online]. Available: <https://fileinfo.com/extension/pt>
- [46] “Django,” Django Project. Accessed: Nov. 12, 2023. [Online]. Available: <https://www.djangoproject.com/>
- [47] M. O. contributors Jacob Thornton, and Bootstrap, “Bootstrap.” Accessed: Nov. 13, 2023. [Online]. Available: <https://getbootstrap.com/>
- [48] Prateek, “Decoding the Confusion Matrix,” KeyToDataScience. Accessed: Jan. 15, 2024. [Online]. Available: <https://keytodatascience.com/confusion-matrix/>
- [49] D. de Frutos Zafra, “Agente inteligente para monitorización de carreteras.” Accessed: Jan. 20, 2024. [Online]. Available: <https://oa.upm.es/75049/>
- [50] L. Simon, R. Webster, and J. Rabin, “Revisiting Precision and Recall Definition for Generative Model Evaluation.” arXiv, May 14, 2019. doi: 10.48550/arXiv.1905.05441.
- [51] P. L. K. Ding, Y. Li, and B. Li, “Mean Local Group Average Precision (mLGAP): A New Performance Metric for Hashing-based Retrieval.” arXiv, Nov. 23, 2018. doi: 10.48550/arXiv.1811.09763.
- [52] M. S. Hossain, J. M. Betts, and A. P. Paplinski, “Dual Focal Loss to address class imbalance in semantic segmentation,” *Neurocomputing*, vol. 462, pp. 69–87, Oct. 2021, doi: 10.1016/j.neucom.2021.07.055.

[53] O. Kit and M. Lüdeke, “Automated detection of slum area change in Hyderabad, India using multitemporal satellite imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 130–137, Sep. 2013, doi: 10.1016/j.isprsjprs.2013.06.009.

[54] “2.1.8.1 Criterios y Técnicas de Usabilidad.” Accessed: Jan. 23, 2024. [Online]. Available:

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro17/2181\\_criterios\\_y\\_tcnicas\\_de\\_usabilidad.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro17/2181_criterios_y_tcnicas_de_usabilidad.html)

## APÉNDICE

### 11.1 Diseño Experimental – Resultados

```

all 24 81 0.829 0.851 0.898 0.591
Epoch GPU_mem box_loss cla_loss dfl_loss Instances Size
200/200 14.1G 1.147 1.149 1.544 31 640: 100% 8/8 [00:08<00:00, 1.08s/it]
Class Images Instances Box(P R mAP50 mAP50-95): 100% 1/1 [00:00<00:00, 1.40it/s]
all 24 81 0.834 0.836 0.901 0.607

200 epochs completed in 0.648 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 136.8MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 136.7MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch=2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
Class Images Instances Box(P R mAP50 mAP50-95): 100% 1/1 [00:00<00:00, 1.24it/s]
all 24 81 0.891 0.803 0.921 0.642
zona_proteccion 24 14 0.915 0.714 0.899 0.589
zona_urbana 24 44 0.842 0.725 0.885 0.607
zona_rural 24 23 0.918 0.969 0.978 0.73

Speed: 0.2ms preprocess, 27.4ms inference, 0.0ms loss, 0.6ms postprocess per image
Results saved to runs/detect/train2
Learn more at https://docs.ultralytics.com/modes/train

```

Figura 32: Salida de entrenamiento Modelo Uno

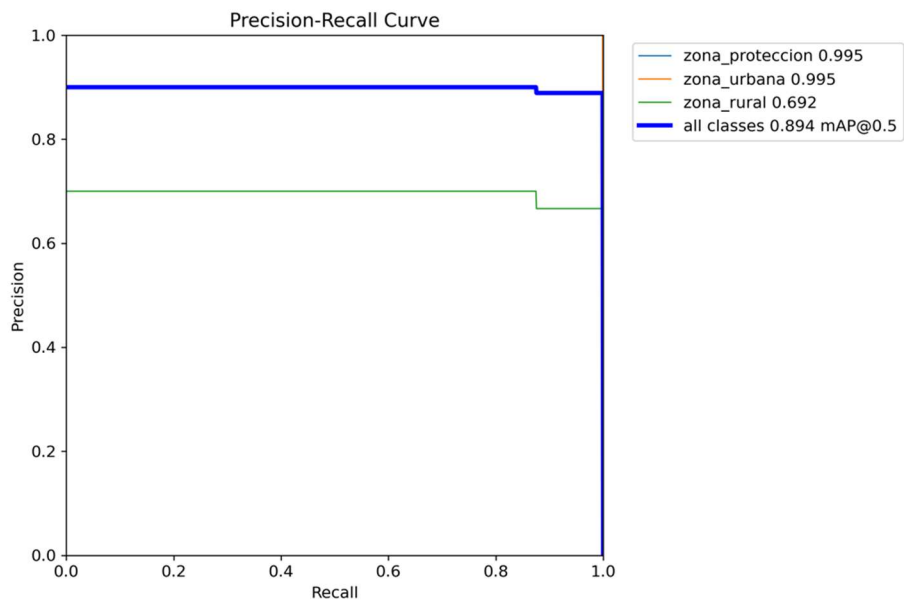


Figura 33: Precision Recall Modelo Dos

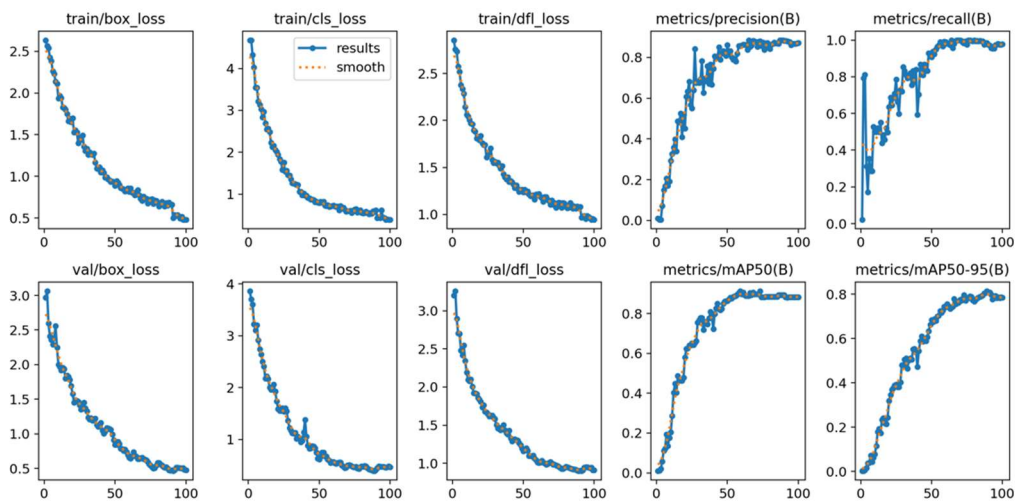


Figura 34: Evaluación de pérdidas y rendimiento del Modelo Dos

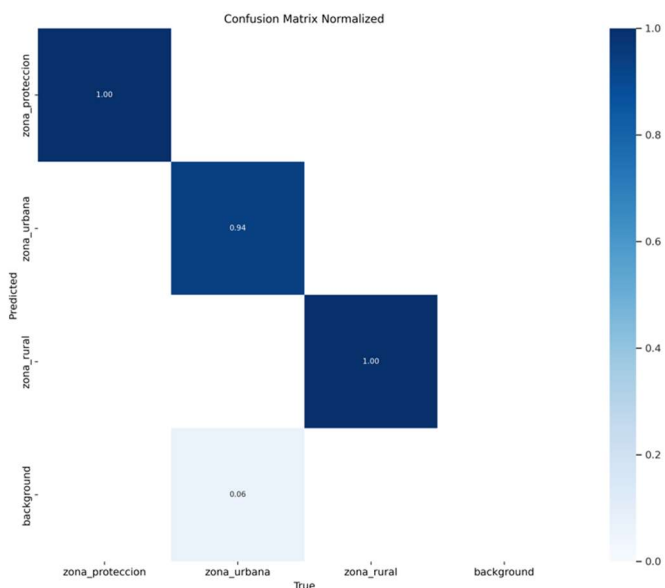


Figura 35: Matriz de confusión Modelo Dos

```

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
100/100 13.8G 0.4797 0.3866 0.9444 3 640: 100% |██████████| 16/16 [00:19<00:00, 1.19s/it]
 Class Images Instances Box(P) R mAP50 mAP50-95): 100% |██████████| 1/1 [00:00<00:00, 1.78it/s]
 all 29 40 0.872 0.978 0.883 0.786

100 epochs completed in 0.630 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 136.7MB
Optimizer stripped from runs/detect/train/weights/best.pt, 136.7MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.200 Python-3.10.12 torch-2.1.0+cud118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
 Class Images Instances Box(P) R mAP50 mAP50-95): 100% |██████████| 1/1 [00:00<00:00, 1.72it/s]
 all 29 40 0.884 0.985 0.894 0.814
 zona_proteccion 29 16 0.994 1 0.995 0.855
 zona_urbana 29 16 1 0.955 0.995 0.928
 zona_rural 29 8 0.656 1 0.692 0.66

Speed: 0.1ms preprocess, 15.4ms inference, 0.0ms loss, 0.9ms postprocess per image
Results saved to runs/detect/train
Ultralytics YOLOv8.0.200 Python-3.10.12 torch-2.1.0+cud118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
val: Scanning /content/drive/MyDrive/Colab Notebooks/train_data/valid/labels.cache... 26 images, 3 backgrounds, 0 corrupt: 100% |██████████|
 Class Images Instances Box(P) R mAP50 mAP50-95): 100% |██████████| 2/2 [00:02<00:00, 1.13s/it]
 all 29 40 0.884 0.985 0.894 0.817
 zona_proteccion 29 16 0.994 1 0.995 0.855
 zona_urbana 29 16 1 0.954 0.995 0.937
 zona_rural 29 8 0.657 1 0.692 0.66

Speed: 0.2ms preprocess, 48.5ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train2

```

Figura 36: Salida de entrenamiento Modelo Dos

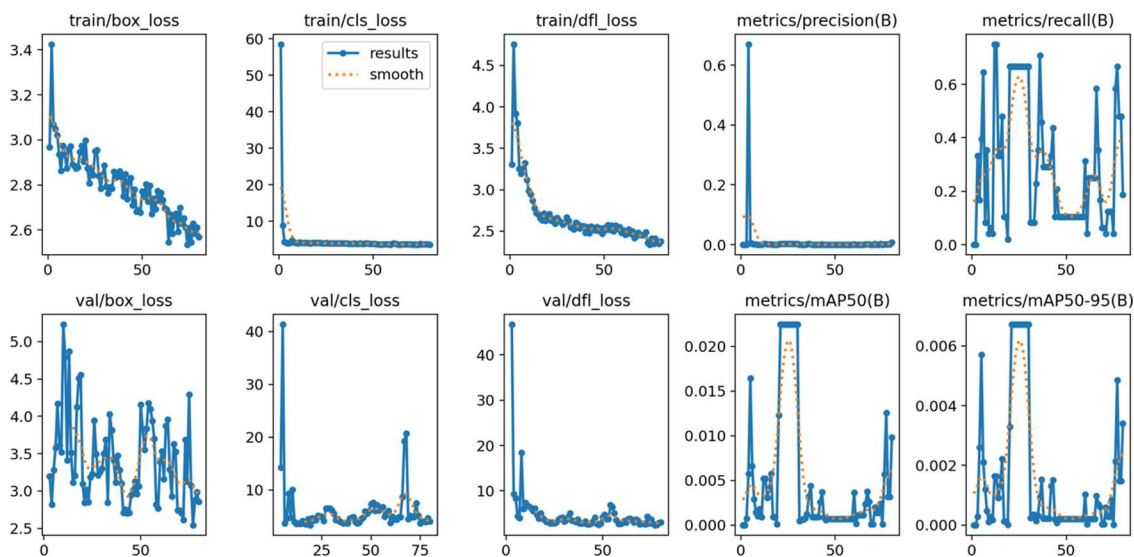


Figura 37: Evaluación de pérdidas y rendimiento del Modelo Tres

```

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
80/200 14G 2.569 3.658 2.379 8 640: 100%|██████████| 16/16 [00:18<00:00, 1.14s/it]
 Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 1/1 [00:00<00:00, 1.39it/s]
 all 29 40 0.00855 0.188 0.00983 0.0034
Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 30, best model saved as best.pt.
To update EarlyStopping(patience=50) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

80 epochs completed in 0.492 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 136.7MB
Optimizer stripped from runs/detect/train/weights/best.pt, 136.7MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.202 Python-3.10.12 torch-2.1.0+cud118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
 Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 1/1 [00:00<00:00, 1.73it/s]
 all 29 40 0.00855 0.188 0.00983 0.0034
 zona_proteccion 29 16 0.00166 0.375 0.00246 0.000829
 zona_urbana 29 16 0.024 0.188 0.027 0.00938
 zona_rural 29 8 0 0 0 0
Speed: 0.1ms preprocess, 16.0ms inference, 0.0ms loss, 0.8ms postprocess per image
Results saved to runs/detect/train
Ultralytics YOLOv8.0.202 Python-3.10.12 torch-2.1.0+cud118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
val: Scanning /content/drive/MyDrive/Colab Notebooks/train_data/valid/labels.cache... 26 images, 3 backgrounds, 0 corrupt: 100%|██████████|
 Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 2/2 [00:02<00:00, 1.01s/it]
 all 29 40 0 0 0 0
Speed: 0.2ms preprocess, 41.5ms inference, 0.0ms loss, 1.5ms postprocess per image
Results saved to runs/detect/train2

```

Figura 38: Salida de entrenamiento Modelo Tres<sup>1</sup>

<sup>1</sup> El modelo al tener un “early stop” no genera curvas de evaluación

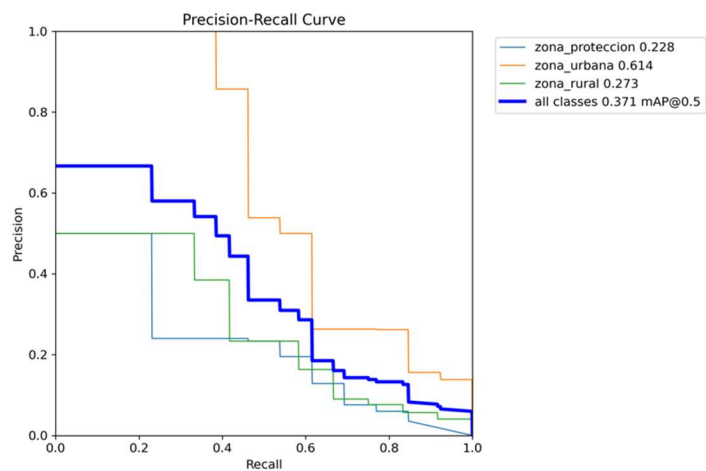


Figura 39: Precision Recall Modelo Cuatro

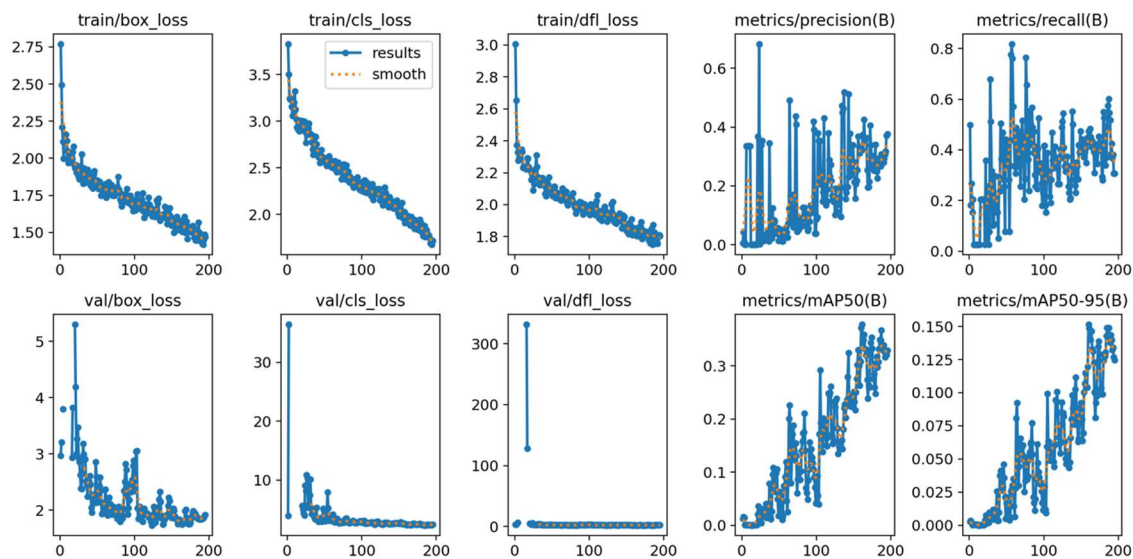


Figura 40: Evaluación de pérdidas y rendimiento del Modelo Cuatro

```

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
195/195 14.1G 1.47 1.72 1.806 51 640: 100% 6/6 [00:07<00:00, 1.24s/it]
 Class Images Instances Box(P) R mAP50 mAP50-95): 100% 1/1 [00:00<00:00, 2.60it/s]
 all 12 38 0.376 0.306 0.329 0.125

195 epochs completed in 0.584 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 136.8MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 136.7MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.209 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68126457 parameters, 0 gradients, 257.4 GFLOPs
 Class Images Instances Box(P) R mAP50 mAP50-95): 100% 1/1 [00:00<00:00, 2.48it/s]
 all 12 38 0.331 0.474 0.371 0.151
 zona_proteccion 12 13 0.232 0.462 0.228 0.0815
 zona_urbana 12 13 0.537 0.462 0.614 0.296
 zona_rural 12 12 0.224 0.5 0.273 0.0764

Speed: 0.2ms preprocess, 28.7ms inference, 0.0ms loss, 1.2ms postprocess per image
Results saved to runs/detect/train2

```

Figura 41: Salida de entrenamiento Modelo Cuatro

## 12.1 Aplicativo Web

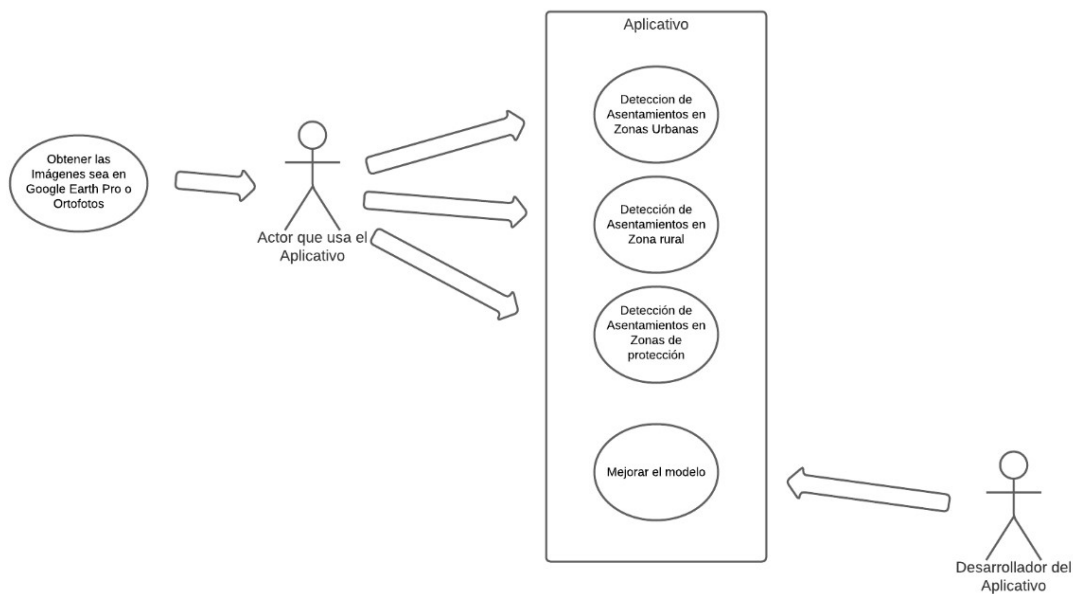


Figura 42: Caso de Uso del Aplicativo en Django

| <b>Caso de Prueba</b>                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nombre:</b>                                                                                                                                                                                                                                                                                            | Acceso y uso del aplicativo                                                                                                                                                                                                                                                                                |
| <b>Autor:</b>                                                                                                                                                                                                                                                                                             | Los Autores                                                                                                                                                                                                                                                                                                |
| <b>Fecha:</b>                                                                                                                                                                                                                                                                                             | 10/10/2023                                                                                                                                                                                                                                                                                                 |
| <b>Descripción:</b> El usuario necesita disponer de imágenes satelitales en formatos (PNG, JPEG, JPG) tomadas a 500 metro de altura con 96dpi, las imágenes deben tener un tamaño preferiblemente de 640x640 píxeles, 1280 x 720, en caso contrario el aplicativo podría presentar predicciones erróneas. |                                                                                                                                                                                                                                                                                                            |
| <b>Actores:</b> Usuario a través del aplicativo Web.                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                            |
| <b>Pre-condiciones:</b> Disponer de las imágenes para procesamiento.                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                            |
| <b>Flujo Normal:</b>                                                                                                                                                                                                                                                                                      | <ol style="list-style-type: none"> <li>1. El usuario accede al aplicativo.</li> <li>2. el usuario carga la imagen.</li> <li>3. el aplicativo procesa y devuelve la imagen procesada.</li> </ol>                                                                                                            |
| <b>Flujo Alternativo:</b>                                                                                                                                                                                                                                                                                 | <ol style="list-style-type: none"> <li>1. El usuario sube archivos en formato diferente al sugerido y el aplicativo no genera salida.</li> </ol>                                                                                                                                                           |
| <b>Excepciones:</b>                                                                                                                                                                                                                                                                                       | <ol style="list-style-type: none"> <li>1. Error en el servidor: en caso de no disponer los servicios para el procesamiento hay un informe de errores que notifica dicho evento.</li> </ol>                                                                                                                 |
| <b>Post Condiciones:</b>                                                                                                                                                                                                                                                                                  | <ol style="list-style-type: none"> <li>1. Resultado del procesamiento: una vez cargada a la imagen el aplicativo muestra la imagen procesada.</li> <li>2. Restauración a estado inicial: una vez procesad una imagen, al cargar una segunda imagen el aplicativo se restaura al estado inicial.</li> </ol> |

Tabla 2: Caso de prueba Aplicativo en Django.

| “Heurísticas de Jakob Nielsen”                       |                                                                                                                                                                                                      |           |           |           |           |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------|-----------|-----------|
|                                                      | Pregunta                                                                                                                                                                                             | usuario 1 | usuario 2 | usuario 3 | usuario 4 |
| Visibilidad del estado del sistema                   | El sistema siempre mantiene al usuario informado de lo que esta pasando en la web a cada momento.                                                                                                    | 5         | 5         | 4         | 5         |
| Consistencia entre el sistema y el mundo real        | El sistema habla el idioma correcto al tuyo (español) usa un lenguaje familia, las imágenes son correctas                                                                                            | 5         | 5         | 5         | 5         |
| Control y libertad del usuario                       | botones de deshacer y rehacer                                                                                                                                                                        | 3         | 3         | 3         | 3         |
| Consistencia y estándares                            | usa estándares en el diseño web                                                                                                                                                                      | 5         | 5         | 5         | 5         |
| Prevención de errores                                | Los buenos mensajes de error son importantes. Pero mejor aún, es crear un diseño que evite que los errores ocurran.                                                                                  | 4         | 5         | 3         | 4         |
| Reconocimiento en lugar de recuerdo                  | el interfaz es intuitivo y por ende el usuario no se obliga a memorizar las cosas                                                                                                                    | 5         | 5         | 5         | 5         |
| Flexibilidad y eficiencia de uso                     | Los usuarios más experimentados, deben poseer atajos y aceleradores para poder realizar sus operaciones mas habituales. De esta forma, tiene alternativas para personalizar sus acciones frecuentes. | 4         | 5         | 4         | 5         |
| Diseño estético y minimalista                        | Las interfaces no deben contener información irrelevante o que rara vez se necesite. Se debe conocer que tipo de contenido necesita el usuario.                                                      | 5         | 5         | 4         | 5         |
| Ayuda a los usuarios a reconocer, diagnosticar y rec | Los mensajes de error deben expresarse en un lenguaje entendible, sin códigos de error.                                                                                                              | 4         | 3         | 5         | 4         |
| Ayuda y documentación                                | Lo ideal es que un sistema no necesite ninguna explicación adicional. Pero en algunos casos, se debe proporcionar una documentación para ayudar a los usuarios a entender cómo completar sus tareas. | 5         | 5         | 5         | 5         |

Tabla 3: Test de Usabilidad basada en Heurística de Jakob Nielsen [54]