



UNIVERSIDAD DE COLIMA

Facultad de Telemática

ANÁLISIS DE LA INFRAESTRUCTURA PEATONAL URBANA
APOYADO POR TÉCNICAS DE RECONOCIMIENTO DE
PATRONES E INTELIGENCIA ARTIFICIAL

Tesis que para obtener el grado de Maestro en Tecnologías de
Internet

Presenta:

Pedro Rincon Avalos

Asesores:

Dr. Mendoza Cano Oliver

Dra. Hevia Montiel Nidiyare

Colima, Col., México, junio 2024

Quisiera expresar mi más profundo agradecimiento y dedicar este trabajo a mi compañera de vida y esposa, Karla Michelle. Su presencia constante y su apoyo incondicional en las diversas etapas y momentos cruciales de mi vida han sido fundamentales para mi crecimiento personal y profesional.

Extiendo mi gratitud a mis padres, hermanos y profesores, quienes me brindaron invaluable oportunidades para desarrollar mis habilidades y fomentaron un ambiente que nutrió mi crecimiento. Su confianza y amor inquebrantable han sido pilares en mi camino hacia la realización de mis metas.

ÍNDICE

RESUMEN	1
Abstract.....	1
1 INTRODUCCIÓN	2
1.1 Problemática	3
1.2 Justificación	4
2 DEFINICIÓN DE OBJETIVOS	7
2.1 Objetivo General.....	7
2.2 Objetivos específicos	7
3 DESARROLLO	8
3.1 Marco Teórico.....	8
3.1.1 Inteligencia Artificial.....	8
3.1.1.1 Aprendizaje de máquina vs Aprendizaje Profundo	9
3.1.1.2 La neurona.....	10
3.1.1.3 Redes neuronales	11
3.1.1.4 Las funciones de activación	12
3.1.2 Arquitecturas para detección de objetos	15
3.1.2.1 YOLO.....	17
3.1.2.2 Fast R-CNN.....	17
3.1.2.3 Transformadores.....	18
3.1.2.4 U-Net.....	19
3.1.3 Python.....	21
3.1.3.1 Keras.....	22
3.1.4 GitHub y Git	22

3.1.5	La nube en IA	23
3.2	Metodología	23
3.2.1	Proceso de trabajo en la clasificación de imágenes con aprendizaje profundo ...	23
3.2.2	Área de estudio	24
3.2.3	Recopilación de las imágenes	25
3.2.3.1	Cámaras	25
3.2.4	Elección de elementos a identificar	27
3.2.5	Preprocesamiento de las imágenes	28
3.2.6	Herramienta de etiquetado	29
3.2.7	Dividir nuestra base de datos	29
3.2.8	Aumento de la base de datos	30
3.2.9	Entrenamiento del modelo	31
3.2.10	Evaluar el modelo	31
3.3	Análisis de resultados	34
3.3.1	Intersección sobre la unión	36
4	CONCLUSIONES	40
4.1	Trabajos Futuros	40
	REFERENCIAS	42

RESUMEN

El documento propone un método para elaborar un inventario de elementos urbanos en una zona de la Ciudad Conurbada Colima-Villa de Álvarez, usando técnicas de reconocimiento de patrones u objetos e inteligencia artificial. El objetivo es mejorar la planificación y el diseño urbano, así como la toma de decisiones y la gestión de emergencias. El documento consta de una introducción, una revisión bibliográfica, una metodología, un análisis de resultados y unas conclusiones. El método consiste en recolectar, preprocesar y etiquetar imágenes de las banquetas, los pasos peatonales y las ciclovías, y segmentarlas con un modelo basado en la arquitectura U-Net. El modelo obtuvo un rendimiento del 83% en el índice de intersección sobre la unión (IoU por sus siglas en inglés) en la segmentación de los elementos urbanos, y el método se considera útil para obtener un inventario de elementos urbanos en ciudades emergentes.

ABSTRACT

The document proposes a method for creating an inventory of urban elements in an area of the Colima-Villa de Álvarez Conurbated City, using pattern recognition techniques or objects and artificial intelligence. The aim is to enhance urban planning and design, as well as decision-making and emergency management. The document consists of an introduction, a literature review, a methodology, an analysis of results, and conclusions. The method involves collecting, preprocessing, and labeling images of sidewalks, pedestrian crossings, and bike lanes, and segmenting them with a model based on U-Net architecture. The model achieved an 83% performance on the Intersection over Union (IoU) index in urban element segmentation, and the method is considered useful for obtaining an inventory of urban elements in emerging cities.

1 INTRODUCCIÓN

Los autores Deng et al (2015); Soto-Cortés (2015); Williamson, (1988) afirman que, en décadas pasadas, el desarrollo de la sociedad obligaba a las personas a emigrar de zonas rurales a urbanas en búsqueda de un mejoramiento en su calidad de vida, situación que provocó que las ciudades crecieran y mejoraran sus capacidades para albergar a la población inmigra.

Esta situación está cambiando con el tiempo. La población empieza a desarrollarse con mayor velocidad en las ciudades de tamaño intermedio, ocasionado por un incremento descontrolado de las urbes y el desarrollo de las zonas rurales, también conocidas como ciudades emergentes, que tienden a mejorar sus condiciones con el objetivo de ofrecer un incremento en la calidad de vida para sus ocupantes, eliminando la necesidad de búsqueda de una mejor calidad de vida en las zonas urbanas (BID, 2016; Terraza et al., 2014).

Esta situación de crecimiento exponencial de las ciudades emergentes ha significado un enorme desafío para los gobiernos en América Latina y en el Caribe (ALC) en donde una mala gestión del crecimiento deriva en una ineficiencia en la capacidad de las ciudades de solventar y abastecer las necesidades que demanda la población de forma adecuada, como lo es la dotación y acceso a los servicios básicos y necesarios para el desarrollo de su población, evitando su crecimiento económico, social y cultural. así como demográficamente (BID, 2016; Terraza et al., 2014).

Entre 1950 y 2014, se registró un incremento considerable de la urbanización de ALC, de un 41 % a un 80 %, tendencia que se espera siga creciendo, que, junto con las condiciones de trabajo, vivienda y movilidad de ALC, se pronostica mayor desigualdad (BID, 2014, 2016; Terraza et al., 2014).Haga clic aquí para escribir texto.

En el Estado de Colima, los municipios de Colima y Villa de Álvarez son los de mayor predominio urbano, continuando una tendencia a un incremento en el área urbanizada. Desde el año 1990 el Nivel de Urbanización cambió de 91.81 % y 94.81 % a 93.52 % y 98.04 % en 2010 para cada ciudad respectivamente (Gobierno del estado de Colima, 2017).

Esto conlleva una problemática donde el desarrollo urbano crece más rápido que la planeación urbana derivando en proyectos poco funcionales, asentamientos irregulares y una incapacidad del gobierno por abastecer de servicios públicos básicos a los nuevos condominios.

Aunado a dicho hecho, la información geoespacial del INEGI solo se actualiza cada 10 años, lo que excluye las regiones más nuevas de las ciudades (Salinas et al., 2017). Esta situación deriva en problemas para las instituciones gubernamentales encargadas de diseñar proyectos dedicados a dotar de nueva infraestructura y mejorar la existente y los servicios urbanos para la ciudad al desconocer las características físicas más actuales de las áreas urbanas.

Realizar un levantamiento topográfico para un sector extenso, como las colonias, puede llevar una gran cantidad de horas de trabajo, un costo elevado, además de la importancia de disponer de un número adecuado de trabajadores para llevar a cabo el registro digital del estado urbano en el área de interés (Chen et al., 2018).

1.1 PROBLEMÁTICA

El crecimiento demográfico exponencial y la mala gestión urbana de los gobiernos en las ciudades de ALC generan desigualdad y bajo crecimiento económico ante la falta de trabajo, malas condiciones de vivienda y una capacidad ineficaz del transporte público, lo que perpetúa el crecimiento descontrolado y bajo crecimiento económico.

Una incorrecta gestión del gasto público conlleva una inadecuada priorización de las modalidades de transporte según lo establecido en la pirámide de movilidad, en donde se establecen las modalidades que deben priorizarse según su grado de sostenibilidad, preferencia vial y de inversión de obras públicas, colocando en la cima a los peatones y ciclistas. La ausencia de acera peatonal, falta de rampas y cruces peatonales bien emplazados, la correcta integración e implementación de ciclovías (Figura 1) son algunas de las problemáticas debidas a una gestión y planificación urbanas ineficientes. Dichos conceptos, son algunos de los elementos que se manejan en el proyecto de Calles Completas, mencionado en el Manual de Calles Completas por el Instituto de Políticas para el Transporte y el Desarrollo (Institute for Transportation and Development Policy – ITDP por sus siglas en inglés) y la Secretaría de Desarrollo Agrario, Territorial y Urbano (SEDATU).



Figura 1 Imagen ilustrativa de la problemática de mala planeación en el emplazamiento de elementos peatonales. Imagen recopilada de (Paredes-Bonilla et al., 2021).

1.2 JUSTIFICACIÓN

Una de las estrategias existentes para atender optimizar la movilidad y funcionamiento de las vialidades en las ciudades es la implementación del concepto de “Calle Completa”. Las vialidades o calles se definen el elemento de mayor importancia para el óptimo funcionamiento de las ciudades, es por este medio en donde la mayor parte de la población tiene acceso a la ciudad que habita, siendo el espacio público más grande y de mayor predominancia en las urbes.

El concepto de Calle Completa busca redistribuir físicamente el espacio vial para que cualquier usuario lo use correctamente sin importar su condición de discapacidad motriz, cognitiva, visual o social. Para lograrlo se deben priorizar los transportes no-motorizados principalmente,

atendiendo a lo visto en la nueva pirámide de movilidad que se puede ver en la Figura 2 (BID, 2014; Leal Vallejo et al., 2019; SEDATU, 2019).

La importancia de usar IA y visión computacional para identificar elementos propios de una Calle Completa, radica en la capacidad de evaluar la accesibilidad de estas, a fin de buscar redistribuir físicamente el espacio vial para que cualquier usuario lo use correctamente sin importar su condición de discapacidad motriz, cognitiva, visual o social. Para lograrlo, se deben priorizar los transportes no-motorizados principalmente, atendiendo a lo visto en la nueva pirámide de movilidad.



Figura 2 Nueva pirámide de movilidad. Imagen recopilada de (SEDATU, 2019).

2 DEFINICIÓN DE OBJETIVOS

2.1 OBJETIVO GENERAL

Generar un modelo de segmentación semántica con IA para elaborar un inventario de elementos urbanos en un tramo de la Avenida Calzada Galván y Boulevard Camino Real, en la Ciudad Conurbada Colima-Villa de Álvarez. Esto permitirá tener una visión detallada de la infraestructura urbana presente en el área (Figura 3), lo que permitirá mejorar la planificación y diseño urbano del área de estudio, así como la toma de decisiones y gestión de emergencias.



Figura 3 Representación esquemática de los resultados esperados al final el proyecto.

2.2 OBJETIVOS ESPECÍFICOS

- Realizar una investigación documental de la literatura existente, referente al uso de inteligencia artificial en las ciudades inteligentes.
- Definición, estructuración y justificación de la metodología para el levantamiento de las imágenes, así como la especificación del modelo de IA (Inteligencia Artificial).
- Capturar el banco de imágenes para la construcción de modelo de inteligencia artificial.
- Entrenamiento, validación y aplicación del modelo de inteligencia artificial.
- Desarrollo de un tablero digital para la exposición de resultados.

3 DESARROLLO

3.1 MARCO TEÓRICO

3.1.1 Inteligencia Artificial

A lo largo de la historia, encontramos que diversos autores han otorgado una descripción diferente al término de Inteligencia Artificial (Figura 4) en (Education, 2021) se define como una herramienta que utiliza computadoras y máquinas para replicar la habilidad de resolver problemas y tomar decisiones que posee la mente humana, en (Russell & Norvig, 2021) donde la IA “no sólo se necesita de la comprensión, sino también de la construcción de entidades inteligentes”, mientras que para este proyecto usaremos la descripción propuesta por (Burns et al., 2022) que la describe como una recreación de procesos de la inteligencia humana mediante simulaciones creadas por las máquinas.

Actualmente en el campo de la programación se definieron 6 disciplinas que tiene que cumplir una computadora para considerar que “puede pensar”: Procesamiento de lenguaje natural con cual la máquina pueda interpretar el lenguaje humano; Representación del conocimiento, actualmente el campo más complicado en el uso de la inteligencia artificial debido a que implica no solo reconocer el lenguaje humano, sino también el manipularlo y darle un sentido (El Naqa & Murphy, 2015); el Razonamiento automatizado busca obtener nueva información del conocimiento abstracto para darle un fin a diversas aplicaciones como la interpretación de videos (Liu et al., 2020); Aprendizaje de máquina que consiste en que la máquina “aprenda de un contexto” y pueda ser capaz de obtener posibles resultados para tareas que aún no se ven (El Naqa & Murphy, 2015); Visión computacional, con la cual se busca que los sistemas autónomos sean capaces de emular el comportamiento de la vista humana y poder reconocer patrones en imágenes (El Naqa & Murphy, 2015); y la robótica con la cual se puedan manipular objetos físicos (Burns et al., 2022).

La detección de objetos involucra tareas de clasificación y localización para ser capaces de usar estas metodologías en entornos reales, donde contamos con más de un elemento de interés en

una escena. Por lo tanto, podemos decir que la detección de objetos involucra dos pasos: el primero de ellos consiste en encontrar objetos en una escena para posteriormente clasificarlos en un segundo paso (Sharma, 2022).

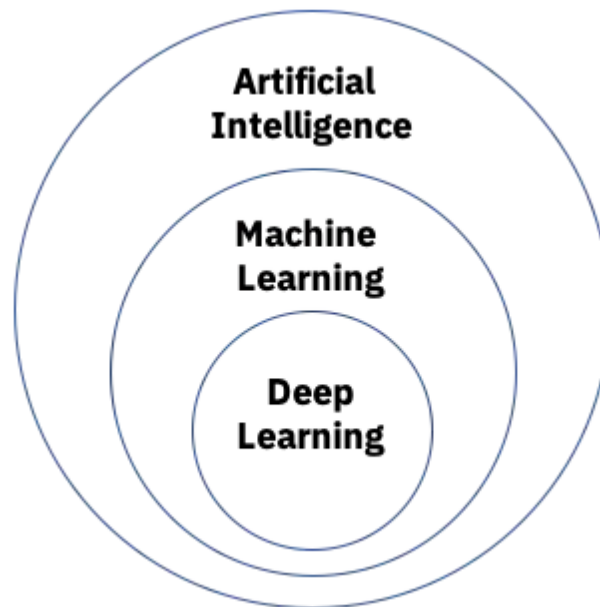


Figura 4 Representación esquemática de los niveles de profundidad que engloban el concepto de Inteligencia Artificial. Imagen recuperada de Education, 2021

3.1.1.1 Aprendizaje de máquina vs Aprendizaje Profundo

El machine learning y el deep learning son dos subconjuntos del campo de la inteligencia artificial. Ambos tienden a enfocarse en enseñar y brindar las pautas para que las máquinas sean capaces de aprender y tomar decisiones con base a un entrenamiento, en donde se analicen casos referenciales por medios del cual pueda tomar decisiones futuras según los parámetros enseñados.

El machine learning se fundamenta en algoritmos que se adaptan a los datos de entrenamiento y utilizan esta información para hacer predicciones y tomar decisiones. Dichos algoritmos pueden ser supervisados, no supervisados o de refuerzo. El aprendizaje de máquina es útil para

una amplia y diversa gama de aplicaciones, como lo es detección de fraudes, la clasificación de imágenes y la predicción de precios.

Estas técnicas son utilizadas en ese tipo de problemas específicos porque aprovechan la capacidad que presentan redes neuronales profundas con el objetivo de extraer y representar las características más relevantes y sobresaliente de los datos. Por ejemplo, en el reconocimiento de voz, el aprendizaje profundo puede capturar los patrones fonéticos, gramaticales y semánticos del lenguaje hablado y traducirlos a texto escrito. En la traducción automática, el aprendizaje profundo puede modelar las relaciones entre los idiomas de origen y destino y generar oraciones coherentes y naturales. En la identificación de objetos dentro de imágenes, el aprendizaje profundo puede detectar las formas, colores y texturas de los objetos y clasificarlos en categorías. Estas tareas requieren un alto nivel de abstracción y generalización que el aprendizaje de máquina convencional no puede lograr con la misma eficiencia y precisión.

En resumen, mientras que el aprendizaje de máquina es una herramienta amplia y versátil que puede aplicarse a una amplia gama de problemas, el aprendizaje profundo es una técnica más especializada que se enfoca en tareas concretas que requieren una gran cantidad de datos y procesamiento. Ambos campos son importantes en el ámbito de la inteligencia artificial y su uso depende del problema que se está intentando resolver.

3.1.1.2 La neurona

Dentro del contexto del aprendizaje profundo y según los autores (Choi et al., 2020; DotCSV, 2018; Montesinos-López et al., 2021), es importante destacar que las neuronas son la unidad básica del procesamiento de información. Cada capa de una red neuronal se encuentra compuesta por neuronas que procesan los datos de entrada y crean una salida. El término "neurona" es una analogía directa a las neuronas biológicas, que se ubican en el cerebro y son responsables del procesamiento de la información sensorial.

Cada neurona en una red neuronal se encuentra conectada a varias otras neuronas a través de conexiones llamadas sinapsis. Estas conexiones permiten que las neuronas trabajen en conjunto con la finalidad procesar la información de entrada y generar una salida. Cada neurona aplica

una función matemática simple a los datos de entrada que recibe, utilizando diferentes parámetros para ajustar su comportamiento.

Además de las neuronas, otro componente importante en una red neuronal son los pesos, que consisten en valores numéricos relacionados a cada una de las conexiones sinápticas entre las neuronas. Estos pesos son ajustados a lo largo del entrenamiento de la red neuronal y su valor final determina la eficacia de la red para brindar solución a un problema dado.

En resumen, las neuronas son un componente fundamental en las redes neuronales y su función esencial es el procesamiento y manejo de la información de entrada y la generación de una salida. El ajuste de los parámetros de cada neurona y de los pesos de las conexiones sinápticas durante el entrenamiento de la red neuronal es lo que permite que la red aprenda a dar solución a un problema específico (Figura 5).

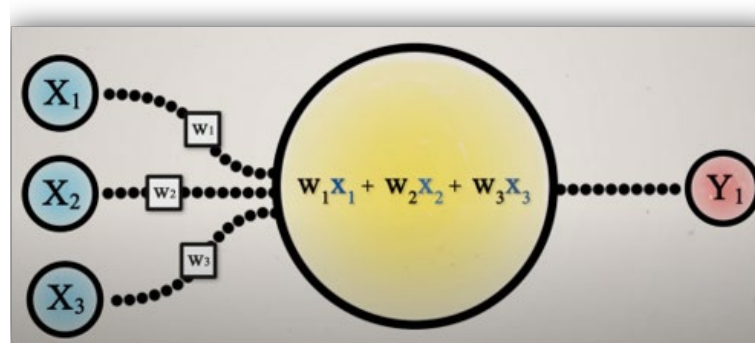


Figura 5 Representación esquemática de una neurona en aprendizaje profundo.

3.1.1.3 Redes neuronales

Las redes neuronales son un algoritmo de aprendizaje automático que imita cómo las neuronas procesan la información en el cerebro humano. Estas redes se encuentran compuestas de nodos interconectados llamados “neuronas”, las cuales trabajan en conjunto para resolver problemas complejos. Hay diversos tipos de redes neuronales, cada uno cuenta con sus propias aplicaciones y características (Heaton, 2018)

La red neuronal más simple es la red neuronal de una capa, también conocida como perceptrón, cuya representación gráfica podemos ver en la Figura 5. La red es de una sola capa de neuronas

y se usa para problemas de clasificación binaria. Sin embargo, la mayoría de los problemas de inteligencia artificial requieren redes más complejas (LeCun et al., 2015).

Las redes neuronales multicapa consisten en redes con múltiples capas de neuronas interconectadas. Estas redes son capaces de procesar datos más complejos y resolver problemas más difíciles que las redes de una sola capa. Algunos ejemplos de redes neuronales multicapa incluyen (Strudel et al., 2021)

Las redes neuronales convolucionales (CNN por sus siglas en inglés) son especialmente útiles para el procesamiento de imágenes, por lo cual son utilizadas en aplicaciones de reconocimiento facial y la detección de objetos. Estas redes utilizan filtros convolucionales para identificar y extraer características específicas de una imagen y reducir la cantidad de datos que la red necesita procesar (Heaton, 2018).

Las redes neuronales recurrentes (RNN) son adecuadas para el procesamiento de datos secuenciales, como el procesamiento del lenguaje natural y la predicción del tiempo. Estas redes utilizan conexiones recurrentes para recordar información anterior y procesar secuencias de datos (Heaton, 2018).

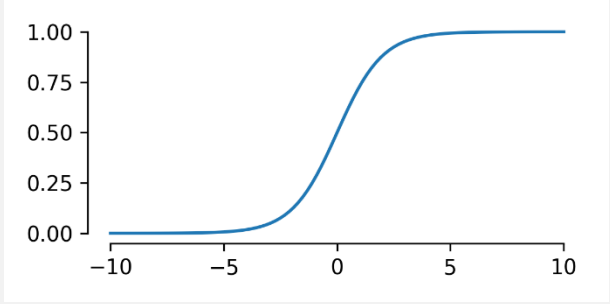
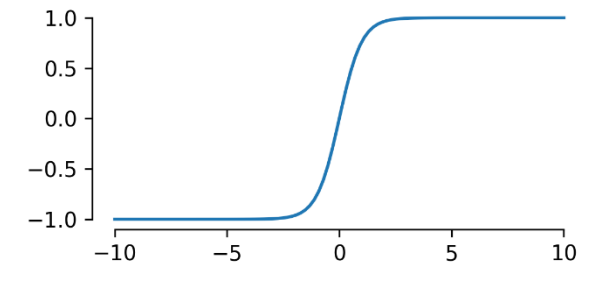
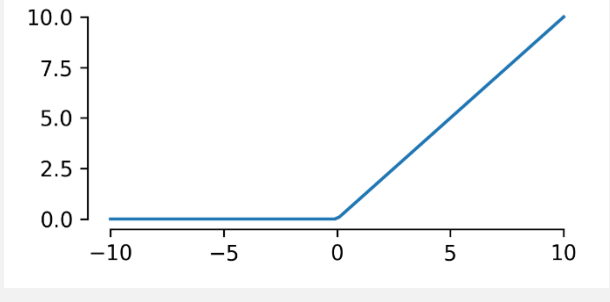
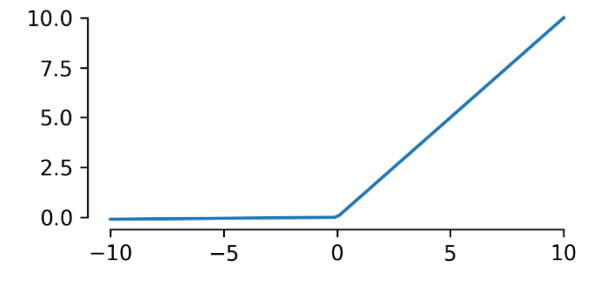
En resumen, cada tipo de red neuronal tiene sus propias fortalezas y debilidades, y se adapta mejor a ciertos tipos de problemas. Por lo tanto, resulta de vital importancia seleccionar la red neuronal adecuada para la tarea específica en cuestión.

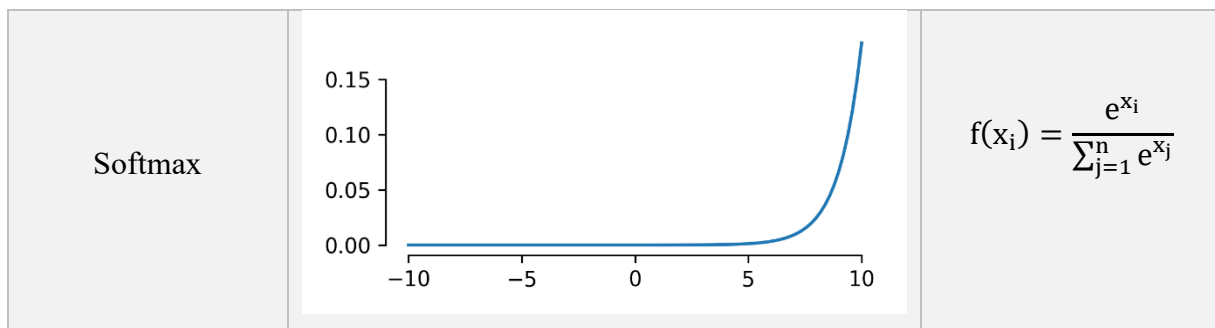
3.1.1.4 Las funciones de activación

En la Inteligencia Artificial, las funciones de activación son un componente clave de las redes neuronales artificiales. Estas funciones se aplican a los valores de entrada de una neurona y determinan la salida o activación de esta. Las funciones de activación son necesarias para incluir no linealidad en los modelos de redes neuronales, lo que permite que las redes sean más expresivas y puedan aproximar funciones más complejas (Heaton, 2018).

(Maas et al., 2013)

Tabla 1 Análisis comparativo de las principales funciones de activación utilizadas en redes neuronales.

Nombre	Imagen	Ecuación
Sigmoide	 <p>The graph shows the Sigmoid function, which is an S-shaped curve. The x-axis ranges from -10 to 10, and the y-axis ranges from 0.00 to 1.00. The curve starts near 0 for negative x, passes through (0, 0.5), and approaches 1 for positive x.</p>	$f(x) = \frac{1}{1 + e^{-x}}$
Tangente hiperbólica	 <p>The graph shows the Hyperbolic Tangent function, which is an S-shaped curve. The x-axis ranges from -10 to 10, and the y-axis ranges from -1.0 to 1.0. The curve starts near -1 for negative x, passes through (0, 0), and approaches 1 for positive x.</p>	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Rectificada lineal (ReLU)	 <p>The graph shows the Rectified Linear Unit (ReLU) function. The x-axis ranges from -10 to 10, and the y-axis ranges from 0.0 to 10.0. The function is zero for all negative x and increases linearly with a slope of 1 for all positive x.</p>	$f(x) = \max(0, x)$
Rectificada lineal con fuga (Leaky ReLU)	 <p>The graph shows the Leaky Rectified Linear Unit (Leaky ReLU) function. The x-axis ranges from -10 to 10, and the y-axis ranges from 0.0 to 10.0. The function is zero for all negative x and increases linearly with a slope of 1 for all positive x.</p>	$f(x) = \max(\alpha x, x)$



La función ReLU consiste en una función no lineal que retorna 0 si la entrada es negativa y la entrada misma si es positiva. A diferencia de otras funciones de activación, ReLU no sufre del problema de desvanecimiento del gradiente, que ocurre cuando la función de activación se satura en regiones en las que la derivada es cercana a cero, lo que puede impedir el entrenamiento efectivo de la red neuronal (Glorot et al., 2011). Además, ReLU tiene un tiempo de cómputo más rápido que otras funciones de activación como lo resulta la tangente hiperbólica, lo que la hace más adecuada para modelos de redes neuronales de gran escala (Krizhevsky et al., 2012).

Otras funciones de activación comúnmente utilizadas (Tabla 1) son la función sigmoide, la tangente hiperbólica y la función de activación lineal rectificada con fuga (Leaky ReLU). La función sigmoide es una función no lineal que retorna un valor entre 0 y 1, lo que la vuelve adecuada para modelar probabilidades o clasificaciones binarias. Sin embargo, la función sigmoide también sufre del problema de desvanecimiento del gradiente y tiene un tiempo de cómputo más lento que ReLU (Glorot et al., 2011). La tangente hiperbólica es una función no lineal que retorna un valor entre -1 y 1, lo que la hace más simétrica y centrada que la función sigmoide. La tangente hiperbólica también tiene el problema de desvanecimiento del gradiente, pero en menor medida que la función sigmoide (Heaton, 2018). La función de activación lineal rectificada con fuga (Leaky ReLU) es una variante de la función ReLU que permite un pequeño valor positivo para las entradas negativas, lo que evita que las neuronas mueran cuando solo reciben entradas negativas. La función Leaky ReLU tiene un mejor rendimiento que ReLU en algunos casos, pero también puede causar inestabilidad numérica si el valor de fuga es demasiado grande (Maas et al., 2013). Finalmente, la función softmax consiste en una función de activación que se emplea para la clasificación multiclase, ya que retorna un vector de

probabilidades que suman 1. La función softmax resulta similar a la función sigmoide, pero generalizada para múltiples clases (Heaton, 2018).

En resumen, las funciones de activación son una parte fundamental de las redes neuronales y la selección de la función correcta puede tener una influencia significativa en el desempeño del modelo. ReLU es una función de activación simple y efectiva que ha demostrado ser adecuada para una gran parte de las aplicaciones de redes neuronales en la actualidad.

3.1.2 Arquitecturas para detección de objetos

La detección y segmentación de objetos es uno de los campos más relevantes en visión computacional y ha sido objeto de investigación durante muchos años. Con el avance de la tecnología y la disponibilidad de grandes conjuntos de datos, se han desarrollado varias arquitecturas para mejorar la precisión y la velocidad de estos sistemas.

Entre las arquitecturas más populares se encuentran las redes neuronales convolucionales (CNN), las cuales han logrado demostrar un gran rendimiento en tareas de detección y segmentación de objetos en imágenes. Una de las arquitecturas más conocidas de CNN es YOLO (You Only Look Once por sus siglas en inglés, Redmon & Farhadi, 2018), que realiza detección de objetos en tiempo real con una sola pasada a través de la red.

Otra arquitectura popular es la red neuronal convolucional en cascada (Cascade RCNN por sus siglas en inglés), que ha demostrado un excepcional rendimiento en la detección de objetos en imágenes con una alta resolución. Cascade RCNN (Cai et al., 2021) usa múltiples etapas de detección para obtener una alta precisión.

Para la segmentación de objetos, una de las arquitecturas más utilizadas es la red totalmente convolucional (FCN por sus siglas en inglés), como ejemplo ilustrativo tenemos la Figura 6. FCN se ha utilizado en varias aplicaciones, como la segmentación de objetos en la clasificación de imágenes y la segmentación de objetos en tiempo real en video.

Además, también existen arquitecturas basadas en el uso de redes neuronales recurrentes (RNN) y redes neuronales generativas adversarias (GAN) para la detección y segmentación de objetos. Estas arquitecturas han demostrado una mayor eficiencia en la detección y segmentación de objetos en escenarios complejos y en la generación de imágenes realistas.

Dentro de la rama de la detección de objetos contamos con dos arquitecturas principales: Detectores de objetos de una capa y Detectores de objetos de dos etapas (Single-Stage Object Detectors y Two-Stage Object Detectors, respectivamente); su principal diferencia radica en que las segundas arquitecturas se utilizan en situaciones donde un elemento en una escena puede contener otros elementos (Sharma, 2022).

En conclusión, las arquitecturas para la detección y segmentación de objetos en visión computacional son diversas y se están investigando continuamente nuevas formas de mejorar su eficacia y eficiencia. Las arquitecturas mencionadas anteriormente son solo algunas de las más populares y utilizadas en la actualidad (Goodfellow et al., 2020; Redmon et al., 2016; Ren et al., 2017)

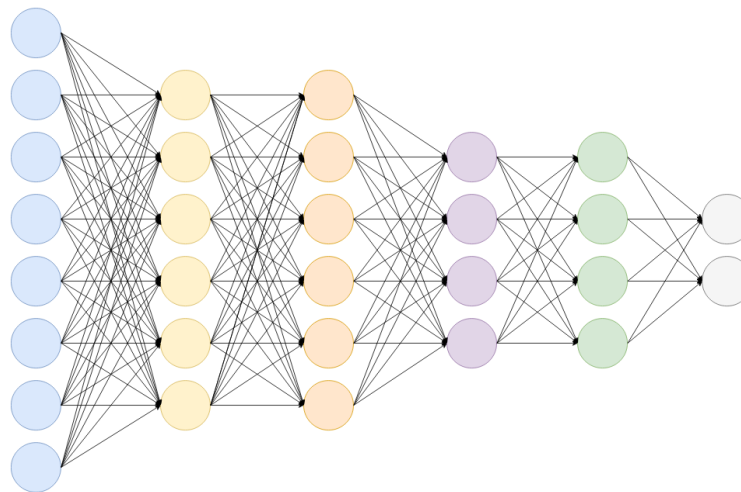


Figura 6 Ilustración de una red completamente conectada. Ilustración extraída de Long et al., 2014

3.1.2.1 YOLO

YOLO (You Only Look Once) es una arquitectura de detección de objetos en tiempo real que se caracteriza por su precisión y velocidad. A diferencia de otras arquitecturas de detección de objetos que requieren una región de interés previa (ROI (región de interés)) para realizar la detección, YOLO segmenta la imagen en una cuadrícula y predice la clase y la ubicación del objeto en cada celda. Esto permite que YOLO sea una arquitectura extremadamente rápida en la detección en tiempo real de objetos.

YOLO también es conocido por su capacidad para detectar múltiples objetos en una sola imagen. Además, a diferencia de otras arquitecturas que requieren múltiples pasadas para la detección de objetos en diferentes tamaños y escalas, YOLO utiliza una sola red para detectar objetos en diferentes tamaños y escalas.

La arquitectura YOLO ha sido utilizada en una diversidad y variedad de aplicaciones, desde la detección de objetos en imágenes médicas hasta la detección de objetos en videos de vigilancia. YOLO ha sido continuamente mejorado y optimizado en diferentes versiones, como YOLOv2, YOLOv3 y YOLOv4 (Redmon et al., 2016; Redmon & Farhadi, 2018).

3.1.2.2 Fast R-CNN

Fast R-CNN (Figura 7) es un enfoque para la detección de objetos en imágenes que emplea una única red neuronal convolucional (CNN) para ejecutar tanto la extracción de características como la detección de objetos. A diferencia de su predecesor, R-CNN, el cual requería la ejecución de la CNN varias veces por imagen, Fast R-CNN realiza la extracción de características de la imagen una sola vez y utiliza esta información para detectar objetos. Esto hace que el enfoque sea mucho más rápido y eficiente en términos de memoria. Además, Fast R-CNN hace uso de una capa RoI (región de interés) que permite detectar objetos en diferentes tamaños y escalas. Esta capa identifica y extrae características de cada región de la imagen propuesta y las utiliza para la detección final de objetos.

El modelo ha demostrado ser altamente preciso en la detección de objetos en imágenes y ha superado a su predecesor en aspectos de velocidad y eficiencia. Fast R-CNN también ha sido utilizado como base para otros enfoques más avanzados, como Faster R-CNN y Mask R-CNN (Girshick, 2015)

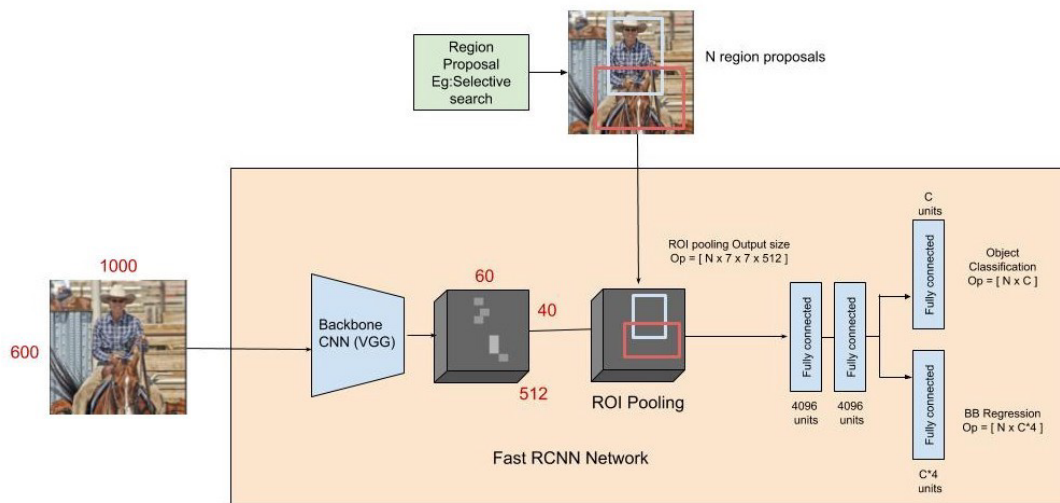


Figura 7 Arquitectura de las Fast R-CNN. Ilustración extraída de Ananth, 2021

3.1.2.3 Transformadores

Los transformadores, que originalmente fueron introducidos en el procesamiento del lenguaje natural, también han demostrado ser eficientes y efectivos en la visión computacional. En particular, se han utilizado en la tarea de detección de objetos y clasificación de imágenes, superando incluso a los métodos tradicionales como las redes neuronales convolucionales (CNN) (Tan et al., 2021)

Un caso de esto es la arquitectura DETR (Transformador de Detección sin Anclaje), que aplica un transformador para identificar objetos en una imagen sin la necesidad de usar anclajes previamente definidos. Este método ha mostrado ser efectivo en la detección de objetos en

tiempo real y en el aumento de la precisión de la detección en objetos pequeños (Xie et al., 2021; Zhu et al., 2021).

Los anclajes son regiones que se seleccionan en una imagen para tratar de encontrar objetos que se parecen a ciertas formas y tamaños. Sin embargo, los anclajes pueden fallar muchas veces y no funcionar bien para objetos que tienen formas raras o que no se conocen de antemano. Por eso, la detección sin anclajes es un método que no usa anclajes y trata de encontrar objetos directamente usando la información que hay en la imagen.

Otra aplicación de los transformadores en la visión computacional es en la generación de texto enriquecido para describir imágenes. Un ejemplo de esto es el modelo ViT (Transformador de Visión), que utiliza un transformador para aprender representaciones de imágenes y generar una descripción acerca del contenido de la imagen en forma de texto (Lin et al., 2022).

En resumen, los transformadores han logrado demostrado ser una herramienta efectiva en la visión computacional, permitiendo la detección de objetos sin anclajes y la generación de descripciones de imágenes. Estas aplicaciones están en constante evolución y mejora, lo que sugiere que los transformadores seguirán siendo una herramienta importante en el campo de la visión por computadora.

3.1.2.4 U-Net

U-Net (Figura 8) es una arquitectura de red neuronal convolucional profundamente supervisada para la segmentación de imágenes, introducida por (Ronneberger et al., 2015). Esta arquitectura se utiliza ampliamente en aplicaciones médicas y biológicas, donde se requiere una segmentación precisa de las imágenes, como la segmentación de células o estructuras anatómicas en imágenes médicas.

La estructura de la red consta de dos partes: la primera mitad actúa como un codificador que utiliza capas convolucionales para extraer características de la imagen de entrada, mientras que la segunda mitad actúa como un decodificador que utiliza capas deconvolucionales para generar una máscara de segmentación precisa. Además, U-Net también utiliza saltos de conexión (skip

connections) para permitir una transferencia de información de nivel superior y así lograr evitar la pérdida de detalles finos durante el proceso de convolución.

La arquitectura U-Net ha mostrado ser muy efectiva en la segmentación de imágenes, y ha sido utilizada en varias aplicaciones médicas y biológicas, como la segmentación de células por medio de imágenes microscópicas, la segmentación de tumores en imágenes de resonancia magnética, entre otros. Además, ha sido objeto de varias mejoras y extensiones, como U-Net++ y Attention U-Net.

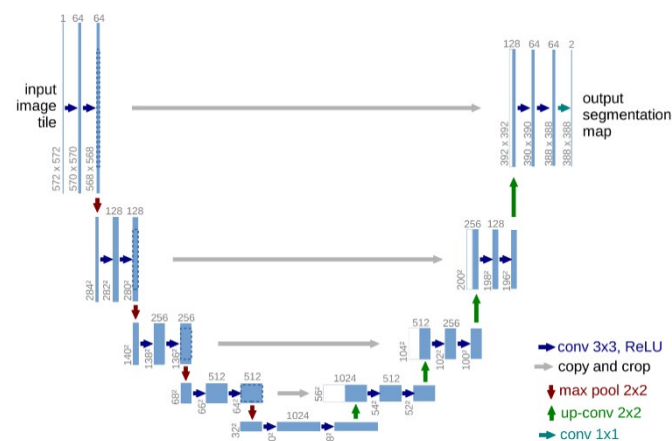


Figura 8 Arquitectura U-Net. Imagen extraída de Davis et al., 1993

Si bien U-Net se popularizó por su alto rendimiento en la segmentación de imágenes médicas, también ha mostrado su eficacia en otras áreas de aplicación. Por ejemplo, en la identificación de defectos en la industria textil (Huang & Xiang, 2022) y en la clasificación de especies de plantas en la agricultura (dos Santos Oliveira et al., 2022). Incluso se ha utilizado en la segmentación de carreteras y edificios en imágenes satelitales (Li et al., 2020).

La arquitectura U-Net se ha demostrado efectiva en una variedad y diversidad de aplicaciones fuera de la medicina debido a su capacidad para realizar segmentaciones precisas incluso en imágenes con ruido y objetos pequeños. Su capacidad para aprender de características en diferentes escalas y la incorporación de capas de convolución transpuestas permiten la recuperación de detalles finos en la imagen de salida (Ronneberger et al., 2015).

3.1.3 Python

Python es un lenguaje de programación ampliamente y comúnmente empleado en la visión computacional debido a su fácil aprendizaje, alto rendimiento y disponibilidad de bibliotecas de código abierto. Entre las bibliotecas más populares para la visión computacional en Python se encuentran OpenCV, Pillow, Scikit-image y TensorFlow.

OpenCV es una biblioteca de visión computacional de código abierto y gratuita que tiene como objetivo proporcionar una infraestructura común para aplicaciones de visión. Es ampliamente utilizada en aplicaciones de detección de objetos, seguimiento, reconocimiento facial y más. Scikit-image es una biblioteca que proporciona herramientas para el procesamiento de imágenes en Python y se centra en la facilidad de uso y la claridad del código. Pillow es una biblioteca de procesamiento de imágenes de Python que se emplea para cargar, manipular y guardar diferentes formatos de imágenes.

TensorFlow es una biblioteca de aprendizaje automático de código abierto desarrollada por Google. Se utiliza comúnmente para crear y entrenar modelos de aprendizaje automático, incluidos modelos para aplicaciones de visión computacional. La biblioteca proporciona una gran cantidad de funciones útiles para trabajar con datos de imágenes, como la creación de redes neuronales convolucionales.

Python y sus bibliotecas para la visión computacional se usan en aplicaciones, como la detección de objetos en tiempo real, la identificación de patrones en imágenes médicas y la automatización de la inspección industrial. Además, la comunidad de Python es activa y proporciona soporte y actualizaciones constantes a las bibliotecas, lo que asegura su relevancia y eficacia en el tiempo.

En conclusión, Python y sus bibliotecas para la visión computacional son una opción popular para el desarrollo de aplicaciones en esta área debido a su facilidad de uso, disponibilidad de bibliotecas de código abierto y su eficacia demostrada en una variedad de aplicaciones prácticas (Abadi et al., 2011; Bradski, 2000; van der Walt et al., 2014).

3.1.3.1 Keras

Keras es una biblioteca de aprendizaje profundo escrita en Python que comúnmente se utiliza para diseñar, entrenar y evaluar redes neuronales. Su facilidad de uso y flexibilidad la hacen muy popular en la comunidad de la visión computacional. Con Keras, los desarrolladores pueden construir modelos de manera rápida y sencilla utilizando una API intuitiva y modular. Además, Keras es compatible con TensorFlow, una biblioteca de código abierto desarrollada por Google para tareas de aprendizaje automático, lo que permite una integración perfecta entre ambas herramientas (Chollet, 2015)

En la visión computacional, Keras se ha utilizado para la clasificación de imágenes, detección de objetos, segmentación de imágenes y otras tareas relacionadas. La biblioteca ha demostrado ser efectiva en la creación de modelos de aprendizaje profundo precisos y eficientes en términos de tiempo y recursos computacionales (Géron, 2019; Reddy, 2018).

3.1.4 GitHub y Git

GitHub es una plataforma en línea que se utiliza para alojar proyectos de software y facilitar la colaboración entre desarrolladores. Los usuarios son capaces de almacenar y compartir código fuente, realizar seguimientos de los cambios en el código y colaborar con otros miembros del equipo. Además, GitHub ofrece herramientas de gestión de proyectos, seguimiento de errores y colaboración en equipo.

Git es un sistema de control de versiones distribuido y gratuito que suele emplearse para rastrear los cambios y diferencias en el código fuente de un proyecto. Permite a los desarrolladores trabajar en el mismo código fuente de forma simultánea y mantener un historial completo de todos los cambios ejecutados en el proyecto. Además, Git permite a los desarrolladores trabajar sin conexión a Internet y sincronizar los cambios en un momento posterior.

GitHub y Git son herramientas esenciales para desarrollar software y son muy utilizados por la comunidad de desarrolladores en todo el mundo. Al usar Git junto con GitHub, los

desarrolladores pueden trabajar juntos en proyectos de software, compartir código y colaborar en tiempo real.

3.1.5 La nube en IA

El uso de la nube en el desarrollo de modelos de IA ha revolucionado la forma en que se aborda el aprendizaje automático y el aprendizaje profundo. Herramientas como Google Colab permiten a los desarrolladores acceder a recursos de cómputo de alta capacidad y almacenar grandes conjuntos de datos en la nube, lo que facilita el desarrollo y entrenamiento de modelos de IA complejos.

Además, el uso de la nube permite a los desarrolladores colaborar en tiempo real y compartir código y datos de manera eficiente. Esto ha permitido el desarrollo de modelos de IA más precisos y sofisticados, y ha acelerado el progreso en el campo de la IA.

En resumen, el uso de la nube en el desarrollo de modelos de IA ha permitido a los desarrolladores acceder a recursos de cómputo de alta capacidad y colaborar en tiempo real, lo que ha acelerado el progreso en el campo de la IA.

3.2 METODOLOGÍA

3.2.1 Proceso de trabajo en la clasificación de imágenes con aprendizaje profundo

En los procesos de construcción de algoritmos de clasificación de imágenes usando aprendizaje profundo, es común seguir una ruta de trabajo que consta de cuatro pasos principales. El primer paso es obtener la base de datos, seguido por la preparación de los datos para su posterior procesamiento. Luego, se procede a la construcción del modelo de aprendizaje profundo, y finalmente se evalúa su desempeño mediante pruebas y ajustes necesarios.

Aunque estos procesos son estándares en la mayoría de los modelos, pueden requerir modificaciones para adecuarse a las necesidades de proyectos específicos. Es importante destacar que una ruta de trabajo clara y bien definida permite una construcción eficiente y efectiva de los algoritmos de clasificación de imágenes.

3.2.2 Área de estudio

La región de estudio se encuentra ubicada en el estado de Colima, que forma parte de los 32 estados que conforman México. La ciudad capital del estado es el municipio de Colima, que está conurbado con Villa de Álvarez, ambos situados entre las coordenadas 19°14'35" latitud norte, 103°43'41" longitud oeste, a una altitud de 485 msnm y una superficie territorial de aproximadamente 66 km², formando la Ciudad Conurbada Colima-Villa de Álvarez (CCCVA). Es dentro de esta área donde se levantará las imágenes.

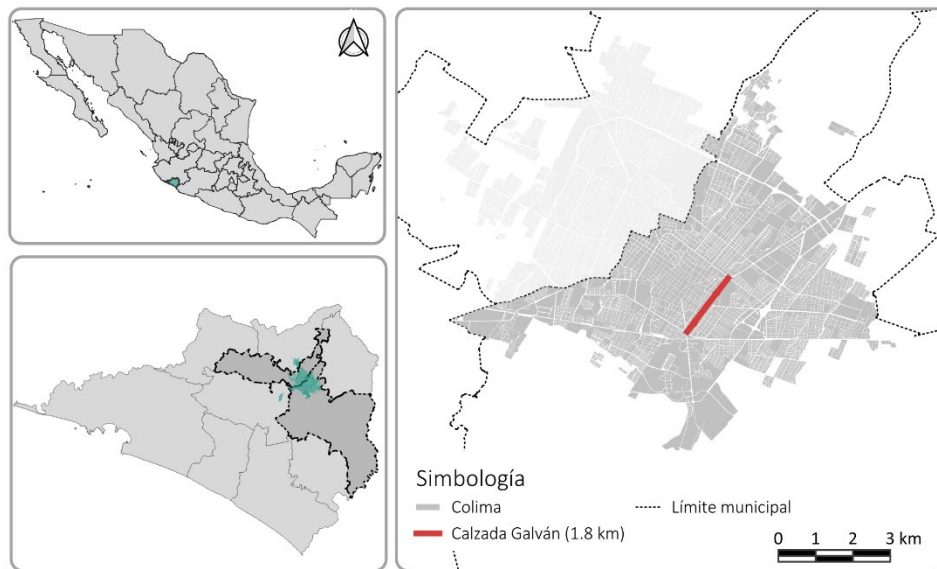


Figura 9 Área de estudio. Elaboración propia a partir de la información del INEGI.

El área de estudio para el levantamiento digital se dividió en dos tramos (Figura 9). El primer tramo es la avenida Calzada Galván, ubicada en el municipio de Colima, limitada por la avenida San Fernando al norte y la avenida 20 de noviembre al sur. Este tramo cuenta con 1.8 km por sentido, lo que representa un recorrido total de 3.6 km. Además, fue intervenido a mediados de 2021 (Figura 10).



Figura 10 Imagen capturada sobre el área de estudio.

El segundo tramo es el Boulevard Camino Real, que se extiende entre la Avenida San Fernando al sur y la Avenida Sevilla del Río al norte, con un recorrido de 0,736 km por sentido, lo que da 1.472 km en este tramo.

Considerando ambos tramos en los que se dividió la zona de estudio, se tiene un recorrido de 5.072 km.

3.2.3 Recopilación de las imágenes

3.2.3.1 Cámaras

GoPro HERO 8 Black



Figura 11 Cámara GoPro Hero 8 Black

Tabla 2 Ficha técnica de la cámara de video GoPro Hero 8 Black. Datos recopilados de GoPro, 2019.

Características	GoPro Hero 8 Black
Resolución de foto	12 megapíxeles
Resolución de video	4K@60, 1080p@240, cámara lenta 8x
Micrófono	Tres micrófonos, con reducción de ruido de viento.
Modos de disparo	Ráfaga, foto nocturna, , Superfoto, fotografía secuencial (estándar y nocturna), grabación en bucle, cámara lenta (8x), vídeo TimeWarp 2.0
GPS	Sí
Estabilización de video	HyperSmooth 2.0 y Boost
Conectividad	WiFi, GPS y Bluetooth

El primer paso de esta ruta de trabajo implica la recolección de datos para el entrenamiento y validación del modelo, junto con el etiquetado de objetos o elementos que se desean identificar. Es fundamental contar con una cantidad equilibrada de etiquetas para cada elemento, evitando el sobre ajuste del modelo. Es importante que el modelo (Figura 11) aprenda a encontrar las diferencias entre los elementos, utilizando una etiquetación adecuada. Este paso es crucial y fundamental para el éxito del modelo de clasificación de imágenes, ya que la calidad y cantidad de los datos recolectados influirá directamente en la precisión y eficacia del algoritmo (Tabla 2).

Para la recolección de material visual en el área de estudio, se llevó a cabo un meticuloso proceso que involucró recorrer las aceras de las vialidades en ambos sentidos. Se procuró cubrir todos los ángulos importantes por medio de un detallado recorrido peatonal, utilizando dispositivos y herramientas de última generación, como una cámara digital y accesorios de geolocalización de las imágenes. Durante el levantamiento de las imágenes, se emplearon accesorios especializados para facilitar la captura de éstas. En primer lugar, se utilizó un arnés que se colocó a la altura del torso para, posteriormente, ubicar la cámara a mitad del pecho (Figura 12).



Figura 12 Arnés de montaje en el pecho de la cámara de video GoPro Hero 8 Black.

Tras completar el proceso de levantamiento del material necesario para el proyecto, se recopilaron alrededor de 40 minutos de video, capturados a una resolución de 1920x1080 pixeles y a un promedio de 60 fotogramas por segundo. En total, se generaron 144,000 fotogramas disponibles para ser empleados en el proceso de entrenamiento y validación del modelo. Todo este riguroso proceso permitió recopilar un material de alta calidad que será clave para obtener resultados precisos y confiables.

3.2.4 Elección de elementos a identificar

Las vías urbanas tienen una importancia crucial en el funcionamiento de una ciudad. No solo son infraestructuras para la movilidad, sino que también juegan un papel importante en la vida pública de las ciudades. Por lo tanto, deberán ser diseñadas como espacios públicos para la

población en general y no solo como vías de tránsito. El diseño de las vías debe procurar un uso equitativo por parte de todos los usuarios, con especial énfasis en los peatones y ciclistas, atendiendo lo especificado en la pirámide de movilidad.

Por esta razón, se ha decidido analizar elementos de infraestructura peatonal y ciclista, seleccionando los siguientes elementos para su detección: aceras, cruces peatonales y ciclovías/ciclo bandas. Estos elementos corresponden a los dos primeros peldaños de la jerarquía de movilidad y son esenciales para garantizar la comodidad y seguridad de los usuarios más vulnerables de las vías urbanas. Analizar estos elementos permitirá mejorar la movilidad urbana, haciendo de las vías espacios más seguros y amigables para los peatones y ciclistas, lo que fomentará el uso de medios de transporte más sostenibles y saludables para la ciudadanía.

3.2.5 Preprocesamiento de las imágenes

El preprocesamiento de las imágenes es una etapa crítica en el desarrollo de un modelo de detección de elementos urbanos, y en este proyecto no fue la excepción. Esta fase consistió en pasos definidos y establecidos para lograr que las imágenes pudieran utilizarse en el entrenamiento y validación del modelo, ya que las capacidades del equipo de cómputo usaban tenía recursos limitados.

El primer paso del proceso fue extraer los fotogramas necesarios de cada video para construir el "banco de materiales". A partir de los archivos exportados de la herramienta de etiquetado CVAT, se comenzó la construcción de archivos tipo .png a modo de matriz multidimensional, según la cantidad de tipos de etiquetas del proyecto para la etapa de entrenamiento. Posteriormente para la realización de máscaras se emplearon archivos .tif, debido a que permitían un manejo más cómodo de las multiclases, eran esenciales para la construcción de los datasets que se usarían para entrenar y validar el modelo.

El segundo paso del proceso de preprocesamiento de las imágenes consistió en reducir la resolución de las imágenes de entrenamiento y validación. Se redujo la resolución de las imágenes de 1920x1080 píxeles a 256x144 píxeles. Esto se hizo para poder cumplir con las limitaciones de recursos del equipo de cómputo utilizado. Aunque esta reducción de resolución

puede resultar en la pérdida de detalles finos de las imágenes, es necesario para poder utilizar un conjunto de datos de un tamaño manejable para el equipo de cómputo.

3.2.6 Herramienta de etiquetado

Entre las diversas herramientas disponibles en internet para el etiquetado de imágenes se tomó la decisión de usar CVAT (<https://openvinotoolkit.github.io/cvat/about/>) por su facilidad para ser usado directamente en el explorador y a su vez permitir que diversas personas en distintos equipos de cómputo puedan participar en el mismo proyecto de etiquetado, asignándoles un grupo de imágenes a cada individuo y poder obtener la mayor cantidad de imágenes etiquetadas, además de ser una herramienta gratuita.

En resumen, el preprocesamiento de las imágenes fue un paso crucial para la construcción de un modelo de detección de elementos urbanos efectivo. Este proceso permitió que las imágenes pudieran ser utilizadas en el entrenamiento y validación del modelo, a pesar de las limitaciones de recursos del equipo de cómputo utilizado. La extracción de los fotogramas necesarios, la construcción de archivos .png y la reducción de la resolución de las imágenes fueron los pasos necesarios para poder avanzar en el proyecto de detección de elementos urbanos.

3.2.7 Dividir nuestra base de datos

El desarrollo de modelos de inteligencia artificial, llámese aprendizaje de máquina o aprendizaje profundo, requieren de dos grupos principales de datos independientes uno de otro:

- Información de entrenamiento
- Información de prueba

El primer grupo consiste en las imágenes que el modelo usará para aprender a identificar los elementos asociados a las etiquetas. Usualmente, este grupo es el que contiene la mayor cantidad de imágenes, aunque no existe un consenso sobre qué porcentaje de imágenes son necesarias, una práctica común es usar dos terceras partes, tres cuartas partes y nueve partes de diez (Figura 13).

El segundo grupo consta de las imágenes que serán usadas para que el modelo autoevalúe su desempeño, comparándolo con las imágenes previamente etiquetadas asociadas a este grupo. El porcentaje de imágenes asignado consta del resto de datos de la base de datos, respectivamente son: Una tercera parte, una cuarta parte y una parte de diez.

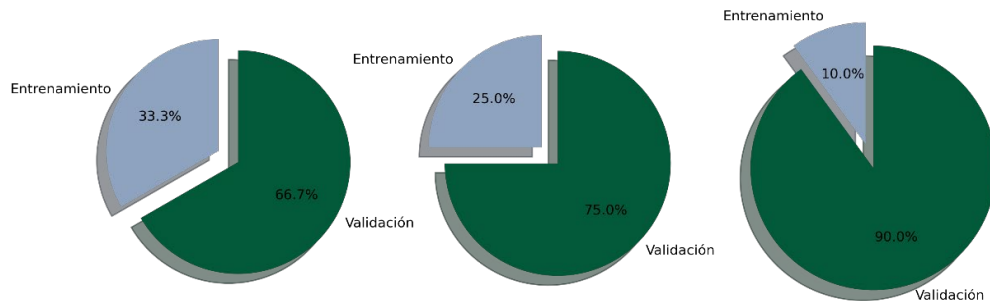


Figura 13 Distintas distribuciones de datos destinados al entrenamiento y validación. Elaboración propia.

3.2.8 Aumento de la base de datos

Para mejorar el rendimiento del modelo y evitar en la medida de lo posible el sobreajuste, se aplicaron técnicas de aumento de datos para expandir el conjunto de imágenes disponibles en nuestra base de datos. El aumento o incremento de datos es una práctica común en el aprendizaje profundo, especialmente en tareas de visión por computadora, que permite generar variaciones de las imágenes originales mediante transformaciones geométricas y de apariencia, aumentando así la diversidad de datos y reduciendo la probabilidad de sobreajuste (Perez & Wang, 2017).

En nuestro estudio, se utilizaron las siguientes técnicas de aumentación de datos:

1. Espejear las imágenes: Consiste en reflejar horizontalmente las imágenes originales, generando una versión "espejada" de cada una. Esta técnica es especialmente útil para aumentar la invariancia del modelo a las diferencias de orientación en las imágenes (Krizhevsky et al., 2012).
2. Rotar las imágenes: Se aplicaron rotaciones a las imágenes originales en intervalos de 2 grados, tanto en sentido horario como antihorario. Específicamente, se generaron

versiones de las imágenes rotadas en ángulos de ± 2 , ± 4 , ± 6 , ± 8 , ± 10 , ± 12 grados, así como espejeando las imágenes. La rotación permite que el modelo sea más robusto frente a pequeñas variaciones en la orientación de los objetos en las imágenes (Simard et al., 2003).

3.

El proceso de aumentación se llevó a cabo de forma iterativa sobre cada imagen del conjunto de datos, según la disponibilidad de memoria. Cabe mencionar que el aumento en la cantidad de imágenes también puede aumentar el tiempo de entrenamiento del modelo, pero se espera que el beneficio en términos de rendimiento y generalización supere este costo adicional (Shorten & Khoshgoftaar, 2019).

3.2.9 Entrenamiento del modelo

Entrenar un modelo de IA es el proceso de ajustar los parámetros de un algoritmo para que pueda aprender patrones y relaciones en un conjunto de datos. Esto se logra mediante el uso de un conjunto de datos de entrenamiento, donde el modelo se ajusta a los datos para minimizar la diferencia entre las predicciones y las respuestas correctas. Durante el entrenamiento, el modelo va ajustando sus parámetros en función del error de predicción para mejorar su capacidad de generalización y poder hacer predicciones más precisas sobre datos nuevos y no vistos anteriormente. El objetivo final del entrenamiento es obtener un modelo que pueda realizar predicciones precisas y confiables sobre nuevos datos que no se utilizaron durante el entrenamiento.

3.2.10 Evaluar el modelo

Una vez que nuestro modelo terminó su primera etapa de entrenamiento, haremos uso del segundo conjunto de datos (datos de prueba) con el que evaluaremos el desempeño de dicho modelo en la clasificación de los elementos en una escena.

Para poder realizar la evaluación correcta en un modelo dedicado a la segmentación semántica contamos con 3 coeficientes principalmente detallados en (Ekin Tiu, 2019):

- **Precisión del píxel:** Como lo dice su nombre, consiste en comparar el porcentaje de píxeles que coinciden entre las imágenes de prueba y las imágenes con predicción. La métrica "precision of pixel" es una medida de evaluación comúnmente utilizada para medir la calidad de la segmentación de imágenes. Esta métrica calcula la proporción de píxeles clasificados correctamente con relación a la cantidad total de píxeles en la imagen. En otras palabras, se trata de una métrica que evalúa la precisión de la segmentación de objetos en una imagen, midiendo cuántos píxeles clasificados como pertenecientes a un objeto realmente lo son.
- **Intersección sobre la unión:** También conocido como el coeficiente de Jaccard, IoU mide la superposición entre la máscara predicha y la máscara verdadera para cada clase. Es el área de intersección dividida por el área de la unión de ambas máscaras (Figura 14). Un valor de IoU más alto indica una mejor coincidencia entre las predicciones y las anotaciones del suelo verdadero (Vineeth S Subramanyam, 2021).

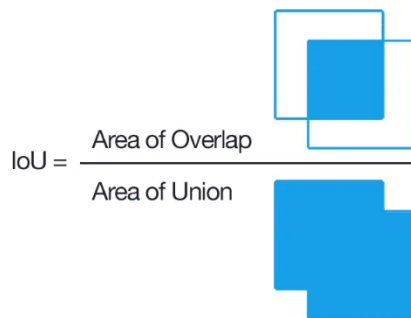


Figura 14 Ilustración de la métrica Intersección

sobre la Unión. Imagen recuperada de Rosebrock, 2022

- **Coeficiente de datos:** Similar al IoU, el coeficiente de Dice compara la superposición entre las máscaras predichas y verdaderas. Se calcula como el doble de la intersección

dividida por la suma de los tamaños de las dos máscaras. Un coeficiente de Dice más alto indica un mejor rendimiento del modelo. (Figura 15).

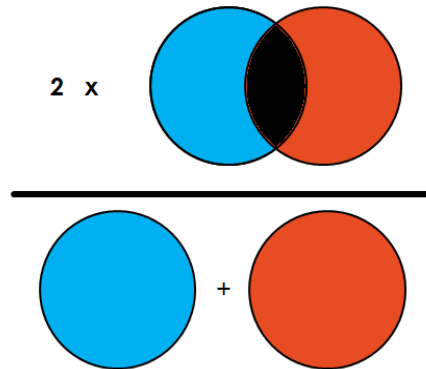


Figura 15 Ilustración de la Coeficiente de datos. Imagen recuperada de .Ekin Tiu, 2019

Para entrenar el modelo U-Net, primero instanciamos la arquitectura utilizando la función `unet_model` con un tamaño de entrada de (256, 256, 3). Luego, compilamos el modelo utilizando el optimizador Adam (Kingma & Ba, 2014) y la función de pérdida de entropía cruzada binaria. Además, utilizamos la precisión como métrica adicional para evaluar el rendimiento del modelo durante el entrenamiento.

Los optimizadores son algoritmos que se utilizan para ajustar los parámetros de un modelo de aprendizaje automático con el fin de minimizar la función de pérdida. Los optimizadores pueden variar en cuanto a la velocidad, la estabilidad y la complejidad de su implementación. El optimizador Adam es uno de los más populares y eficientes, ya que combina las ventajas de dos otros optimizadores: el momento adaptativo (Momentum) y la tasa de aprendizaje adaptativa por parámetro (RMSProp). El momento adaptativo ayuda al optimizador a escapar de los mínimos locales y a acelerar la convergencia, mientras que la tasa de aprendizaje adaptativa ajusta la magnitud de los pasos de actualización de cada parámetro según la escala de sus gradientes. El optimizador Adam también tiene un mecanismo de corrección de sesgo que evita que los primeros pasos sean demasiado pequeños.

La función de pérdida de entropía cruzada binaria se utiliza para medir la discrepancia entre dos distribuciones de probabilidad binarias: la verdadera y la predicha. En el contexto de la

segmentación de imágenes, la distribución verdadera es una máscara binaria que indica si cada píxel pertenece o no al objeto de interés, y la distribución predicha es una máscara de probabilidades que indica la confianza del modelo en asignar cada píxel al objeto de interés. La función de pérdida de entropía cruzada binaria penaliza las predicciones que se alejan de la verdad, y se reduce a cero cuando las predicciones coinciden con la verdad. Esta función de pérdida es adecuada para problemas de clasificación binaria, como la segmentación semántica de un solo tipo de objeto.

El modelo se entrenó utilizando el conjunto de entrenamiento con un tamaño de lote de 16 y durante 50 épocas. También se empleó el conjunto de validación para monitorear el rendimiento del modelo en datos no vistos y prevenir el sobreajuste (Srivastava et al., 2014). Durante el entrenamiento, se calculó el Índice de Jaccard o Intersección sobre la Unión (IoU) en el conjunto de validación para evaluar el rendimiento del modelo en términos de solapamiento entre las máscaras verdaderas y las predicciones.

3.3 ANÁLISIS DE RESULTADOS

Una vez completado el entrenamiento, evaluamos el rendimiento del modelo en el conjunto de validación utilizando diversas métricas para obtener una visión más completa de su efectividad y precisión en la segmentación de imágenes de infraestructura urbana.

Las métricas de evaluación utilizadas en este análisis incluyen la pérdida de entropía cruzada y el índice de Jaccard. La pérdida de entropía cruzada es una medida comúnmente utilizada en tareas de clasificación para cuantificar la discrepancia entre las etiquetas verdaderas y las predicciones del modelo. En el contexto de la segmentación semántica, esta métrica identifica qué bien el modelo puede asignar píxeles a sus clases. Valores más bajos de la pérdida de entropía cruzada indican un mejor rendimiento del modelo en la tarea de clasificación de píxeles.

IoU (Intersección sobre la Unión, también conocida como Jaccard): Esta métrica se utiliza ampliamente en la evaluación de modelos de segmentación semántica, ya que mide el solapamiento entre las máscaras verdaderas y las predicciones del modelo (Ronneberger et al.,

2015). Un valor de IoU más alto nos indica un mejor rendimiento del modelo en términos de coincidencia de áreas de interés.

Ambas métricas, pérdida de entropía cruzada y el índice de Jaccard, son útiles para evaluar el rendimiento del modelo en la tarea de segmentación semántica, ya que proporcionan información complementaria sobre la calidad de las predicciones del modelo. Mientras que la pérdida de entropía cruzada se centra en la exactitud de la clasificación de píxeles, el índice de Jaccard evalúa la calidad de la segmentación en términos de solapamiento entre las áreas segmentadas.

Además de calcular estas métricas, visualizamos las predicciones del modelo en comparación con las máscaras verdaderas para identificar áreas donde el modelo está funcionando bien o mal. Esta visualización puede realizarse utilizando bibliotecas de Python como Matplotlib, como se mencionó anteriormente.

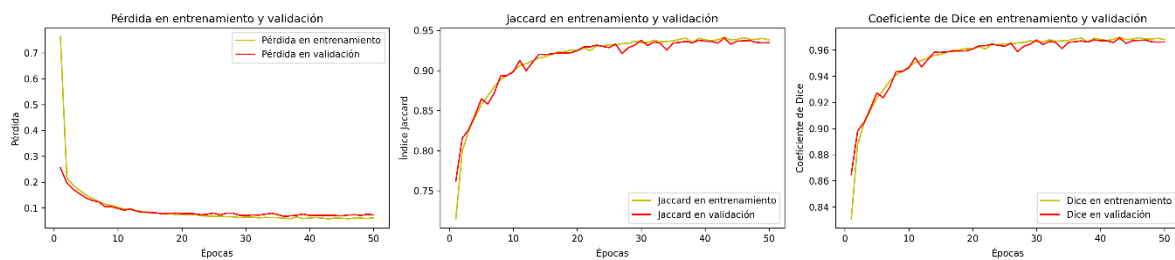


Figura 16 Resultados de pérdida y métrica Jaccard en entrenamiento y validación durante el proceso de entrenamiento.

Los resultados obtenidos del entrenamiento del modelo de inteligencia artificial demuestran una mejora progresiva en las métricas de rendimiento a lo largo del tiempo. Tanto el coeficiente de Dice como el índice de Jaccard, que son métricas comunes para evaluar la calidad de la segmentación semántica (Milletari et al., 2016; Taha & Hanbury, 2015), muestran un aumento notable en las primeras 15 épocas en donde alcanzó un índice de 0.90, con un crecimiento más moderado en las épocas siguientes con un incremento de 0.04 aproximadamente entre la época 15-50.

Inicialmente, el coeficiente de Dice y el índice de Jaccard comienzan en valores cercanos a 0.85 y 0.73, respectivamente. A medida que avanza el entrenamiento, ambos indicadores mejoran, alcanzando valores entre 0.95 y 0.97 al final del proceso. Este comportamiento es consistente tanto en las líneas de entrenamiento como en las de validación, lo que sugiere que el modelo generaliza bien a los datos no vistos.

En cuanto a la función de pérdida, tanto en entrenamiento como en validación, los valores iniciales son de aproximadamente 0.8 y 0.25, respectivamente. Durante las primeras cinco épocas, la pérdida disminuye rápidamente hasta alrededor de 0.25 en la línea de entrenamiento, mientras que la línea de validación ya comenzó en ese nivel. Posteriormente, ambas líneas experimentan una reducción más gradual a lo largo de las 50 épocas, alcanzando valores cercanos a 0.1 al final del proceso.

Estos resultados (Figura 16) indican que el modelo de IA se ha ajustado adecuadamente a los datos y ha aprendido a segmentar las imágenes de manera efectiva. La mejora continua en las métricas de rendimiento y la disminución de la función de pérdida respaldan la efectividad del modelo y su capacidad para aplicarse en casos prácticos (Milletari et al., 2016; Taha & Hanbury, 2015).

3.3.1 Intersección sobre la unión

Los resultados obtenidos de la evaluación del modelo de segmentación semántica muestran un Índice de Jaccard promedio (Mean IoU) de 0.8386048. El Índice de Jaccard, también conocido como Intersección sobre la Unión (IoU), es una métrica ampliamente utilizada para evaluar la calidad de la segmentación en comparación con las áreas de las máscaras verdaderas (Long et al., 2014). Un valor más alto de IoU indica un mayor solapamiento entre las áreas segmentadas y las máscaras verdaderas, lo que implica una segmentación más precisa.

Al desglosar el IoU para cada clase (Figura 17), se observa que la clase 'Fondo' obtiene el mejor rendimiento con un IoU de 0.9110635, lo que indica que el modelo es altamente preciso en la segmentación de esta clase. La clase 'Cruce peatonal' muestra un rendimiento moderado con un IoU de 0.70221066, mientras que la clase 'Ciclovía' presenta un rendimiento superior con un

IoU de 0.8317835. Por último, la clase 'Acera' presenta un rendimiento cercano al de 'Fondo', con un IoU de 0.90936154.

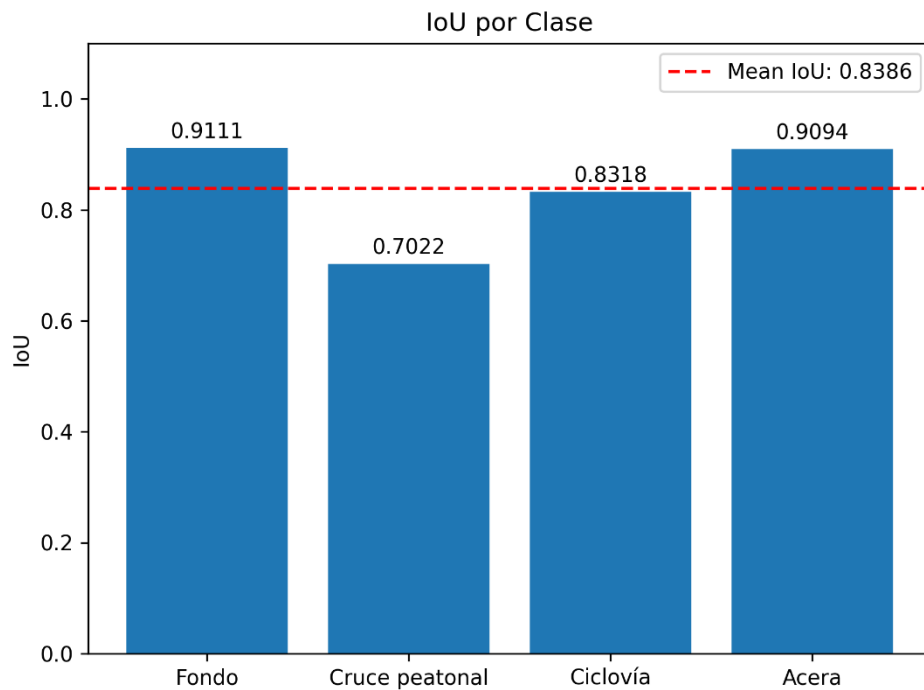


Figura 17 Comparación del Índice de Jaccard (IoU) por Clase en la Segmentación Semántica

Estos resultados indican que el modelo de segmentación semántica puede realizar segmentaciones precisas para algunas clases, aunque su rendimiento varía entre las diferentes. Este análisis puede ser útil para identificar áreas de mejora en la arquitectura del modelo o en el proceso de entrenamiento, con el objetivo de incrementar la precisión de la segmentación para todas las clases (Garcia-Garcia et al., 2017).

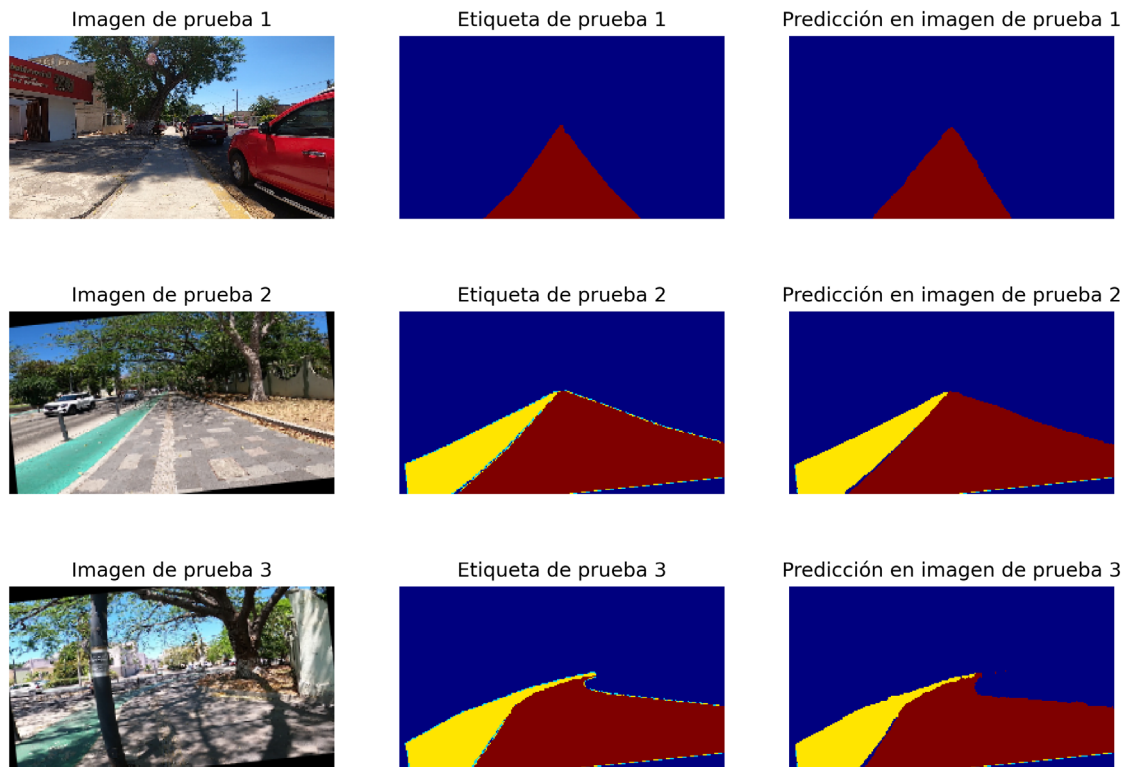


Figura 18 Resultados de pérdida y métrica Jaccard en entrenamiento y validación durante el proceso de entrenamiento.

La Figura 18 se compone de tres filas, cada una representando un ejemplo de prueba distinto. En la primera columna, se muestran las imágenes de prueba originales, tituladas como "Imagen de prueba 1", "Imagen de prueba 2" y "Imagen de prueba 3". Estas imágenes sirven como entrada para el modelo.

Tabla 3 Comparación de Índice de Jaccard

Clase	IoU
Fondo	0.9111
Cruce peatonal	0.7022
Ciclovía	0.8318
Acera	0.9094
Mean IoU	0.8386

En la segunda columna, se presentan las etiquetas de prueba reales asociadas a cada imagen de prueba. Estas etiquetas, tituladas como "Etiqueta de prueba 1", "Etiqueta de prueba 2" y "Etiqueta de prueba 3", representan la segmentación semántica verdadera de las imágenes y se utilizan como referencia para comparar con las predicciones del modelo.

La tercera columna muestra las predicciones generadas por el modelo de segmentación semántica para cada imagen de prueba. Estas imágenes, tituladas como "Predicción en imagen de prueba 1", "Predicción en imagen de prueba 2" y "Predicción en imagen de prueba 3", ilustran cómo el modelo segmenta las imágenes de prueba en función de las clases aprendidas durante el entrenamiento.

Al comparar visualmente las etiquetas de prueba reales con las predicciones del modelo, se observa que el comportamiento de segmentación es correcto en las imágenes seleccionadas, lo que sugiere que está listo para implementarse en aplicaciones prácticas, como se muestra en la Tabla 3. Es importante considerar que este análisis visual solo incluye tres ejemplos y puede ser necesario evaluar más con métricas cuantitativas y un conjunto más amplio de imágenes de prueba antes de llegar a conclusiones definitivas sobre el rendimiento del modelo.

4 CONCLUSIONES

En el caso del estudio presentado en el documento, se puede concluir que el uso de técnicas de reconocimiento de patrones e inteligencia artificial para el análisis de la infraestructura peatonal urbana es una herramienta valiosa para mejorar la planificación y el diseño urbano, así como la toma de decisiones y gestión de emergencias.

El modelo de segmentación semántica desarrollado en este estudio demostró ser efectivo en la identificación de elementos urbanos como aceras, cruces peatonales y ciclovías, lo que permitió obtener un inventario detallado de la infraestructura urbana en el área de estudio. Además, el empleo de técnicas de aumento de datos y la selección cuidadosa de la arquitectura del modelo permitieron mejorar el rendimiento y la precisión del modelo en la segmentación de imágenes.

En resumen, este estudio demuestra el potencial de las técnicas de reconocimiento de patrones e inteligencia artificial para establecer una base cuantitativa sobre el estado de las calles, siendo una oportunidad para la planificación y gestión del mejoramiento de la infraestructura urbana. Se recomienda continuar investigando y desarrollando modelos más avanzados y precisos para aplicaciones en ciudades emergentes y en constante crecimiento.

4.1 TRABAJOS FUTUROS

Este proyecto tuvo como objetivo desarrollar un modelo de segmentación semántica que pudiera identificar y clasificar diferentes elementos de la infraestructura urbana a partir de imágenes capturadas con dispositivos de uso cotidiano. El modelo consistió en una red neuronal convolucional, que se entrenó y validó con una base de datos capturados en la zona de interés. Los resultados obtenidos mostraron que el modelo fue capaz de segmentar las imágenes con una alta precisión y un buen nivel de acuerdo con las etiquetas reales.

Algunas opciones de trabajo a futuro para darle seguimiento al proyecto son:

- Ampliar la base de datos con imágenes de más ciudades y regiones, especialmente de países en desarrollo, donde la infraestructura urbana es más heterogénea y desafiante.

- Incorporar más clases de objetos urbanos, que puedan ser relevantes para el análisis de la accesibilidad y la movilidad urbana.
- Explorar otras arquitecturas de redes neuronales convolucionales, que puedan ofrecer mejores resultados o mayor eficiencia computacional.
- Implementar el modelo en una plataforma web o móvil, que permita a los usuarios cargar sus propias imágenes y obtener la segmentación semántica de forma interactiva y rápida.

REFERENCIAS

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Research, G. (2011). TensorFlow: Large-scale machine learning. *GPU Computing Gems Emerald Edition*, November, 277–291. www.tensorflow.org.
- Ananth, S. (2021). *Fast R-CNN for object detection - Towards Data Science*. <https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022>
- BID. (2014). Urbanización rápida y desarrollo. *Cumbre de América Latina y China de políticas y conocimiento: impacto de la urbanización rápida y la prosperidad nacional*, 71.
- BID. (2016). *Guía Metodológica Programa de Ciudades Emergentes y Sostenibles* (Tercera ed).
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Burns, E., Laskowski, N., & Tucci, L. (2022). *What is artificial intelligence (AI)?* <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>
- Cai, Y., Li, Q., Fan, Y., Zhang, L., Huang, H., & Ding, X. (2021). An automatic trough line identification method based on improved UNet. *Atmospheric Research*, 264, 105839. <https://doi.org/10.1016/j.atmosres.2021.105839>
- Chen, L. C., Collins, M. D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., & Shlens, J. (2018). Searching for efficient multi-scale architectures for dense image prediction. *Advances in Neural Information Processing Systems, 2018-Decem*(NeurIPS), 8699–8710.
- Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Peter Campbell, J. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science and Technology*, 9(2). <https://doi.org/10.1167/tvst.9.2.14>
- Chollet, F. (2015). *Keras*. Github. <https://github.com/keras-team/keras>

- Davis, R., Shrobe, H., & Szolovits, P. (1993). What Is a Knowledge Representation? *AI Magazine*, 14(1), 17. <https://doi.org/10.1609/aimag.v14i1.1029>
- Deng, X., Huang, J., Rozelle, S., Zhang, J., & Li, Z. (2015). Impact of urbanization on cultivated land changes in China. *Land Use Policy*, 45, 1–7. <https://doi.org/10.1016/j.landusepol.2015.01.007>
- dos Santos Oliveira, W. C., Braz Junior, G., Lima Gomes Junior, D., Cardoso de Paiva, A., & Sousa de Almeida, J. D. (2022). *A Two-Stage U-Net to Estimate the Cultivated Area of Plantations* (pp. 346–357). https://doi.org/10.1007/978-3-031-06427-2_29
- DotCSV. (2018, marzo 19). *¿Qué es una Red Neuronal? Parte 1 : La Neurona*. Youtube. <https://www.youtube.com/watch?v=MRIV2IwFTPg&t>
- Education, I. C. (2021). *Artificial Intelligence (AI)*. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- Ekin Tiu. (2019, agosto 9). *Metrics to Evaluate your Semantic Segmentation Model*. Towerdsdatascience.
- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? En *Machine Learning in Radiation Oncology* (pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). *A Review on Deep Learning Techniques Applied to Semantic Segmentation*. <http://arxiv.org/abs/1704.06857>
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc.
- Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. En G. Gordon, D. Dunson, & M. Dudík (Eds.), *Proceedings of the Fourteenth International*

Conference on Artificial Intelligence and Statistics (Vol. 15, pp. 315–323). PMLR.
<https://proceedings.mlr.press/v15/glorot11a.html>

Gobierno del estado de Colima. (2017). *Programa Sectorial de Desarrollo Urbano y Ordenamiento Territorial 2016-2021*. 93.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144. <https://doi.org/10.1145/3422622>

GoPro. (2019). *Cámara de acción sumergible con estabilización HERO8 Black | GoPro*.
<https://gopro.com/es/es/shop/cameras/hero8-black/CHDHX-801-master.html>

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines*, 19(1–2), 305–307.
<https://doi.org/10.1007/s10710-017-9314-z>

Huang, Y., & Xiang, Z. (2022). RPDNet: Automatic Fabric Defect Detection Based on a Convolutional Neural Network and Repeated Pattern Analysis. *Sensors*, 22(16).
<https://doi.org/10.3390/s22166226>

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.
<http://arxiv.org/abs/1412.6980>

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. En F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25). Curran Associates, Inc.
<https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>

Leal Vallejo, A., Viramontes Fabela, Y., Benítez, Aguirre, B., SEDATU, & ITDP. (2019). *Calles Completas*.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- Li, T., Jiang, C., Bian, Z., Wang, M., & Niu, X. (2020). Semantic Segmentation of Urban Street Scene Based on Convolutional Neural Network. *Journal of Physics: Conference Series*, 1682(1), 012077. <https://doi.org/10.1088/1742-6596/1682/1/012077>
- Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, 3, 111–132.
<https://doi.org/10.1016/j.aiopen.2022.10.001>
- Liu, Z., Jiang, M., & Lin, H. (2020). *A graph-based spatial temporal logic for knowledge representation and automated reasoning in cognitive robots*.
<http://arxiv.org/abs/2001.07205>
- Long, J., Shelhamer, E., & Darrell, T. (2014). *Fully Convolutional Networks for Semantic Segmentation*. <http://arxiv.org/abs/1411.4038>
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 28.
- Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*. <http://arxiv.org/abs/1606.04797>
- Montesinos-López, O. A., Montesinos-López, A., Pérez-Rodríguez, P., Barrón-López, J. A., Martini, J. W. R., Fajardo-Flores, S. B., Gaytan-Lugo, L. S., Santana-Mancilla, P. C., & Crossa, J. (2021). A review of deep learning applications for genomic selection. En *BMC Genomics* (Vol. 22, Número 1). BioMed Central Ltd. <https://doi.org/10.1186/s12864-020-07319-x>
- Perez, L., & Wang, J. (2017). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. <http://arxiv.org/abs/1712.04621>

- Reddy, R. (2018). *Keras for beginners: Implementing a convolutional neural network*. Medium.com. <https://towardsdatascience.com/keras-for-beginners-implementing-a-cnn-cfeab764dbf9>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. <http://arxiv.org/abs/1804.02767>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28
- Rosebrock, A. (2022). *Intersection over Union (IoU) for object detection*.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence A Modern Approach* (4th Edition). En *Pearson Series*. <https://books.google.com.br/books?id=koFptAEACAAJ>
- Salinas, L. C., Rodríguez Laguna, R., René, J., Lazalde, V., Arturo, O., Sandoval, A., Icela, R., & Hernández, B. (2017). Detección del crecimiento urbano en el estado de Hidalgo mediante imágenes Landsat Monitoring of urban growth in the state of Hidalgo using Landsat images. *Investigaciones Geográficas: Boletín del Instituto de Geografía*, 2017, 64–73.
www.investigacionesgeograficas.unam.mx recibido: 15/09/2015. Aprobado: 11/03/2016. Publicado en línea
- SEDATU. (2019). Manual de calles - Diseño vial para ciudades mexicanas. En *Angewandte Chemie International Edition*, 6(11), 951–952.

- Sharma, A. (2022). Introduction to the {YOLO} Family. En D. Chakraborty, P. Chugh, A. R. Gosthipaty, J. Haase, S. Huot, K. Kidriavsteva, R. Raha, & A. Thanki (Eds.), *PyImageSearch*.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 958–963. <https://doi.org/10.1109/ICDAR.2003.1227801>
- Soto-Cortés, J. J. (2015). El crecimiento urbano de las ciudades: enfoques desarrollista, autoritario, neoliberal y sustentable. *Paradigma económico*, 7(1), 1–23. [file:///C:/Users/Usuario/Downloads/Dialnet-ElCrecimientoUrbanoDeLasCiudades-5926288 \(5\).pdf](file:///C:/Users/Usuario/Downloads/Dialnet-ElCrecimientoUrbanoDeLasCiudades-5926288%20(5).pdf)
- Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. En *Journal of Machine Learning Research* (Vol. 15).
- Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). *Segmenter: Transformer for Semantic Segmentation*. <http://arxiv.org/abs/2105.05633>
- Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15(1). <https://doi.org/10.1186/s12880-015-0068-x>
- Tan, H., Chen, C., Luo, X., Zhang, J., Seibold, C., Yang, K., & Stiefelhagen, R. (2021). *Flying Guide Dog: Walkable Path Discovery for the Visually Impaired Utilizing Drones and Transformer-based Semantic Segmentation*. <http://arxiv.org/abs/2108.07007>
- Terraza, H., Blanco, D. R., Vera, F., & BID. (2014). De ciudades emergentes a ciudades sostenibles. *Educatio Siglo XXI*, 32(1), 287–290.

- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & the scikit-image contributors. (2014). scikit-image: image processing in Python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>
- Vineeth S Subramanyam. (2021, enero 17). *IOU (Intersection over Union)*. Medium.com. <https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef>
- Williamson, J. G. (1988). Chapter 11 Migration and urbanization. *Handbook of Development Economics*, 1, 425–465. [https://doi.org/10.1016/S1573-4471\(88\)01014-9](https://doi.org/10.1016/S1573-4471(88)01014-9)
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*. <http://arxiv.org/abs/2105.15203>
- Zhu, F., Zhu, Y., Zhang, L., Wu, C., Fu, Y., & Li, M. (2021). *A Unified Efficient Pyramid Transformer for Semantic Segmentation*. <http://arxiv.org/abs/2107.14209>