Visión por computadora para mejorar la movilidad y desplazamiento de personas ciegas en ciudades

Integrantes

Juan Nicolás Carvajal Useche
Gabriela María Castro Beltrán

Profesor:

Flavio Augusto Prieto Ortiz

Universidad Nacional de Colombia
Facultad de Ingeniería
Visión de maquina
Bogotá, Colombia
Septiembre, 2023

Índice general

Introducción

La movilidad segura y autónoma en entornos urbanos es esencial para la calidad de vida de todas las personas, pero representa un desafío significativo para aquellos con discapacidades visuales. Las personas ciegas a menudo enfrentan obstáculos, falta de información accesible y dependencia de otros para la orientación al navegar por las calles de la ciudad. La visión por computadora, al procesar datos visuales y proporcionar información en tiempo real, se presenta como una herramienta poderosa para superar estas barreras y permitir que las personas ciegas se desplacen con mayor seguridad y autonomía.

1. Contextualización del problema

La movilidad segura y autónoma es fundamental para la calidad de vida de todas las personas, pero representa un desafío significativo para aquellos con discapacidad visual, quienes enfrentan desventajas en entornos urbanos. La discapacidad visual, que abarca la ceguera y la baja visión, afecta la percepción y la orientación, generando obstáculos, especialmente en áreas urbanas densamente pobladas como Bogotá.

En Colombia, según cifras del Instituto Nacional para Ciegos (INCI), se estima que 1.9 millones de personas presentan algún grado de discapacidad visual, siendo la ciudad de Bogotá uno de los epicentros de esta población. Además, diversos estudios han revelado que la prevalencia de esta condición varía significativamente, con cifras que oscilan entre un 0,03 % y un 1,09 % para la ceguera y del 7,76 % al 17,4 % para la discapacidad visual moderada. [1] Aún más, los obstáculos que estas personas enfrentan al desplazarse se agravan debido a la falta de infraestructuras y servicios adecuados para satisfacer sus necesidades. Los desafíos pueden manifestarse en diversas formas, desde el tráfico vehicular hasta desniveles o huecos en aceras sin señalización, así como salientes que obstruyen la zona de paso a una altura que no puede ser detectada con los bastones generalmente utilizados. Estas condiciones resaltan la urgente necesidad de mejorar la infraestructura urbana y la accesibilidad en Bogotá para garantizar una movilidad segura y equitativa para todas las personas, independientemente de su discapacidad. Cabe resaltar que la inseguridad también plantea preocupaciones adicionales para las personas con discapacidad visual, se han reportado incidentes en los que los bastones utilizados por estas personas son objeto de robo, lo que no solo les priva de una herramienta vital para su movilidad, sino que también genera una sensación de vulnerabilidad y desamparo. Además, los semáforos sonoros y los pavimentos podotáctiles, que son fundamentales para proporcionar información y orientación a las personas con discapacidad visual, a menudo son vandalizados o dañados, lo que compromete aún más su seguridad y autonomía en la ciudad.

2. Antecedentes

En los últimos años, se han realizado avances significativos en el desarrollo de tecnologías asistenciales diseñadas para mejorar la movilidad y la calidad de vida de las personas con discapacidad visual. Estos esfuerzos se centran en la creación de sistemas tecnológicos que se

adapten a las tareas de movilidad con el propósito de facilitarlas y promover la autonomía de quienes enfrentan esta condición.

Uno de ellos es el prototipo sónico de ayuda a la ceguera, creado por investigadores de la Universidad Carlos III de Madrid. Este sistema utiliza un procesador de estereovisión para medir la diferencia de imágenes captadas por dos cámaras y calcular la distancia a los objetos en el entorno. Luego, transmite esta información al usuario a través de una serie de sonidos codificados que indican la posición y distancia de los obstáculos. Este prototipo busca complementar el uso del bastón o el perro guía y ha sido diseñado para reducir su tamaño y ser más accesible en términos de costos. [2]

Figura 1.1: Prototipo sónico de ayuda a la ceguera. [2]

Además de los dispositivos, también se han desarrollado aplicaciones móviles como Seeing AI, creada por Microsoft para dispositivos iOS. Esta aplicación utiliza la cámara del dispositivo para identificar personas y objetos, proporcionando descripciones auditivas a personas con discapacidad visual. Sus capacidades abarcan la descripción de texto, documentos, productos, individuos, monedas, escenarios, colores, escritura a mano y niveles de iluminación. Además, la aplicación puede escanear códigos de barras para brindar información sobre productos y hasta estimar la edad, género y estado emocional de las personas. Aunque algunas de sus funciones se ejecutan en el dispositivo, tareas más complejas, como la descripción de escenas y el reconocimiento de escritura a mano, requieren una conexión a Internet. [3]

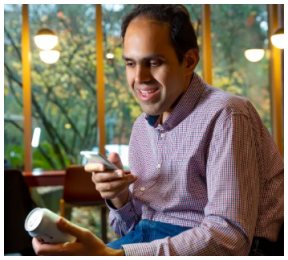

Figura 1.2: Aplicativo móvil de inteligencia artificial. [3]

Por otro lado, el Instituto Nacional para Ciegos (INCI) de Colombia está anunciando la disponibilidad de las innovadoras gafas inteligentes OrCam MyEyes en el país. Estas gafas, desarrolladas por la prestigiosa marca OrCam, están equipadas con una cámara de alta resolución de 13 megapíxeles y láseres integrados que permiten dirigir la lectura de textos o señalar áreas específicas de un documento mediante una voz digitalizada. Además, estas gafas tienen la capacidad de reconocer colores y billetes. Estas gafas han tenido éxito en Colombia y en otros 41 países, brindando mayor autonomía a personas ciegas o con baja visión y per-

mitiéndoles realizar actividades sin depender de otros. [4]

Figura 1.3: Gafas inteligentes para la lectura. [4]

Más tarde, Panasonic y Biel Glasses han colaborado en el desarrollo de gafas inteligentes para personas con discapacidad visual, presentando esta solución en el CES 2023 en Las Vegas. Las gafas combinan la tecnología de realidad virtual de Panasonic con la tecnología para baja visión de Biel Glasses. Estas gafas utilizan inteligencia artificial y robótica para detectar obstáculos y peligros en tiempo real, permitiendo a los usuarios desplazarse de manera segura e independiente. Los optometristas ajustan las funciones según las necesidades de cada persona, mejorando la visión y reduciendo la carga durante el uso. Panasonic y Biel Glasses se comprometen a seguir desarrollando soluciones tecnológicas para apoyar a las personas con baja visión en el futuro. [5]

Figura 1.4: Gafas inteligentes para personas con discapacidad visual. [5]

3. Solución propuesta

Partiendo de los conceptos vistos y desarrollados en el curso, en donde se destacan técnicas para el preprocesamiento de imágenes, detección y clasificación de objetos y algoritmos de aprendizaje profundo, se propone el desarrollo de un programa escrito en Python que capture imágenes de la vida cotidiana (imágenes en la calle mayoritariamente), que haga el debido preproceso y que logre detectar diferentes objetos clave para una persona invidente, como lo son pasos de cebra, semáforos, señales de tránsito, entre otras. Así mismo, el algoritmo debe ser capaz de avisarle al usuario si se está acercando a uno de estos objetos clave y así mismo comunicar de qué objeto se trata. Esta herramienta proporcionará un apoyo a personas con discapacidad visual ya que lo mantendrá conectado con el entorno y al tanto de estos objetos clave que se encuentran en la cotidianidad.

Este informe tiene como objetivo mostrar el desarrollo de este programa, su implementación y resultados, partiendo inicialmente de la revisión del estado del arte en donde se

discutirán varias alternativas para la creación de los algoritmos, se observarán sus posibles limitaciones y su compatibilidad con lo requerido en este proyecto, después de esto se planteará la solución y los algoritmos a usar según lo revisado y discutido, comenzando con diagramas de flujo que describan el proceso que se desea y así lograr implementar algoritmos y un programa estructurado, después, se evaluarán los resultados obtenidos por el programa mediante pruebas de este y verificar los resultados deseados, finalmente se darán conclusiones y se expondrán las limitaciones y posibles campos de mejora.

4. Alcance

Este proyecto se enfocará en lugares urbanos comunes y al aire libre, como lo son andenes, carreteras, puentes, entre otros. Además, se enfocará únicamente en los objetos clave, como son señales de tránsito, semáforos, pasos de cebra, vehículos y algunos símbolos como los de los baños, o símbolos de peligro y precaución. Finalmente, el proyecto se enmarcará en los conceptos vistos en el curso, por lo que se aplicará procesamiento de proyecciones bidimensionales del espacio. Así mismo, el proyecto tomará en cuenta diversos estados de tiempo y horas del día.

Estado del arte

Las aplicaciones de visión por computadora para el desarrollo de prototipos de asistencia a personas ciegas o con problemas de vista se han visto en numerosos documentos y papers de investigación. En general, las soluciones a este problema se dan en 2 áreas principales.

1. Aplicaciones en detección de obstáculos

Para la detección de obstáculos en el entorno, generalmente se emplean dos enfoques distintos: por un lado, los sensores de distancia, como el ultrasonido, LiDAR o la triangulación infrarroja (IR), y por otro lado, las técnicas basadas en cámaras, como las cámaras monoculares o las cámaras RGB-D. En ocasiones, se recurre a la combinación de ambas técnicas para lograr una mayor precisión en la detección. Uno de los desafíos al desarrollar soluciones de asistencia para personas ciegas radica en determinar la distancia precisa entre un objeto u obstáculo y el usuario, ya que la ubicación tridimensional de dicho objeto en el mundo real a menudo se infiere a partir de una imágen bidimensional capturada por una cámara monocular. Para abordar este problema, se han utilizado diversas soluciones, que incluyen sensores ultrasónicos, nubes de puntos y capturas de imágenes RGB-D a través de cámaras de visión estéreo, así como enfoques basados en cálculos matemáticos. En esta sección, se analizan y describen estos enfoques de detección de obstáculos.

1.1. Técnicas basadas en cámaras

En el contexto en el que se basa en imágenes de profundidad capturadas por una cámara RGB-D. Estas imágenes RGB-D son obtenidas mediante cámaras que colaboran con sensores de detección de distancia. Los Stixels dividen los elementos en la imágen que rodea al usuario en áreas verticales en función de su diferencia de profundidad en el entorno. Posteriormente, empleando técnicas de identificación de objetos, los Stixels clasifican de manera semántica los objetos presentes en la escena. [6] También existe la opción de calcular la distancia de un objeto respecto a la cámara utilizando un enfoque matemático, como se describe en el artículo [7].

$$Z = \frac{f * k_v * h}{v - v_o}$$

En esta ecuación, f representa la distancia focal de la cámara, k_v corresponde a la densidad de píxeles (píxeles por metro), h indica la altura de la cámara desde el suelo, v_o es la coordenada central de la imágen formada, v denota la distancia desde la cámara hasta las coordenadas del suelo del objeto que se está observando, y Z representa la distancia entre el objeto en cuestión y la ubicación de la cámara. En el artículo [7], se destacó que este método posee una alta precisión en la medición de distancias y puede determinar la distancia de un objeto incluso a más de 10 metros de distancia.

Además de las técnicas previamente mencionadas, existe la posibilidad de estimar la profundidad a partir de imágenes monoculares utilizando enfoques basados en aprendizaje profundo. En [8], se desarrolló una red de predicción de profundidad que genera un mapa de

profundidad a partir de una única imágen RGB. Estas predicciones son compatibles con imágenes capturadas por diversos modelos de cámaras. Se han implementado varios tipos de redes neuronales, como CNNs y RNNs, lo que ha demostrado la eficacia de la estimación de la profundidad en imágenes monoculares.

1.2. Técnicas basadas en sensores de distancia

El empleo de sensores ha sido una estrategia más extendida en la detección de obstáculos en comparación con las técnicas basadas en cámaras. Entre la variedad de sensores disponibles, los sensores ultrasónicos se destacan por su popularidad debido a su precisión, asequibilidad, eficiencia energética y facilidad de implementación.

Los sensores ultrasónicos funcionan mediante la emisión de ondas sonoras a una frecuencia demasiado alta para ser percibida por el oído humano. Posteriormente, el receptor del sensor espera la reflexión de estas ondas sonoras, y en función de esta información, se calcula la distancia hasta el obstáculo. Estos sensores presentan ventajas significativas en la detección de objetos transparentes en comparación con sensores basados en luz o tecnologías de radar. Por ejemplo, en el estudio mencionado en el artículo [9], se emplearon sensores ultrasónicos para detectar obstáculos ubicados a nivel de la rodilla y objetos de baja altura. Sin embargo, es importante mencionar que los sensores ultrasónicos tienen limitaciones, como su alcance limitado y su capacidad para detectar principalmente obstáculos cercanos, lo que los hace más apropiados para entornos interiores.

1.3. Técnicas combinadas de sensores de distancia y cámaras

En tiempos recientes, algunos investigadores han optado por combinar ambas metodologías, es decir, sensores de distancia y cámaras. En el artículo mencionado en [10], se hizo uso de sensores ultrasónicos en conjunto con cámaras RGB-D para llevar a cabo la detección de obstáculos. Su dispositivo de asistencia electrónica (ETA) procesa los datos provenientes de una cámara RGB-D utilizando una Raspberry Pi 3 B+ que está equipada con sensores ultrasónicos para la medición de distancias. La fusión de estos dos enfoques resulta en una detección de obstáculos notablemente más precisa.

2. Aplicaciones en detección y clasificación de objetos

Para implementar una solución de reconocimiento de objetos, existen diferentes enfoques. Uno de los enfoques comunes es ejecutar el proceso de reconocimiento de forma remota en servicios en la nube como Google Cloud Vision, Microsoft Azure Computer Vision, Amazon Rekognition, entre otros. Estos servicios ya están entrenados con enormes conjuntos de datos que permiten un mejor rendimiento. En [9] propusieron un sistema móvil que utiliza Google Cloud Vision para reconocer objetos, texto y rostros. También existen soluciones que proporcionan sus propios algoritmos de procesamiento de imágenes en la nube.

Otro enfoque es el procesamiento de imágenes local, utilizado en muchas soluciones, que realiza los cálculos relacionados con el reconocimiento de objetos en el lado del cliente. Sin embargo, este enfoque suele estar limitado a un número reducido de objetos debido a las limitaciones de hardware. El Grupo de Geometría Visual (VGG16), que es un modelo de red

neuronal convolucional, es la red base del algoritmo SSD, seguida de una capa de características multi-escala para la predicción de categorías de objetos y cajas delimitadoras. SSD genera cajas de anclaje en diversas tallas y predice objetos en función de su tamaño. El tiempo de inferencia del método SSD512 es de 22 milisegundos con aproximadamente un 76.8 % de Precisión Promedio Mínima (mAP) en el conjunto de datos Pascal VOC2007 de imágenes, lo que demuestra su competencia y velocidad en la detección de objetos.

YOLO es una técnica de detección de objetos basada en CNN y utiliza Darknet, un marco de red de código abierto escrito en C y CUDA. YOLO divide una imágen en cuadrículas de tamaño SxS y genera B cajas delimitadoras para cada una de ellas. Luego, predice la probabilidad de clases para los objetos y sus correspondientes cajas delimitadoras. En [13] la implementación de software se basa en el uso de las bibliotecas OpenCV de Python y se incorpora un proceso de aprendizaje automático. La unidad principal de procesamiento es una Raspberry Pi, que escanea y detecta los contornos faciales mediante la cámara Pi, mientras que los objetos en la imágen se capturan y reconocen utilizando la cámara de un dispositivo móvil. El proceso de detección de objetos funciona a una velocidad de procesamiento de 6-7 FPS con una precisión del 63-80 %, mientras que el proceso de identificación facial alcanza una precisión del 80-100

Se han utilizado diferentes versiones (YOLO v3 (Tiny), YOLO v2, YOLO 9000) en diferentes investigaciones. La precisión y el número de objetos que se pueden detectar varían para cada versión. Por ejemplo, TinyYolo está diseñado para dispositivos móviles y puede reconocer un número menor de objetos en comparación con las otras versiones. YOLO y SSD son muy populares porque logran un equilibrio entre precisión y velocidad.

En [12] se compararon los resultados experimentales de CNN, SVM, YOLO2 y YOLO3 en función de su precisión y la cantidad de cuadros generados por segundo. Se observó que YOLO3 obtuvo una precisión del 46.8 %, superando a los otros métodos, y en términos de la generación de cuadros por segundo, YOLO3 fue el más rápido, alcanzando hasta 18 cuadros por segundo.

En [14] el resultado de la prueba indica que el método desarrollado utilizando el modelo YOLO-v5 tiene una precisión del 95.71 % en la detección y del 100 % en el reconocimiento. Este sistema opera de manera independiente y en tiempo real.

Implementación

1. Base de datos

El conjunto de datos COCO (Common Objects in Context) es una extensa colección de imágenes ampliamente empleada en tareas de visión por computadora. La versión de 2017 de este conjunto de datos, que será utilizada en el desarrollo del proyecto, consta de 40,670 imágenes de prueba, 118,287 imágenes de entrenamiento y 5,000 imágenes de validación. Además de estas imágenes, el conjunto de datos proporciona anotaciones detalladas para diversas tareas, como detección de objetos, segmentación, detección de puntos clave, coordenadas de caja delimitadora y descripciones de texto para las imágenes. COCO es una elección común en la evaluación y capacitación de modelos de visión por computadora, siendo una referencia crucial en la investigación de este campo.

Además de esto, es ampliamente reconocido por su diversidad, ya que el conjunto de datos engloba una extensa variedad de objetos y situaciones cotidianas, abarcando hasta 80 categorías distintas, que incluyen elementos como vehículos, semáforos, cruces peatonales, personas y otros objetos de relevancia crítica para el proyecto. Esta diversidad es de gran importancia, ya que permite que el modelo de detección de objetos se entrene y adapte a una amplia gama de escenarios reales. Esto resulta esencial para ayudar a las personas ciegas a identificar y navegar de manera efectiva en diferentes entornos.

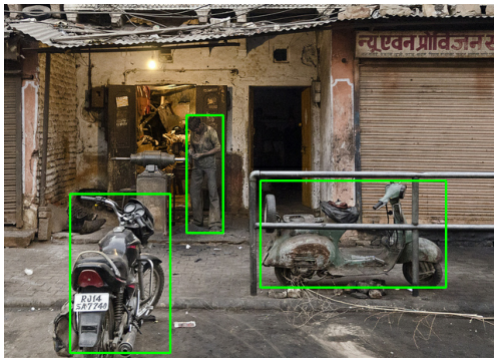

Figura 3.1: Ejemplo de una imágen de la base de datos COCO junto con la caja delimitadora.

Sin embargo, debido a la naturaleza del conjunto de datos COCO, que contiene un total de 80 categorías, muchas de las cuales no son pertinentes para la aplicación en cuestión, y a su complicado manejo en un entorno de ejecución como Colab, se tomó la decisión de utilizar un conjunto de datos alternativo disponible en Kaggle, denominado "CARLA DATASET ".

Esta elección se fundamenta en varias razones clave. En primer lugar, el dataset "CARLA DATASET "destaca por su contenido altamente pertinente, ya que se enfoca en la detección de objetos y situaciones específicas en entornos de conducción. Esto incluye la identificación precisa de elementos cruciales, como señales de tráfico, luces de tráfico y otros componentes relevantes para aplicaciones relacionadas con la seguridad vial y objetos cotidianos en las ca-

lles.

A diferencia del conjunto de datos COCO, reconocido por su amplitud y diversidad, que puede resultar en una carga considerable de datos difícil de gestionar, especialmente en entornos de desarrollo y pruebas, el dataset "CARLA DATASET "se presenta como una alternativa más manejable. Este dataset tiene un tamaño de tan solo 103 MB, lo que facilita considerablemente la experimentación y el desarrollo de algoritmos en un marco de tiempo más eficiente.

Una característica esencial que distingue a este dataset es la precisión de sus anotaciones. Cada píxel en las imágenes se encuentra etiquetado con una categoría semántica específica, lo que permite llevar a cabo una segmentación semántica precisa. Además, el dataset incluye Bounding Boxes (cajas delimitadoras) que se encuentran debidamente anotadas, lo que resulta fundamental para la detección y localización precisa de objetos en las imágenes. Estas anotaciones detalladas no solo contribuyen a la exactitud de los resultados, sino que también agilizan el proceso de implementación de algoritmos de visión por computadora, ahorrando tiempo y esfuerzo significativos.

El dataset se organiza en 10 clases, cada una representando diferentes elementos que una persona ciega o con discapacidad visual podría encontrar en su entorno mientras se desplaza. Estas categorías abarcan desde semáforos y señales de tráfico hasta vehículos y personas.

- **Detección de Semáforos por Color:**
  El dataset es especialmente útil para detectar semáforos en base a sus colores. Esto es crucial para que las personas ciegas puedan identificar cuándo deben cruzar una calle de manera segura, ya que podrán conocer cuándo la luz del semáforo está en rojo, amarillo o verde.

- **Detección de Señales de Tráfico por Velocidad:**
  Además, este dataset es valioso para identificar señales de tráfico que indican límites de velocidad. Esto permite a las persona conocer las restricciones de velocidad en áreas específicas.

- **Simplificación de Categorías de Vehículos:**
  En lugar de diferenciar entre automóviles, camiones y otros tipos de vehículos, el dataset los agrupa bajo una única categoría "vehicles " . Esta simplificación facilita la detección y el seguimiento de vehículos, lo que es relevante para evitar obstáculos en la movilidad de las personas con discapacidad visual.

- **Inclusión de Bicicletas, Motocicletas y Personas:** El dataset también se preocupa por la seguridad de las persona al incluir categorías específicas para la detección de bicicletas, motocicletas y personas. Identificar estos elementos es esencial para garantizar una movilidad segura y la interacción sin peligros en las vías públicas.

2. Preprocesamiento de las imágenes

El preprocesamiento de imágenes desempeña un papel fundamental en la visión por computadora, incluyendo tareas como la detección de objetos. Este proceso es esencial para normalizar los datos de entrada, garantizar tamaños de imágen consistentes y optimizar la eficiencia

computacional. Además, el preprocesamiento mejora la calidad de los datos al corregir problemas en las imágenes y eliminar datos no deseados. También asegura la compatibilidad con los requisitos específicos del modelo, como formato y rango de datos. Al aplicar técnicas de aumento de datos durante el preprocesamiento, se enriquece la diversidad de los datos de entrenamiento, lo que contribuye a la robustez y la generalización del modelo.

En primer lugar, se colocan las imágenes en escala de grises. Esto reduce la cantidad de datos al eliminar la información de color, lo que puede acelerar el procesamiento. Las imágenes en escala de grises se enfocan en la textura y forma de los objetos, lo que es beneficioso para la detección de objetos independientemente de su color. Además, su procesamiento es menos intensivo en recursos, lo que es útil en dispositivos con limitaciones de cómputo. Estas imágenes son más robustas frente a variaciones de iluminación y condiciones ambientales y simplifican tareas específicas, como la detección de bordes.

Luego, para garantizar un procesamiento óptimo de las imágenes de entrada, se requiere que todas posean un tamaño uniforme. Es recomendable que estas imágenes tengan dimensiones cuadradas y, preferiblemente, un tamaño específico. Por lo tanto, es necesario ajustar las imágenes a esta dimensión particular, comúnmente 416x416 píxeles o 608x608 píxeles. La elección de un tamaño cuadrado, en este caso, 416 x 416 píxeles, es de vital importancia para asegurar la eficiencia del modelo en el procesamiento de imágenes y preservar la precisión en las detecciones. Para lograr este redimensionamiento, se utiliza una técnica de interpolación del vecino más cercano, garantizando así la calidad de las imágenes transformadas.

El siguiente paso implica normalizar los valores de píxeles en las imágenes. Esto significa ajustar los valores de los píxeles para que estén en un rango específico, comúnmente entre 0 y 1. La normalización es crucial para que el modelo pueda aprender de manera efectiva durante el entrenamiento. Los valores normalizados permiten que las operaciones matemáticas se realicen de manera más eficiente y contribuyen a una convergencia más rápida del modelo. Del mismo modo se realiza la normalización de las cajas delimitadoras y se ajustan al formato de la red neuronal que en este caso será YOLO-V5 asegurandose que coincidan con las imágenes preprocesadas.

Para que YOLO funcione eficazmente, todas las imágenes del dataset deben estar organizadas en un directorio específico. Este directorio puede tener subdirectorios separados para conjuntos de datos diferentes, como entrenamiento, validación y prueba. Mantener una estructura de directorios ordenada facilita la gestión de los datos. Cada imágen del dataset debe tener un archivo de anotaciones correspondiente. Este archivo de anotaciones comparte el mismo nombre que la imágen, pero con una extensión de archivo diferente, comúnmente ".txt" .Estas anotaciones contienen información sobre los objetos presentes en la imágen, incluyendo la clase del objeto y las coordenadas del bounding box.

Dentro del archivo de anotaciones, cada línea representa un objeto presente en la imágen. Cada línea debe incluir la clase del objeto, que se representa con un número entero, seguido de las coordenadas del bounding box del objeto. Las coordenadas se presentan como:

[Número de categoría] [centro del objeto en X] [centro del objeto en Y] [ancho del objeto en X] [ancho del objeto en Y].

Esto es esencial para evaluar y visualizar las detecciones de objetos correctamente.

El archivo de clases es un componente fundamental del dataset. Define las clases a las que corresponden los números en los archivos de anotaciones. Cada número que representa una clase en los archivos de anotaciones se asocia con una etiqueta legible definida en el archivo de clases. Este archivo de clases es una referencia clave para la interpretación de las clases en los archivos de anotaciones. Para este dataset las cetegorias estan dadas de la siguiente manera:

```
# Classes
nc: 11  # number of classes
names: ['Superclase','bike','motobike','person','luzverde','luzamarilla','
    luzroja','30km','60km','90km','vehiculo']  # class names
```

Finalmente, se llevó a cabo la carga de las imágenes de entrenamiento y validación junto con sus respectivas etiquetas. Estas imágenes se organizaron en carpetas y subcarpetas especialmente creadas en el espacio de almacenamiento en Google Drive, con el propósito de construir un nuevo dataset estructurado y eficiente. Durante este proceso, se implementó una práctica de depuración al eliminar aquellas imágenes que no contaban con anotaciones, asegurando que solo las imágenes relevantes y anotadas fueran incluidas en el nuevo conjunto de datos.

Además, se realizó un paso adicional para mejorar la calidad visual de las imágenes. Se aplicó un filtro gaussiano, una técnica de procesamiento de imágenes, con el fin de reducir el ruido presente en las imágenes. El filtro gaussiano se utilizó para suavizar las imágenes y eliminar imperfecciones visuales no deseadas, mejorando así la legibilidad y la claridad de las anotaciones. Este proceso de filtrado contribuyó a obtener resultados más limpio y coherente, lo que es esencial para garantizar una precisión y una detección efectiva de objetos en las imágenes.

3. Algoritmo de detección

De conformidad con lo estudiado en el estado del arte, se escoge como algoritmo de detección a YoloV5.

3.1. YoloV5

YOLOv5 es una iteración más reciente de la serie YOLO, desarrollada por Ultralytics. Ha ganado popularidad por su rendimiento, precisión y velocidad mejorados en comparación con las versiones anteriores. YOLOv5 utiliza arquitecturas de redes neuronales convolucionales más profundas y técnicas de entrenamiento avanzadas para detectar objetos en imágenes en tiempo real. Este modelo ha sido utilizado en una variedad de aplicaciones, como la detección de objetos en cámaras de seguridad, automóviles autónomos, sistemas de asistencia para la conducción, drones y más. Además, YOLOv5 ha demostrado ser efectivo en la detección de múltiples clases de objetos en una sola imagen.

3.2. Arquitectura de YOLOv5

YOLO posee una arquitectura que se caracteriza por su eficiencia y velocidad, y ha experimentado varias iteraciones a lo largo del tiempo, siendo YOLOv4 y YOLOv5 algunas de las versiones más conocidas. En la figura 3.2 se observa visualmente la arquitectura.

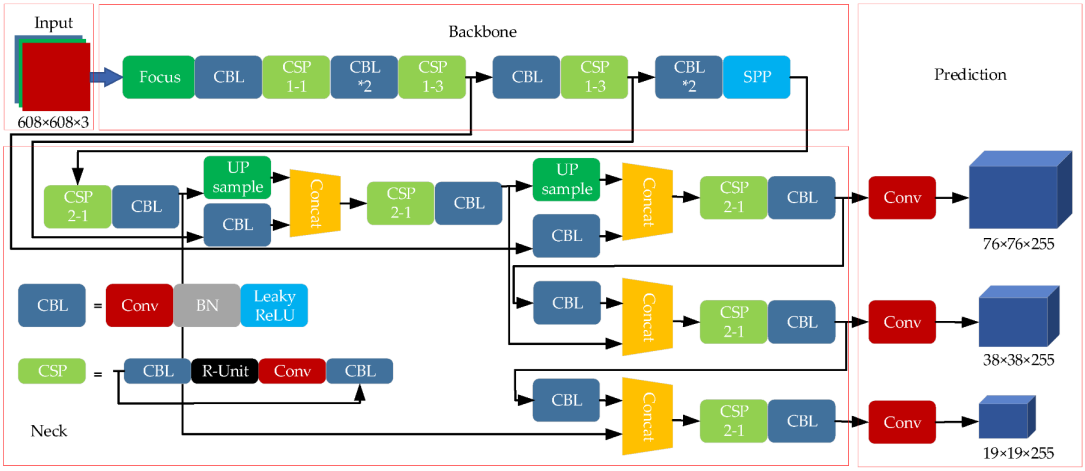

Figura 3.2: Arquitectura de YoloV5.

A continuación, se explica la arquitectura general de YOLOv5, que es una versión moderna de esta serie.

- Entrada de imagen: La entrada a la red es una imagen de un tamaño fijo. En YOLOv5, el tamaño comúnmente utilizado es 640x640 píxeles, pero esto puede variar según la configuración.
- Backbone convolucional: YOLOv5 utiliza una red convolucional como "backbone"para extraer características de la imagen. En versiones recientes, se emplean backbones eficientes y potentes como CSPDarknet53 o CSPDarknet53-PANet.
- Encabezados de múltiple: YOLOv5 utiliza múltiples encabezados (headers) para realizar detecciones en diferentes escalas. Cada encabezado produce un conjunto de detecciones. En YOLOv5, hay 3 encabezados de detección que trabajan en diferentes resoluciones de la imagen.
- Predicciones de detección: Cada encabezado genera predicciones de detección que consisten en cajas delimitadoras (bounding boxes), clases de objetos y confianza (score) para cada detección. Estas predicciones se realizan en términos de anclajes (anchors) y son ajustadas de acuerdo a la forma de la cuadrícula en la que se realizan las detecciones.
- Supresión de NMS: Después de la detección, se aplica un algoritmo de NMS para eliminar detecciones redundantes y mantener solo las detecciones más confiables.
- Salida: La salida final de la red son las detecciones de objetos, que incluyen las coordenadas de las cajas delimitadoras, las clases de objetos detectados y las puntuaciones de confianza asociadas.

3.3. Entrenamiento

YOLOv5 se entrena utilizando un conjunto de datos etiquetado que contiene imágenes con coordenadas de cajas delimitadoras y etiquetas de clases. El modelo ajusta sus pesos mediante el proceso de retropropagación para minimizar la pérdida entre las predicciones y las etiquetas reales. Las variables de diseño usadas son:

- Epochs: Un epoch es una pasada completa a través de todo el conjunto de datos de entrenamiento. El número de epochs determina cuántas veces se entrenará la red en todo el conjunto de datos de entrenamiento. Se debe tener cuidado con la posibilidad de sobreajuste. Este valor es el que se suele iterar para obtener mejores resultados.
- Batch size: El batch size determina cuántas imágenes se utilizan en cada paso de entrenamiento. Un tamaño de lote más grande puede acelerar el entrenamiento, pero requerirá más memoria.

3.3.1. Procedimiento

Se implementó un código en google colab usando el repositorio de ultralytics que incluyen la librería de Pytorch y Comet. Con esta instrucción se importan las funciones para el entrenamiento y para correr el algoritmo. El siguiente paso fue cargar el dataset preprocesado que se explicó en la anterior sección. Luego, se escogen los epochs y el batch size de entrenamiento. Se verifica la ruta del archivo *data.yaml* y del dataset, se verifica el tamaño de la imagen del parámetro $--img$ y la ruta de destino para los pesos de la red. Se ejecuta el código correspondiente y se puede ir observando el progreso de entreno.

3.3.2. Resultados

Durante las pruebas y evaluaciones, nuestro modelo demostró una alta tasa de precisión al asignar objetos a las categorías correspondientes en condiciones ideales. Esto valida la efectividad de nuestro enfoque de entrenamiento y la robustez del modelo en la identificación de objetos en situaciones estándar.

Figura 3.3: Ejemplo de una imágen tomada de internet.

Sin embargo, es importante destacar que, a medida que enfrentamos desafíos más complejos, como cambios de perspectiva, hemos observado una disminución en la capacidad de nuestro modelo para identificar las categorías en su totalidad. Estos cambios de perspectiva pueden incluir variaciones en la orientación, iluminación o ángulo de visión de la cámara. En tales casos, el modelo puede experimentar dificultades para asignar con precisión categorías a objetos. Como se puede ver en la figura 3.4 no se logra detectar la señal de límite de velocidad de 30 km/h, ya que el dataset se entreno con imagenes en donde la perspectiva era de frente.

Figura 3.4: Ejemplo de una imágen tomada en Normandia.

Un componente crucial para mejorar la precisión del modelo radica en el preprocesamiento de las imágenes. La mayoría de las imágenes, especialmente aquellas capturadas con cámaras no profesionales, suelen contener un nivel significativo de ruido. Para abordar este desafío, se emplean técnicas de filtrado de imágenes, siendo el filtro de Gauss una de las opciones preferidas en nuestro enfoque. Este proceso de filtrado tiene un impacto significativo en la calidad de las imágenes al reducir el ruido, permitiendo que las imágenes de entrada sean más limpias y coherentes. Como resultado, se logra una mejora sustancial en la precisión de la clasificación, dado que el modelo trabaja con datos de entrada más refinados y consistentes.

Figura 3.5: Ejemplo de una imágen tomada en la Universidad Nacional de Colombia.

A pesar de estas limitaciones, este proyecto representa un paso significativo en la dirección de la automatización y la mejora de la eficiencia en la identificación de categorías. Los resultados obtenidos subrayan la importancia de seguir perfeccionando y ajustando el modelo para abordar situaciones de cambio de perspectiva y aumentar su robustez.

Una parte fundamental del proceso de evaluación de nuestro modelo de inteligencia artificial consistió en probar su rendimiento en una amplia variedad de condiciones. Esto incluyó la evaluación de videos tanto diurnos como nocturnos, y los resultados obtenidos son especialmente alentadores. Durante las pruebas con videos diurnos, el modelo demostró una habilidad sobresaliente para detectar y clasificar objetos en diversas categorías. La luminosidad adecuada y las condiciones de iluminación consistentes durante el día no representaron un desafío significativo para la precisión de las predicciones del modelo.

En cuanto a las grabaciones nocturnas, es importante señalar que el modelo enfrentó desafíos adicionales debido a la falta de luz natural. A pesar de estos desafíos, el modelo logró mantener un nivel de detección, aunque no tan alto ni tan preciso como en las condiciones diurnas. Este resultado resalta que, si bien el modelo es versátil, aún existe margen para mejorar su capacidad de detección en entornos de poca luz.

En general, los resultados del entrenamiento se evidencian en las gráficas de la figura 3.6

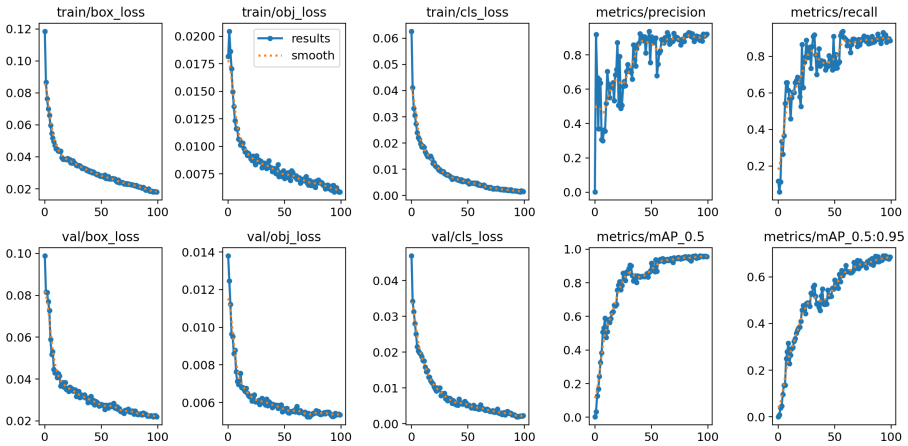

Figura 3.6: Resultados del entrenamiento.

Se observa que el mAP es bastante aceptable, tanto para un umbral de IoU de 0.5 como para el de 0.95 se observa que logra llegar, pero se sigue observando que para el caso de 0.95 llega a un valor más bajo, esto es evidente dado que es una métrica más exigente, pero igual reporta datos aceptables (0.65). En cuanto a las métricas de presición y recall, se observa un grado muy bueno (0.8) de valor para estas, lo cual sugiere que el modelo logra detectar las imágenes correctamente y con buena frecuencia. Por otra parte, se observa que la pérdida de cajas (box loss) se logra reducir hasta una saturación muy cercana a 0, sugiriendo que el algoritmo está bien capacitado para encontrar los centros del objeto, así mismo, para el caso de pérdida de objetos (obj loss) se reduce también mucho (casi a 0) sugiriendo que se logra detectar el objeto, finalmente, la misma tendencia ocurre con la pérdida en la clasificación (cls loss), donde se observa que el algoritmo logra clasificar los objetos identificados correctamente en sus etiquetas.
Para el caso de los datos de validación ocurren las mismas tendencias, por lo que se concluye de manera general que el algoritmo entrenado es aceptable.

4. Conclusiones

De este segundo avance se concluye que:

- El preproceso de las imágenes del data set para reducir su tamaño es de vital importancia cuando se usan servicios en la nube con almacenamiento limitado, además permite un entreno más rápido.
- Se observa que el algoritmo pierde robustez cuando la perspectiva de los objetos cambia, sin embargo, sigue siendo muy buena. Una forma de solucionar esto es añadiendo al dataset imágenes de los objetos en distintas perspectivas y volver a entrenar.
- Para la futura entrega se recomienda evaluar también imágenes en diferentes estados de tiempo y desde más perspectivas, así como observar y redefinir las clases para que sean únicamente las clases de interés de esta aplicación.

Bibliografía

[1] "Los Ciegos en el censo 2018: Instituto Nacional Para Ciegos," www.inci.gov.co, https://www.inci.gov.co/blog/los-ciegos-en-el-censo-2018 (accessed Sep. 24, 2023).

[2] Noticias de la Ciencia, "Nuevo Prototipo Sónico de Ayuda a La Ceguera," Noticias de la Ciencia y la Tecnología (Amazings® / NCYT®), https://noticiasdelaciencia.com/art/9286/nuevo_prototipo_sonico_de_ayuda_a_la_ceguera (accessed Sep. 24, 2023).

[3] "Seeing AI app from Microsoft," Seeing AI App from Microsoft, https://www.microsoft.com/en-us/ai/seeing-ai (accessed Sep. 24, 2023).

[4] "Llegan a Colombia las gafas inteligentes: Instituto Nacional para Ciegos," www.inci.gov.co, https://www.inci.gov.co/blog/llegan-colombia-las-gafas-inteligentes (accessed Sep. 24, 2023).

[5] Panasonic, "Gafas Inteligentes para las personas con Discapacidad Visual," Blog de Panasonic España, https://blog.panasonic.es/innovacion/gafas-inteligentes-discapacidad-visual-biel-glasses/ (accessed Sep. 24, 2023).

[6] Presti, G., Ahmetovic, D., Ducci, M., Bernareggi, C., Ludovico, L., Baratè, A., Avanzini, F., Mascetti, S.: Watchout: obstacle sonifcation for people with visual impairment or blindness. In: The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 402–413 (2019)

[7] Lin, B.-S., Lee, C.-C., Chiang, P.-Y.: Simple smartphone-based guiding system for visually impaired people. Sensors 17(6), 1371 (2017)

[8] Facil, J.M., Ummenhofer, B., Zhou, H., Montesano, L., Brox, T., Civera, J.: Cam-convs: camera-aware multi-scale convolutions for single-view depth. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11826–11835 (2019)

[9] Bharatia, D., Ambawane, P., Rane, P.: Smart electronic stick for visually impaired using android application and Google's cloud vision. In: 2019 Global Conference for Advancement in Technology (GCAT), pp. 1–6. IEEE (2019)

[10] Hakim, H., Fadhil, A.: Navigation system for visually impaired people based on RGB-D camera and ultrasonic sensor. In: Proceedings of the International Conference on Information and Communication Technology, ICICT '19, pp. 172–177. Association for Computing Machinery, New York (2019). ISBN 9781450366434

[11] Dosi, S., Sambare, S., Singh, S., Lokhande, N., Garware, B.: Android application for object recognition using neural networks for the visually impaired. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1–6. IEEE (2018)

[12] M. I. Thariq Hussan, D. Saidulu, P. T. Anitha, A. Manikandan and P. Naresh (2022), Object Detection and Recognition in Real Time Using Deep Learning for Visually Impaired People. IJEER 10(2), 80-86. DOI: 10.37391/IJEER.100205.

[13] Joshi, R. C., Yadav, S., & Dutta, M. K. (2020, February). YOLO-v3 based currency detection and recognition system for visually impaired persons. In 2020 International Conference on Contemporary Computing and Applications (IC3A) (pp. 280-285). IEEE.

[14] Rahman, F., Ritun, I. J., Farhin, N., & Uddin, J. (2019, January). An assistive model for visually impaired people using YOLO and MTCNN. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (pp. 225-230).