

# **Informe para proyecto de Aprendizaje de Máquina**

**Safecross: Una propuesta de aplicación auxiliar  
para cruce de calles**

## **Integrantes:**

- Nahomi Bouza Rodríguez
- Francisco Vicente Suárez Bellon
- Carlos Arturo Pérez Cabrera
- Diana Laura Pérez Trujillo

# Contents

<b>Introducción</b>	<b>2</b>
Motivación . . . . .	2
Problemática . . . . .	3
Problemática General . . . . .	3
Problemática Específica . . . . .	3
Objetivos Generales y específicos. Preguntas científicas . . . . .	3
Objetivo General . . . . .	3
Objetivos Específicos . . . . .	4
<b>Estado del Arte y Preliminares</b>	<b>5</b>
Avances Tecnológicos y Herramientas de Asistencia . . . . .	5
Tecnología de Asistencia Visual [ <b>Islam2019DevelopingWA</b> ] . . . . .	5
Detección de Objetos en Imágenes [ <b>Zhao2018ObjectDW</b> ] . . . . .	6
Detección de Peatones . . . . .	8
YOLO [ <b>Terven2023ACR</b> ], [ <b>Redmon2015YouOL</b> ] . . . . .	8
<b>Propuestas de Solución</b>	<b>9</b>
<b>Implementación de la solución, Experimentación y Resultados</b>	<b>10</b>
Detección de luces de semáforos . . . . .	10
Segundo Modelo entrenado . . . . .	14
Cambio de resolución de las imágenes . . . . .	22
Fotos Nocturnas . . . . .	26
Detección de Cruce de Peatones . . . . .	33
Primer Modelo . . . . .	33
Segundo Modelo . . . . .	34
Detección de Autos . . . . .	37
Resultados Finales . . . . .	38
<b>Discusión de los Resultados. Repercusión ética de las soluciones</b>	<b>41</b>
<b>Conclusiones y trabajo futuro</b>	<b>43</b>
<b>Bibliografía</b>	<b>43</b>

## **Introducción**

Cruzar la calle puede ser especialmente desafiante para las personas ciegas o con discapacidad visual, ya que dependen de sus otros sentidos y de la ayuda de otras personas para navegar de manera segura en entornos públicos. Esto puede llevar a situaciones peligrosas y a una menor independencia y autonomía. En este proyecto se propone realizar una investigación acerca de cómo se podría utilizar la rama del Aprendizaje de Máquinas para intentar crear opciones viables y seguras para el cruce de calles y avenidas para personas con discapacidad visual.

Llevará de la mano un proceso de experimentación y resultados que finalmente serán analizados y servirán para vislumbrar futuros caminos de investigación y mejoras.

## **Motivación**

Según datos aportados por la Organización Mundial de la Salud (OMS), en el Reporte Ceguera y Discapacidad Visual del año 2023, existen en el mundo, al menos, 2200 millones de personas con deterioro de la visión cercana o lejana. Según otros datos, del año 2020, existían 43 millones de personas con ceguera total, 295 millones con deficiencia visual de moderada a grave y 258 millones de personas con deficiencias leves.

En Cuba, según datos del 2016 publicados por el Informe Nacional de Censo de Población y Viviendas, del 5% de las personas con alguna discapacidad en el país, un 38% presentan discapacidad visual y un 3% presentan ceguera total.

Una de las dificultades que estas condiciones de vida pueden presentar es con respecto a la seguridad de la persona en las calles. Particularmente, en los pasos de vehículos y avenidas.

¿Por qué no aprovechar las bondades de la tecnología en este siglo para mejorar esta situación? Sería valioso encontrar una manera de garantizar un cruce de calle seguro para personas que no sean capaces de distinguir o ver las señalizaciones y semáforos, la cercanía de vehículos o personas.

## **Problemática**

### **Problemática General**

Para garantizar un cruce seguro, es necesario realizar un análisis visual del entorno y responder las siguientes interrogantes: ¿Está libre el paso?, ¿Hay vehículos cerca?, ¿Hay personas cerca?, ¿Está el semáforo en verde?, etc. Una persona con discapacidad visual no puede responder a estas preguntas de manera efectiva.

Una aplicación auxiliar que realice este análisis mediante imágenes y responda estas interrogantes. Nótese que se comenta de un problema bastante extenso. Existen calles y avenidas con cruces de 4 o 6, rotundas y cada uno tiene sus características específicas, existen cruces donde los vehículos tienen vía libre siempre que vayan por una senda, otros donde no es permitido girar hacia un lado, entre otras particularidades del diseño de las leyes y reglas del tráfico de cada lugar. Incluso, existen muchos tipos de semáforos, diferentes no solo en su enfoque (peatones, vehículos) sino en su estilo, orden, etc. (Ejemplo: en Berlín hay un tipo de semáforo con 13 señales diferentes y en Japón en algunas localidades existen semáforos con luz azul).

El problema tiene muchas vertientes y especificaciones en dependencia de muchos factores. Por ello el presente trabajo propone un enfoque más específico y busca resolver dicho subproblema.

### **Problemática Específica**

Como parte del problema general, se busca resolver aquellos casos donde la persona se encuentra frente a un cruce peatonal que es, en realidad, la opción más segura de cruce para una persona con discapacidad visual.

Para resolver este caso, es necesario que de alguna manera se detecten la presencia de cruces peatonales, vehículos que puedan ir transitando por el cruce, y distinguir entre las luces del semáforo. Con toda esta información, realizar una conclusión acerca de si es seguro y viable efectuar el cruce o no.

## **Objetivos Generales y específicos. Preguntas científicas**

### **Objetivo General**

El objetivo general de este proyecto resulta en la implementación de una aplicación auxiliar para personas que padecan ceguera total o discapacidad visual

severa, que dado una imagen, y el posterior análisis interno de la misma, se decida si es seguro realizar el cruce, si no lo es o si no es determinable.

### **Objetivos Específicos**

- Realizar un análisis de las imágenes para detectar cruces peatonales.
- Realizar un análisis de las imágenes para detectar vehículos.
- Realizar un análisis de las imágenes para detectar luces de semáforos.
- Realizar una conclusión sobre si es seguro cruzar o no.

# Estado del Arte y Preliminares

## Avances Tecnológicos y Herramientas de Asistencia

### Tecnología de Asistencia Visual [Islam2019DevelopingWA]

La tecnología de asistencia visual se puede dividir en tres áreas principales: mejora de la visión, sustitución de la visión y reemplazo de la visión. Entre estas, las ayudas electrónicas de viaje (ETA) que recopilan información del entorno mediante sensores, cámaras y smartphones son las más prometedoras. Las ETAs están disponibles en formatos portátiles y de mano y ayudan a las personas ciegas a detectar obstáculos y orientarse.

#### Dispositivos Basados en Sensores:

Los dispositivos basados en sensores son comunes debido a su simplicidad y eficacia en entornos específicos. Entre ellos, los bastones electrónicos han sido una solución popular. Por ejemplo:

- **Sensory Guidance System:** Utiliza sensores ultrasónicos para detectar obstáculos y emitir vibraciones cuando el usuario se aproxima a uno.
- **NavGuide:** Combina sensores ultrasónicos con retroalimentación acústica y vibratoria para ayudar en la navegación.

Aunque estos dispositivos son efectivos, suelen tener un alcance de detección limitado y su efectividad puede verse comprometida por factores ambientales como la temperatura y la humedad.

#### Tecnología de Radar

- **Dispositivo de Cardillo et al.:** Utiliza radar de microondas para detectar obstáculos.
- **Herramienta de Kwiatkowski et al.:** Emplea radar FMCW para transformar la distancia a obstáculos en señales acústicas tridimensionales.

Estas tecnologías también tienen limitaciones, como un alcance limitado y la complejidad en el diseño de la antena.

### **Dispositivos Basados en Visión por Computadora:**

Los dispositivos basados en visión por computadora representan una de las tecnologías más avanzadas en este campo. Ejemplos incluyen:

- **Sistema de Yang et al.:** Utiliza cámaras RGB e IR junto con procesadores de imagen RealSense y auriculares de conducción ósea para proporcionar retroalimentación auditiva detallada.
- **Sistema de Bai et al.:** Combina cámaras de profundidad con sensores ultrasónicos para generar señales de audio que ayudan en la navegación.

Sin embargo, estos sistemas son altamente dependientes de las condiciones de iluminación y pueden enfrentar problemas con sombras y reflejos, lo que afecta su precisión y eficacia.

### **Dispositivos Basados en Smartphones:**

Los avances en los smartphones han permitido el desarrollo de asistentes de caminata basados en estos dispositivos, ofreciendo una solución flexible y accesible. Ejemplos incluyen:

- **Sistema de Alghamdi et al.:** Utiliza tecnología RFID para ayudar en la navegación tanto en interiores como en exteriores, comunicándose con el usuario a través de señales de audio.
- **EyeMate de Tanveer et al.:** Usa gafas conectadas a un smartphone para proporcionar navegación mediante señales de voz en bengalí o inglés.

A pesar de su conveniencia, estos sistemas enfrentan limitaciones en el rango de detección y pueden ser menos efectivos en entornos ruidosos, además de que la operación de smartphones puede ser un desafío para algunos usuarios.

### **Detección de Objetos en Imágenes [Zhao2018ObjectDW]**

Para comprender completamente las imágenes, es crucial enfocarse en la detección de objetos, que implica clasificar y localizar objetos dentro de las imágenes. Esta tarea es fundamental para aplicaciones como la clasificación de imágenes, análisis del comportamiento humano, reconocimiento facial y conducción autónoma.

**Pipeline de Modelos Tradicionales:** El pipeline de modelos tradicionales

de detección de objetos generalmente incluye tres etapas: selección de regiones informativas, extracción de características y clasificación.

- **Selección de Regiones Informativas:** Tradicionalmente emplea enfoques de ventana deslizante multi-escala para explorar imágenes en busca de objetos, equilibrando entre cobertura exhaustiva y eficiencia computacional.
- **Extracción de Características:** Técnicas como Transformación de Características Invariantes a Escala (SIFT), Histograma de Gradientes Orientados (HOG) y características tipo Haar proporcionan representaciones visuales robustas pero enfrentan dificultades con apariencias y fondos diversos.
- **Clasificación:** Métodos como Máquinas de Soporte Vectorial (SVM), AdaBoost y modelos basados en partes deformables (DPM). Los DPMs destacan en el manejo de deformaciones mediante modelado basado en partes y aprendizaje gráfico.

#### **Avances Recientes con Redes Neuronales Profundas (DNNs) [Damini2024AnalysisOO]**

La emergencia de las Redes Neuronales Profundas (DNNs) ha revolucionado la detección de objetos al abordar varios desafíos de large data. A diferencia de los enfoques tradicionales dependientes de descriptores diseñados manualmente y modelos superficiales, las DNNs, especialmente las Redes Neuronales Convolucionales (CNNs), ofrecen arquitecturas más profundas capaces de aprender características complejas de manera autónoma. Este cambio comenzó con las Regiones con Características de CNN (R-CNN), marcando una mejora significativa en la precisión de la detección de objetos.

Modelos Destacados: R-CNN, Fast R-CNN, Faster R-CNN y YOLO. Cada uno ha contribuido a avances significativos en precisión y eficiencia de detección. Desafíos: Variaciones en el punto de vista, occlusiones y condiciones de iluminación.

## Detección de Peatones

### Desafíos Específicos:

- **Instancias Pequeñas:** En escenarios típicos de detección de peatones (como conducción automática y vigilancia inteligente), hay muchas instancias pequeñas de peatones. La aplicación de capas de RoI pooling en el pipeline genérico de detección de objetos puede resultar en características "planas" debido a la agrupación de bins.
- **Predicciones Falsas:** La principal causa de predicciones incorrectas es la confusión con instancias de fondo difíciles.

### Enfoques Avanzados:

- **IModificaciones del Faster R-CNN:** Zhang et al. intentaron adaptar el Faster R-CNN genérico a la detección de peatones. Modificaron el clasificador downstream agregando bosques potenciados a mapas de características convolucionales compartidas de alta resolución y empleando una RPN para manejar instancias pequeñas y ejemplos negativos difíciles.
- **IModelos Basados en Partes:** Tian et al. propusieron DeepParts, que toma decisiones basadas en un conjunto de detectores de partes extensas, manejando occlusiones parciales.
- **ICombinación de Fuentes de Datos:** Liu et al. propusieron redes neuronales profundas multi-espectrales para la detección de peatones, combinando información complementaria de imágenes en color e imágenes térmicas. Tian et al. propusieron una CNN asistente de tarea (TA-CNN) para aprender conjuntamente múltiples tareas con múltiples fuentes de datos y combinar atributos de peatones con atributos semánticos de la escena.

## YOLO [Terven2023ACR], [Redmon2015YouOL]

YOLO (You Only Look Once) es un algoritmo de detección de objetos de aprendizaje profundo que ha revolucionado el campo gracias a su eficiencia y precisión. Su arquitectura única y su capacidad para realizar la detección en una sola pasada lo convierten en una herramienta poderosa para diversas aplicaciones.

### Arquitectura:

YOLO utiliza una red neuronal convolucional (CNN) para dividir la imagen de entrada en una cuadrícula. Cada celda de la cuadrícula es responsable de predecir la presencia de objetos dentro de su área, así como su clase y posición. Para lograr esto, la red predice una serie de cuadros delimitadores (bounding boxes) junto con probabilidades de confianza para cada clase.

#### Ventajas:

- **Velocidad:** YOLO procesa imágenes a altas velocidades, lo que lo convierte en ideal para aplicaciones en tiempo real. Esto se debe a que realiza la detección de objetos en una sola pasada, a diferencia de los métodos tradicionales que requieren múltiples etapas.
- **Precisión:** YOLO ha demostrado ser preciso en la detección de objetos, con una alta tasa de precisión en diversas tareas.
- **Generalización:** YOLO es capaz de generalizar bien a diferentes tipos de objetos y escenarios, lo que lo convierte en un algoritmo versátil.
- **Sensibilidad a pequeñas variaciones:** YOLO es más sensible a pequeñas variaciones en el tamaño y la posición de los objetos en comparación con otros métodos.

## Propuestas de Solución

Como se mencionó anteriormente, se tienen varios objetivos específicos a cumplir, constituyendo cada uno un subproblema a resolver.

El primer paso de la solución sería identificar la presencia o no de un cruce peatonal. Para ello se propone el uso del modelo Yolov8m, al que se le ha realizado un proceso de fine tuning, para mejorar la detección de cruces peatonales más ajustados al caso en cuestión. Para la detección de las luces de semáforos, se propone la utilización del modelo Yolov8m. Con dicho modelo, realizar un proceso de fine-tuning para mejorar la detección de las luces de los semáforos peatonales (rojo, verde) y en caso de no haber semáforo peatonal con luz visible (dígase no hay semáforo peatonal en la imagen o se encuentra apagado) no se realizan detecciones. Se tendrán en cuenta casos como fotos nocturnas, donde la luz se difumina bastante y casos con imágenes con diferentes resoluciones.

Finalmente se propone una aplicación que determina, basada en los resultados de los modelos si es seguro cruzar, si no se puede cruzar o si no se puede definir. En la siguiente sección se detallará más el proceso de la solución.

## **Implementación de la solución, Experimentación y Resultados**

### **Detección de luces de semáforos**

Como se comentó en la sección anterior, se utilizó un modelo pre-entrenado Yolov8m, al que se le realizó fine-tuning, pues el problema específico que se quería resolver era el de detección de luces de semáforos peatonales.

Los experimentos próximamente mostrados, fueron desarrollados en un equipo con las siguientes características: Intel Core i5-13420H, 32GB de RAM, NVIDIA GeForce RTX 3050. En principio, se utilizó un conjunto de datos preexistente con imágenes clasificadas en dos categorías: rojo y verde. Este dataset permitía identificar el color de la luz del semáforo cuando estaba presente. Utilizando este dataset se pasó por varias fases. La distribución de las imágenes es la siguiente: Se utilizaron 2979 imágenes para el entrenamiento, 1273 imágenes de validación y 3100 imágenes para test. En primera instancia se entrenó el modelo, utilizando un valor de epochs = 20. La fase del entrenamiento tuvo una duración de aproximadamente 7h.

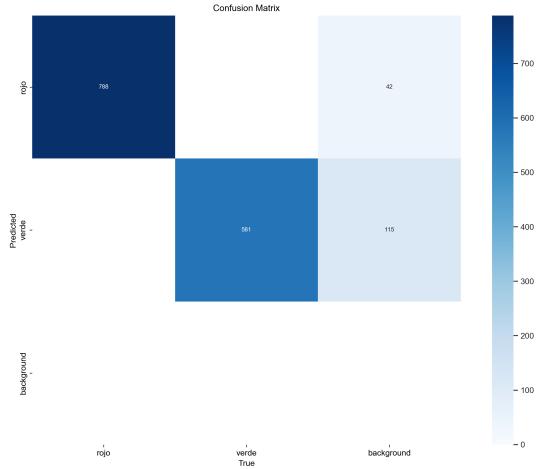


Figure 1: Matriz de confusión de modelo entrenado con 20 épocas

Se puede observar buenos resultados en cuanto a la diferenciación entre verde y rojo, no confundiendo una luz de un semáforo rojo e identificándola con una luz de semáforo verde. Sin embargo se observa en algunos casos que detecta objetos en el fondo de la imagen como luces de semáforos, que en este caso no son deseadas.

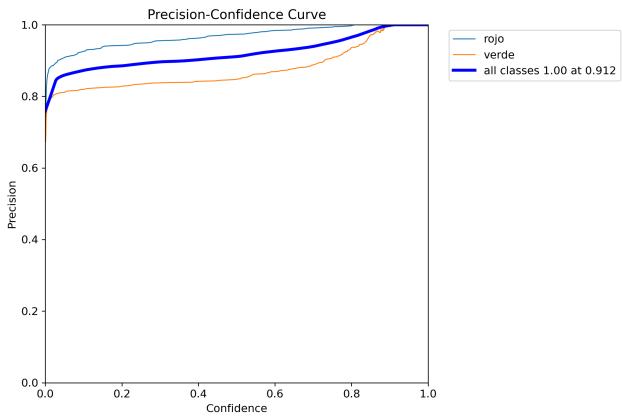


Figure 2: Curva precisión-confianza

En la gráfica de Curva precisión-confianza se observa que la precisión aumenta a medida que aumenta la confianza y que es para la clase roja donde se obtiene una mayor precisión.

En la siguiente imagen se muestra un ejemplo del desempeño del modelo con las predicciones:

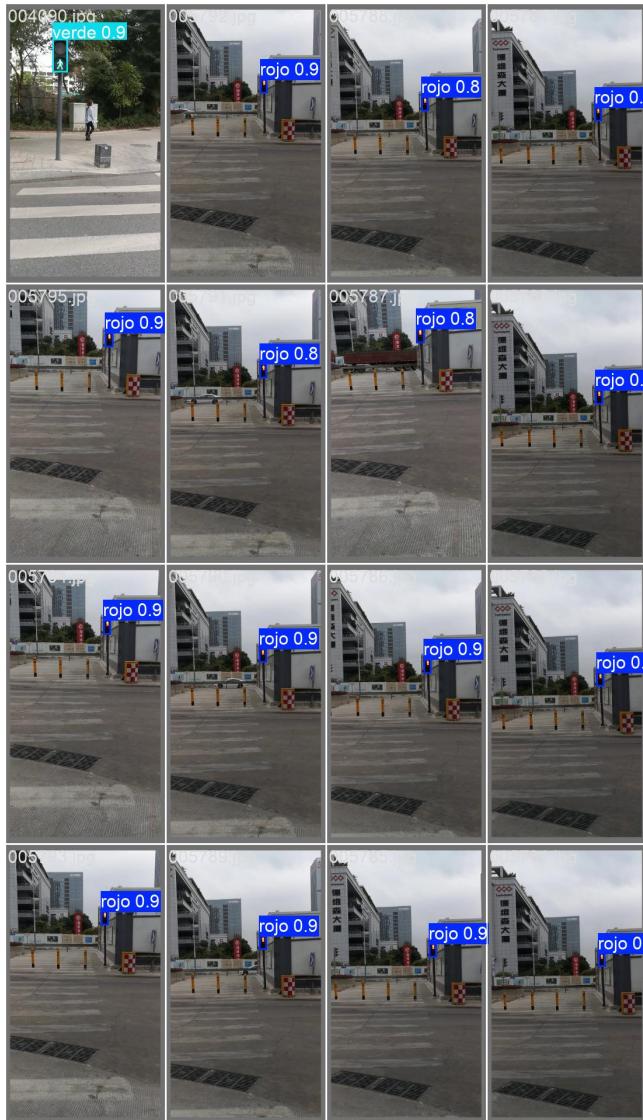


Figure 3: Ejemplo de predicciones

### **Segundo Modelo entrenado**

Se experimentó con un segundo modelo, esta vez entrenado con un número de epochs=80. Esta vez, tuvo una duración de aproximadamente 26h. Se buscaba una mejoría en los resultados, sobre todo en los casos donde el anterior modelo confundía objetos del fondo con luces de semáforos peatonales.

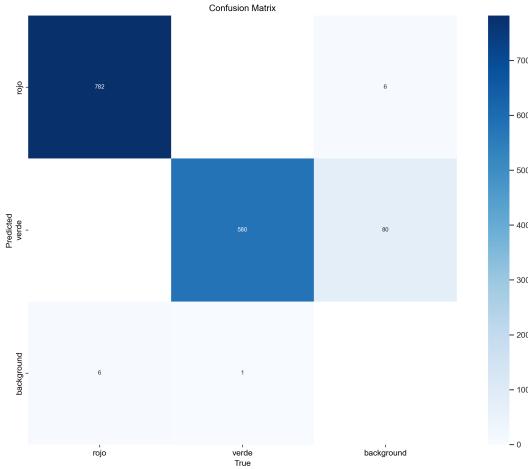
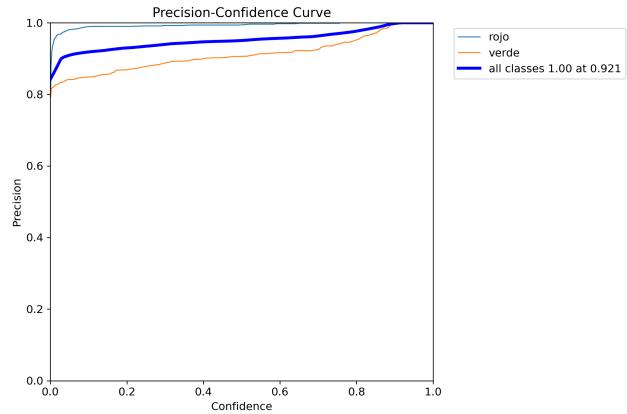


Figure 4: Matriz de confusión de modelo entrenado con 80 épocas

En la matriz de confusión se observa una mejora en la detección de luces de semáforos peatonales, la cantidad de falsos positivos disminuyó y la cantidad de detectados positivos se mantuvo casi igual (1 caso menos en la clase de los verdes y 6 casos menos en la clase de los rojos). Se considera este modelo mejor que el anterior presentado, pues es preferible no poder determinar qué luz hay en el semáforo en alguna ocasión aislada, que confundir objetos del fondo con indicadores de semáforo.



En este caso el análisis es similar al anterior. la precisión aumenta a medida que aumenta la confianza, y es en la clase rojo donde mejor se obtiene.

Las siguientes imágenes son ejemplos del desempeño del modelo en las predicciones.



Figure 5: Ejemplo de predicción de luz peatonal verde

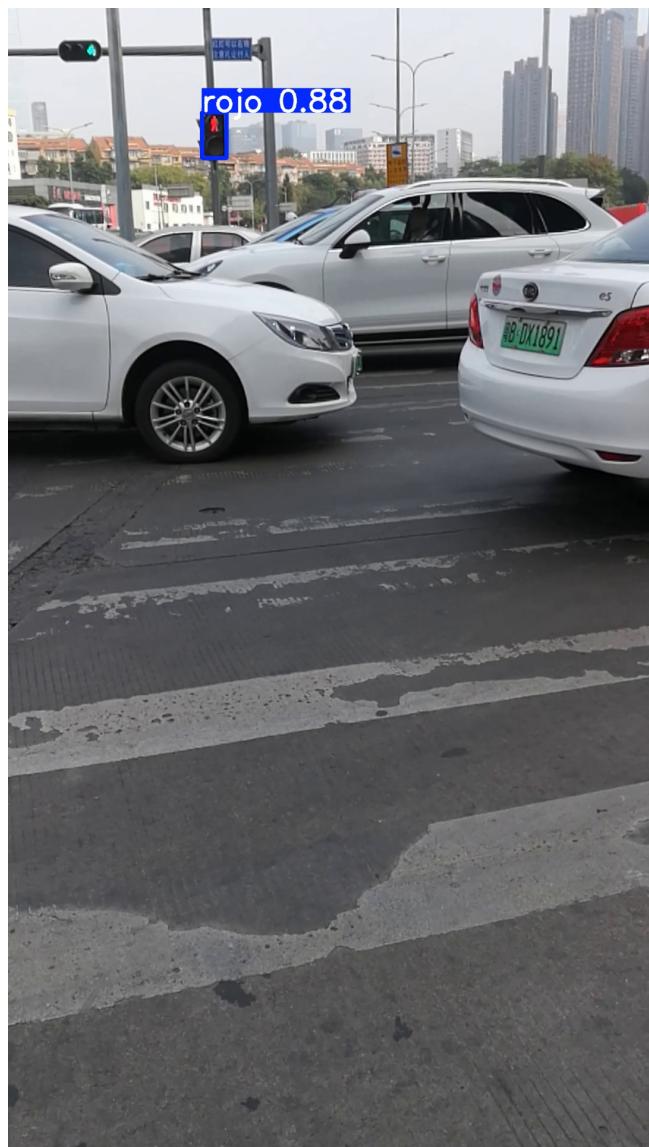


Figure 6: Ejemplo de predicción de luz peatonal roja. Nótese que solo identifica el semáforo peatonal.

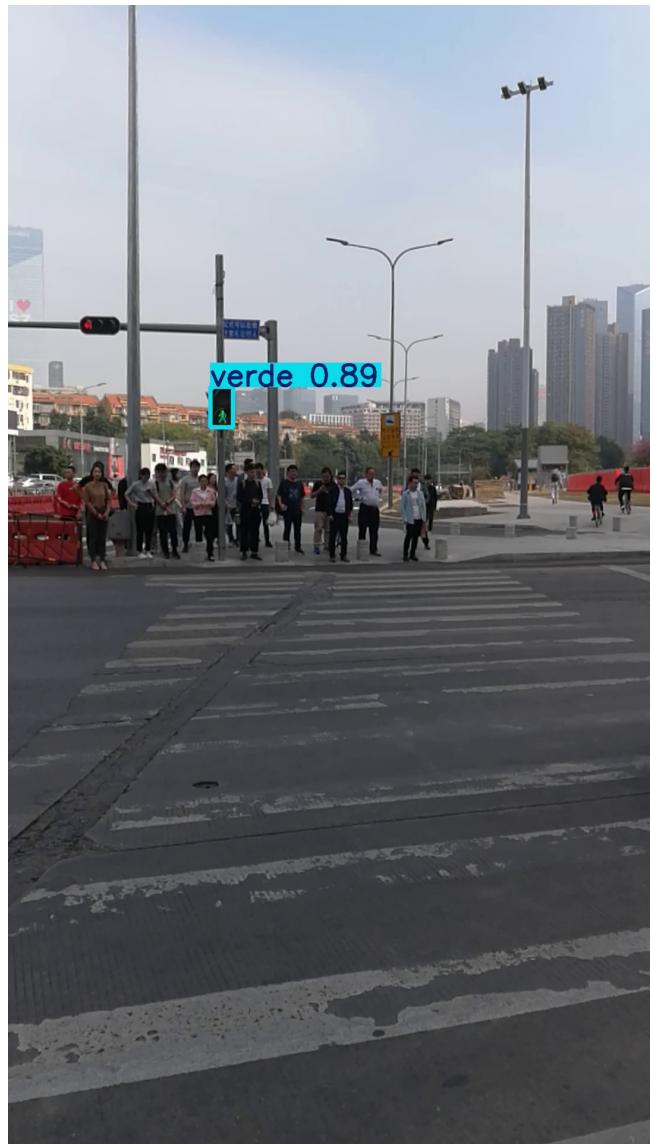


Figure 7: Ejemplo de predicción de luz peatonal verde.

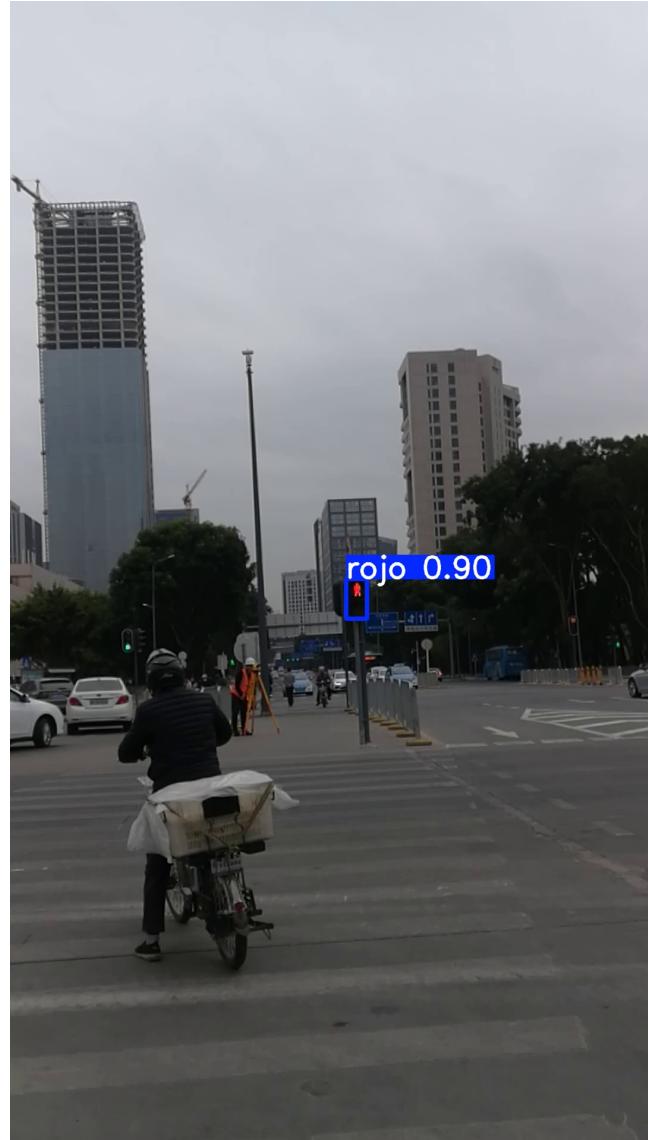


Figure 8: Ejemplo de predicción de luz peatonal roja



Figure 9: Ejemplo donde detecta un semáforo cercano y uno lejano.

En los casos como el último mostrado, en realidad a la hora de realizar el cruce, lo que más interesa es el estado del semáforo cercano.

### **Cambio de resolución de las imágenes**

En general, las imágenes utilizadas en los modelos, tenían una buena calidad. Se quiso probar qué pasaría si se redujera dicha calidad, qué tan bien el modelo lo reconocería o no. Esto podría ser un caso de la vida real donde el teléfono o dispositivo conque se tomó la imagen podría no tener buena resolución.

El primer experimento se llevó a cabo con imágenes a las que se les disminuyó la resolución hasta 480x600. En estos casos las imágenes se tornaron menos nítidas, pero el modelo supo identificar bien en su gran mayoría las luces.





En un segundo experimento con la resolución de las imágenes, se disminuyó a 320x200. Hubo varios casos donde el modelo no pudo identificar luces de semáforos (muchas veces difícil de detectar para el ojo humano).



Figure 10: Ejemplo donde sí detectó la luz.

### Fotos Nocturnas

En el conjunto de imágenes que se tenía, prevalecían las imágenes de día, con buena luz donde los semáforos y sus formas se veían bastante detalladas. Por ello surge la duda, qué pasaría con imágenes oscuras, tomadas de noche con poca luz.

Se creó un conjunto pequeño de imágenes tomadas de semáforos del Vedado (23 y 12, 23 y 26, 23 y Paseo y Línea y 12) desde varias perspectivas, de noche, con teléfonos de gama media (Xiaomi Redmi Note 8 Pro y Xiaomi Redmi 10 Pro).

Lo primero a resaltar es que los 4 semáforos tomados, presentaban algunas diferencias entre sí. Ejemplo, el de Línea y 12 no tenía numeros, uno de los de 23 y 12 estaban con los dibujos parcialmente apagados, en 23 y Paseo dos no funcionaban bien, entre otros detalles de ese estilo.

Con respecto a las luces, en los semáforos del Vedado, las rojas se ven mucho más definida que las verdes, debido a la intensidad de esta. En total se habían tomado aproximadamente 500 imágenes y 6 videos. Sin embargo, solo fueron viables finalmente 145 imágenes, de las cuales, la mayoría constituían luces rojas. En las imágenes descartadas la luz se veía demasiado intensa, al punto de perder la silueta y parecer un foco brillante circular. En el caso de los videos tomados, no se rescató nada valioso (Fueron tomados con la cámara de menor calidad).



Figure 11: Ejemplo donde no detecta la luz.

En un primer experimento, se decidió ver qué tal se desempeñaba el modelo que se tenía, con estas imágenes. Los resultados no fueron satisfactorios, así que se tomó la decisión de realizar fine-tuning nuevamente, encima de este modelo (el que ya se tenía entrenado con 80 épocas anteriormente).

Ante esta idea, surgió el problema de cómo distribuir las imágenes. A pesar de que se intentó variar las perspectivas de las fotos y rescatar todas las posibles, el conjunto era pequeño y algunas imágenes similares. Para la distribución de las imágenes se utilizó K-fold, finalmente con  $k=5$ . En esta ocasión y por la cantidad de imágenes, se escogió un valor de  $\text{epochs}=15$ .

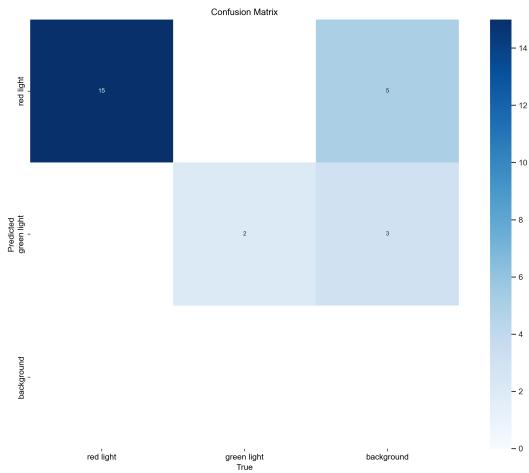


Figure 12: Matriz de confusión para modelo entrenado con 15 épocas y fotos nocturnas

Ejemplos del desempeño con fotos nocturnas:



Figure 13: Ejemplo de luz roja nocturna con los números incompletos

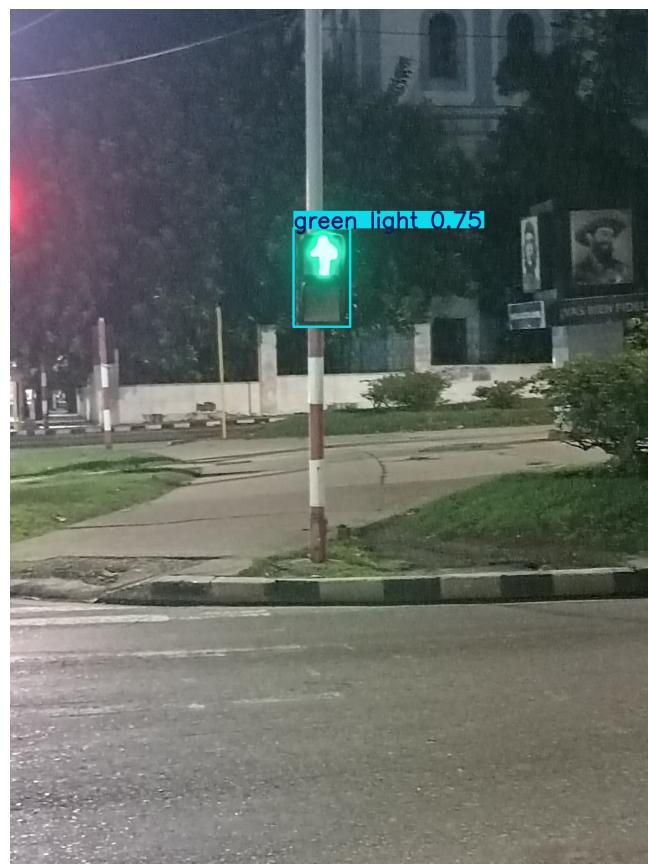


Figure 14: Ejemplo de luz verde nocturna sin números

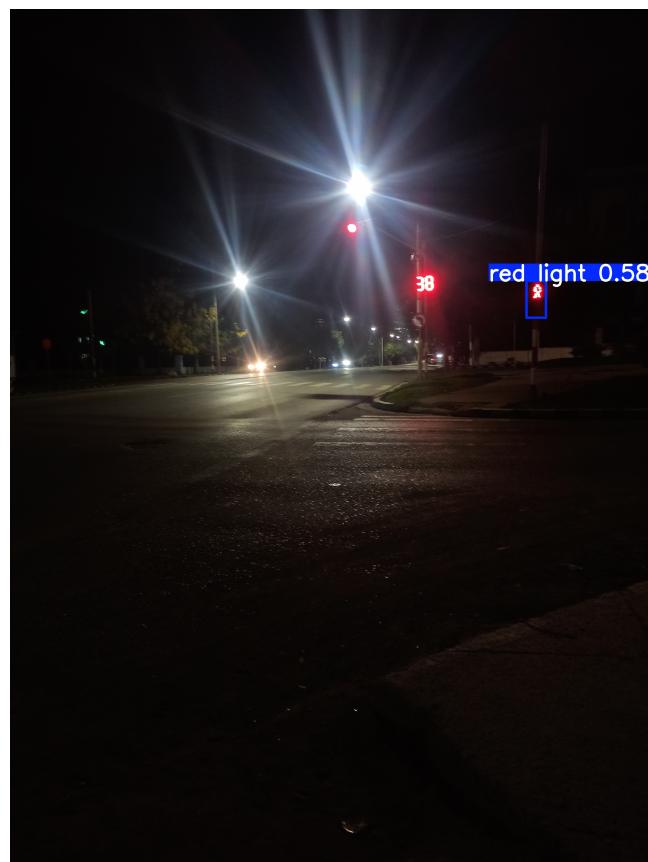


Figure 15: Ejemplo de luz roja nocturna sin números

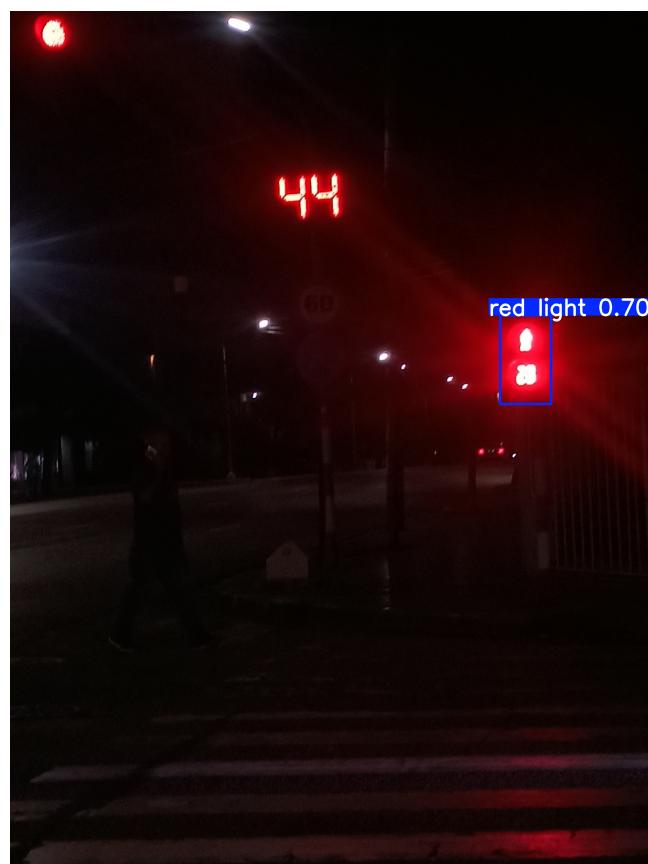


Figure 16: Ejemplo de luz roja nocturna con los números

Concluyendo la parte de detección de luces de semáforos peatonales, se considera que el modelo ha arrojado en general buenos resultados, siendo los peores casos que no detecte el semáforo (cuyo caso no va a repercutir gravemente sino que se tomará como indefinido).

## Detección de Cruce de Peatones

El problema a resolver es dado una imagen reconocer el paso de peatones en esta con el fin de poder establecer si puede cruzar o no.

Se estarán observando dos métricas importantes: la precisión dado que es importante que esté lo más cercano a 1 posible (en este caso un Falso positivo podría resultar mortal) y el recall pues se quiere que identifique correctamente las clases positivas.

Para la selección del dataset se buscó que contuvieran imágenes de cruce de peatones con su delimitador (para poder ser utilizado con el modelo), que estuvieran reflejadas varias perspectivas y que existieran ejemplos con personas o automóviles pasando por encima de ellos.

Para la tabla de test tienen 224 imágenes las cuales han sido seleccionadas para que existan representadas las distintas condiciones posibles así como ángulos de cámara, este modelo de test fue usado en todos los modelos. Para la correcta utilización de este se utilizaron modelos más grandes como Florence 2 para contrastar los resultados de los labels.

## Primer Modelo

El modelo se entrenó en una HP Omen 2019, CPU i7-9750H 16 GB de RAM, GPU Nvidia geforce 1660ti 6gbvram.

Se utilizó un dataset que contiene más de 2000 imágenes de cruce de peatones. Se le hizo un análisis de requerimientos con respecto a: calidad de la imagen, cantidad de cruces peatonales, contornos y sombras, brillo de la imagen.

Se distribuyeron 2179 imágenes de train y 314 de validación.

Este modelo mostró muy baja precisión con un máximo de 0.13 sobre la tabla de test. Después de un profundo análisis sobre este se determinó que dada la calidad de las imágenes de entrenamiento y validación era muy complejo que el modelo generalizara correctamente.

El entrenamiento se realizó desde 50 épocas como máximo hasta 150 con un valor de patience de 10. Su mejor resultado de recall fue con 40 épocas.

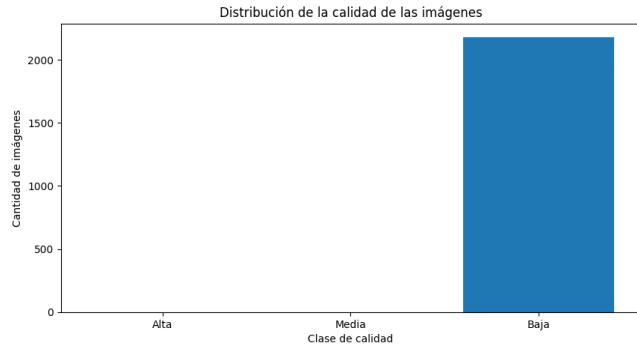


Figure 17: Distribución de Calidad dataset 1

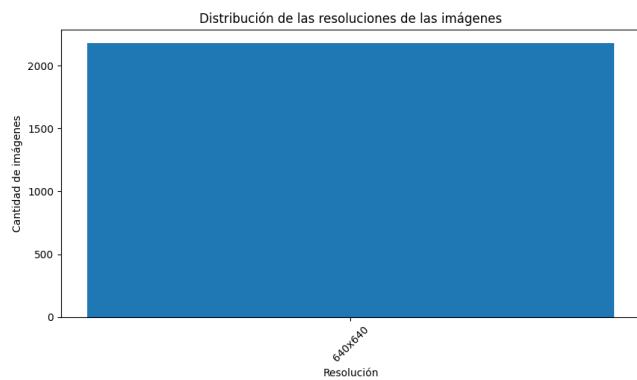


Figure 18: Resolución de las imágenes

## Segundo Modelo

El modelo se entrenó en una HP Omen 2019, CPU i7-9750H 16 GB de RAM, GPU Nvidia geforce 1660ti 6gbvram. Además, en esta ocasión se utilizó Notebooks de Kaggle con 24GB de RAM, 2 NVIDIA T4 16GB VRAM.

El dataset utilizado cuenta con 1046 imágenes de entrenamiento y 224 de validación.

Se revisó que el dataset cubriera diferentes escenarios: horarios variados (día, tarde, noche), condiciones climatológicas más usuales, niveles de desgaste del pavimento y la pintura del cruce (Dado que las fotos principalmente eran de Europa se procedió a aumentar con algunas muestras por el nivel de desgaste que se tiene) y colores del paso de peatones.

El entrenamiento se realizó siguiendo la API de ultralytics, con un máximo

de 300 épocas, 10 de patience. Inicialmente, la media fue de 50 épocas, con lo que se obtuvo una precisión de 96% y un recall de 0.93.

Para aumentar la precisión se tomó la decisión de dividir los bounding boxes en varios de estos, con el fin de hacer que la región a detectar sea menor y exista menos falsos positivos. Ello establece mayor prioridad a los box más cercanos a la imagen pudiendo distinguir con mayor claridad, además de ser útil para poder establecer distancias, evitar confusión con otros cruces. Se recomienda tomar la foto con una inclinación de la cámara de 45 grados para no evitar detecciones en zonas donde existan muchos pasos de peatones.

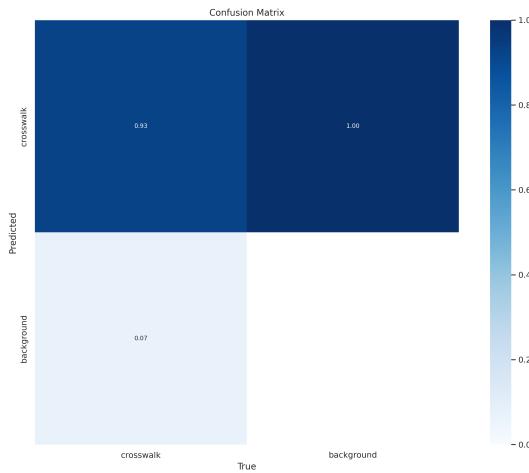


Figure 19: Matriz de confusión normalizada

En este modelo solo se cuenta con el label: crosswalk, que indica si es un cruce de peatones o no. (es decir que las únicas etiquetas que existirán serán de ese tipo). Se observa que en la mayoría de los casos detecta un cruce como tal y en la mínima no lo detecta.

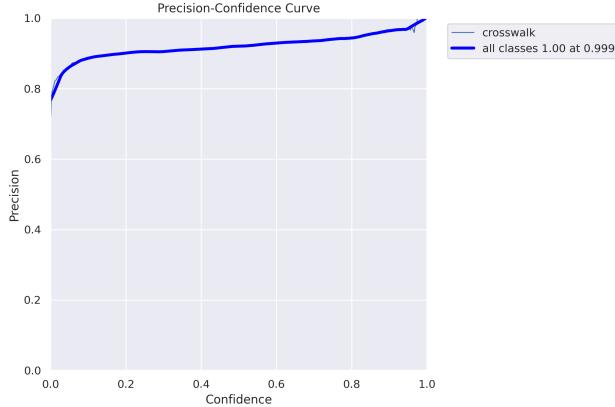


Figure 20: Curva precisión-confianza

Según se observa en la gráfica de precisión-confianza, el modelo tiende a ser preciso si tiene un valor de confianza alto.

## Detección de Autos

Como primer paso, se realizó una exhaustiva búsqueda de un dataset apropiado para el desarrollo del modelo de detección de objetos. Se consideraron diferentes factores como la cantidad de imágenes, la variedad de objetos, la calidad de las etiquetas y la disponibilidad del dataset. Finalmente, se seleccionó un dataset que cumplía con los requisitos del proyecto y ofrecía un buen balance entre cantidad y calidad de datos.

El dataset seleccionado utilizaba un formato de etiquetas XML, mientras que YOLO requiere un formato TXT. Para solucionar esta incompatibilidad, se desarrolló un script personalizado que convertía las etiquetas XML a TXT. El script extraía las coordenadas de los cuadros delimitadores (bounding boxes) de los archivos XML y las transformaba a un formato compatible con YOLO, generando un archivo TXT por imagen con las etiquetas correspondientes.

Para evaluar el rendimiento del modelo, se dividió el dataset en dos conjuntos: entrenamiento (training) y prueba (test). La proporción de imágenes destinada a cada conjunto se determinó de acuerdo con las prácticas comunes en el campo del aprendizaje profundo, con un enfoque en asegurar una distribución representativa de datos en ambos conjuntos.

Para evaluar la generalización del modelo y evitar el sobreajuste (overfit-

ting), se implementó una validación cruzada de 5 pliegues en el conjunto de entrenamiento. El conjunto de entrenamiento se dividió en 5 partes iguales, utilizando cada parte como conjunto de validación en un turno diferente. Este proceso permitió obtener una estimación más precisa del rendimiento del modelo en datos no vistos durante el entrenamiento.

Se utilizó la arquitectura YOLOv8n para el entrenamiento del modelo de detección de objetos. Se fijó un tamaño de imagen de 480 píxeles para todas las imágenes del dataset. Se realizaron tres rondas de entrenamiento para cada división del 5-fold, con 50, 75 y 150 épocas respectivamente. El proceso de entrenamiento se basó en el "fine-tuning" de la arquitectura YOLOv8n, utilizando las imágenes y etiquetas convertidas como entrada para optimizar los parámetros del modelo.

## Resultados Finales

Como resultados tenemos 3 predictores independientes entre si que pueden complementarse para tomar decisiones sobre el problema original, todos ellos han sido testados contra conjunto de imágenes para rastrear estos, todos los conjuntos utilizados se trata de tener la mayoría de las imágenes desde la perspectiva de un paso de peatones, todos han dado una precisión superior al 90% lo cual dado que se utiliza un modelo el cual está diseñado para dispositivos de bajos recursos computacionales, además del escaso tiempo para poder profundizar en los detalles de cada predictor.

Después de la investigación y experimentación llevada a cabo sobre este punto en particular del problema de cruce de calles para personas con dificultades visuales concluimos que es probable mediante recursos a la mano de un ciudadano medio poder tener mayor autonomía mediante el uso de soluciones de visión computacional, hemos podido afirmar que aunque YOLOV8n tiene buenos resultados para generalizar con relativamente pequeños volúmenes de datos, puede existir complicaciones, en caso de tener una mala selección del dataset, así como la aplicación como K-fold para poder sortear las dificultades de la poca disponibilidad de datos no es necesario altos números, con  $k=3$  y  $k=5$  ha arrojado muy buenos resultados, dado los escasos recursos, puede ser útil cambiar la perspectiva hacia que predecir como fue en el caso de los cruces de peatones los cuales haciendo que detectara solo porciones de este arrojaba mayor probabilidad así como existía un descenso del nivel de Falsos Positivos los cuales es nuestro objetivo disminuir tanto como nos sea posible.

Para poder dar una aproximación a una solución del subproblema hemos tomado los tres modelos y creado una serie de reglas sencillas donde establece que se tiene que cumplir al mismo tiempo.

Se aconseja cruzar, extremando las precauciones en caso de :

- Existencia del cruce peatonal,luz verde y la no existencia de ningun auto sobre el cuadro delimitador del cruce de peatones.

Se aconseja repetir la captura en caso de (Si en varios intentos no se a logrado respuesta contundente, pedir ayuda) :

- Existencia de cruce peatonal, luz amarilla o roja o existencia de un vehiculo de grandes dimensiones en el cruce.

- No existencia del cruce de peatones o luces de peatones

Ejemplo de una imagen completamente procesada:



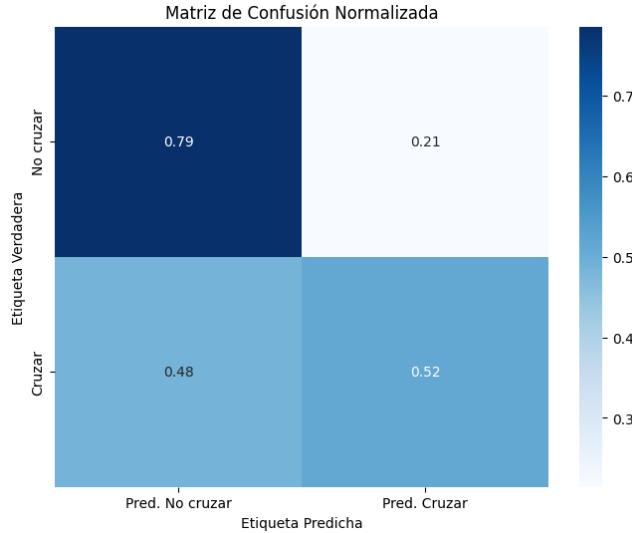


Figure 21: Matriz de confusión

## Discusión de los Resultados. Repercusión ética de las soluciones

Se ha logrado una buena precisión en la detección de cruces peatonales y luces de semáforos peatonales. Para determinar si cruzar o no cruzar, se obtuvo un por ciento aunque bajo, mejorable, de casos donde se indica cruzar cuando en realidad no se podía.

En el caso de las luces de semáforos peatonales, no se confunde un color con el otro, esto significa que no se dará el caso de que el semáforo haya sido detectado (cumpliendo que la confianza sea mayor o igual que el umbral propuesto) pero con un color equivocado (lo que sería el caso de que esté en rojo y la aplicación permita cruzar).

Otro caso problemático es cuando el modelo detecta un objeto en el fondo y lo confunde con una luz de semáforo. En los ejemplos donde ha sucedido, se ve un valor de confianza realmente bajo, por lo que no sería tomado en cuenta.

En los casos donde se detecten varios semáforos en una misma imagen, se toman en cuenta una medida de proximidad (esto evita guiarse por la luz de un semáforo que este lejano pero por la perspectiva de la foto se vea).

Hemos logrado el objetivo de que nuestros predictores funcionen como un todo, minimizando los casos de los Falsos Positivos, en nuestro experimento no

encontramos ninguno, ni en pruebas dirigidas donde entre el equipo conociendo las deficiencias de cada predictor tratamos de buscar escenarios complejos para ello y aun ha sido resistente.

Este sistema es muy primitivo aun y necesita muchos hiperparámetros a ajustar para poder ser una aplicación usable dado que aun mantenemos una tasa de **48%** respecto a la predicción de no cruzar cuando se podía.

El trabajo en este proyecto desarrollado puede ser un primer acercamiento a una aplicación más ajustada a las características del país y que puede ser útil en un futuro con refinamiento para las personas débil visuales. Siempre se trata de ante una duda sobre si cruzar o no, dejarlo indefinido, puesto que un falso positivo en algunos casos puede resultar fatal. El equipo de trabajo considera que con refinamiento, abarcando más casos podría ser una idea provechosa y valiosa a nivel social y de la comunidad.

## **Conclusiones y trabajo futuro**

Es evidente que el problema tratado tiene muchas vertientes para trabajos futuros. En el subproblema específico en el que se ha centrado, quedan hilos por donde seguir trabajando para lograr mejores resultados y situaciones más abarcadoras.

Se propone continuar con la investigación para mejorar la detección de los semáforos peatonales, diferenciándolos de los dedicados a vehículos. Además, abarcar más los casos de las fotos nocturnas e incluir ciertas condiciones climatológicas.

Mejorar la determinación de proximidad de objetos, que puede ayudar a dislumbrar el cruce peatonal y semáforo más cercano (que debe ser el que la persona desea analizar).

Profundizar la investigación acerca del tema, puesto que el desarrollo por imágenes resulta un poco engorroso, al menos por sí solo.

## **Bibliografía**

- Traffic light detection and intersection crossing using mobile computer vision
- Documentación de ultralytics
- How to Train YOLO v5 on a Custom Dataset
- Pedestrian Crossing Sensing Based on Hough Space Analysis to Support Visually Impaired Pedestrian
- Tools and Technologies for Blind and Visually Impaired Navigation Support: A Review