

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Sistema de Monitoreo Visual para el Control de Acceso Vehicular

PROYECTO INTEGRADOR

Previo la obtención del Título de:

**Nombre de la titulación
Ingeniero en Telemática**

Presentado por:

Maximiliano Divar González Monard

GUAYAQUIL - ECUADOR

Año: 2023

DEDICATORIA

Maximiliano González

Dedico este trabajo a mi familia que ha sido mi soporte y mi ayuda a lo largo de mi vida universitaria, quiero agradecerles a las personas que conocí a lo largo de toda mi vida universitaria y que contribuyeron para poder alcanzar este logro.

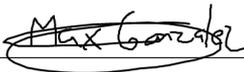
AGRADECIMIENTOS

Maximiliano González

Agradezco a mis abuelos y a mis padres por su gran esfuerzo. Agradezco a Pamela Valdivieso por ser un pilar fundamental en mi vida, y ser una pieza fundamental al iniciar esta carrera. Fanny Pacheco que siempre ha estado a mi lado durante toda mi vida universitaria. Agradezco a mi tutora y profesores por su valiosa orientación y conocimiento a lo largo de este camino.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Maximiliano Divar Gonzalez Monardy doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

A handwritten signature in black ink, appearing to read "Max Gonzalez", is written over a horizontal line. The signature is somewhat stylized and cursive.

Maximiliano González

EVALUADORES

Ignacio Marín

PROFESOR DE LA MATERIA

María José Ramírez

PROFESOR TUTOR

RESUMEN

Este proyecto aborda la creación de un sistema de monitoreo vehicular utilizando una Raspberry Pi y cámaras de alta definición, enfocado en mejorar la seguridad y gestión de accesos en parqueaderos urbanos. Se busca desarrollar una solución adaptable, eficiente y económica que permita un registro detallado del ingreso de vehículos, utilizando tecnologías accesibles y de código abierto.

En el desarrollo, se empleó el modelo YOLOV5 para la detección de placas, complementado por Roboflow para el procesamiento de imágenes, y se configuró una Raspberry Pi con RTSP(Real Time Streaming Protocol) para la transmisión en tiempo real. Las pruebas demostraron la eficacia del sistema en el reconocimiento de placas, utilizando la librería easyocr para la conversión de imágenes en texto y MongoDB para el almacenamiento de datos.

Los resultados indican una alta eficiencia en la detección y registro de placas vehiculares, demostrando la viabilidad de sistemas compactos y de bajo costo en aplicaciones de seguridad y monitoreo. La implementación de MongoDB facilitó la gestión eficiente de datos, siendo escalable y adaptable a las necesidades del proyecto.

En conclusión, la combinación de Yolo V5, Roboflow y easyocr en una Raspberry Pi representa un avance significativo en el control de accesos vehiculares. Este sistema no solo mejora la seguridad, sino que también abre posibilidades para futuras innovaciones en la gestión de tráfico y estacionamientos. Se recomienda explorar soluciones de almacenamiento y procesamiento de datos más eficientes para manejar grandes volúmenes de información, así como la actualización continua y mantenimiento del sistema. Líneas futuras incluyen la integración de AI (Artificial Intelligence) y aprendizaje profundo más avanzados para mejorar la precisión y adaptabilidad del sistema en diferentes condiciones ambientales.

Palabras Clave: Raspberry Pi, Monitoreo Vehicular, YOLOV5, Roboflow

ABSTRACT

This project addresses the development of a vehicle monitoring system using a Raspberry Pi and high-definition cameras, focusing on enhancing security and access management in urban parking spaces. The goal is to create an adaptable, efficient, and cost-effective solution that allows detailed recording of vehicle entries, utilizing accessible and open-source technologies.

During the development phase, the YOLOV5 model was used for license plate detection, supplemented by Roboflow for image processing. A Raspberry Pi was configured with RTSP for real-time transmission. Testing demonstrated the system's effectiveness in recognizing license plates, using the easyocr library to convert images to text and MongoDB for data storage.

The results indicate high efficiency in detecting and registering vehicle plates, proving the feasibility of compact, low-cost systems in security and monitoring applications. The implementation of MongoDB facilitated efficient data management, scalable and adaptable to project needs.

In conclusion, the combination of Yolo V5, Roboflow, and easyocr on a Raspberry Pi marks a significant advancement in vehicular access control. This system not only enhances security but also opens possibilities for future innovations in traffic and parking management. Recommendations include exploring more efficient data storage and processing solutions for handling large volumes of information, as well as continuous updating and maintenance of the system. Future lines of development involve integrating more advanced AI and deep learning to improve the system's accuracy and adaptability under various environmental conditions.

Keywords: Raspberry Pi, Vehicle Monitoring, YOLOV5, Roboflow

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	iii
ABREVIATURAS	vii
SIMBOLOGÍA	ix
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
1 INTRODUCCIÓN	1
1.1 Definición de la problemática	1
1.2 Justificación	2
1.3 Objetivos	2
1.4 Alcances y Limitaciones	3
1.5 Marco teórico	3
1.6 Estado del arte	5
2 METODOLOGÍA	9
2.1 Materiales	10
2.2 Metodología	13
3 RESULTADOS Y ANÁLISIS	15
3.1 Simulación	15
3.2 Pruebas	18
3.3 Análisis	21

4 CONCLUSIONES Y LÍNEAS FUTURAS	22
4.1 Conclusiones	22
4.2 Recomendaciones	24
4.3 Líneas Futuras	24
BIBLIOGRAFÍA	26
APÉNDICES	28

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
OCR	Optical Character Recognition
CNN	Convolutional Neural Network
JSON	Notacion de objetos JavaScript
GPIO	General-Purpose Input/Output
YOLO	You Only Look Once
CV	Computer Vision
ROI	Region of Interest
RTSP	Real Time Streaming Protocol

SIMBOLOGÍA

V	Voltaje
mm	Milimetro
m	Metro
MG	Mega bytes
GB	Gyga bytes

ÍNDICE DE FIGURAS

2.1	Arquitectura del sistema	10
2.2	Raspberry Pi 3	11
2.3	Tarjeta MicroSD de 16 GB	11
2.4	Cargador de Raspberry Pi 3	12
2.5	Cámara USB	13
3.1	Fotografía de vehículo en el parqueadero.	16
3.2	Reconocimiento de placa vehicular.	16
3.3	Placa vehicular convertida a string.	17
3.4	Número de placa almacenado con la fecha y hora.	18
3.5	Reproducción en página web	18
3.6	Modelo de reconocimiento de placa funcionando.	19
3.7	Roboflow en funcionamiento	19
3.8	Raspberry Funcionando y ejecutando el script	20
3.9	Numero de placa almacenado en MongoDB	20
3.10	Ejecución de script, modelo de reconocimiento de placa funcionando.	21
3.11	A la izquierda se muestra la predicción y a la derecha el auto analizado.	21

ÍNDICE DE TABLAS

1	Costo de implementación para el prototipo	30
---	---	----

CAPÍTULO 1

1. INTRODUCCIÓN

En el contexto actual de creciente urbanización y demanda de espacios de estacionamiento, la eficiente gestión y supervisión de ingresos vehiculares se vuelve una prioridad. La tecnología ofrece soluciones innovadoras para abordar este desafío, y una de las herramientas más versátiles en este sentido es la Raspberry Pi. Esta plataforma de bajo costo y alto rendimiento ha demostrado ser un catalizador para el desarrollo de sistemas de monitoreo y automatización. En este contexto, el presente trabajo se centra en el diseño y desarrollo de un sistema de monitoreo basado en Raspberry Pi, en conjunción con cámaras de alta definición, para supervisar el ingreso de vehículos.

Este proyecto busca una solución tecnológica eficaz, sostenible y adaptable, que podría ser implementada en diversos contextos urbanos y contribuir significativamente a la seguridad en ingresos vehiculares. En este sentido, esta tesis busca abordar de manera integral la problemática actual de control y supervisión de acceso vehicular.

1.1 Definición de la problemática

La problemática radica en mejorar la seguridad proporcionando una herramienta para vigilar el ingreso de vehículos en tiempo real. Esto puede disuadir a posibles intrusos o malintencionados y proporcionar evidencia visual en caso de incidentes. También se desea obtener un registro del ingreso de los vehículos, con su número de placa y hora en que el vehículo realiza el ingreso, lo que nos permitirá llevar un registro detallado y preciso.

1.2 Justificación

En la actualidad, la gestión eficiente y segura de accesos vehiculares es de suma importancia en diversos contextos, como centros comerciales, empresas y zonas residenciales. Sin embargo, muchas soluciones existentes son costosas, complejas o carecen de la flexibilidad necesaria para adaptarse a diferentes entornos. El uso de tecnologías de bajo costo y accesibles, como la Raspberry Pi y cámaras, ofrece una oportunidad prometedora para desarrollar un sistema de monitoreo visual a medida que responda a las necesidades.

Este proyecto se justifica en virtud de la necesidad de implementar un sistema de control de acceso vehicular eficiente, confiable y económico, que permita la identificación rápida y segura de vehículos, así como el registro de los mismos. La combinación de una cámara de video con una Raspberry Pi abre la posibilidad de desarrollar un sistema robusto y flexible, capaz de adaptarse a diferentes condiciones de iluminación y entornos. Además, la plataforma Raspberry Pi ofrece una amplia variedad de recursos de desarrollo, lo que facilita la implementación y escalabilidad del sistema.

Al abordar este tema, se busca contribuir al campo de la seguridad y gestión de accesos, proporcionando una solución tecnológica accesible y versátil que pueda ser implementada en una variedad de escenarios. Asimismo, se pretende fomentar la adopción de tecnologías de código abierto y de bajo costo en aplicaciones de seguridad y control de accesos.

1.3 Objetivos

El objetivo general de dicho proyecto es: Desarrollar un sistema de monitoreo utilizando Raspberry Pi y una cámara de vigilancia para supervisar y registrar de manera efectiva el ingreso de vehículos. Para desarrollar el objetivo general mencionado se han desarrollado tres objetivos específicos:

1. Seleccionar y configurar el software necesario para la gestión y procesamiento de las imágenes de la cámara.

2. Establecer una base de datos para almacenar los registros de ingreso de vehículo incluyendo la fecha y hora.
3. Configurar y desplegar una interfaz web en un servidor que presente el monitoreo en tiempo real

1.4 Alcances y Limitaciones

Es relevante considerar que en este análisis no se incluirá la creación de una aplicación para la visualización de los datos recopilados, se seleccionará y configurará el software necesario para la gestión y procesamiento de las imágenes capturadas por la cámara, además, el análisis de la duración del dispositivo no va a ser incluido en esta investigación, debido al tiempo prolongado que tomaría realizar dicha investigación. El proyecto se enfoca en desarrollar un sistema de monitoreo de placas de vehículos para ingresos vehiculares, no va a llevar un registro sobre los espacios disponibles del parqueadero, ya que está enfocado en controlar el ingreso de los vehículos. Las conclusiones que se obtengan van a depender de la fase de simulación y resultados, ya que nos va a decir como quedo la funcionalidad del proyecto.

En cuanto a las limitantes, la dirección ip del dispositivo que transmite a través de una cámara tiene que ser alcanzable, una dirección pública, solo se registrarán los datos de ingreso de vehículos mas no de salida, por lo tanto, no habrá un total control de vehículos con respecto a espacios disponibles. La cámara del Raspberry Pi 3 capturará la placa del vehículo y la hora de ingreso vehicular y se almacenarán para el análisis. Adicional, solo se utilizará un tipo de Raspberry Pi para la captura y monitoreo de datos.

1.5 Marco teórico

La Raspberry Pi, un miniordenador económico del tamaño de una tarjeta de crédito, se conecta a monitores o TVs y funciona con teclado y ratón estándar. Funciona con Linux, permitiendo a usuarios de todas las edades explorar la computación y aprender lenguajes como Scratch y Python. Realiza funciones de un ordenador de escritorio, como navegar

en internet, reproducir vídeos HD, manejar documentos de oficina y jugar. (Raspberry Pi, 2021)

La Raspberry Pi 3, el modelo elegido para este proyecto, se destaca por sus múltiples interfaces. Incluye puertos USB, un conector ethernet, entrada de audio de 3.5mm, puerto HDMI, ranura para tarjeta microSD y un puerto micro-USB para la alimentación. Además, está equipada con puertos especiales para conectar una cámara y una pantalla. Es fundamental mencionar que la Raspberry Pi 3B+ cuenta con un conjunto de 40 pines GPIO, que son esenciales para la interacción con el entorno externo, permitiendo conectar sensores y actuadores. Cabe destacar que estos pines GPIO operan a un nivel de voltaje de 3.3V. (Raspberry Pi, 2021)

Un algoritmo a utilizar es You Only Look Once (YOLO) que propone utilizar una red neuronal de extremo a extremo que realiza predicciones de cuadros delimitadores y probabilidades de clases al mismo tiempo. Esto difiere del enfoque adoptado por algoritmos previos de detección de objetos, que reutilizaban clasificadores para llevar a cabo la detección. El algoritmo YOLO toma una imagen como entrada y luego utiliza una red neuronal convolucional profunda simple para detectar objetos en la imagen. (Kundu, 2023)

Las primeras 20 capas de convolución del modelo están pre-entrenadas utilizando ImageNet al conectar una capa temporal de agrupación promedio y una capa totalmente conectada. Luego, este modelo pre-entrenado se convierte para llevar a cabo la detección, ya que investigaciones previas demostraron que agregar capas de convolución y conectadas a una red pre-entrenada mejora el rendimiento. La última capa totalmente conectada de YOLO predice tanto las probabilidades de clase como las coordenadas de los cuadros delimitadores.

YOLO transforma una imagen en una retícula de $S \times S$, donde cada celda detecta un objeto si su centro cae en ella. Este proyecto utiliza YOLO v5, que se basa en una versión más eficiente de EfficientNet, superando a la EfficientDet de YOLO v5 en eficiencia computacional. YOLO v5 también presenta un avance en la generación de cajas de anclaje, conocidas como cajas de anclaje densas.

Otro algoritmo de detección es Detectron, el cuál es el sistema de software de Facebook AI Research que implementa algoritmos de detección de objetos de vanguardia, incluido Mask R-CNN. Está escrito en Python y funciona con el marco de aprendizaje profundo Caffe2 y será potenciado por el marco de aprendizaje profundo PyTorch 1.0. (Meta Platforms, 2024)

TrOCR es un modelo OCR basado en Transformer de extremo a extremo para el reconocimiento de texto, con modelos CV y NLP pre-entrenados. A diferencia de los modelos de reconocimiento de texto existentes, TrOCR es un modelo simple pero efectivo que no utiliza una CNN como columna vertebral. Primero redimensiona la imagen de texto de entrada a 384×384 y luego divide la imagen en una secuencia de parches de 16×16 que se utilizan como entrada para los transformadores de imagen. (Li et al., 2023)

Para entrenar de manera efectiva el modelo TrOCR, el codificador puede inicializarse con modelos pre-entrenados al estilo ViT, mientras que el decodificador puede inicializarse con modelos pre-entrenados al estilo BERT respectivamente (Sierra et al., 2006) . Por lo tanto, TrOCR tiene tres ventajas. Primero, TrOCR utiliza modelos Transformer de imagen y texto pre-entrenados, que aprovechan los datos no etiquetados a gran escala para la comprensión de imágenes y el modelado del lenguaje, sin necesidad de un modelo de lenguaje externo.

Segundo, TrOCR no requiere ninguna red convolucional como columna vertebral y no introduce ningún sesgo inductivo específico de la imagen, lo que hace que el modelo sea muy fácil de implementar y mantener. Finalmente, los resultados experimentales en conjuntos de datos de referencia de OCR muestran que TrOCR puede lograr resultados de vanguardia en imágenes de texto impresas, manuscritas y de escena sin necesidad de pasos de pre/post-procesamiento complejos.

1.6 Estado del arte

De acuerdo con la investigación por en la ciudad de Cuenca (Monard, 2024), se trabajó un modelo usando redes neuronales convolucionales para detectar placas de vehículos en movimiento. Este proceso inició con el etiquetado manual de las imágenes de vehículos, señalando las placas y sus caracteres. Luego, se utilizó la estructura de red neuronal COCO inception V2 de Google como base para entrenar un modelo especializado en el

reconocimiento de placas.

La investigación realizada en Colombia (Suaza, 2023), se desarrolló una app web que ayude a facilitar la situación a cualquier conductor en la ciudad de Villavicencio. El sistema desarrollado fue capaz de ajustarse a cualquier estacionamiento y contó con la capacidad de los resultados basándose en los datos acumulados semanalmente, incluyendo predicciones de disponibilidad de espacios. Con la detección de placas vehiculares, el aplicativo web realiza un registro instantáneo de los vehículos, almacenando información relevante como la hora de ingreso, el tipo de vehículo y la ubicación del puesto asignado. Además, se llevará un control preciso de la capacidad del parqueadero y se generarán alertas en caso de exceder la capacidad máxima. Utilizó tecnologías como HTML, CSS y JavaScript para la interfaz de usuario, así como frameworks y bibliotecas especializadas en reconocimiento de imágenes procesamiento de datos.

Adicionalmente, otro estudio llevado a cabo de Tacna, Perú (Aguilar, 2022), En este estudio, se comenzó usando la librería OpenCV para desarrollar un prototipo de reconocimiento de placas vehiculares. El proceso involucró 30 detecciones de vehículos de carga pesada. Inicialmente, se aplicó una prueba a un grupo de procesos antes de administrar el tratamiento experimental, que es el prototipo mencionado. Luego del tratamiento, se realizó una prueba posterior para evaluar los resultados.

Otro estudio realizado en México (Morales et al., 2019), Para crear un sistema de reconocimiento automático de placas vehiculares, se capturaron imágenes con cámaras CCD, centrándose en las placas como región de interés (ROI). Se combinaron técnicas de Inteligencia Artificial como redes neuronales, máquinas de soporte vectorial y OCR. Los caracteres de las placas se detectaron con el detector de bordes de Canny y se utilizó el modelo K vecinos más cercanos (K-NN) para su reconocimiento y separación en la ROI.

Un artículo escrito (Yousif et al., 2020) , Se propuso una metodología innovadora para reconocer matrículas (LP), combinando procesamiento de imágenes y un conjunto neutrosófico (NS) optimizado por algoritmo genético (GA). Se emplearon técnicas como detección de bordes y morfología para localizar las LP, seguido de la optimización de

NS mediante GA para reducir la ambigüedad en las imágenes. Se utilizó el algoritmo k-means para segmentar los caracteres y el análisis de componentes conectados (CCLA) para identificar y extraer eficientemente cada carácter de las LP.

De acuerdo con un estudio de titulación realizado por dos estudiantes de ESPAMMFL (López, 2019), con el objetivo de mejorar la administración de los vehículos públicos en el GADM de Chone, se creó un sistema informático que se basa en el reconocimiento de placas para el control de acceso. El proceso se inició con la captura y el pre-procesamiento de las imágenes de las placas, seguido de la segmentación para identificar tanto la placa como sus caracteres. Estos elementos se transformaron en datos numéricos comprensibles para la computadora. La etapa final del proceso incluyó la utilización de técnicas de visión por computadora para reconocer los caracteres y dígitos de las placas, resultando en la creación del modelo de reconocimiento.

Un estudio realizado en la provincia de Santa Elena (Córdova, 2022), busca mejorar el proceso de acceso vehicular al estacionamiento del Edificio “El Velero Azul” mediante técnicas de visión artificial y reconocimiento óptico de caracteres en las placas de los vehículos. La autorización se determina comparando la matrícula con la base de datos. Si hay coincidencia, se abre la puerta, de lo contrario, se cierra. Este proyecto busca presentar un prototipo de control de acceso al servicio de estacionamiento, mejorando un estudio que evalúa las debilidades en este proceso .

Un artículo desarrollado en la Universidad de El Salvador (Morán, 2023), brinda una propuesta de sistema de reconocimiento de placas salvadoreñas para la Universidad de El Salvador. Para llegar a esta propuesta se buscaron candidatos basados en software libre en los repositorios de software más populares de internet. Los candidatos seleccionados fueron OpenALPR y la combinación de YOLOv5 con PaddleOCR, y se realizó su instalación en un ambiente de prueba. Luego se creó un conjunto de imágenes de placas salvadoreñas para entrenar y comparar los modelos. Se obtuvieron las inferencias de cada candidato para cada imagen del conjunto, así como los aciertos, el tiempo utilizado y la certeza. A partir de esta información se compararon los candidatos entre sí. En cada elemento de comparación, el candidato con mejor desempeño fue la

combinación de YOLOv5 con PaddleOCR, convirtiéndose en la propuesta de sistema de reconocimiento de placas salvadoreñas.

En un trabajo de titulación desarrollado en la Escuela Politécnica Nacional (León, 2022) , se desarrolló un sistema utilizando el framework Scrum para reconocer placas vehiculares ecuatorianas. Las pruebas demostraron una precisión del 85% en la localización de la placa y una lectura de caracteres al 100% de efectividad. El desarrollo se dividió en tres sprints, con evaluaciones al final de cada uno, permitiendo la detección y corrección temprana de errores para una mejora continua .

CAPÍTULO 2

2. METODOLOGÍA

Para el desarrollo del presente proyecto se seleccionó el modelo YOLOV5 previamente configurado para tareas de segmentación y clasificación de imágenes. Para su elección, se tomaron en cuenta diversos factores, entre ellos el lenguaje de programación utilizado, este modelo trabaja con el lenguaje Python, además de, su velocidad de procesamiento, fácil uso e implementación y compatibilidad con diferentes hardwares entre ellos el Raspberry PI. La arquitectura del proyecto se muestra en la Figura 2.1. Luego, se recopilamos imágenes de diferentes vehículos con sus placas en una base de datos utilizando una cámara de video.

Se utilizó la plataforma de inteligencia artificial ROBOFLOW para el entrenamiento del modelo. Para ello, se insertaron las imágenes previamente recopiladas y se seleccionaron de manera manual los recuadros de las placas de los carros, con la finalidad de enseñar al modelo la detección de letras y números en estas.

Se configuró la raspberry PI con el protocolo RTSP (Real Time Streaming Protocol) para transmitir en tiempo real el video capturado por la cámara. Además, se integró el servidor Flask para obtener la transmisión del flujo RTSP y mostrarla en la dirección del servidor.

Posterior a esto, se elaboró el código en Python, mismo que importa el modelo previamente entrenado y la transmisión del servidor Flask. Luego, detecta la región reconocida como placa y utiliza la librería easyocr en la fase de OCR para extraer el número de placa y almacenarlo en una base de datos junto con la fecha y hora de ingreso.

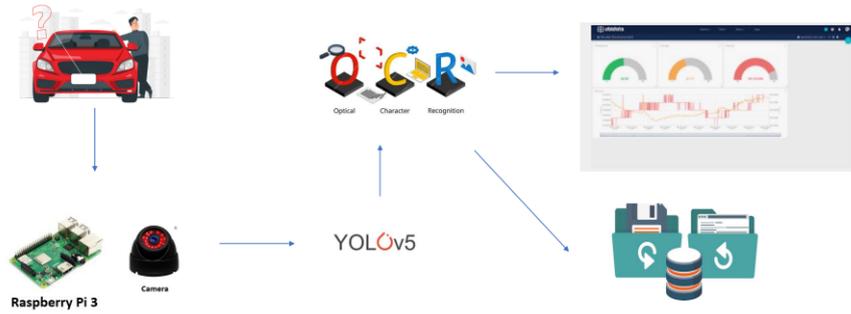


Figura 2.1: Arquitectura del sistema

2.1 Materiales

En la creación de este proyecto se han considerado tanto los recursos físicos como digitales, teniendo en consideración su disponibilidad, facilidad de implementación, además de, la información recopilada en el marco teórico y el estado del arte.

Raspberry Pi 3: Es un controlador de bajo costo desarrollado por Raspberry Pi Foundation, incorpora una CPU Cortex-A53 de cuatro núcleos, 64bit y 1.2Ghz, una unidad de procesamiento (GPU), memoria integrada, y alimentación de 5V. Además, un puerto para cámara y puertos de entrada y salida compatibles con diferentes componentes electrónicos como se muestra en la Figura 2.2. (Raspberry Pi, 2023) La Raspberry Pi 3 es compatible con accesorios de modelos anteriores gracias al mismo conector GPIO. Además, ofrece Wifi 802.11n y Bluetooth 4.1, eliminando la necesidad de Ethernet y permitiendo conexiones inalámbricas con dispositivos y periféricos. Posee 1 Puerto Ethernet 10/100, 1 conector de vídeo/audio HDMI, 1 Conector 3.5mm audio/video compuesto, 4 Puertos USB 2.0, 40 Pines GPIO, conector de pantalla DSI, y ranura de tarjeta microSD. (Raspberry Pi, 2021)



Figura 2.2: Raspberry Pi 3

Tarjeta microSD: Tarjeta microSD con al menos 16 GB de almacenamiento para poder almacenar el sistema operativo de la Raspberry Pi y los programas necesarios para la captura de imágenes de las placas vehiculares como se muestra en la Figura 2.3. Además, las tarjetas que pueden alcanzar velocidades UHS-I Clase 10 de 95 MB/s permiten un almacenamiento sencillo y transferencias rápidas (Raspberry Pi, 2023).



Figura 2.3: Tarjeta MicroSD de 16 GB

Hugging Face TrOCR: Hugging Face es una empresa de tecnología dedicada al desarrollo de herramientas y plataformas de procesamiento de lenguaje natural o NLP basadas en inteligencia artificial (Hugging, 2023). Dentro de Hugging Face se encuentra TrOCR, el cual fue propuesto en "TrOCR: Reconocimiento óptico de caracteres basado en transformadores con modelos preentrenados" por Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei. TrOCR consta de un codificador de transformador de imagen y un decodificador de texto autorregresivo basado en transformadores para realizar el reconocimiento óptico de caracteres (OCR)

(Hugging, 2023).

Cargador Raspberry Pi 3: Cargador USB-C que proporciona un voltaje de salida de 5 voltios (V) que es el requerido por la Raspberry Pi 3 como se muestra en la Figura 2.4. Adicionalmente, la corriente de salida debe ser de 2.5 amperios (A) para satisfacer sus demandas de energía para cargar (Raspberryporg, 2023)



Figura 2.4: Cargador de Raspberry Pi 3

Cámara Web USB: Cámara con domo de aluminio que se puede instalar directamente en interiores y exteriores de hogares, oficinas, autobuses, taxis, escuelas, hospitales, etc. Es resistente al agua; posee una resolución HD de megapíxeles de hasta 1080X720P. Adicional, posee visión nocturna gracias a luces infrarrojas y es compatible con Android, Linux, Windows, Mac y Raspberry Pi como se muestra en la Figura 2.5.



Figura 2.5: Cámara USB

Bibliotecas: Se utilizó el sistema operativo compatible con Raspberry Pi, ya sea Raspbian o cualquier otro compatible con Linux. Se usarán ciertas bibliotecas como: flask, opencv, roboflow, easyocr y time, además de un software específico para el procesamiento de imágenes como Yolo.

2.2 Metodología

Primeramente, se llevó a cabo un análisis comparativo entre los frameworks de detección de objetos en imágenes Yolo y Detectron. Al realizar la comparación entre ambos, se concluyó que Yolo era el más adecuado para la detección de las placas de vehículos. Esto se debe a su velocidad en tiempo real; es un framework eficaz y simple que utiliza una sola red neuronal convolucional para predecir las cajas delimitadoras y las clases de los objetos en una sola pasada a través de la imagen. Adicionalmente, posee gran facilidad de implementación y uso debido a que posee una arquitectura pre-entrenada y simple.

Posteriormente, en el sitio web Hugging Face se pudo obtener el código necesario de TrOCR, el cual permite utilizar un conjunto de datos etiquetados que contenga imágenes de placas vehiculares y las correspondientes transcripciones de texto, además se trabajó con una librería de python que es easyocr, que nos permite realizar la misma función que TrOCR sin necesidad de entrenarse. Los pasos que normalmente sigue TrOCR en su fase de entrenamiento es la siguiente:

1. **Recopilación de datos:** Se recopilan imágenes de placas vehiculares junto con las transcripciones del texto presente en cada imagen.

2. **Preprocesamiento de imágenes:** Las imágenes pueden requerir preprocesamiento para mejorar la calidad, ajustes de contraste, reducción de ruido, normalización de tamaño, etc.
3. **División de datos:** El conjunto de datos se divide en dos partes: un conjunto de entrenamiento y un conjunto de prueba o validación. El conjunto de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se utiliza para evaluar su rendimiento.
4. **Extracción de características:** Se extraen características relevantes de las imágenes, como bordes, esquinas, texturas, etc. En el caso de TrOCR, también se pueden usar métodos específicos para el reconocimiento de texto.
5. **Diseño del modelo:** Se elige una arquitectura de red neuronal (por ejemplo, una red convolucional, o CNN) adecuada para el problema.
6. **Entrenamiento del modelo:** Durante el entrenamiento, el modelo aprende a mapear las características de las imágenes a las transcripciones de texto.
7. **Evaluación del modelo:** Se utiliza el conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos. Esto ayuda a determinar si el modelo generaliza bien a nuevas imágenes.
8. **Ajustes:** Si el rendimiento no es satisfactorio, se pueden realizar ajustes en la arquitectura del modelo o en los hiperparámetros y se vuelve a entrenar.
9. **Despliegue y prueba en tiempo real:** Una vez que el modelo tiene un rendimiento aceptable, se puede implementar en un entorno de producción y probar en situaciones reales.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En este capítulo se llevaron a cabo simulaciones y pruebas para el sistema de reconocimiento de placas vehiculares usando Jupyter Lab y YOLO. Se enfatizó la capacidad de YOLO para detectar placas en tiempo real y su integración efectiva con una Raspberry Pi. Utilizando MongoDB para la gestión de datos y Flask con RTSP para la transmisión en tiempo real, las pruebas demostraron la efectividad del sistema en el reconocimiento y registro de placas. Este enfoque resultó ser una solución eficiente y adaptable para aplicaciones de monitoreo y proyectos de IoT.

3.1 Simulación

En el contexto de este proyecto, se procedió a realizar las simulaciones usando Jupyter Lab como se muestra en la Figura 3.1 y Figura 3.2, que es un entorno de desarrollo interactivo de código abierto diseñado para la comunicación interactiva, la ciencia de datos, la visualización de datos y la programación en varios lenguajes. Su interfaz de usuario permite a los usuarios crear documentos que pueden contener código, texto enriquecido, ecuaciones, visualizaciones y demás.

Debido a que, se escogió Yolo en lugar de Detectron, se simuló el código respectivo en Jupyter Lab. Entre sus ventajas más destacadas están su entorno interactivo y flexible que permite ejecutar código en celdas individuales y ver su resultado de inmediato; proporciona soporte multiplataforma; admite múltiples lenguajes de programación; se puede visualizar datos de forma interactiva usando herramientas como Matplotlib, Plotly, Bokeh y otras bibliotecas de visualización; la arquitectura de JupyterLab permite la creación de extensiones y personalizaciones, lo que brinda la oportunidad de adaptar el entorno a las necesidades específicas del usuario y muchas otras.

Se procedió a tomar fotografías de varios automóviles para poder simular el código. Una vez subida la imagen, el sistema pudo reconocer la placa vehicular y posteriormente transformarla a string como se muestra en la Figura 3.3.

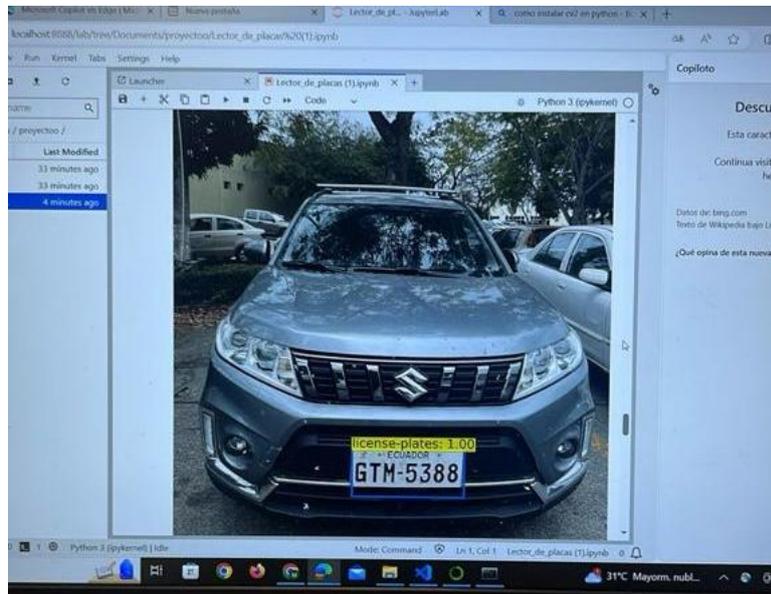


Figura 3.1: Fotografía de vehículo en el parqueadero.

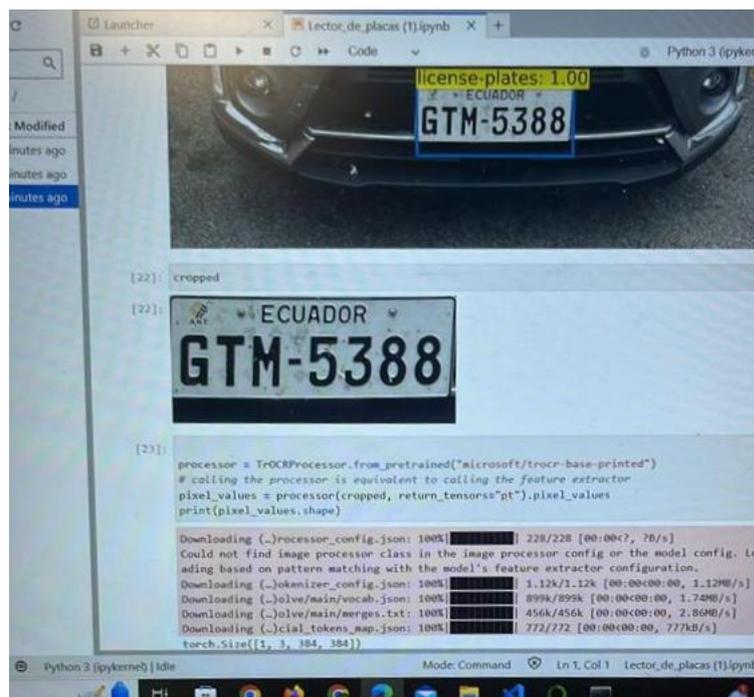


Figura 3.2: Reconocimiento de placa vehicular.

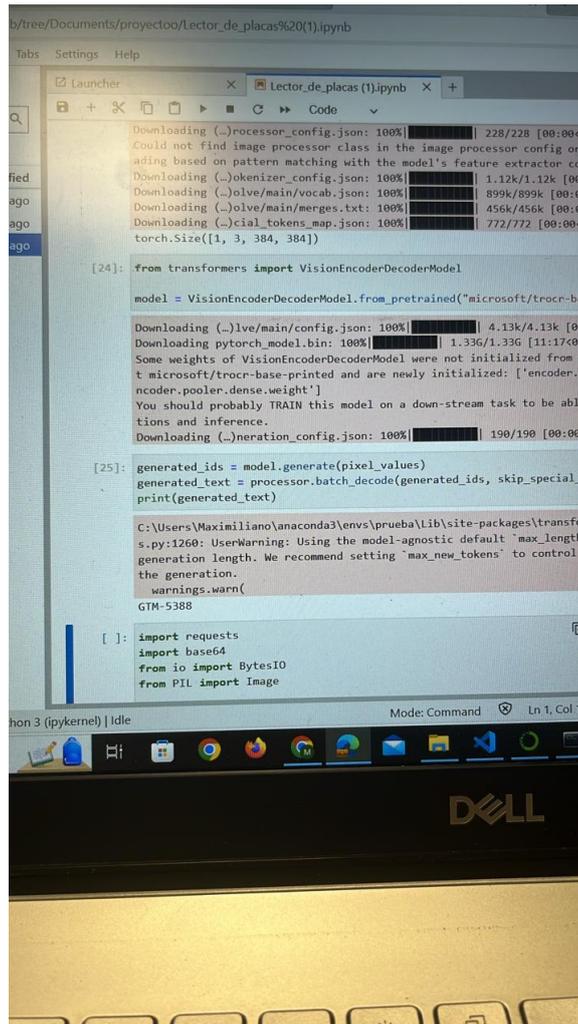


Figura 3.3: Placa vehicular convertida a string.

Con respecto al segundo objetivo del proyecto, que es establecer una base de datos para registrar los ingresos de los vehículos con fecha, hora y la placa del vehículo, se escogió la herramienta MongoDB como se muestra en la Figura 3.4. Es un servicio de base de datos NoSQL, este servicio proporciona una base de datos flexible y escalable para el desarrollo de aplicaciones web y móviles.

Entre sus ventajas mas destacables están que es una base de datos en tiempo real, lo que significa que sincroniza los datos entre los clientes de manera instantánea; es escalable horizontalmente, esto quiere decir que puede manejar grandes volúmenes de datos y una alta concurrencia de manera eficiente sin requerir la gestión de infraestructura por parte del desarrollador; es compatible con diversas plataformas, incluyendo aplicaciones web, Android, iOS y más; ofrece reglas de seguridad flexibles y configurables a nivel de base de datos, lo que permite controlar quién tiene acceso y qué operaciones pueden realizar.



Figura 3.4: Número de placa almacenado con la fecha y hora.

La integración de Flask con RTSP demuestra ser una solución robusta y eficiente para aplicaciones de monitoreo en tiempo real. La Figura 3.5 muestra la ventana del navegador donde se reproduce el video en tiempo real, capturado y transmitido a través del protocolo RTSP.

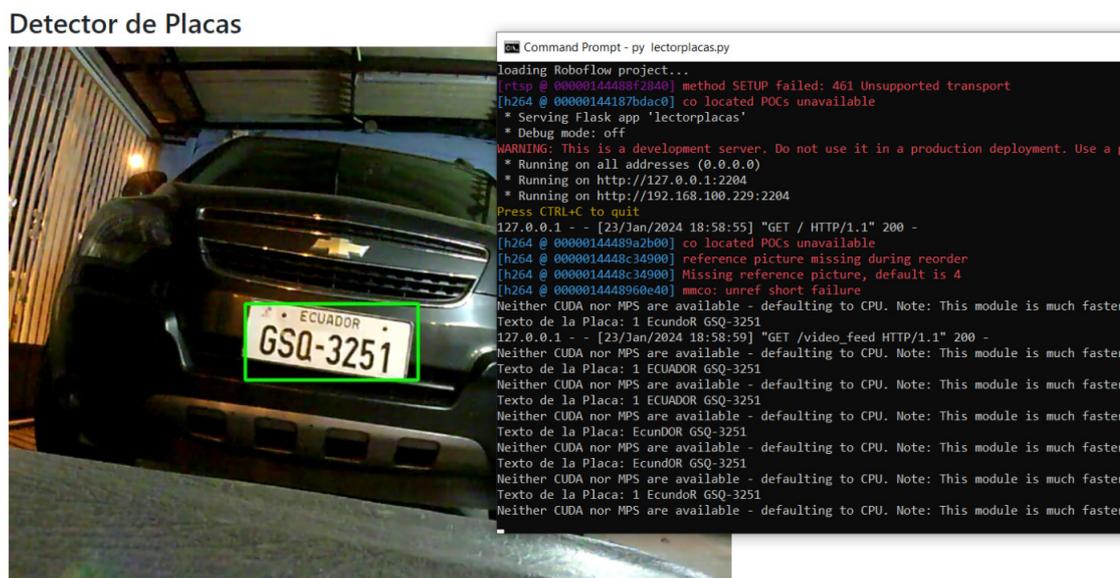


Figura 3.5: Reproducción en página web

3.2 Pruebas

Se realizaron varias pruebas, las primeras se realizaron con el Notebook de google como se muestra en la Figura 3.6, el cual nos permite correr nuestro código y probar el funcionamiento del modelo

Resultado de la ejecución del modelo con otra imagen como se muestra en la Figura

3.8

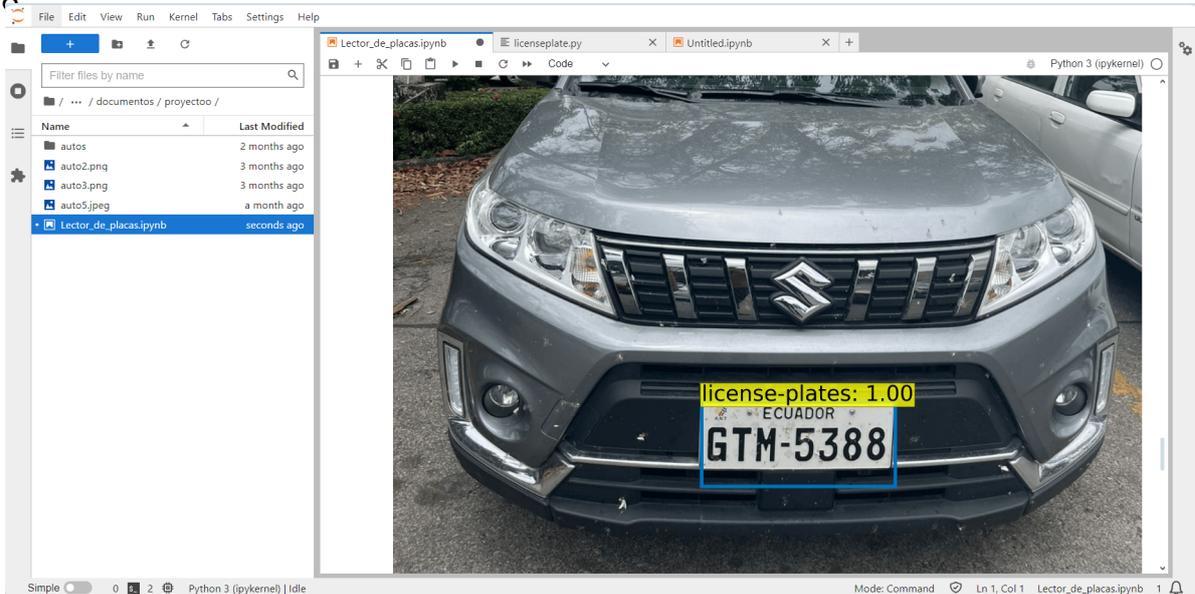


Figura 3.6: Modelo de reconocimiento de placa funcionando.

Las pruebas con Yolo V5 en una Raspberry Pi, complementadas por Roboflow como se muestra en la Figura 3.7, han resultado ser extraordinariamente prometedoras. Yolo V5, conocido por su eficacia en la detección de objetos en tiempo real, se adapta perfectamente a la Raspberry Pi, aprovechando su tamaño compacto y capacidad de procesamiento. Integrar Roboflow, una plataforma que asiste en la creación, gestión y despliegue de modelos de visión por computadora, ha permitido a los usuarios mejorar significativamente tanto la precisión como el rendimiento de sus modelos.

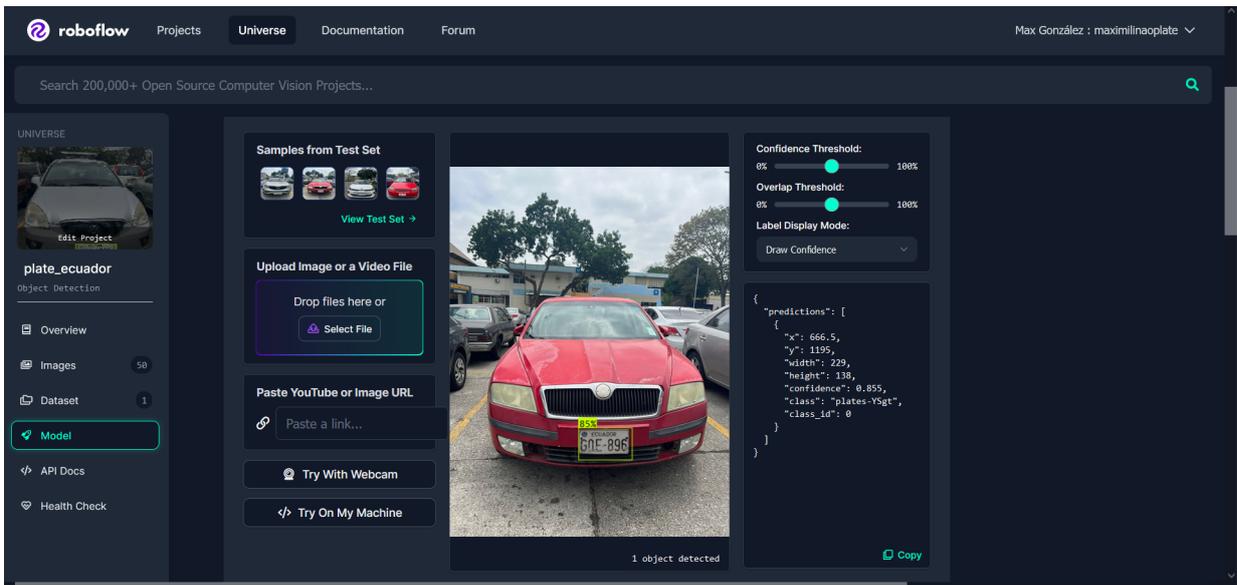


Figura 3.7: Roboflow en funcionamiento

Esta sinergia tecnológica facilita a investigadores y entusiastas la implementación de soluciones de visión por computadora eficientes y económicas. Gracias a Roboflow, la preparación y optimización de los conjuntos de datos para Yolo V5 se simplifica, resultando en una detección de objetos más rápida y precisa como se muestra en la Figura 3.8. Además, la Raspberry Pi, por ser un dispositivo portátil y de bajo consumo, es ideal para proyectos de IoT y aplicaciones móviles.

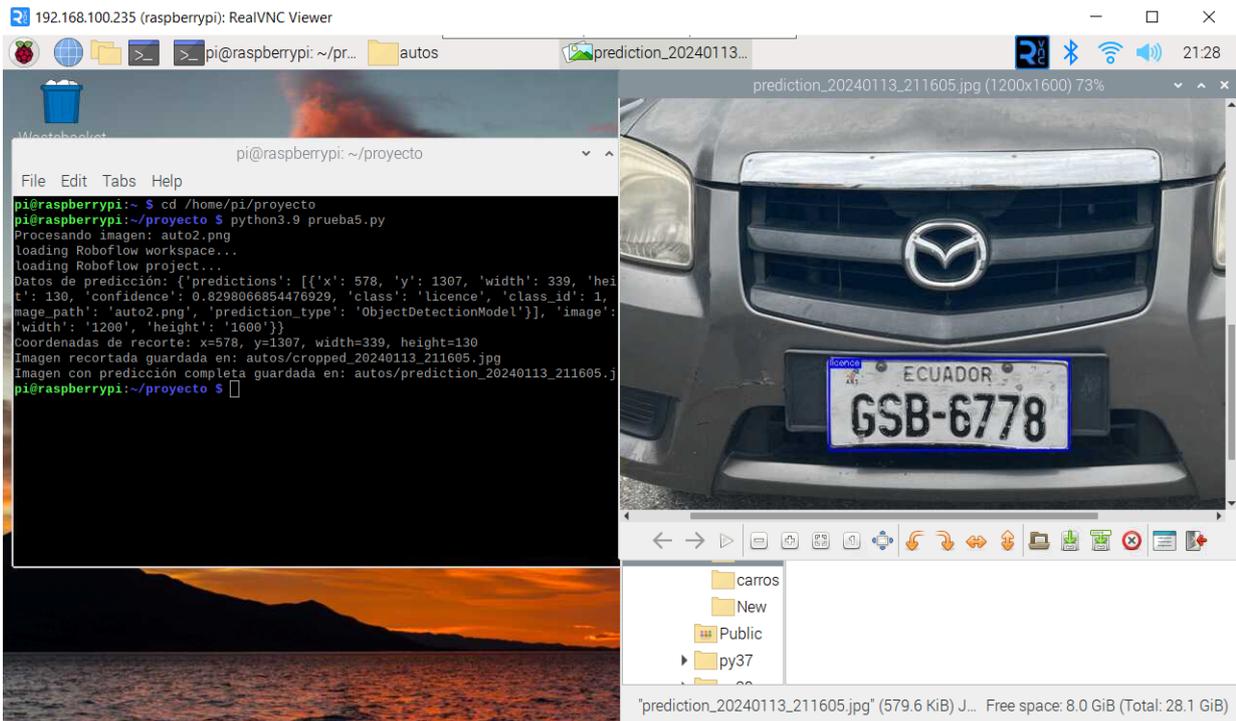


Figura 3.8: Raspberry Funcionando y ejecutando el script

Pruebas realizadas con la base de datos mongoDB, mostrando el registro de los números de placas de los vehículos, con la fecha y hora de ingreso, como se muestra en la Figura 3.9.

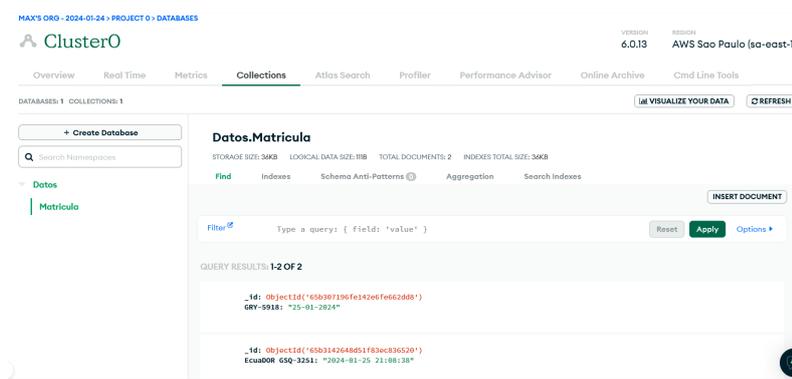


Figura 3.9: Numero de placa almacenado en MongoDB

3.3 Análisis

La cantidad de imágenes es un factor crucial en el entrenamiento y validación de modelos de visión por computadora. Un número mayor de imágenes suele permitir un aprendizaje más profundo y una generalización más efectiva.

YOLOV5 es conocido por su velocidad de detección en tiempo real, lo que lo hace adecuado para aplicaciones que requieren respuestas rápidas. También es relativamente más fácil de implementar y optimizar en dispositivos con recursos limitados, como una Raspberry Pi

El tamaño de las imágenes puede afectar significativamente el rendimiento y la precisión de los modelos de visión por computadora. Imágenes más grandes pueden proporcionar más detalles, pero también requieren más recursos de procesamiento

Se realizaron pruebas con diferentes imágenes para corroborar el correcto funcionamiento del modelo, como se muestra en la Figura 3.10 y la Figura 3.11.

```
pi@raspberrypi:~ $ cd /home/pi/proyecto
pi@raspberrypi:~/proyecto $ python3.9 prueba5.py
Procesando imagen: auto2.png
loading Roboflow workspace...
loading Roboflow project...
Datos de predicción: {'predictions': [{'x': 578, 'y': 1307, 'width': 339, 'height': 130, 'confidence': 0.8298066854476929, 'class': 'licence', 'class_id': 1, 'image_path': 'auto2.png', 'prediction_type': 'ObjectDetectionModel'}], 'image': {'width': '1200', 'height': '1600'}}
Coordenadas de recorte: x=578, y=1307, width=339, height=130
Imagen recortada guardada en: autos/cropped_20240113_211605.jpg
Imagen con predicción completa guardada en: autos/prediction_20240113_211605.jpg
pi@raspberrypi:~/proyecto $ python3.9 prueba5.py
Procesando imagen: auto3.png
loading Roboflow workspace...
loading Roboflow project...
Datos de predicción: {'predictions': [{'x': 578, 'y': 1307, 'width': 339, 'height': 130, 'confidence': 0.8298066854476929, 'class': 'licence', 'class_id': 1, 'image_path': 'auto3.png', 'prediction_type': 'ObjectDetectionModel'}], 'image': {'width': '1200', 'height': '1600'}}
Coordenadas de recorte: x=578, y=1307, width=339, height=130
Imagen recortada guardada en: autos/cropped_20240113_220750.jpg
Imagen con predicción completa guardada en: autos/prediction_20240113_220750.jpg
```

Figura 3.10: Ejecución de script, modelo de reconocimiento de placa funcionando.

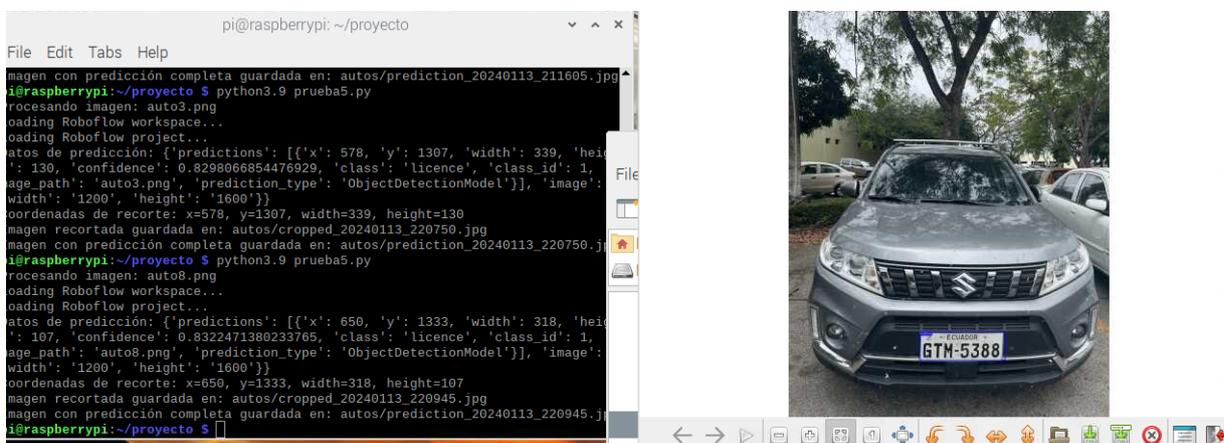


Figura 3.11: A la izquierda se muestra la predicción y a la derecha el auto analizado.

CAPÍTULO 4

4. CONCLUSIONES Y LÍNEAS FUTURAS

Durante el transcurso del proyecto, hemos desarrollado un sistema de control de ingreso vehicular basado en el reconocimiento de placas, utilizando una Raspberry Pi y una cámara. Este sistema combina la visión por computadora y el procesamiento de imágenes para identificar de manera eficiente y precisa las placas de los vehículos, facilitando un control de acceso automatizado y seguro. La implementación y las pruebas del sistema han demostrado su capacidad para operar de forma efectiva.

4.1 Conclusiones

1. La integración de Yolo V5 y Roboflow en una Raspberry Pi ha sido clave para gestionar y procesar las imágenes de la cámara, cumpliendo uno de nuestros objetivos. Esta combinación ha permitido, de manera económica y eficiente, aplicaciones avanzadas de visión por computadora, como el reconocimiento preciso de placas vehiculares con easyocr, ampliando el espectro de posibilidades desde la seguridad hasta soluciones pioneras en IoT. Este enfoque demuestra que es posible lograr un alto rendimiento en el procesamiento de imágenes en sistemas compactos y accesibles
2. La integración de easyocr en nuestro sistema para el control de ingreso ha sido fundamental en la automatización del registro de vehículos. Gracias a su eficaz reconocimiento óptico de caracteres, easyocr convierte las imágenes de las placas en texto, facilitando una identificación rápida y precisa de cada vehículo que ingresa. Las simulaciones y pruebas realizadas han evidenciado la robustez y fiabilidad de easyocr, demostrando su capacidad para procesar y analizar con éxito una

amplia gama de imágenes en diferentes condiciones, lo que asegura un rendimiento consistente y confiable en el reconocimiento y registro automático de vehículos

3. La implementación de un sistema de control de ingreso vehicular utilizando una Raspberry Pi y cámaras ha demostrado ser una solución eficiente y escalable. Este enfoque ha permitido no solo un monitoreo constante y en tiempo real, sino también una gestión más inteligente y automatizada. Además, a través de simulaciones realizadas con una variedad de imágenes, hemos corroborado la efectividad y fiabilidad del sistema
4. Este proyecto ha demostrado el potencial de la visión por computadora y el aprendizaje automático en el contexto del control de accesos vehiculares. La integración exitosa de estos elementos tecnológicos en un sistema funcional abre el camino para futuras innovaciones en el campo del control de tráfico y la gestión de infraestructuras de estacionamiento, marcando un hito en la aplicación práctica de estas tecnologías en entornos urbanos
5. La implementación de MongoDB ha sido esencial para crear una base de datos eficiente que registre los ingresos de vehículos, incluyendo detalles cruciales como fecha y hora. Esta solución, altamente escalable y adaptable, no solo maneja la información vehicular de manera precisa, sino que también garantiza un acceso rápido y seguro a los datos, vital para la toma de decisiones y el seguimiento en tiempo real. Así, MongoDB cumple con las demandas de nuestro proyecto de control de ingreso vehicular, asegurando una gestión de datos eficaz y adaptada a nuestras necesidades operativas.
6. Pruebas realizadas mostraron que la implementación del sistema es capaz de obtener los resultados esperados en el reconocimiento de placas de vehículos. Esta capacidad de reconocimiento confiable es esencial para la automatización efectiva del control de acceso vehicular y representa un avance importante hacia la mejora de los sistemas de seguridad y gestión de tráfico en infraestructuras urbanas.

4.2 Recomendaciones

1. Para manejar de manera efectiva un alto volumen de datos de reconocimiento de placas, se sugiere implementar soluciones de almacenamiento y procesamiento escalables. Esto podría incluir el uso de bases de datos más eficientes o la implementación de sistemas en la nube para el almacenamiento y análisis de datos
2. Para garantizar la continuidad del servicio, es importante considerar sistemas de redundancia y planes de recuperación ante fallos. Esto podría incluir la implementación de cámaras de respaldo, sistemas de alimentación ininterrumpida y procedimientos para la restauración rápida del sistema en caso de interrupciones
3. Establecer un plan de actualizaciones regulares y mantenimiento preventivo es esencial para mantener la efectividad del sistema a largo plazo. Esto debería abarcar tanto el hardware (Raspberry Pi, cámaras) como el software (algoritmos de Yolo V5, Pytesseract), asegurando así que el sistema permanezca actualizado con las últimas mejoras en tecnología y seguridad
4. Es esencial dedicar esfuerzos en la revisión y optimización del código subyacente del sistema de control de ingreso vehicular. Esto implica depurar y refinar los scripts de Python utilizados para la integración de Yolo V5 y easyocr, asegurando que operen de manera más eficiente en la Raspberry Pi

4.3 Líneas Futuras

Una de las líneas futuras más importantes involucra la integración de inteligencia artificial y aprendizaje profundo más avanzados para mejorar la precisión y la velocidad del reconocimiento de placas. Esto podría incluir el uso de redes neuronales más sofisticadas que puedan aprender de un conjunto de datos en constante crecimiento, adaptándose a nuevas condiciones y tipos de placas.

Otra área de desarrollo futuro es la implementación de un sistema más robusto y adaptable para diferentes condiciones ambientales, como cambios en la iluminación o en las condiciones meteorológicas. Esto garantizará que el sistema mantenga un alto nivel de precisión sin importar las circunstancias externas

Las posibilidades de crecimiento y mejora en este proyecto son amplias y variadas, ofreciendo un potencial significativo para revolucionar la forma en que se gestionan y monitorean los accesos vehiculares en entornos urbanos y privados

BIBLIOGRAFÍA

- Aguilar, J. P. (2022). Prototipo de reconocimiento de placas vehiculares para detección de vehículos alertados en el complejo de control aduanero de tomasiri, tacna-2022.
- Córdova, A. G. (2022). *Propuesta de mejora al acceso de vehículos autorizados y no autorizados mediante el reconocimiento de placas, tratamiento de imagen y automatización al edificio el velero azul* [B.S. thesis]. La Libertad: Universidad Estatal Península de Santa Elena. 2022.
- Hugging. (2023). *Hugging face documentation on trocr*. Retrieved November 12, 2023, from https://huggingface.co/docs/transformers/model_doc/trocr
- Kundu, R. (2023). YOLO: Algorithm for Object Detection Explained [+Examples] [Último acceso: [fecha en que accediste al sitio]].
- León, J. L. (2022). *Desarrollo de una aplicación para detección de placas vehiculares del ecuador*. [B.S. thesis]. Quito: EPN, 2022.
- Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). Trocr: Transformer-based optical character recognition with pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11), 13094–13102.
- López, F. (2019). *Sistema informático de control de ingreso y salida de vehículos mediante detección y reconocimiento de placas* [B.S. thesis]. Calceta: ESPAM MFL.
- Meta Platforms. (2024). *Meta ai tools - detectron* [Último acceso: enero de 2024]. <https://ai.meta.com/tools/detectron/>
- Monard, M. D. G. (2024). DISEÑO DE UN ALGORITMO DE RECONOCIMIENTO DE PLACAS VEHICULARES ECUATORIANAS USANDO REDES NEURONALES CONVOLUCIONALES. <https://doi.org/10.5281/zenodo.4147629>
- Morales, S. J. M., Hilario, V. Á., Peralta, E. R., & Carmona, E. S. (2019). Desarrollo de un sistema de reconocimiento de placas vehiculares usando operaciones morfológicas y ocr. *Revista Innova Ingeniería*, 1(4), 5–5.
- Morán, J. C. P. (2023). Propuesta de un sistema de reconocimiento de placas salvadoreñas para la universidad de el salvador. *Revista Multidisciplinaria de Investigación-REMI*, 2(1).
- Raspberry Pi. (2021). Raspberry pi 4 model b [Último acceso: 15 de noviembre de 2023]. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b>
- Raspberry Pi. (2023). Raspberry pi documentation [Accessed: 12-11-2023].
- Raspberrypio. (2023). *Rs components - raspberry pi* [Acceso en línea: 19 de julio de 2023]. Retrieved July 19, 2023, from <https://www.rs-components.com/raspberrypi>

- Sierra, E., Gómez, G., Suárez, A., & Pérez, E. (2006). Sistema para el entrenamiento paralelo de redes neuronales de propagación hacia atrás. *Ingeniería Industrial*, 27(1), 5.
- Suaza, R. (2023). Desarrollo de aplicativo web con reconocimiento de placas vehiculares para parqueaderos.
- Yousif, B. B., Ata, M. M., Fawzy, N., & Obaya, M. (2020). Toward an optimized neutrosophic k-means with genetic algorithm for automatic vehicle license plate recognition (onkm-avlpr). *IEEE Access*, 8, 49285–49312.

APÉNDICES

Campo	Descripción	Precio Unitario (USD)	Cantidad	Costo Total (USD)
Hardware	Raspberry Pi	\$100.00	1	\$100.00
	Cámara	\$70.00	1	\$70.00
	Cable	\$10.00	1	\$10.00
Software	Implementación	\$1000.00	1	\$1000.00
Mano de Obra	Ajustes	\$350.00	2	\$350.00
Costo				\$1530.00

Tabla 1: Costo de implementación para el prototipo