

Desarrollo de un sistema-prototipo para reconocimiento y detección de placas vehiculares usando visión e inteligencia artificial para el control de ingreso al parqueadero de la Gobernación de Sucre.

Design of a system-prototype for license plate recognition and detection using vision and artificial intelligence to control the entrance of cars to the parking lot at the Governor's Office of Sucre, Colombia.



Orlando Díaz Conde

Ingeniería Electrónica

Departamento de Ingeniería Electrónica

Facultad de Ingenierías y Arquitectura

Junio 2022

Resumen

Tradicionalmente los controles de entrada a parqueaderos son realizados por personal de vigilancia, lo que puede generar problemas de seguridad por errores o negligencia humana. Por esto se busca diseñar, construir y validar un sistema-prototipo que identifique placas vehiculares mediante el uso de visión e inteligencia artificial. La visión artificial permite extraer información de imágenes para tomar acciones basadas en esos datos. Para el hardware el sistema utiliza el computador de una sola placa (SBC) Raspberry Pi 3b+ en conjunto con una Pi Camera de conexión serial. En la parte de software, el procesamiento de imágenes usado para el reconocimiento está dividido en tres etapas: la detección de placas, el preprocesamiento y el reconocimiento de caracteres. Los resultados del sistema describen un porcentaje de detección de placas del 100% con un reconocimiento del 58,33% de caracteres.

Palabras clave: Visión Artificial, Reconocimiento de caracteres, Detección de placas

Abstract

Traditionally, parking lot entry controls are performed by security personnel, which can generate security problems due to human error or negligence. For this reason, we seek to design, build and validate a prototype system that identifies license plates through the use of vision and artificial intelligence. Artificial vision allows extracting information from images to take actions based on that data. Hardware wise, the system uses the Raspberry Pi 3b+ single board computer (SBC) in conjunction with a parallel-connected Pi Camera. On the software side, the image processing used for recognition is divided into three stages: plate detection, preprocessing and character recognition. The system results describe a plate detection rate of 100% and a character recognition rate of 58,33%.

Keywords: Machine Vision, Character Recognition, License Plate Detection

Índice general

Resumen.....	2
Abstract	3
1. Introducción	7
2. Justificación	9
3. Delimitación.....	9
3.1 Objetivo General.....	9
3.2 Objetivos Específicos.....	10
3.3 Acotaciones.....	10
4. Cargo que desempeño en la empresa y personas a cargo.	10
5. Marco teórico.....	11
5.1 Características de la placa	11
5.2 Procesamiento de imágenes	12
5.2.1 Segmentación.....	12
5.2.2 Operaciones morfológicas	13
5.2.3 Transformaciones Geométricas	14
5.2.4 Extracción de características.....	14
5.3 Detección de objetos	14
5.3.1 Deep Learning.....	14
5.4 Reconocimiento y clasificación	15

5.4.1 Redes neuronales artificiales.....	15
5.5 Antecedentes	16
6. Metodología	17
6.1 Obtención de elementos y prediseño del sistema.	17
6.2 Técnicas de visión artificial para el sistema ALPR.	21
6.2.1 Técnicas tradicionales.....	21
6.2.2 Deep learning	22
6.3 Elección de técnicas para el sistema.	23
6.4 Generación de patrones fotográficos	24
6.5 Creación de web app y base de datos.....	25
6.6 Pruebas de funcionamiento, mejoramiento y análisis de calidad.	26
7. Diseño e implementación.....	27
7.1 Elementos de Hardware	27
7.2 Elementos de Software	28
7.3 Detección de placa	30
7.4 Preprocesamiento	32
7.5 Reconocimiento de caracteres.....	33
7.6 Sistema informático e interfaz grafica	34
8. Resultados.	37
8.1 Controlados	37

8.2 No controlados	39
9. Análisis e impacto.....	41
10. Conclusiones	43
11. Recomendaciones	44
12. Presupuesto económico.....	45

Introducción

El reconocimiento automático de placas vehiculares, ALPR, por sus siglas en inglés, es un método de vigilancia que utiliza el reconocimiento óptico de caracteres (OCR) y la detección de objetos para leer las placas de vehículos de transporte (Baggio, et al., 2012). Históricamente fue creada a finales del siglo XX, en el año 1976 en la subdivisión de investigación de la policía (PSDB) del Reino Unido. Por esto sus primeros usos fueron la detección de automóviles robados, desde entonces su uso ha avanzado considerablemente y se utiliza para estudios de tráfico, control de acceso y parqueo, y cumplimiento de leyes de tránsito. (ANPR International Ltd)

La Gobernación de Sucre es una entidad gubernamental ubicada en Sincelejo, capital del departamento de Sucre, en sus instalaciones cuenta con un parqueadero privado usado por los empleados adscritos al ente. Con el uso del sistema ALPR se busca reconocer la placa única nacional de Colombia descrita por el ministerio de transporte, utilizando como base los vehículos de tipo automóvil que brindan un servicio particular. La placa contiene 6 caracteres ubicados horizontalmente constituidos por 3 letras y 3 números.

El trabajo está dividido en tres partes principales: investigación, experimentación y resultados. La investigación busca reconocer métodos de visión computacional ya sean tradicionales o modernos, que permiten llevar a cabo la detección, procesamiento y reconocimiento de las placas vehiculares. Luego de la investigación se aplican prácticamente los métodos encontrados y se comparan para obtener técnicas que se adecuen a las restricciones computacionales del sistema usado. Los resultados mostrarán los caracteres reconocidos en espacios controlados y no controlados, con descripciones de velocidad de procesamiento y porcentajes de éxito.

Introduction

Automatic license plate recognition (ALPR) is a surveillance method that uses optical character recognition (OCR) and object detection to read transportation vehicles license plates (Baggio, et al., 2012). Historically it was created at the end of the 20th century, in the year 1976 in the police research subdivision (PSDB) of the United Kingdom. For this reason, its first uses were the detection of stolen cars, since then its use has advanced considerably and it is used for traffic studies, access and parking control, and traffic law enforcement. (ANPR International Ltd).

The Governor's Office of Sucre is a government entity located in Sincelejo, capital of the department of Sucre, in its facilities it has a private parking lot used by its employees. The ALPR system is used to recognize Colombia's unique national license plate as described by the Ministry of Transportation, using as a basis the cars that provide private services. The plate contains 6 characters located horizontally, consisting of 3 letters and 3 numbers.

The work is divided into three main parts: research, experimentation and results. The research aims to recognize traditional and modern computer vision methods that allow the detection, processing and recognition of vehicle license plates. After the research, the methods found are applied in practice and compared to obtain techniques that fit the computational constraints of the used system. The results will show the characters that are recognized in controlled and uncontrolled environments, with descriptions of processing speed and success rates.

1. Justificación

Teniendo en cuenta que para el año 2021 según la Asociación del Sector Automotriz (ASOAPRTEs) los robos a vehículos se incrementaron en un 10,21% (Rodríguez, 2021). Las entidades estatales como las gobernaciones o alcaldías, deben tener esquemas de seguridad acordes a la importancia pública de sus sedes. Uno de los puntos clave para mantener ese orden es el control de los ingresos tanto de vehículos como de peatones. Con previo conocimiento del tamaño del parqueadero de las instalaciones, los cupos son repartidos solo a un grupo de trabajadores y solo estos pueden ingresar al parqueadero, si llegasen a entrar vehículos no registrados se generarían problemas de seguridad y/o desorden, por lo que es de gran importancia reconocer los automóviles ya sea por el conductor a bordo, su carnet o el número de la placa.

El desarrollo de un sistema-prototipo para el reconocimiento y detección de placas en el parqueadero de la Gobernación de Sucre hace que el acceso al estacionamiento no dependa de errores o negligencia humana, además sirve como herramienta para que el personal de seguridad tenga un control de llegada y salida de mayor calidad. El uso de visión e inteligencia artificial para obtener información de imágenes y videos ha sido amplio dentro de distintos campos, en el control de ingreso permite la automatización de una tarea repetitiva y la optimización de los recursos humanos.

2. Delimitación

2.1 Objetivo General.

Desarrollar un sistema-prototipo que identifique placas vehiculares mediante el uso de visión e inteligencia artificial para el control de ingreso al parqueadero de la Gobernación de Sucre.

2.2 Objetivos Específicos.

- Diseñar y construir un sistema-prototipo de reconocimiento de placas vehiculares usando el computador de pequeña escala Raspberry PI.
- Implementar técnicas de inteligencia artificial para reconocer los caracteres de las placas vehiculares.
- Evaluar la calidad del sistema-prototipo a partir de los porcentajes de éxito de la detección y reconocimiento.
- Realizar un sistema informático para administrar el control de ingreso al parqueadero.

2.3 Acotaciones.

El proyecto será desarrollado en 3 meses utilizando la computadora a pequeña escala Raspberry Pi con una cámara de bajo costo e implementación en el lenguaje Python. Teniendo en cuenta el lugar de ejecución las horas activas del sistema serán diurnas y no se realizarán cambios en la infraestructura del parqueadero.

3. Cargo que desempeño en la empresa y personas a cargo.

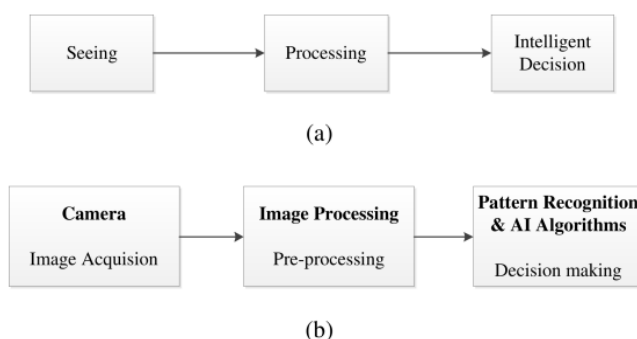
Se desempeña el cargo de ingeniero de soporte técnico, con objetivos de brindar soporte de hardware y software a infraestructura informática y equipos de cómputo de la Gobernación de Sucre y el cargo ingeniero de visión artificial con el objetivo de desarrollar un sistema-prototipo que identifique placas vehiculares. El personal a cargo son el ingeniero Carlos Mauricio Diaz Royerth jefe de la oficina de sistemas y Stephanie del Carmen Bermejo Altamar subsecretaria de gestión de recursos físicos.

4. Marco teórico.

La visión artificial es un campo de las ciencias informáticas, y su objetivo es permitir que los computadores procesen e identifiquen imágenes y vídeos del mismo modo que lo hace la visión humana. La visión computacional pretende imitar el sistema visual humano. El objetivo es construir sistemas artificiales que puedan extraer información de las imágenes, es decir, el objetivo es hacer que los ordenadores entiendan las imágenes y los vídeos. (Bhuyan, 2019)

Figura 1

Sistema visual humano vs. visión artificial (Bhuyan, 2019)



Los datos de la imagen pueden ser una secuencia de vídeo, imágenes de profundidad, vistas de varias cámaras o datos multidimensionales de sensores de imagen. El objetivo principal de la visión por computador es describir una escena del mundo real en una o más imágenes e identificar y reconstruir sus propiedades, como las características de color, la información de forma, las características de textura, la iluminación de la escena, etc.

4.1 Características de la placa

La placa única nacional, descrita en la ficha técnica MT 001 del Ministerio de Transporte, es un instrumento de identificación externa y privativo de los vehículos de transporte. Es intransferible, único, permanente y de validez en todo el territorio de Colombia. Está clasificado

según el tipo de vehículo: ya sean automóviles y similares o motocicletas y similares; y el tipo de servicio: particular, publico, oficial, entre otros. (Ministerio de Transporte)

En el aspecto físico lleva de forma horizontal, 6 caracteres en relieve: 3 letras y 3 dígitos en dos grupos separados por el logotipo del Ministerio de Transporte. La placa tiene un tamaño de 330 mm de largo por 160mm de ancho, sus características de color son un fondo amarillo, caracteres, logotipo y marco negro para vehículos particulares. El tamaño máximo de los caracteres es de 80 mm de alto por 40 mm de ancho, y varía según la morfología de los números y las letras. (Ministerio de Transporte)

Figura 2

Representación de placa única nacional de Colombia. (Ministerio de Transporte)



4.2 Procesamiento de imágenes

4.2.1 Segmentación

Binarización

La binarización es uno de los enfoques ingenuos de la segmentación de imágenes.

En general, la segmentación basada en el umbral da como resultado una imagen binaria. Donde la imagen de entrada está representada por $f(x; y)$, y la salida viene dada por:

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T \end{cases} \quad (1)$$

El método de Otsu encuentra el valor óptimo del umbral global óptimo. Busca un umbral que minimice la varianza intraclase de las imágenes segmentadas. El umbral de Otsu se encuentra

buscando en todo el rango de valores de los píxeles de la imagen hasta que la varianza intraclase se acerque a su mínimo. (Bhuyan, 2019)

4.2.2 Operaciones morfológicas

Las operaciones morfológicas se denotan para dos imágenes. La imagen que se procesa es la imagen activa, y la segunda imagen se denomina núcleo o elemento estructurante. Cada elemento estructurante tiene una forma preestablecida, que se aplica como un filtro en la imagen activa. (Bhuyan, 2019)

Dilatación

Dado un conjunto A y el elemento estructurante B, la dilatación de A con B se define como:

$$A \oplus B = \{x/A \cap (\hat{B})_x \neq \phi\} \quad (2)$$

Generalmente el tamaño de B es menor que el de A. Si B se coloca en la frontera de A, entonces el tamaño de A aumenta para incluir los puntos de B. Por lo tanto, todos los puntos que tocan la frontera se incluirán debido a la dilatación. (Bhuyan, 2019)

Erosión

La erosión de A con B viene dada por:

$$A \ominus B = \{x/B_x \subseteq A\} \quad (3)$$

En esta operación, el elemento estructurante B debe estar completamente dentro del objeto A, por lo que no se incluyen los píxeles de frontera. Dos regiones casi conectadas serán separadas por la operación de erosión. Cualquier hueco dentro del objeto se incrementará y el límite de un objeto puede ser suavizado por la operación de erosión. (Gonzalez & Woods, 2018)

Convex Hull (Aproximación de polígonos)

Un conjunto, S, de puntos en el plano euclidiano se dice que es convexo si, y sólo si, un segmento de línea recta que une dos puntos cualesquiera de S está enteramente dentro de S.

El convex hull, H , de S es el conjunto convexo más pequeño que contiene a S . (Gonzalez & Woods, 2018)

4.2.3 Transformaciones Geométricas

Las transformaciones geométricas modifican la disposición espacial de los píxeles de una imagen. (Gonzalez & Woods, 2018)

Redimensionamiento

El redimensionamiento es un tipo de transformación afín que esta descrita por la siguiente matriz:

$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

La transformación anterior mueve las coordenadas de los píxeles de una imagen a nuevas ubicaciones. Para completar el proceso, se deben que asignar valores de intensidad a esas ubicaciones. Esta tarea se lleva a cabo mediante la interpolación de la intensidad. (Gonzalez & Woods, 2018)

4.2.4 Extracción de características

Detección de esquinas de Harris

El detector de esquinas Harris considera los gradientes locales en las direcciones horizontal y vertical alrededor de cada punto. El objetivo es encontrar puntos en la imagen en los que la intensidad de la imagen varía en ambas direcciones (una esquina) en lugar de en una dirección (un borde), o en ninguna (una región plana plana). (Prince, 2012)

4.3 Detección de objetos

4.3.1 Deep Learning

El aprendizaje profundo (también conocido como aprendizaje estructurado profundo) forma parte de una familia más amplia de métodos de aprendizaje automático basados en redes

neuronales artificiales con aprendizaje de representación. El aprendizaje puede ser supervisado, semisupervisado o no supervisado. (Bengio, LeCun, & Hinton, 2015)

Detector SSD

El single-shot multibox detector (SSD) es un modelo de detección de objetos de una red neuronal profunda. Este funciona discretizando el espacio de salida de los cuadros delimitadores en un conjunto de cuadros por defecto sobre diferentes relaciones de aspecto y escalas por ubicación del mapa de características. En el momento de la predicción, la red genera puntuaciones para la presencia de cada categoría de objeto en cada caja por defecto y produce ajustes en la caja para que se ajuste mejor a la forma del objeto. Además, la red combina las predicciones de múltiples mapas de características con diferentes resoluciones para manejar de forma natural objetos de diversos tamaños. (Liu, et al., 2016)

4.4 Reconocimiento y clasificación

4.4.1 Redes neuronales artificiales

LSTM

Long Short-Term Memory (LSTM) es una red neuronal artificial utilizada en los campos de la inteligencia artificial y el aprendizaje profundo. A diferencia de las redes neuronales estándar, la LSTM tiene conexiones de retroalimentación. Una red neuronal recurrente de este tipo puede procesar no solo puntos de datos individuales (como imágenes), sino también secuencias enteras de datos (como el habla o el vídeo). Por ejemplo, la LSTM es aplicable a tareas como el reconocimiento de la escritura a mano no segmentada y conectada, el reconocimiento del habla, la traducción automática, el control de robots, los videojuegos y la atención sanitaria. La LSTM se ha convertido en la red neuronal más citada del siglo XX. (Hochreiter & Schmidhuber, 1997)

4.5 Antecedentes

En el trabajo “License plate recognition for campus auto-gate system” Yaacob et. al. utilizan la tecnología de detección de placas vehiculares para el monitoreo de la entrada y salida de los vehículos en las puertas de acceso de una universidad. El sistema propuesto tendrá la habilidad de capturar, verificar y reconocer las placas utilizando técnicas de procesamiento de imágenes, los resultados del sistema dan una exactitud de detección de 91.58%. (Yacob & Alkahtani, 2021)

En el trabajo “Desarrollo e implementación de un sistema automático de reconocimiento de placas vehiculares” Cruz y Rodríguez presentan la implementación de un sistema de reconocimiento de placas de vehículos colombianos para administrar el acceso de residentes al Conjunto Multifamiliar Ciudadela Comfenalco. El autor baso su programación en Python. El autor presenta un nivel de precisión del 85.7%, una sensibilidad del 87.5% y una especificidad de 75%, describiendo un rendimiento aceptable. (Cruz & Rodriguez, 2021)

En el trabajo “Identificación de caracteres en placas de carros colombianos utilizando diferentes técnicas para su estudio comparativo” el autor realiza una comparación de técnicas de visión artificial para el reconocimiento de caracteres de placas vehiculares, usando Machine Learning y Deep Learning. Inicialmente, se realiza un proceso de segmentación usando binarización y detección de contornos. Las técnicas de clasificación usadas son las máquinas de soporte vectorial, random forest y KNN, y como patrones característicos se tienen los histogramas de gradientes, las características HAAR y los local binary patterns. Los resultados presentados determinan cuales técnicas usadas dan una mayor probabilidad de éxito de reconocimiento. (España, 2021)

5. Metodología

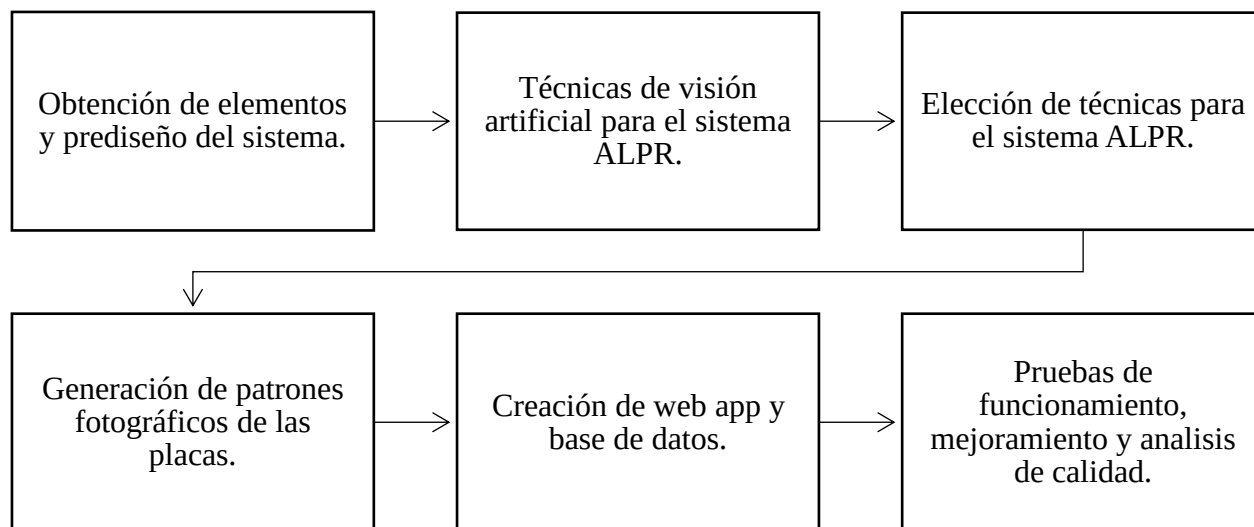
Para la metodología, se utiliza un enfoque exploratorio y cuantitativo, con una búsqueda de descripción experimental y resultados cuantitativos de experimentos controlados y no controlado.

El reconocimiento automático de placas vehiculares es una tecnología que utiliza la visión artificial para leer o extraer los números de registro de las placas, los algoritmos básicos para esta lectura requieren la detección, el preprocesamiento y el reconocimiento de caracteres.

Los pasos para llegar a estos algoritmos se presentan en la figura 3, a partir de este diagrama, se describen en los índices posteriores las metodologías que engloban la obtención de elementos de hardware, la elección de las técnicas de visión artificial, la creación de la aplicación web y las pruebas de funcionamiento.

Figura 3

Tareas de la metodología



5.1 Obtención de elementos y prediseño del sistema.

En la obtención de elementos de hardware y el prediseño del sistema, se realiza la elección del sistema embebido y la cámara, y se define la estructura general del sistema de reconocimiento.

En la tabla 1 se observan los posibles sistemas embebidos a usar en el reconocimiento de placas, la tabla describe las especificaciones, la disponibilidad y el precio de los sistemas.

Tabla 1

Comparación de sistemas embebidos

Sistema	Especificaciones			Disponibilidad	Precio
	Procesador	RAM	Conectividad		
Arduino 33 BLE	Nrf52840 32-bit ARM 64MHz	256KB SRAM	Bluetooth, USB	Media	22,3
Arduino Uno	Microcontrolador ATmega328P	32Kb SRAM	USB	Muy alta	25,3
Raspberry Pi 3b+	1.4GHz 64-bit quad-core Cortex-A53	1 GB LPDDR2	Wireless Lan, Ethernet, USB, Bluetooth	Media	39
Raspberry Pi 4b	1.5GHz 64-bit quad-core Cortex-A72	1/2/4/8GB LPDDR4	Wireless Lan, Ethernet, USB, Bluetooth	Media	43,5
Jetson Nano DK	Quad-core ARM A57 @ 1.43 GHz + 128-core Maxwell GPU	4 GB 64-bit LPDDR4	Ethernet, M.2 Key E Wireless slot	Media	59
Coral AI DB	Quad-core Cortex-A53 + acelerador ML Edge TPU coprocessor	1 GB LPDDR4	Wireless Lan, Bluetooth, USB, Ethernet	Baja	130

Nota: Los precios presentados están en dólares estadounidenses y representan los valores de los sitios web oficiales.

Los sistemas descritos están divididos en 3 grupos principales: los de costo bajo, costo medio y costo alto. Los dispositivos de costo bajo son los de la plataforma Arduino, estos dispositivos usan microcontroladores y procesadores con velocidades menores a 100MHz, con memorias de acceso aleatorio (RAM) estática y no tienen conectividad Ethernet ni Wi-Fi. Para los de costo medio se tienen los dispositivos Raspberry Pi en sus versiones 3B+ y 4B, estos contienen procesadores capaces de correr sistemas operativos y tienen arquitectura de x64, con RAM

dinámicas de mínimo 1GB y conectividad Ethernet y Wi-Fi. Los dispositivos de alto costo en el aspecto de capacidad de procesamiento son bastante parecidos a las tarjetas Raspberry Pi, pero estos contienen unidades de procesamiento grafico (GPU) dedicadas, que tienen alta utilidad en la aceleración de algoritmos deep learning.

El sistema embebido de elección es la Raspberry Pi 3B+ ya que este permite correr modelos de deep learning a velocidades aceptables, comparadas con los dispositivos de bajo costo, aunque no permite la aceleración por GPU. Además, cuenta con una disponibilidad y precio medio dentro del mercado. En la conectividad en comparación con la Jetson Nano esta tiene capacidad Ethernet y Wi-Fi, y si se entiende que el sistema tendrá capacidades IoT esto facilita el diseño de hardware.

La tabla 2 presenta las posibles cámaras para el sistema ALPR, en esta se describen 3 tipos de cámara: las originales, vendidas por la Fundación Raspberry Pi, las USB y las vendidas por fabricantes terceros. La tabla describe la resolución de captura, el sensor usado, la resolución del sensor y el precio.

Tabla 2

Comparación de cámaras

Cámara	Resolución captura	Sensor	Resolución sensor	Precio
Pi Camera v1	5 MP	OmniVision OV5647	2592 × 1944	30
Pi Camera v2	8 MP	Sony IMX219	3280 × 2464	30
Pi HQ Camera	12.3 MP	Sony IMX477	4056 x 3040	50
Waveshare Pi Camera	5 MP	OmniVision OV5647	2592 × 1944	16
USB: Logitech C505	1.2 MP	N/A	1280 x 720	70

Nota: Los precios presentados están en dólares estadounidenses

Las cámaras oficiales corresponden a las versiones Pi Camera V1, V2 y HQ; según sus fechas de fabricación. Las Pi Camera V1 tiene una resolución de captura de 5MP, la V2 de 8MP

y la HQ de 12.3MP, por lo que la HQ es la que toma las fotos con mayor detalle por un amplio margen, aunque esta cuesta el doble que sus predecesores.

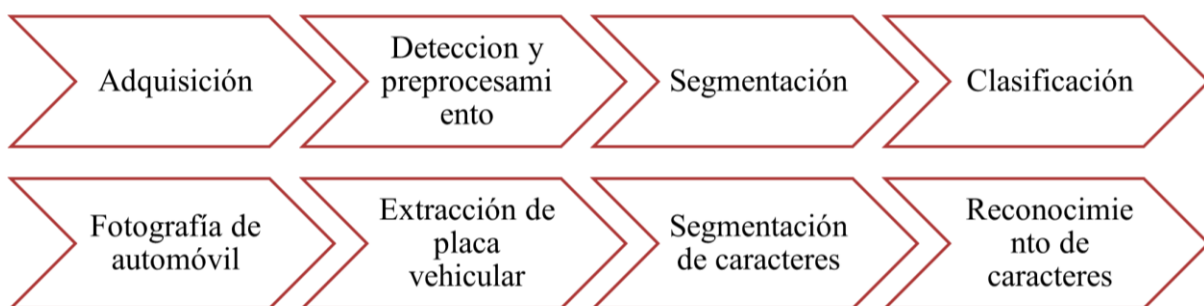
Una opción adicional a las cámaras de conexión serial son las cámaras USB vendidas para el uso en videoconferencias. Estas cámaras ofrecen una facilidad de conexión mayor y un posprocesamiento de mayor calidad, pero son más lentas en la conversión a datos primarios y tienen una menor resolución. Además, son más costosas ya que están diseñada para usuarios finales y no desarrolladores.

Reconociendo el alto costo de la cámara USB de la compañía Logitech y que la cámara de fabricante terciario Waveshare, tiene las mismas especificaciones que la Pi Camera V1 con la mitad de precio. Se elige la cámara RPi Camera (B), Adjustable-Focus como el hardware para la toma de fotografías de los automóviles.

En la figura 4 como parte del prediseño se presentan las 4 etapas un sistema de visión artificial junto a su acción específica en el sistema ALPR, la adquisición corresponde a la toma de la fotografía del automóvil, en la detección y preprocesamiento se encuentra la ubicación de la placa en la imagen, se extrae con las coordenadas y se preprocesa para obtener mejores resultados etapas de segmentación y clasificación, en estas dos últimas etapas se realiza la lectura del número de registro de la placa.

Figura 4

Etapas de la visión artificial y el reconocimiento de placas vehiculares



Los posibles algoritmos y técnicas para completar las etapas descritas, se presentan en los siguientes incisos.

5.2 Técnicas de visión artificial para el sistema ALPR.

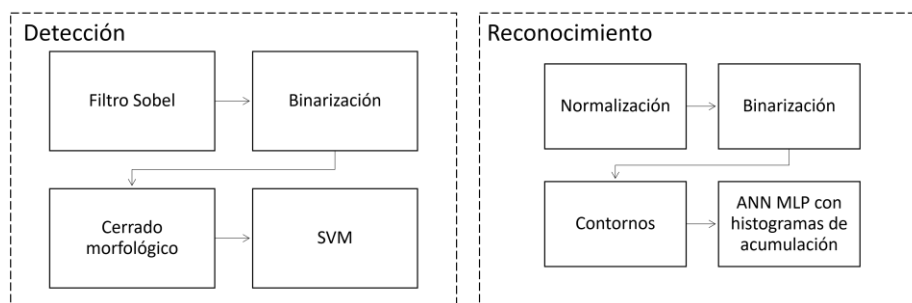
Luego de una revisión bibliográfica las posibles técnicas para el sistema están caracterizadas en dos grupos: técnicas tradicionales y deep learning. A continuación, se describen los algoritmos utilizados para los dos grupos teniendo en cuenta las características más importantes.

5.2.1 Técnicas tradicionales

En la figura 5, se presentan los algoritmos utilizados para las pruebas de técnicas tradicionales. Están divididas en la etapa de detección y reconocimiento, y cada de una de estas cuentan con algoritmos de preprocesamiento.

Figura 5

Diagrama ALPR con técnicas tradicionales



El sistema está basado en el trabajo de (Baggio, et al., 2012), implementar este sistema de reconocimiento ayuda a entender las etapas de diseño necesarias para obtener un sistema ALPR.

Uno de los aspectos más importantes del diseño son las características morfológicas de la placa vehicular como lo son el tamaño, color y configuración textual. A partir de estos se pueden realizar comprobaciones de la detección y segmentación. El software implementado está programado en C++, utilizando la librería OpenCV. La detección está dividida en dos pasos, la segmentación, en donde se aplican distintos métodos clásicos como los filtros Sobel, la

binarización, las operaciones morfológicas, entre otros y la clasificación del segmento en donde se utilizan las máquinas de soporte vectoriales. (SVM).

En la etapa de reconocimiento de caracteres se utiliza una red neuronal artificial, tanto para el entrenamiento y la inferencia, se tiene un proceso de segmentación de los caracteres de la placa y una extracción de características basada en histogramas de acumulación verticales y horizontales. Para la inferencia se utiliza un perceptrón multicapa que después del entrenamiento es capaz de reconocer las letras y números de las placas.

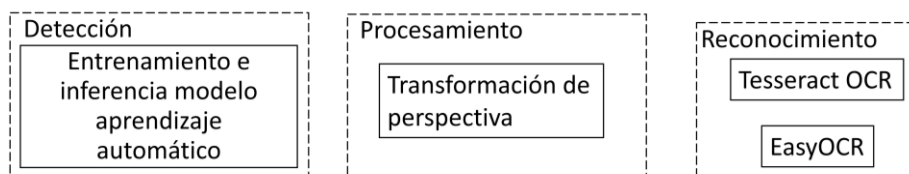
Aunque la implementación del sistema permite distinguir los aspectos importantes de las etapas de ALPR, el fin del experimento 1 es puramente académico ya que las velocidades del sistema son muy bajas para la restricción de hardware de la Raspberry Pi 3B+.

5.2.2 Deep learning

En el caso del deep learning se utilizan modelos más orientados a ser sistemas blackbox, en la figura 6 se observan las técnicas de ALPR usando deep learning y están caracterizadas en 3 etapas, en la detección se realiza el entrenamiento de un modelo SSD MobileNet, que sea capaz de ubicar dentro de la imagen capturada las coordenadas que permitan extraer solo la placa cuando se realice la inferencia. Para el preprocesamiento se tienen algoritmos de procesamiento de imágenes que permitan eliminar rotaciones e inclinaciones en la placa encontrada antes de que esta ingrese a la etapa de reconocimiento.

Figura 6

ALPR con técnicas de deep learning



En el reconocimiento se realizan pruebas con EasyOCR y TesseractOCR, librerías de uso libre, para lograr el reconocimiento los 6 caracteres ubicados en la placa, con el fin de enviar los resultados a la aplicación web. La librería elegida para el diseño final es TesseractOCR, ya que tiene mayor velocidad en dispositivos que no cuentan con GPUs.

5.3 Elección de técnicas para el sistema.

En la tabla 3, se presenta un cuadro comparativo entre las técnicas tradicionales y el deep learning, los aspectos tomados en cuenta son: la extracción de características, la cantidad de datos de entrenamiento, los tipos de modelos, respuesta a cambios en la entrada, la mejorabilidad y la reusabilidad.

Tabla 3

Elección de técnicas para el sistema ALPR

Descripción	Tradicional	Deep learning
Extracción de características	Humana	Algorítmica
Librerías	Sckit-learn	TensorFlow, Pytorch
Datos para entrenamiento	Generalizado	Mucha cantidad
Tipos de modelos	Grey box	Black box
Tiempo de entrenamiento	Bajo	Alto
Respuesta a cambios de entrada	Media	Alta
Mejorabilidad	Re-training	Transferencia de conocimiento
Reusabilidad	Baja	Alta

Para los métodos tradicionales la librería más usada es Scikit-learn, estas técnicas, cuentan con una extracción de características humana, lo que implica que un ingeniero o diseñador debe estudiar la información para encontrar los patrones importantes de las fotografías. Lo anterior explica el menor tiempo de entrenamiento comparado con las técnicas de deep learning. Para los datos de entrenamiento se permiten una cantidad generalizada, por lo que depende mucho del algoritmo usado y lo que se espere de los resultados, además se reconocen como un sistema

greybox ya que el diseñador puede aplicar mejoramientos cambiando los patrones o la configuración del algoritmo.

Para el deep learning hay una mayor cantidad de librerías o frameworks usados, los principales son TensorFlow y Pytorch. Estas librerías permiten al usuario entrenar, configurar y ejecutar inferencias, con modelos de deep learning. Este tipo de técnicas cuenta con una extracción de características algorítmica, el usuario no debe reconocer los patrones claves previamente al entrenamiento del modelo de inferencia. Por esta razón la cantidad de entrenamiento debe ser alta y se caracterizan por ser un sistema blackbox. En todos los aspectos faltantes, ignorando el tiempo de entrenamiento, el deep learning cuenta con ventajas en su respuesta a cambios de entrada, mejorabilidad y reusabilidad.

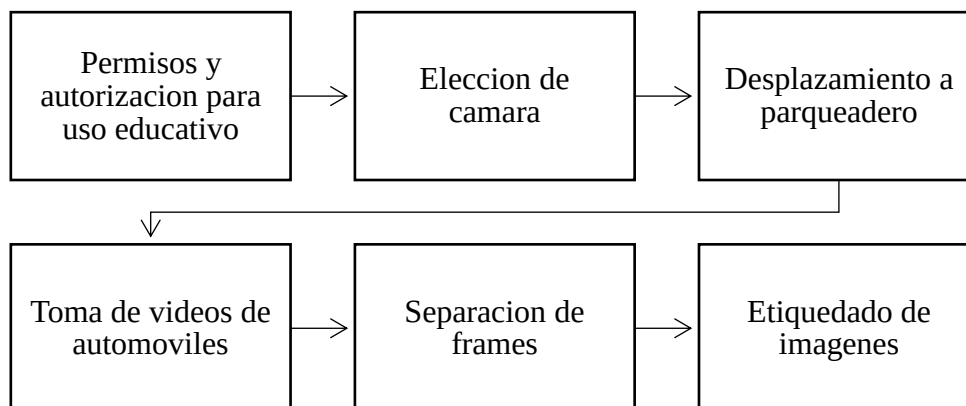
5.4 Generación de patrones fotográficos

La generación de patrones fotográficos se decide realizar en el lugar de ejecución y no se utilizan bases de datos públicas. Las etapas de la generación se presentan en la figura 7, la primera etapa corresponde a la creación del acta para la autorización del uso de las imágenes de manera educativa, en esta se piden los permisos necesarios tanto al personal administrativo como al personal vigilancia para lograr capturar las imágenes de los vehículos que ingresan y salen del parqueadero, aunque se permite la captura de los videos el personal de la entidad recomienda no tomar las firmas de autorización, argumentando que es suficiente con el permiso del personal de vigilancia. A partir de este permiso se elige una cámara del teléfono personal Realme 6, con soporte para video de 1080p y 30fps, y se realiza el desplazamiento tomando los videos en 3 jornadas: mañana, tarde y noche, que son las horas en las que se da mayor movimiento de vehículos y con mayor variación de iluminación en las horas activas del sistema. Luego de la toma de videos se procede a una separación de los cuadros del video utilizando el software abierto FFMPEG cada

500 ms, y se seleccionan las imágenes representativas de cada video. Finalmente, estas imágenes son llevadas al etiquetado previo al entrenamiento del algoritmo de deep learning.

Figura 7

Etapas para la generación de patrones fotográficos



5.5 Creación de web app y base de datos

A partir de la elección de la Raspberry Pi 3B+, se elige que el sistema completo este desarrollado en el lenguaje Python, por lo que la elección del framework para la creación de la aplicación web está basada en el criterio de que el lenguaje de programación sea Python, las dos posibles opciones se presentan en la tabla 4, y son los framework Django y Flask.

Tabla 4

Comparación para la elección del framework web

Caracteristica	Django	Flask
Tipo de framework	Full Stack	Micro
Flexibilidad	Repleto	Completa
Uso de ORM	Integrada	SQLAlchemy
Diseño	Listo para usar	Minimalista
Estilo de trabajo	Monolitico	Diversificado

En aspectos generales la gran diferencia entre los dos framework es la cantidad opciones y características que traen si se comparan las instalaciones limpias. Django es un framework full stack, que tiene funcionalidades tanto para el manejo de los servidores, como para el diseño de la

vista de usuario, su característica implica una baja flexibilidad, alta integralidad con bases de datos y facilidad de desarrollo. En comparación Flask ofrece un framework tipo micro, que el usuario podrá configurar para el uso requerido, lo que implica una flexibilidad alta, un diseño minimalista y la importación de librerías terceras para que este encuentre utilidad en la creación de las aplicaciones.

5.6 Pruebas de funcionamiento, mejoramiento y análisis de calidad.

Las pruebas del prototipo están divididas en dos partes, las de funcionamiento y las de mejoramiento: Las de funcionamiento corresponden a encontrar resultados del sistema, que permitan detectar la placa y reconocer las letras del sistema, los resultados de estas pruebas son descritos en resultados controlados y no controlados, en la tabla 5 se presentan los aspectos importantes que describen las pruebas en ambiente controlado y en ambiente no controlado, las diferencias abarcan el sistema de procesamiento usado para la inferencia, la adaptación de los algoritmos, el lugar de ejecución y la similitud entre los patrones de imágenes de entrenamiento y prueba.

Tabla 5

Tipos de pruebas de funcionamiento

Ambiente controlado	Ambiente no controlado
Inferencia en computador con sistema operativo Windows	Inferencia en lugar de ejecución con sistema embebido Raspberry Pi
Resultados adaptados a cambios en el preprocesamiento	No adaptabilidad del preprocesado
Fotografías de prueba y entrenamiento tomadas en las mismas condiciones climatológicas	Diferentes condiciones climatológicas que las fotografías de entrenamiento

Por otra parte, las pruebas de mejoramiento se basan en aspectos como:

- La cantidad de fotografías en la generación de patrones.

- Variaciones en la distancia de enfoque de la cámara.
- Configuración del tamaño del batch para el entrenamiento del modelo deep learning.
- Distancia de la fotografía tomada.
- Porcentaje de fondo en las fotografías.
- Velocidad del sistema.

Finalmente, el análisis de calidad esta descrito por:

- La prueba unitaria
- La prueba funcional
- La interoperabilidad
- El desempeño: velocidad y precisión
- La fiabilidad
- La autonomía

6. Diseño e implementación

A continuación, se presenta el diseño final y la implementación del sistema de reconocimiento de placas. En el capítulo se exponen los elementos de hardware y software, y se describen los algoritmos y procesos para la detección de la placa, el preprocesamiento, el reconocimiento de caracteres y el sistema informático.

6.1 Elementos de Hardware

Figura 8

Construcción del hardware



La construcción física del sistema se encuentra en la figura 8, en esta se observan la base de soporte de la cámara y 2 componentes de hardware: la Pi Camera y la SBC Raspberry Pi.

Cámara

La cámara usada en el sistema es el módulo RPi Camera (B), Adjustable-Focus, el cual contiene un sensor OV5647 de 5 megapíxeles con las siguientes especificaciones técnicas:

- Tamaño del CCD: 1/4 de pulgada
- Apertura (F): 2.0
- Distancia focal: 6mm
- Ángulo de visión (diagonal): 60,6 grados
- Mejor resolución del sensor: 1080p
- Dimensión: 32mm x32mm

Raspberry Pi 3b+

Raspberry Pi es una serie de pequeños ordenadores de placa única (SBC) desarrollados en el Reino Unido por la Fundación Raspberry Pi en asociación con Broadcom. El proyecto Raspberry Pi se orientó originalmente a la promoción de la enseñanza de las ciencias informáticas básicas en las escuelas y en los países en desarrollo. (Raspberry Pi Foundation, n.d.) La Raspberry Pi 3b+ cuenta con las siguientes especificaciones técnicas:

- Broadcom BCM2837B0, SoC Cortex-A53 (ARMv8) de 64 bits a 1,4 GHz.
- 1GB LPDDR2 SDRAM
- LAN inalámbrica IEEE 802.11.b/g/n/ac a 2,4 GHz y 5 GHz, Bluetooth 4.2, BLE
- Gigabit Ethernet sobre USB 2.0 (rendimiento máximo de 300 Mbps)

6.2 Elementos de Software

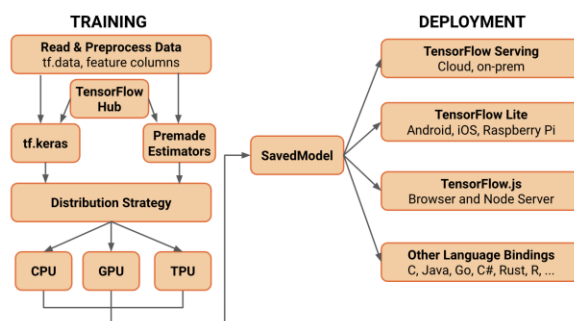
TensorFlow

TensorFlow es una biblioteca de software gratuita y de código abierto para el aprendizaje automático y la inteligencia artificial. Se puede utilizar en toda una serie de tareas, pero se centra

especialmente en el entrenamiento y la inferencia de redes neuronales profundas. (Abadi, et al., 2016) En la figura 9 se presenta la arquitectura para las etapas de entrenamiento y deployment.

Figura 9

Arquitectura de TensorFlow 2 (TensorFlow Team, 2019)



OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de funciones de programación destinada principalmente a la visión por ordenador en tiempo real. Desarrollada originalmente por Intel, fue apoyada posteriormente por Willow Garage y luego por Itseez. La biblioteca es multiplataforma y de uso gratuito bajo la licencia Apache 2 de código abierto. A partir de 2011, OpenCV incorpora la aceleración de la GPU para las operaciones en tiempo real. (Pulli, Baksheev, Korniyakov, & Eruhimov, 2022)

Tesseract OCR

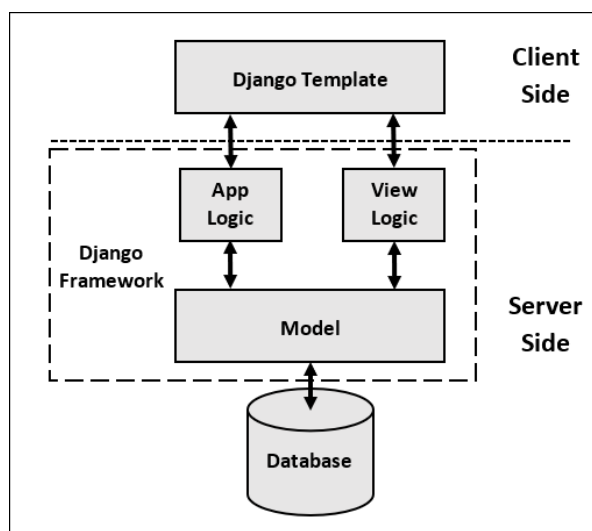
Tesseract es un motor de reconocimiento óptico de caracteres para varios sistemas operativos. Es un software libre, publicado bajo la licencia Apache. Desarrollado originalmente por Hewlett-Packard como software propietario en la década de 1980, fue liberado como código abierto en 2005 y su desarrollo ha sido patrocinado por Google desde 2006. (Announcing Tesseract OCR, 2006)

Django y Sqlite

Django es un framework web basado en Python, gratuito y de código abierto, que sigue el patrón arquitectónico modelo-plantilla-vista (MTV). Su mantenimiento corre a cargo de la Django Software Foundation (DSF), una organización independiente constituida en EE.UU. como entidad sin ánimo de lucro 501(c)(3). (Django Software Foundation , n.d.) La arquitectura básica de Django esta presentada en la figura 10.

Figura 10

Arquitectura Django (Mastering Django, n.d.)



SQLite es un motor de base de datos escrito en lenguaje C. No es una aplicación independiente, sino una biblioteca que los desarrolladores de software integran en sus aplicaciones. Como tal, pertenece a la familia de las bases de datos integradas. Es el motor de base de datos más extendido, ya que es utilizado por varios de los principales navegadores web, sistemas operativos, teléfonos móviles y otros sistemas integrados. (SQLite, n.d.)

6.3 Detección de placa

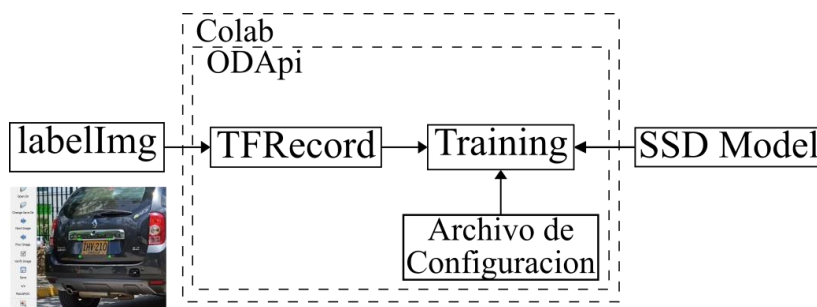
La detección es la etapa que permite ubicar el área de interés de la placa en la imagen tomada al vehículo, está dividida en dos partes: el entrenamiento, donde se realiza la extracción de

características de la placa y la inferencia en donde se aplica el modelo encontrado a imágenes de prueba. Dentro del entrenamiento se tienen en cuenta aspectos físicos de la captura de la imagen que busquen disminuir efectos no esperados en la inferencia, la velocidad del vehículo debe ser lenta e idealmente el automóvil debe detenerse, el porcentaje de fondo en la imagen debe ser mínimo y la distancia de captura de la imagen debe la mayor repetibilidad posible.

La detección se realiza usando el framework de aprendizaje profundo TensorFlow (TF), más específicamente la librería móvil TFLite. Para el entrenamiento se utilizan 93 imágenes que fueron previamente etiquetadas utilizando el software abierto labelImg, a partir de las etiquetas se pueden crear los archivos TFRecord, estos contienen la información de las imágenes y del etiquetado en un formato binario. Estos archivos serán la entrada a la API de detección de objetos de TF (ODAPI), que fue montada en el sistema de notebooks Google Colab, en este se puede ejecutar código Python en el navegador, y permite el uso de entornos de hardware con tarjetas gráficas. El uso de estos entornos gratuitos aumenta entre 2 o 3 veces la velocidad de aprendizaje de los modelos. La ODAPI brinda los archivos de configuración y scripting para el entrenamiento y la exportación del modelo de detección.

Figura 11

Entrenamiento de modelo de aprendizaje profundo



Los modelos de deep learning se construyen a partir de algoritmos específicos, en donde algunos algoritmos ofrecen ventajas específicas en velocidad y porcentaje de éxito sobre los otros,

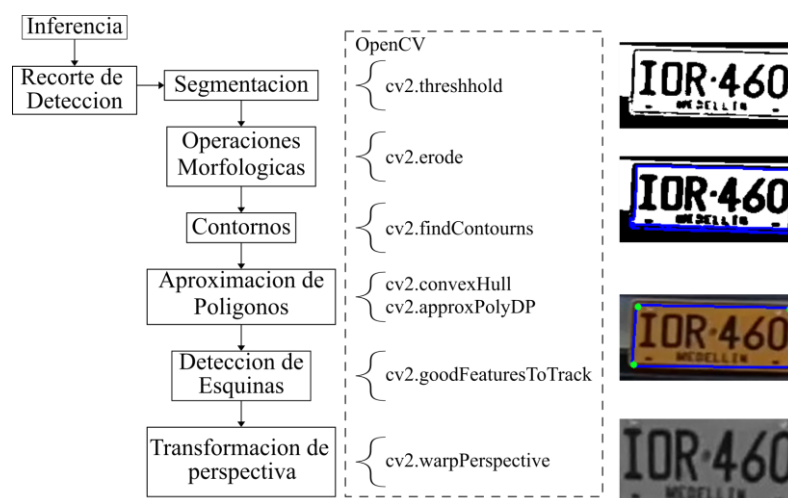
para la inferencia de la placa se utiliza un modelo MobileNet, basado en algoritmo Single-Shot Detection (SSD). Este modelo ofrece alta velocidad en dispositivos móviles, y esta pre entrenado en el dataset COCO 2017.

6.4 Preprocesamiento

El preprocesamiento es la segunda etapa del sistema, esta une a la detección con el reconocimiento de caracteres, en esta etapa se comprueba que la placa detectada cumple con las propiedades morfológicas y también se realiza la transformación de perspectiva de la placa usando transformaciones geométricas.

Figura 12

Etapa de preprocesamiento



Las propiedades morfológicas de la placa se toman en cuenta para determinar si los contornos encontrados corresponden a los esperados, en la detección se utiliza para determinar que el contorno encontrado puede ser utilizado para transformar la perspectiva y en el reconocimiento para la segmentación de los caracteres y el conteo de caracteres. Las propiedades utilizadas en ambos casos, son la relación entre la altura y el ancho, el área mínima y máxima, la altura mínima y máxima, y el conteo de píxeles blancos.

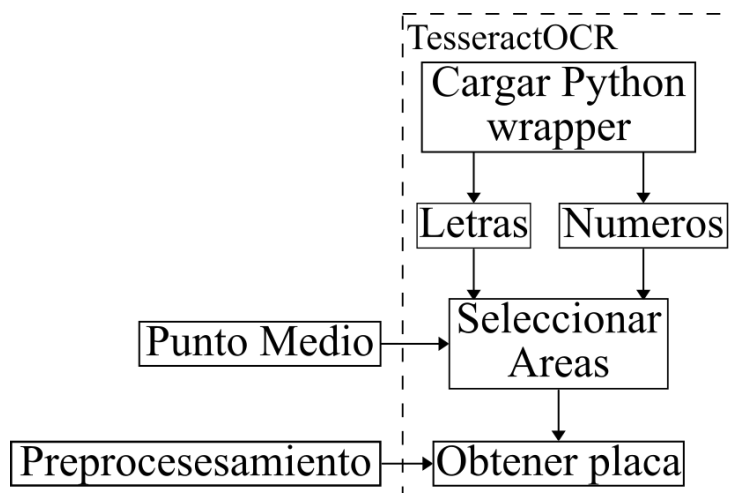
Los pasos del procesamiento se presentan en la figura 12, para la transformación de perspectiva es necesario encontrar los puntos de las esquinas la placa, una manera de realizarlo es encontrando los contornos y utilizando el detector de esquinas de Harris, el cual utiliza derivadas espaciales y otros algoritmos para encontrar la variación de intensidad más grande en imágenes en escala de grises o binarias. Para encontrar el contorno de la placa la imagen recibida del detector se segmenta utilizando el método Otsu, se le aplica dilatación para para rellenar pixeles alrededor de los posibles contornos y se encuentran los contornos uniendo los puntos continuos que tienen el mismo color. Antes de aplicar Harris, se realiza una aproximación del polígono del contorno, usando la función convex hull y realizando una aproximación de polígono por medio del algoritmo Douglas-Peucker.

6.5 Reconocimiento de caracteres

El reconocimiento de caracteres permite a partir de la imagen preprocesada saber cuáles letras y números componen la placa leída, la librería Tesseract OCR es capaz de reconocer caracteres en imágenes a partir de métodos clásicos basados en morfología y métodos modernos como las redes neuronales usando la red artificial de tipo long short-term memory (LSTM).

Figura 13

Etapas de reconocimiento



Aunque la librería es muy completa y tiene soporte para muchos lenguajes, esta requiere un preprocesamiento alto de la imagen de entrada, ya que los caracteres deben estar sin rotaciones y con tamaños de pixeles constrictivos. Se utiliza el modelo pre entrenado en idioma inglés y se agregan limitaciones en los caracteres posibles, solamente permitiendo resultados alfanuméricos exceptuando la Ñ. En la experimentación temprana se realizan pruebas ingresando la imagen directamente a la API, separando la entrada en los caracteres numéricos y las letras, y con procesamientos previos de filtrado y binarización de la imagen de la placa y redimensionamiento de los caracteres. Como se muestra en la figura 13, se realiza una separación en el punto medio, las letras y los números en áreas de interés distintas en el momento que se realiza el reconocimiento, por lo que se decide que el procesamiento de segmentación y filtrado sea hecho por la librería TesseractOCR.

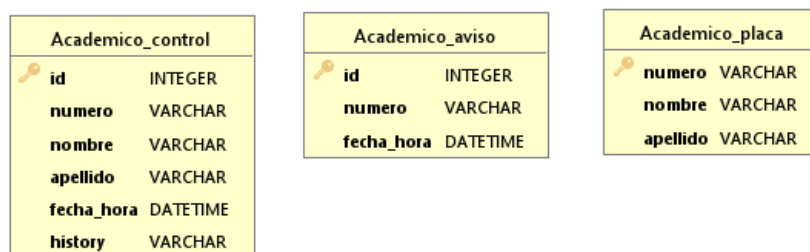
6.6 Sistema informático e interfaz grafica

La interfaz de control de acceso para el sistema se implementa con el framework Django, este ofrece funcionalidades de alto nivel para la manipulación de los datos, el procesamiento de las solicitudes de usuario y la presentación de la información.

Base de datos

La base de datos está construida en el entorno SQLite, y está integrada al framework Django. Su configuración de entidad está compuesta por 3 tablas, la tabla principal es Academico_placa en esta se mantienen las placas ingresadas por el usuario, que pertenecen a los usuarios permitidos. La lista Academico_aviso contiene la información de placas no permitidas y los horarios en las que estos equipos ingresan al parqueadero, y finalmente la lista Academico_control contiene los horarios de entrada y salida de los usuarios permitidos.

\

Figura 14*Entidades y referencias*

Aplicación web

La aplicación web está constituida por 3 módulos, el principal corresponde a la administración de placas, en este se listan las placas y se gestionan la eliminación, guardado y edición. Los valores mostrados al usuario son la placa, el nombre y el apellido del usuario.

Figura 15*Página principal*

The screenshot shows the 'Gestión de Placas' (License Plate Management) page. It includes a search bar and a table listing license plates with associated user information and action buttons.

#	Placa	Nombre	Apellido	Opciones
1	IHV210	Orlando	Diaz	Editar Eliminar
2	IOR460	Sebastian	Flores	Editar Eliminar
3	IVK167	Sandra	Samper	Editar Eliminar
4	HZY809	Evelyn	Simon	Editar Eliminar
5	AXM255	Rodrigo	Pizarro	Editar Eliminar
6	JIS008	Maria	Montero	Editar Eliminar
7	OGX073	Andres	Perez	Editar Eliminar

El módulo de control presenta la información de horarios de entrada y salida de los usuarios que pertenecen a la lista mostrada en el módulo principal. Los datos mostrados son el número de placa, el nombre y apellido, la fecha y hora, y si el vehículo entro o salió del parqueadero.

Figura 16*Control de entrada y salida***Listado de Control**

#	Placa	Nombre	Apellido	Fecha y Hora	Entrada o Salida
1	OGX073	Andres	Perez	17 de Mayo de 2022 a las 16:10	Entrada
2	IOR460	Sebastian	Flores	17 de Mayo de 2022 a las 16:10	Entrada
3	IYK167	Sandra	Samper	17 de Mayo de 2022 a las 16:10	Entrada
4	HZY809	Evelyn	Simon	17 de Mayo de 2022 a las 16:10	Entrada
5	IHV210	Orlando	Diaz	17 de Mayo de 2022 a las 16:10	Entrada
6	AXM255	Rodrigo	Pizarro	17 de Mayo de 2022 a las 16:10	Entrada
7	JJS008	Maria	Montero	17 de Mayo de 2022 a las 16:10	Entrada

En el módulo de aviso se muestran los vehículos que entran o salen del parqueadero que no pertenecen a la lista del módulo principal, este es el listado para verificaciones de seguridad, se muestra el numero de la placa no aceptada y la fecha y hora que registro el ingreso.

Figura 17*Avisos de seguridad***Listado de Avisos**

#	Placa	Fecha y Hora
1	FHN147	17 de Mayo de 2022 a las 16:10
2	INJ921	17 de Mayo de 2022 a las 16:10
3	TSW128	17 de Mayo de 2022 a las 16:10

Seguridad

La seguridad para la aplicación web está basada en el sistema de autenticación de Django, este permite el control de usuarios, permisos y guardado de contraseña por medio de hashing. En la figura 18 se presenta la página de ingreso de usuarios, con esta se pueda ingresar a la interfaz principal, la de control, y la de avisos.

Figura 18*Página de ingreso a la plataforma*

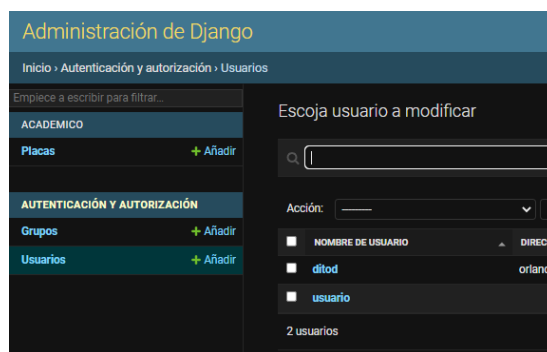
Gobernacion de Sucre

Usuario

Contraseña

Entrar

La creación de usuarios permitidos puede ser administrada desde el panel de administración de Django, como se muestra en la figura 19.

Figura 19*Panel de administración de usuario*

7. Resultados.

Los resultados están divididos en dos partes: controlados y no controlados. Los controlados toman en cuenta las 10 imágenes de prueba seleccionadas del conjunto de 93 imágenes de la etapa de entrenamiento. Los no controlados implican las pruebas in situ, que se realizaron en el parqueadero de la gobernación.

7.1 Controlados

Las imágenes captadas para los resultados controlados se tomaron en horarios de mañana, mediodía y tarde. En la tabla 6 se presentan los resultados de las 10 imágenes de prueba, en esta se observan las diferencias entre la placa real y la placa leída. El porcentaje de acierto es calculado dividiendo los caracteres verdaderos entre los 6 caracteres totales de la placa.

Para las imágenes utilizadas el porcentaje de detección es de 100% y el de reconocimiento es de 91,9%, la duración promedio de una inferencia es de 1,2 segundos.

Tabla 6

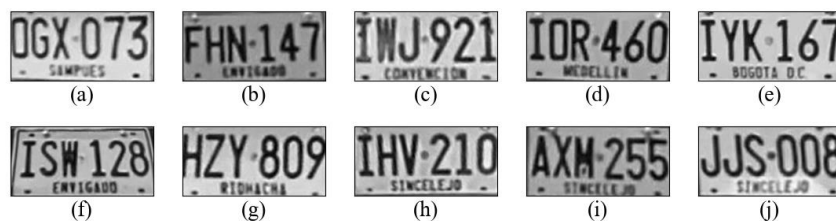
Resultados de inferencia controlada

Placa real	Placa leída	Porcentaje	Duración (s)
IWJ921	IHJ921	0,84	1,34
IYK167	IYK167	1	1,16
ISW128	TSW128	0,84	1,19
OGX073	OGX073	1	1,21
IOR460	TOR460	1	1,23
FHN147	FHN147	1	1,18
HZY809	HZY809	1	1,18
JJS008	JJS008	1	1,12
AXM255	AXH255	0,84	1,15
IHV210	THY210	0,67	1,18
Promedios		91,90%	1,19

En la figura 20 se presentan las imágenes de las placas preprocesadas, están correctamente alineadas y con texto horizontal. Los caracteres erróneos son las letras I, V, M y W, estos casos negativos son explicados por la similitud entre letras y la rareza de sus morfologías. Los caracteres W y M de la placas (c) y (i), demuestran poca separación en sus líneas internas por lo que se perciben demasiado rellenos. En la similitud de letras, se tienen igualdades I-T, W-H-M y V-Y, lo cual implica resultados negativos en las 4 placas que contienen errores.

Figura 20

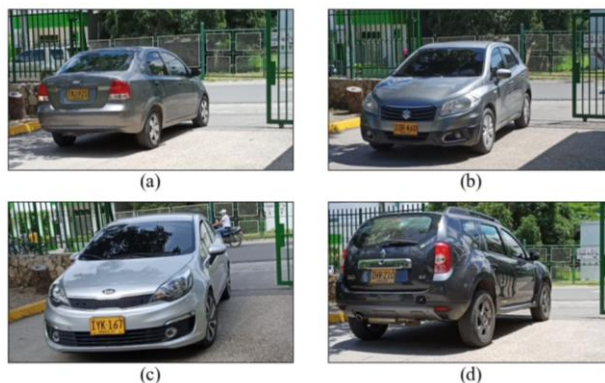
Imágenes después del preprocesamiento resultado controlado



La figura 21 muestra las imágenes de entrada a la inferencia, estas son claras, con placas legibles, fondos constantes y no están afectadas por difuminaciones (blur) o la velocidad de obturación de la cámara (shutter speed).

Figura 21

Imágenes de entrada resultado controlado



7.2 No controlados

Para los resultados no controlados se tienen 10 imágenes tomadas in situ, los resultados se presentan en la tabla 7, el porcentaje de detección es de 100% y el de reconocimiento de 58,33% y la duración promedio de inferencia es de 1,2 segundos. Si no se incluyen las placas OGX062, la cual físicamente no es aceptable de manera legal y la placa N/A que no es descriptible al ojo humano el porcentaje de reconocimiento aumenta a 73%.

Tabla 7

Resultados de inferencia no controlada

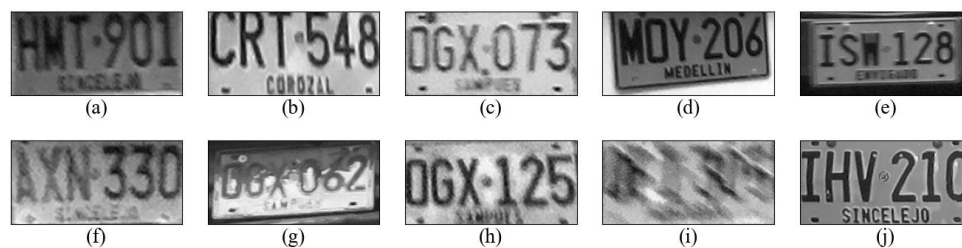
Placa real	Placa leída	Porcentaje	Duración (s)
HMT901	HAT7	33,33%	1,63
CRT548	UNI040	0,00%	1,14
OGX073	OGK073	83,33%	1,29
MOY206	MOY208	83,33%	1,25
ISW128	ISW126	83,33%	1,35
AXN330	AXN330	100,00%	1,31
OGX062		0,00%	1,25

OGX125	OGX125	100,00%	1,30
N/A		0,00%	1,36
IHV210	IHV210	100,00%	1,25
Promedios		58,33%	1,32

En la figura 22 se presentan las imágenes preprocesadas, además de los errores por similitud de letras y rarez morfológica, se tienen nuevos casos causados por características no esperadas al momento de la prueba en lugar. Hay varias razones para la disminución del porcentaje de éxito, en la placa (a) se observa difuminación alta en comparación con los recortes controlados, esto hace que los bordes de las letras sean más difíciles de registrar. La placa (b) cuenta con una característica llamada sobre recorte, por lo que se observa que la parte superior de las letras esta por fuera del área de recorte del preprocesamiento, al ingresar el resultado a Tesseract este no responde de buena manera al recorte. La placa (g) cuenta con una mala segmentación causada por alguna pintura o decoloración observada en la imagen de entrada en el área de la placa. La placa (i) cuenta con una placa no legible en la entrada.

Figura 22

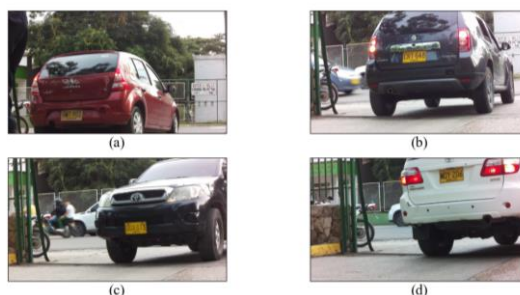
Imágenes después del preprocesamiento resultado no controlado



La figura 23 muestra las imágenes de entrada con resultados apropiados, estas son claras, con placas legibles, no están afectadas por difuminaciones, la velocidad de obturación, sobre recortes o placas no legales.

Figura 23

Imágenes de entrada no controlado con resultados apropiados



Al contrario, en la figura 24 se presentan las imágenes de entrada con resultados no apropiados, y en estas se muestran la placa (a) con decoloracion y la placa (b) con problemas de velocidad de obturacion.

Figura 24

Imágenes de entrada no controlado con resultados negativos



8. Análisis e impacto.

La implementación de un sistema ALPR en el parqueadero de la Gobernación de Sucre, es un trabajo que utiliza la inteligencia artificial para reconocer las placas vehiculares y definir si pertenecen a un listado de placas autorizadas. La entrada al sistema es una fotografía tomada al vehículo y en la salida se obtiene la placa en su formato alfanumérico para guardarla en la base de datos en el servidor del sistema, la información será mostrada en una interfaz tipo web app, que podrá ser navegada por el administrador y los usuarios del sistema.

Los usuarios esperados son el personal de vigilancia y la jefatura de la subsecretaria de recursos físicos, que intercambian datos con el dispositivo IoT donde se implementa el sistema

ALPR, la comunicación de información está dividida en los datos insertables y los datos visibles: los insertables serán el número de placa y la información personal de los usuarios del parqueadero autorizados, los visibles serán la placa captada, las actualizaciones en tiempo real con fechas y horas, y avisos de automóviles no permitidos.

En el aspecto operativo el sistema aumenta la seguridad del parqueadero, brinda a los vigilantes una herramienta para administrar los vehículos que ingresan y salen, y responsabiliza al personal para no aceptar la entrada de vehículos no autorizados. Comparado con los sistemas comerciales, en lo presupuestal, el sistema brinda una reducción de hasta 5 veces el precio de sistemas comerciales de empresas como HKVision y Tattile.

Para el análisis de calidad, se realizan 6 pruebas que determinan la solvencia y el desempeño del sistema construido. La prueba unitaria resulta en que los algoritmos individuales del sistema cumplen con la función esperada, la prueba funcional describe que las etapas de detección, preprocesamiento y reconocimiento dan los resultados esperados, la calidad del sistema tiene alta dependencia del lugar de ejecución por lo que su interoperabilidad es baja. En el aspecto de desempeño, se tiene una velocidad aceptable de procesamiento si se tiene en cuenta las restricciones del hardware, la precisión de detección es adecuada y la de reconocimiento es dependiente de los caracteres de la fotografía. La fiabilidad no pudo ser medida, ya que esta depende de un tiempo de ejecución mayor del sistema. Finalmente se tiene un sistema con baja autonomía ya que este obliga a la construcción de un tejido eléctrico y de telecomunicaciones.

Para la implementación final, el sistema deberá cumplir con la obligación legal de proteger las leyes de la protección de datos personales, con objetivo de que la información del sistema no se use ilegalmente.

9. Conclusiones.

La visión por computador es un campo de la inteligencia artificial, que a través de cámaras y la captura de fotografías o videos, los computadores pueden extraer información para tomar acciones específicas. Algunas tareas de la visión artificial son la clasificación de imágenes, la detección de objetos, el seguimiento de objetos, entre otros.

El trabajo está dividido en 2 etapas principales, el reconocimiento de la placa y el sistema informático. El reconocimiento se divide en 3 subetapas: la detección, el preprocesamiento y el reconocimiento de caracteres (OCR). La detección implica el encontrar instancias de objetos de una clase específica dentro de una imagen, lo que en el sistema sería encontrar las coordenadas que describen la ubicación de la placa en la fotografía del automóvil. Esta tarea fue realizada utilizando el método de aprendizaje de máquina llamado deep learning de tipo supervisado, el entrenamiento fue realizado en el API de detección de objetos (ODAPI) del framework TensorFlow, utilizando un modelo de tipo SSD (Single-Shot Detector) el cual, a partir del etiquetado de las imágenes de entrenamiento, permite el entrenamiento del modelo en la plataforma Colab.

El preprocesamiento es una etapa intermedia que tiene el fin de transformar la perspectiva de la placa en casos de que esta sufra de ángulos de inclinación, rotaciones o profundidades no apropiadas para el OCR. Esta transformación requiere la detección de las 4 esquinas de la placa, para que desde la transformación de perspectiva matricial se obtengan caracteres ubicados en una dirección horizontal y con un tamaño constante. Para obtener estos puntos se utilizan métodos de segmentación, operaciones morfológicas, detectores de contorno y detectores de esquina.

Para el reconocimiento de caracteres se utiliza el motor de código libre TesseractOCR como un sistema de caja negra, con una caracterización de su funcionamiento interno

conceptualizada en la combinación de algoritmos basados en morfología y una red neuronal de tipo LSTM. La API de Tesseract tiene procedimientos para la lectura, la elección de los caracteres reconocibles y la selección del área de interés. La imagen de la placa se divide a la mitad para separar las letras y los números de la placa, luego se obtienen los caracteres de manera separada y al final se vuelven a unir. El sistema de informático permite a los usuarios realizar en control de entrada, ingresar la información de las placas autorizadas y obtener datos de ingresos no autorizados.

Los resultados de la implementación son evaluados en el porcentaje de acierto de la detección y el OCR, y en la velocidad del reconocimiento. En los experimentos controlados con las imágenes de entrenamiento y prueba, se obtiene un 100% de acierto en la detección de la placa y un 91,9% en el reconocimiento de caracteres. Para los experimentos no controlados realizados con el sistema construido en el parqueadero, se obtiene un porcentaje de acierto del 100% en la detección y un 58,33% en el OCR.

10. Recomendaciones

Para la etapa de detección se recomienda aumentar a un valor mayor de 300 las imágenes de entrenamiento y optimizar la cuantización del modelo TFLite, además la detección mejoraría si la toma de imágenes fuera con la misma cámara tanto en el entrenamiento como en la inferencia. El preprocesamiento requiere una reducción de las funciones usadas y un aumento de la velocidad para encontrar las esquinas. Para mejorar el OCR se recomienda usar un modelo propio a partir de la segmentación de las imágenes utilizadas en el entrenamiento de detección, con una búsqueda de la disminución del error causado por caracteres similares y letras con morfologías específicas. Para el sistema de información se recomienda

construir un mejor modelo de entidades y referencias en la base de datos y agregar filtros de información en la web app.

De manera general se recomienda mejorar la velocidad hasta llegar a un sistema de tiempo real, que no requiera sensores externos para tomar las capturas de las placas. En el hardware se recomienda utilizar una versión más avanzada de la Raspberry Pi, como por ejemplo la Pi 4B y una cámara de 8MP. Finalmente, se recomienda utilizar un sistema de energía solar fotovoltaica para alimentar el hardware del ALPR y tener un sistema completamente autónomo.

11. Presupuesto económico.

El presupuesto está constituido por los costos de materiales y los costos de mano de obra, en la tabla 8 se encuentran los costos de materiales. Para el prototipo se tiene una fuente de financiación propia que incluye el sistema embebido y la cámara.

Tabla 8

Costo de Materiales

Descripción	Unidad	Valor Unitario	Cantidad	Valor Parcial
Raspberry Pi 3b+	GL	\$ 147.120	1	\$ 147.120
RPi Camera (B), Adjustable-Focus	GL	\$ 81.000	1	\$ 81.000
Adaptador 5V RPi	GL	\$ 25.000	1	\$ 42.000
Anclaje metálico	GL	\$ 25.000	1	\$ 25.000
Cable UTP Cat 6	Mts	\$ 3.500	10	\$ 35.000
Cable de cobre	Mts	\$ 4.000	5	\$ 20.000
Caja impermeable	GL	\$ 20.000	1	\$ 20.000
			Total	\$ 370.120

El costo total de los materiales para la implementación final es de \$370.120 pesos, con un costo de mano de obra esperado de \$200.000. No hay costos requeridos para el servidor del sistema informático, ya que se espera que se realice en los servidores de la oficina de Sistemas de la Gobernación de Sucre.

Referencias

- Django Software Foundation . (s.f.). *FAQ: General*. Obtenido de <https://docs.djangoproject.com/en/dev/faq/general/>
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Devin, M. (2016). TensorFlow: A System for Large-Scale. Savannah.
- Announcing Tesseract OCR. (2006). Obtenido de Google code updates: <https://web.archive.org/web/20061026075310/http://google-code-updates.blogspot.com/2006/08/announcing-tesseract-ocr.html>
- ANPR International Ltd. (s.f.). *History of ANPR*. Obtenido de ANPR International: <http://www.anpr-international.com/history-of-anpr/>
- Baggio, D. L., Emami, S., Millán, D., Levgen, K., Mahmood, N., Saragih, J., & Shilkrot, R. (2012). *Mastering OpenCV with Practical Computer*. Birmingham: Packt Publishing Ltd.
- Bhuyan, M. K. (2019). *Computer vision and image processing : fundamentals and applications*. Boca Raton: CRC Press/Taylor & Francis Group.
- Cruz, J., & Rodriguez, W. (2021). *Desarrollo e implementación de un sistema automático de reconocimiento de placas vehiculares*.
- Ministerio de Transporte. (s.f.). Ficha tecnica MT 001: Placa unica nacional. Bogota D.C., Colombia.
- Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2022). Realtime Computer Vision with OpenCV: Mobile computer-vision technology will soon become as ubiquitous as touch interfaces.
- Raspberry Pi Foundation. (s.f.). *About us*. Obtenido de Raspberry Pi Foundation: <https://www.raspberrypi.org/about/>

SQLite. (s.f.). *About SQLite*. Obtenido de SQLite: <https://sqlite.org/about.html>

Yacob, N., & Alkahtani, A. (2021). *License plate recognition for campus auto-gate system*.