



**INSTITUTO POLITÉCNICO NACIONAL**



**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y  
ELÉCTRICA. SECCIÓN DE ESTUDIOS DE POSGRADO E  
INVESTIGACIÓN. UNIDAD ZACATENCO.**

**“DISEÑO DE UN SISTEMA DE  
ASISTENCIA VEHICULAR  
AVANZADO”**

**T E S I S**  
QUE PARA OBTENER EL GRADO ACADÉMICO DE  
MAESTRO EN CIENCIAS DE INGENIERÍA EN  
TELECOMUNICACIONES PRESENTA:  
**ING. RAFAEL ASRAEL MIRANDA LEÓN**

**DIRECTORES DE TESIS: DR. MARCO ANTONIO ACEVEDO  
MOSQUEDA  
DRA. MARIA ELENA ACEVEDO MOSQUEDA**

**MÉXICO, D.F. 7 DE ENERO 2019**

---

---

## Resumen

El presente trabajo se centra en diseñar un sistema de visión por computadora el cual pueda dar apoyo en la conducción a choferes y policías, con el fin de facilitar un tránsito seguro en las vialidades. El sistema consiste en dos partes fundamentales: detectar carriles vehiculares y reconocer el número de placas vehiculares. El sistema está implementado mediante una computadora con la ayuda de programas que utilizan las librerías de visión por computadora "OpenCV" y la librería "Tesseract-OCR" para poder realizar dichas tareas.

Se explica que es un sistema para la asistencia vehicular, así como las tecnologías existentes y el estado del arte para la detección de carriles y reconocimiento del número de matrículas. También se describe del software, que se usó en el trabajo.

La detección de carriles consiste en indicar al conductor que se encuentra dentro de un segmento confinado, el cual está diseñado para que solo circule un vehículo dentro de una fila de vehículos, con la ayuda de indicadores visuales como el uso de LEDS que indiquen la detección del carril completo (línea izquierda y línea derecha) o mostrando en una pantalla el entorno tomado mediante una cámara frontal señalando la presencia de carriles o como alternativa usar un indicador auditivo. Gracias a un sistema de detección de carriles se reduce la tendencia de choques, que generalmente son causados por no poner atención a las indicaciones vehiculares como son los carriles. También el detector de carriles puede ser usado en vehículos inteligentes.

El reconocimiento del número de placas vehiculares consiste en reconocer el número de matrícula vehicular por medio de una video cámara, las cuales son el registro que usan los vehículos para su identificación y circulación legal en todo el territorio nacional, dentro de sus aplicaciones se encuentra el control vial de los vehículos, el robo de vehículos, etc. Además de detectar la placa vehicular se requiere asilar dicha placa y extraer sus caracteres los cuales son una combinación de números y letras y reconocer cada carácter de la matrícula, con el fin de mostrar el número de matrícula en pantalla o almacenarla para usarla en alguna futura aplicación como en las mencionadas anteriormente.

Se presenta el desarrollo realizado de los algoritmos de detección de carriles y el reconocimiento del número de matrículas vehiculares, detallando su elaboración paso por paso de cada uno.

Así como las pruebas realizadas a los programas en diferentes lugares, horas del día e iluminación. Con la finalidad de observar la precisión y las debilidades que tengan

---

---

los programas, determinando si pueden ser usados dentro de un sistema para la asistencia vehicular y proponer futuras mejoras.

---

---

## **Abstract**

The present work focuses on designing a computer vision system which could give some support to drivers. The system consists of two fundamental parts: lane detection and number plate recognition. The system uses computer algorithms with the support of computer vision libraries "OpenCV" and "Tesseract-OCR" to perform these tasks.

It is explained in detail what is a vehicular assistance system as well as the existing technologies and state of the art for lane detection and license plate number recognition. It also describes software, used at work.

The lane detection indicates to a driver who is confined in a specific lane, this lane is designed so that only one vehicle circulates within a row of vehicles, using LEDs that the entire track has been detected (left line and right line) or showing on a screen the environment took by a front camera displaying lanes marked or alternatively using a hearing indicator. Collisions are reduced thanks to a lane detection system.

The number plate recognition consists of detecting a license plate through a video camera. A license plate is the registration number used by vehicles for identification and legal circulation throughout the national territory, within its applications are the road control of vehicles, car theft, etc. In addition to detecting a license plate, it is necessary to isolate the license plate and recognize its number which is a combination of numbers and letters, recognize each character to show the license plate number on screen or store it to use in any future application such as those mentioned above.

As well as tests carried out on the programs in different places, daytime, and lighting. With the purpose of observing the accuracy and weaknesses that have both programs determining if they can be used for a vehicular assistance system and propose future improvements.

---

---

## **AGRADECIMIENTOS**

Al Instituto Politécnico Nacional (IPN) por mi formación académica en mis estudios de posgrado.

Al Dr. Marco Antonio Acevedo Mosqueda y a la Dra. María Elena Acevedo Mosqueda, por su asesoría, conocimiento, apoyo y dedicación brindada durante la elaboración el desarrollo del presente trabajo.

A todos los profesores del posgrado que me impartieron las materias necesarias para el desarrollo de mi trabajo.

A mi madre y a mis hermanos, por su apoyo durante el desarrollo de este trabajo.

A mis compañeros de la sección de procesamiento digital de señales en el área del posgrado en telecomunicaciones, por su apoyo y consejos brindados durante el desarrollo del trabajo presente.

---

---

## CONTENIDO

<b>CAPITULO 1: ESTADO DEL ARTE</b> .....	16
Introducción .....	16
1. Conceptos generales .....	17
1.1. Visión por computadora .....	17
1.2. Procesamiento digital de señales .....	21
1.3. Procesamiento digital de imágenes .....	21
1.4. Imagen digital .....	21
1.5. Video Digital .....	24
1.6. Software empleado en el trabajo .....	24
1.7. Tecnologías existentes .....	25
1.8. Estado del conocimiento en la detección de carriles vehiculares .....	27
1.8.1. Carriles vehiculares .....	27
1.8.2. Anchura del carril vehicula .....	27
1.8.3. Color de las marcas del carril .....	28
1.8.4. Detección de carriles por visión por computadora .....	28
1.9. Estado del conocimiento en el reconocimiento de las placas vehiculares .....	35
1.9.1. Placas Vehiculares .....	35
1.9.2. Reconocimiento Óptico de Caracteres .....	36
1.9.3. Reconocimiento automático de matrícula .....	37
<b>CAPITULO 2: DETECCION DE CARRILES</b> .....	41
Introducción .....	41
2.1. Algoritmo desarrollado para la detección de carriles .....	41
2.1.1. Adquisición de la imagen .....	43
2.1.2. Selección de región de interés .....	44
2.1.3. Cambio de espacio de color RGB a escala de grises .....	46
2.1.4. Filtrado de la imagen .....	47
2.1.5. Segmentación de la imagen .....	48
2.1.6. Detección de bordes .....	50
2.1.7. Transformada Hough .....	52
2.1.8. Selección de las líneas que no tiendan a ser horizontales .....	54
2.1.9. Clasificación de líneas (DER/IZQ) .....	54
2.1.10. Aplicación de método de mínimos cuadrados .....	56
2.1.11. Ecuación de la recta para formar una sola línea .....	57
2.1.12. Unir las líneas a la imagen original .....	59
<b>CAPITULO 3: RECONOCIMIENTO DEL NÚMERO DE MATRÍCULAS</b> .....	60
Introducción .....	60
3.1. Algoritmo desarrollado para el reconocimiento del número de matrícula .....	60
3.2. Detección de la matricula vehicular .....	60
3.2.1. Adquisición de la imagen .....	61
3.2.2. Procesamiento de la imagen .....	62

---

---

3.2.3. Umbralización de la imagen .....	64
3.2.4. Operación morfológica de clausura .....	66
3.2.5. Detección de contornos .....	67
3.2.6. Algoritmo de relleno .....	68
3.2.7. Extracción de la matrícula vehicular .....	69
3.3. Extracción del número de la matrícula .....	71
3.3.1. Detección de contornos .....	71
3.3.2. Cortar el número de la matrícula .....	72
3.4. Reconocimiento óptico del número de matrícula .....	73
3.4.1. Umbralización del texto .....	75
3.4.2. Aplicando Tesseract-OCR para el reconocimiento del número de matrícula .....	75
<b>C A P I T U L O 4: PRUEBAS Y RESULTADOS .....</b>	<b>77</b>
Introducción .....	77
4.1. Pruebas al detector de carriles .....	77
4.2. Pruebas al reconocedor del número de matrícula .....	84
4.3. Limitaciones del trabajo.....	86
<b>C A P I T U L O 5: CONCLUSIONES .....</b>	<b>87</b>
Introducción .....	87
5.1. Conclusiones .....	87
<b>REFERENCIAS .....</b>	<b>89</b>
<b>ANEXOS .....</b>	<b>92</b>

---

---

## Lista de figuras

### Capítulo 1

Figura 1.1. Diagrama en bloques de un sistema de visión por computadora -----	17
Figura 1.2. Representación de una imagen digital en forma de una matriz $m \times n$ ---	21
Figura 1.3. Imagen monocromática con sus respectivos ejes de coordenadas $x$ y $y$ ---	22
Figura 1.4. Imagen en modo monocromático -----	23
Figura 1.5. Imagen en modo escala de grises -----	23
Figura 1.6. Imagen en modo RGB -----	24
Figura 1.7. Detectando un vehículo que se encuentra delante de otro vehículo por medio del radar incorporado -----	25
Figura 1.8. Detectando las líneas de carril por medio de una video cámara -----	26
Figura 1.9. Implementación de la tecnología ARNP en patrullas policíacas -----	26
Figura 1.10. Diagrama en bloques de sistema de advertencia de salida de carril ---	29
Figura 1.11. Algunos ejemplos de problemas a afrontar en la detección de carriles (a) Variaciones de iluminación con sombras (b) Presencia de lluvia (c) Carretera en la noche (d) Obstrucción de las líneas de carril debido a la presencia de un vehículo cercano -----	30
Figura 1.12. Diagrama en bloques de sistema Reconocimiento automático de matrícula -----	38

### Capítulo 2

Figura 2.1. Diagrama en bloques de la detección de carriles -----	42
Figura 2.2. Adquisición de la imagen (video tomado de la plataforma YouTube) ----	43
Figura 2.3. Proceso de la transformación perspectiva de una imagen: colocar la imagen plana en el espacio 3D y la proyección perspectiva de la imagen en el espacio -----	45
Figura 2.4. Aplicando la transformada perspectiva a la imagen para obtener la región de interés donde se encuentra el carril frontal -----	45
Figura 2.5. Resultado de aplicar la transformación inversa a la perspectiva producida por la cámara, utilizando la nueva perspectiva de la imagen se puede realizar mediciones u obtener parámetros que en el caso de la imagen original podrían ser más difíciles de obtener -----	45
Figura 2.6. Cambio de espacio de color RGB a escala de grises en la imagen donde se encuentra los carriles -----	47
Figura 2.7. Aplicando filtro gaussiano en la imagen para reducir el ruido -----	47
Figura 2.8. A pesar de utilizar en las dos imágenes la segmentación en el círculo rojo y en los vasos sanguíneos en sus respectivas imágenes no utilizan el mismo	

---



---

método: en el primer caso se utiliza la segmentación por color y el otro caso se utiliza umbralización a partir del histograma. Además de que el segmentar el círculo rojo es más fácil que segmentar vasos sanguíneos ya que estos tienen mayor complejidad de forma y se encuentran en un fondo complejo -----	48
Figura 2.9. Escala de luminosidad donde el color amarillo se aproxima a los niveles de intensidad del color blanco -----	49
Figura 2.10. Resultado de aplicar la umbralización binaria inversa donde los carriles (objetos de interés) son de valor 255 mientras lo demás (el fondo) es cero -----	50
Figura 2.11. Resultado de aplicar el método Canny para detección de bordes a la imagen previamente segmentada -----	51
Figura 2.12. Resultado de aplicar la transformada Hough probabilística para detectar las líneas del carril -----	53
Figura 2.13. Diagrama de flujo de la clasificación de líneas -----	55
Figura 2.14. División de las líneas para determinar el sentido de las líneas del carril - -----	56
Figura 2.15. Obtención de las líneas con sus respectivas coordenadas -----	58
Figura 2.16. Resultado final de la detección de carriles -----	59

### Capítulo 3

Figura 3.1. Diagrama del reconocimiento del número de matrícula -----	60
Figura 3.2. Diagrama de bloques del procedimiento para poder detectar la matrícula - -----	61
Figura 3.3. Fotografía tomada desde un celular Motorola G -----	61
Figura 3.4. Detección de bordes mediante el operador de Sobel utilizando la primera deriva horizontal -----	63
Figura 3.5. Resultado de umbralizar la imagen -----	64
Figura 3.6. Resultado de aplicar la operación de cierre -----	66
Figura 3.7. Mascara de la imagen donde solo se selecciona la región de la matrícula vehicular -----	69
Figura 3.8. Resultado de la detección de la matrícula dentro de la imagen -----	69
Figura 3.9. Ejemplo de una transformación afines -----	70
Figura 3.10. Extracción de la matrícula vehicular -----	70
Figura 3.11. Diagrama de bloques de la extracción del número de la placa vehicular - -----	71
Figura 3.12. Pasos para detectar los caracteres de la matrícula: a) detección de contornos (en azul). b) encerrando los contornos dentro de rectángulos (color verde).c) selección de los rectángulos que contengan a los caracteres (color naranja) -----	72
Figura 3.13. Detección del número de la matrícula -----	73
Figura 3.14. Imagen del número de la matrícula -----	73
Figura 3.15. Diagrama de bloques de la detección del número de matrícula -----	74

---

---

Figura 3.16. Imagen binarizada del número de matrícula para facilitar la detección y reconocimientos de los caracteres .....	75
Figura 3.17. Resultado final del reconocimiento del número de matrícula .....	76

## Capítulo 4

Figura 4.1. Primera prueba realizada .....	78
Figura 4.2. Precisión de la detección de carriles .....	79
Figura 4.3. Segunda prueba realizada .....	80
Figura 4.4. Dificultad de detectar la línea del carril izquierdo debido a la curvatura de la carretera. ....	80
Figura 4.5. Precisión de la detección de carriles. ....	81
Figura 4.6. Fotografías de la colocación de la video cámara .....	82
Figura 4.7. Cámara web modelo: Logitech B525 HD WEBCAM .....	82
Figura 4.8. Tercera prueba realizada .....	83
Figura 4.9. Precisión de la detección de carriles .....	83
Figura 4.10. Ejemplo de captura de una matrícula .....	85

---

---

## LISTA DE TABLAS

### Capítulo 1

Tabla 1.1. Métodos para detección y rastreo de carriles -----	30
Tabla 1.2. Ventajas y desventajas del proyecto de tesis con respecto a las referencias consultadas-----	32
Tabla 1.3. Métodos para reconocimiento de placas vehiculares -----	38
Tabla 1.4. Ventajas y desventajas del proyecto de tesis con respecto a las referencias consultadas-----	39

### Capítulo 3

Tabla 3.1. Resoluciones de las cámaras de celulares -----	62
---	----

### Capítulo 4

Tabla 4.1. Evaluación de la detección de carriles en un minuto -----	85
--	----

### Anexos

Tabla. Pruebas realizadas al reconocedor del número de matrícula -----	92
--	----

---

---

## **Justificación**

En los últimos años en las grandes ciudades alrededor del mundo se han incrementado el número de vehículos que las recorren. Este fenómeno ha conducido al incremento del número de carreteras por lo que la tendencia de que exista un accidente vial, robos vehiculares han aumentado además de la necesidad de tener un control dentro de las vialidades en cuestiones ambientales y de tráfico. Para reducir estos problemas se requiere de dispositivos electrónicos que puedan dar apoyo tanto a los conductores y policías, en cuestiones de asistencia vehicular, delitos, infracciones vehiculares, etc.

Con base al aumento de estos problemas y tener un control vial surge la necesidad de diseñar un sistema de asistencia vehicular avanzado que pueda ser usado tanto para conductores de vehículos privados, choferes de autobuses, policías y con posibilidades de poder implementar la idea en un vehículo autónomo. Donde el sistema cuente con un detector de carriles para dar asistencia al conductor de que está cambiando de carril reduciendo la tendencia de choques.

Finalmente, el sistema es capaz de reconocer el número de las matrículas vehiculares con propósitos de control vial y asistencia policiaca.

---

---

## **Objetivo**

Diseñar un sistema para la asistencia vehicular utilizando visión por computadora que permita la detección de las líneas vehiculares que se encuentran en el asfalto y reconozca el número de las matrículas de los vehículos.

## **Objetivos particulares**

- (1) Desarrollar un algoritmo para la detección de carriles.
- (2) Desarrollar un algoritmo para la detección de placas vehiculares, la extracción del número de las placas, así como el reconocimiento del mismo número.
- (3) Realizar pruebas en ambos algoritmos.

## CAPÍTULO 1

### ESTADO DEL ARTE

#### Introducción

Gracias al desarrollo humano han surgido grandes avances científicos, mejorando la calidad de vida del ser humano y facilitando la elaboración de muchas tareas cotidianas al menor tiempo posible. Debido a las grandes necesidades requeridas han surgido nuevas ramas de la ciencia e investigación, como es el caso de la rama de las ciencias de computación la cual continuamente se está actualizando constantemente e innovando con nuevos desarrollos tecnológicos.

Uno de esos grandes avances de la computación es el campo de la inteligencia artificial la cual es otorgar a las máquinas como son las computadoras, celulares, robots, etc., la capacidad de tomar decisiones y resolver los problemas sin la necesidad de la ayuda del ser humano. La visión por computadora es un campo de la inteligencia artificial enfocado en la extracción de información de las imágenes adquiridas por una cámara de video ofreciendo soluciones a problemas de la vida cotidiana.

Actualmente la visión por computadora está siendo usada en diferentes áreas laborales como es en el automovilismo, donde se han desarrollado sistemas para la asistencia vehicular apoyando a los conductores a tener un viaje más seguro al igual que se han desarrollado sistemas que permitan detectar un vehículo a partir de obtener la información de la placa del mismo vehículo.

Estos sistemas utilizan video cámaras con el fin de captar el medio que rodea al vehículo o las carreteras o calles y de esta forma obtener la información necesaria del entorno a analizar. Para analizar, extraer, procesar y desplegar resultados se requiere de un dispositivo que realice todos estos procesos como puede ser una computadora o un celular.

A pesar de la existencia de estos sistemas de video aún existen problemas al momento de ser utilizados debido a los problemas de iluminación y clima, de esta forma se considera que estos sistemas son de apoyo, pero no pueden aun desplazar por completo al ser humano.

En el presente capítulo, se describen los fundamentos teóricos necesarios para poder comprender todo el contexto del proyecto de tesis como es el software y hardware utilizado, que es la visión por computadora, el procesamiento de imágenes, un sistema para la asistencia vehicular y algunos dispositivos comerciales utilizados.

## 1. Conceptos generales

En el apartado siguiente se definen conceptos generales que serán útiles para la comprensión del presente trabajo.

### 1.1. Visión por computadora[1][2][3][4]

La visión es la ventana al mundo para la gran variedad de seres vivos en el planeta. Su función principal es reconocer y localizar los objetos del ambiente mediante el procesamiento de imágenes. La visión por computadora es el estudio de estos procesos, para lograr comprenderlos y replicarlos en máquinas diseñadas por el humano a partir del modelado matemático para poder generar programas computacionales que otorguen dichas capacidades.

A continuación, se muestra un diagrama con las etapas para desarrollar una aplicación de visión por computadora:

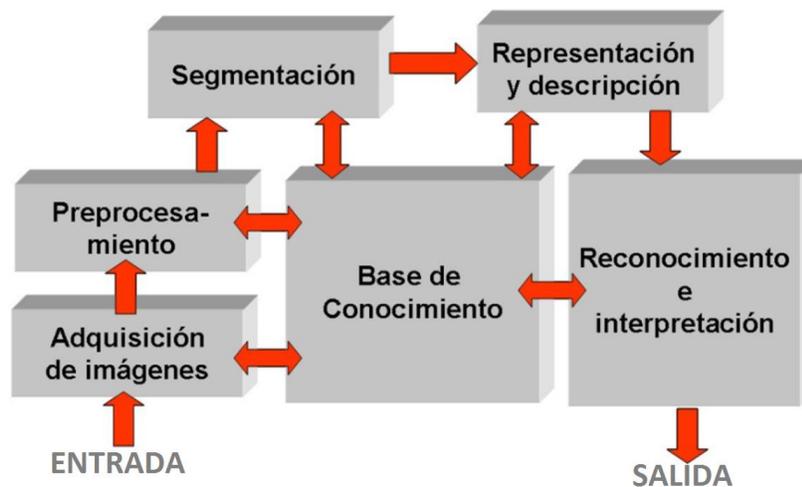


Figura 1.1. Diagrama en bloques de un sistema de visión por computadora.

Con base al diagrama en bloques de la figura 1.1. se describirán las etapas básicas de una aplicación de visión por computadora:

-Adquisición de imágenes: Simplemente es la etapa donde se adquirieren las imágenes ya digitalizadas.

-Preprocesamiento: Etapa donde la imagen es manipulada para tener un resultado más adecuado que el de la imagen original con el propósito de usarla para una aplicación en específico.

-Segmentación: Etapa que consiste en diferenciar el objeto u objetos de interés de una imagen con el fondo. Esta etapa puede ser muy compleja, aunque si se logra hacer una buena segmentación se logra un mejor reconocimiento.

-Representación y descripción: A partir de una imagen segmentada se obtiene un contorno de un objeto de interés gracias a esto se puede obtener un conjunto de parámetros caracterizados como son el perímetro, grado de circularidad, momentos de orden-n, etc.

Estos rasgos hacen posible la distinción de objetos mediante el uso reducido de propiedades.

-Reconocimiento e interpretación: El reconocimiento es el proceso en el que se asigna una etiqueta a un objeto basado en sus descriptores. La interpretación implica asignar significado a un conjunto de objetos reconocidos.

-Base de conocimiento: El conocimiento sobre un dominio del problema está codificado en un sistema de visión por computadora de procesamiento de imágenes como una base de datos de conocimiento. Este conocimiento puede ser tan simple como detallar las regiones de una imagen donde se sabe que se ubica información de interés, limitando así la búsqueda que ha de realizarse para hallar tal información. La base de conocimiento interactúa con todas las etapas previas mencionadas.

Cada aplicación que utilice visión por computadora puede basarse en este diagrama, aunque no siempre contará con todas las etapas que contiene el diagrama.

Configuración de un sistema de visión por computadora.

Una configuración de un sistema de visión por computadora dependerá de dos cuestiones:

-El propósito por el cual se elige la imagen o imágenes a procesar.

-Característica de la imagen de entrada.

Estos aspectos serán configurados dentro de un sistema de visión por computadora y el sistema deberá contener los siguientes dispositivos de manera general:

-Dispositivo de captación de imágenes: Todo dispositivo que permita capturar una imagen por medio de un sensor como puede ser una cámara fotográfica o un escáner.

-Tarjeta digitalizadora: Actualmente las cámaras incluyen estos dispositivos internamente.

-Computadora o unidad de procesamiento: Cualquier dispositivo que tenga gran poder de procesamiento y capacidad de almacenamiento.

-Software: Para que una aplicación de visión funcione requiere de un algoritmo de programación que cumpla con las tareas que debe ejecutar el sistema. Gracias al desarrollo de librerías especializadas al procesamiento digital de imágenes y visión por computadora, desarrollar una aplicación no requiere empezarla desde cero ya que por lo general contienen funciones que ya están establecidas para facilitar el trabajo.

-Dispositivo de salida: Desde una pantalla LCD hasta una impresora láser.

Importancia de la iluminación.

La fuente de luz toma un papel importante en todo sistema de visión por computadora ya que la luz natural interfiere en cualquier escena que sea captada por una video cámara y también a la escena se le puede inducir luz artificial para mejoramiento de la escena o por ausencia de luz. Cuando la luz no puede ser controlada como en el caso del sol a este escenario se le conoce entorno no controlado y en el caso cuando el escenario se encuentra aislado de una fuente de luz natural y se le induce una fuente de iluminación que se pueda controlar su iluminación se le conoce al escenario entorno controlado.

Costo/Beneficio de un sistema de visión.

El costo de un sistema de visión por computadora dependerá de la complejidad de hardware y software que se utilice para lograr una aplicación. Una consideración fundamental al momento de diseñar un sistema de visión por computadora es el dispositivo de captación de imágenes, como por ejemplo una video cámara, es que hay que tomar en cuenta la calidad y características de imagen que permite el dispositivo, considerando las capacidades de procesamiento del dispositivo de procesamiento y el objetivo de la aplicación.

El uso de visión por computadora puede causar un gasto computacional muy costoso en las unidades de procesamiento y la mayoría de las aplicaciones de visión por computadora deben ejecutarse en tiempo real, de tal manera que un simple cuadro (frame) a procesar debe estar dentro de los 30 a 40 milisegundos de tiempo para mostrar la imagen en un dispositivo, esta consideración puede causar problemas para un celular o un computador de placa reducida, en ocasiones es posible sacrificar la calidad por la cantidad de cuadros por segundo.

El uso de visión por computadora, dentro de aplicaciones tecnológicas proporciona beneficios:

- Gracias a la visión por computadora se pueden detectar fenómenos, características u objetos que el ojo humano no puede hacer debido a sus limitaciones.
- Permitir que robots puedan ser autónomos otorgándoles una cámara para que puedan ver.
- Realizar varias tareas que pueden ser tediosas o peligrosas para el humano.

No obstante, así como usar visión por computadora proporciona beneficios también puede haber dificultades al momento de utilizarla.

Al momento de implementar un sistema de visión por computadora es inevitable la presencia de inconvenientes que afecten el rendimiento de la aplicación del sistema.

En ambiente controlado se presentarán menos inconvenientes en las aplicaciones de visión por computadora mientras en un ambiente no controlado el cual además de tener una fuente de iluminación no controlada, la presencia de fenómenos naturales y la presencia de objetos que no son de interés que salen y entran a la región interés afectan al escenario donde se está haciendo la captura para la aplicación.

Aplicaciones.

- Control de calidad: Inspección de productos, identificación de piezas, etiquetados, inspección de circuitos impresos, control de calidad de alimentos, etc.
- Robótica: Control de soldaduras, vehículos no tripulados, etc.
- Biomédicas: Análisis de imágenes de microscopía, resonancias magnéticas, tomografías, genoma humano, etc.
- Astronomía: Exploración en el espacio.

- Reconocimiento de caracteres: Control de cheques, inspección de textos, etc.
- Control de tráfico: Tráfico diario, detector de matrículas vehiculares, etc.
- Agricultura: Interpretación de fotografías aéreas, control de plantaciones, etc.
- Militares: Seguimiento de objetos, vigilancia por satélites, etc.

### 1.2. Procesamiento digital de señales[5]

El procesamiento digital de señales trata de la representación, transformación y manipulación de señales y de su importancia que contienen al momento de usar las para un uso específico.

### 1.3. Procesamiento digital de imágenes[6]

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

### 1.4. Imagen digital[1][7][8][9]

Es un tipo de imagen que puede ser manipulada por medio de un dispositivo electrónico como es una computadora. El proceso que permite tener una imagen digital es la digitalización que convierte una imagen analógica a digital.

Esta imagen se caracteriza por números que se encuentran dentro de una matriz gracias a los procesos de cuantificación y muestreo. La matriz que representa a la imagen de dimensiones  $m \times n$  como se muestra en la figura 1.2.

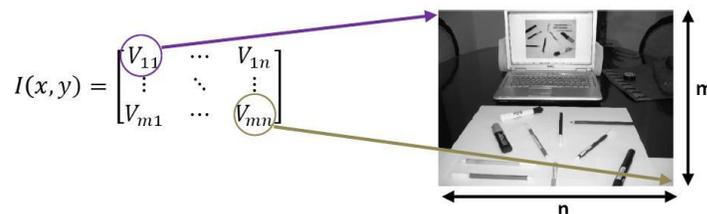


Figura 1.2. Representación de una imagen digital en forma de una matriz  $m \times n$ .

Dentro del concepto de imagen digital se encuentra los siguientes conceptos:

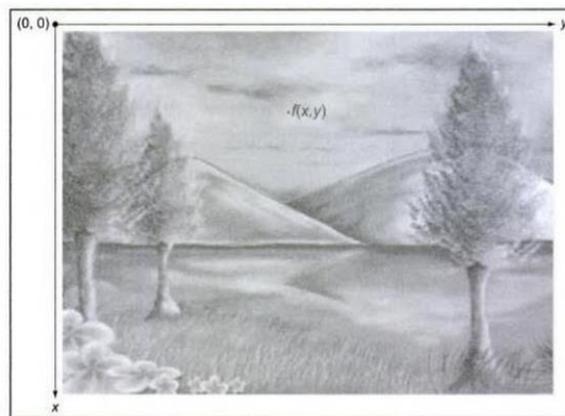
-El píxel es la unidad más pequeña que se encuentra en una imagen, la combinación de varios pixeles forma una imagen y entre mayor sea la cantidad de ellos más definida será la imagen.

-El tamaño de una imagen esta dado en pixeles y este da las dimensiones de una imagen (ancho y alto).

-La resolución es el producto del ancho y alto de una imagen digital dada en pixeles, gracias a la resolución se puede reproducir fielmente los detalles de una imagen.

-La profundidad de color es el número de bits utilizados para describir el color de cada píxel de la imagen, entre más grande sea la profundidad del color de una imagen mayor cantidad de colores tendrá esa imagen.

-Cualquier imagen monocromática, puede ser representada por una función bidimensional de intensidad de luz  $f(x,y)$  donde  $x$  y  $y$  son coordenadas espaciales y el valor de  $x$  en cualquier punto de  $(x,y)$  es el nivel de gris o de brillo de una imagen en ese punto. La convención de ejes el origen de la imagen se encuentra en la esquina superior izquierda y la línea horizontal es considerada el eje  $y$  mientras la vertical es considerado el eje  $x$ , en la figura 1.3 se muestra una figura monocromática con sus respectivos ejes de coordenadas.



---

Figura 1.3. Imagen monocromática con sus respectivos ejes de coordenadas  $x$  y  $y$ .

-La intensidad de una imagen es la cantidad de luz que hay presente.

-El brillo es la intensidad de luz relativa de una parte de una imagen, típicamente de un solo píxel.

-El contraste es la variación de intensidades que puede tener una imagen en sus distintas regiones.

-Modos de color.

Un modo de color es un sistema de coordenadas que describen el color de cada píxel de la imagen usando valores numéricos.

Entre los modos de color más conocidos están:

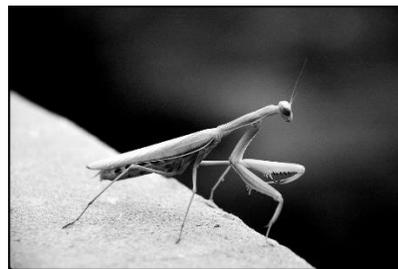
- Modo monocromático: La figura 1.4 contiene una profundidad de color de 1 bit por lo que la imagen está compuesta de pixeles blancos y negros.



---

Figura 1.4. Imagen en modo monocromático.

- Modo escala de grises: En este modo solo maneja un color el negro para poder trabar con imágenes monocromáticas de 256 tonos de grises entre el blanco y negro, como en la figura 1.5 se muestra una imagen en este modo de color.



---

Figura 1.5. Imagen en modo escala de grises.

- Modo RGB: En este modo, cada color se forma por combinación de tres colores, estos colores corresponden a un color primario: rojo, verde y azul. A cada píxel se le asigna un valor de intensidad que oscila de 0 al 255. La figura 1.6 es un ejemplo de una imagen RGB, presentando una gran cantidad de colores dentro de la imagen.



Figura 1.6. Imagen en modo RGB.

### 1.5. Video Digital[10][11]

Un video se considera una secuencia de imágenes fijas tomadas sucesivamente en un tiempo determinado con el fin de dar la ilusión de movimiento en una imagen al ojo humano.

Un fotograma o cuadro es una imagen que forma parte de un conjunto de imágenes sucesivas que son parte de una animación o video. La tasa de fotogramas por segundo dependerá de la aplicación a usar, aunque lo más común usar entre 15 a 30 cps.

### 1.6. Software empleado en el trabajo

OpenCV[12][13].

Es una librería abierta de visión por computadora, la librería está escrita en c y c++ para los sistemas operativos Linux, Windows y Mac OS X. Esta librería está enfocada para tener aplicaciones eficientes y en tiempo real. A través de los años OpenCV se ha utilizado para el desarrollo de trabajos de investigación y aplicaciones para la sociedad.

Qt creator[13][14].

Es un entorno de desarrollo integrado IDE multiplataforma, desarrollado para la elaboración de aplicaciones en los sistemas operativos Linux, Windows y Mac OS.

Existen versiones de pago y también una versión gratuita la cual es muy útil para el desarrollo de muchas aplicaciones. Sus principales características que resaltar son:

-Contiene un avanzado editor de código c++, que permite desplazarse rápidamente dentro del código, resaltar la sintaxis y autocompletar el código.

-Tiene la facilidad de ejecutar el código instrucción por instrucción, revisar puntos de interrupción y la posibilidad de detener la ejecución del programa.

-Permite desarrollar aplicaciones desde la consola o por medio de una interfaz gráfica de usuario (GUI por sus siglas en inglés).

### 1.7. Tecnologías existentes

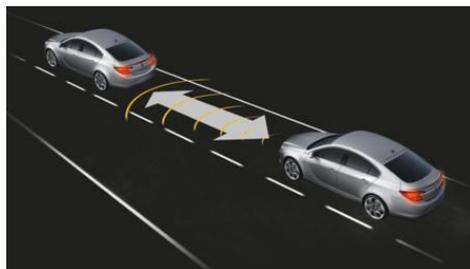
El siguiente apartado es un resumen de las tecnologías existentes en el ámbito de tránsito y automóviles.

Sistema ADAS[15].

El sistema avanzado para la asistencia vehicular es un asistente que proporciona seguridad al automovilista en el momento de conducir un carro de esta forma reducir los accidentes vehiculares.

El sistema ADAS permite integrar varias tecnologías que proporcionan seguridad a los conductores, entre las tecnologías que se encuentran integradas en un sistema ADAS se destacan:

-Control de Crucero Adaptativa (ACC)[16]: Sirve para evitar la colisión entre vehículos que se encuentran inmediatamente adelante uno del otro, indicando la distancia adecuada para no sufrir ninguna colisión a través de un radar de onda milimétrica. En la figura 1.7 se muestra un esquema del funcionamiento del control de crucero.



---

Figura 1.7. Detectando un vehículo que se encuentra delante de otro vehículo por medio del radar incorporado.

-Sistema de mantenimiento de carril o sistema de alerta de abandono de carril[17]: Consiste en indicar al conductor si está desviándose del carril al momento de conducir, gracias a este sistema se reduce los accidentes vehiculares debido a descuidos al conducir. En la figura 1.8 se muestra el sistema de mantenimiento de carril en acción detectando las líneas del carril.



Figura 1.8. Detectando las líneas de carril por medio de una video cámara.

Actualmente existe más tecnologías que se están desarrollando e incorporando en los sistemas ADAS. Aunque estos sistemas aparecen más frecuentemente en los vehículos más modernos y costosos, se puede incorporar el sistema ADAS en vehículos que no lo tengan.

Reconocimiento automático de matrículas[18].

El reconocimiento automático de matrículas (ANPR) o detector de placas vehiculares (LPR) es una tecnología que se usa en diferentes aplicaciones de tránsito. Consiste en capturar una imagen de la matrícula de un vehículo, transformado la imagen en caracteres alfanuméricos usando reconociendo óptico de caracteres, comparando el número de placa adquirido con una base de datos que incluya información de vehículos de interés y alertando al oficial de policía si el vehículo es el que está buscando. Esta tecnología ha servido de gran ayuda a los policías a detectar delincuentes que roban vehículos, secuestros, asesinatos, etc. Anqué no solo puede servir para encontrar delincuentes sino sirve también para multar a conductores que no cumpla con el reglamento de tránsito. En la figura 1.9 se presenta la aplicación del ANPR dentro de una patrulla policiaca:



Figura 1.9. Implementación de la tecnología ANPR en patrullas policiacas.

## 1.8. Estado del conocimiento en la detección de carriles vehiculares

En este apartado se explicará que son los carriles vehiculares y sus características, así como las metodologías usadas para lograr la detección de carriles por visión por computadora en previos trabajos de investigación.

### 1.8.1. Carriles vehiculares[19]

Un carril es una subdivisión de una calzada en forma de una banda longitudinal diseñado con las dimensiones adecuadas para permitir una circulación de una sola fila de vehículos. Los carriles sirven como un elemento de clasificación tipológica de vías, entre vías de dos carriles y multicarril.

Visualmente un carril se puede distinguir mediante marcas viales o separadores de tráfico en el pavimento según el grado seguridad necesaria y sentido de circulación.

La cantidad de carriles que se encuentre en una calzada o carretera dependerán en ocasiones de la topología en donde se encuentren las vías y la infraestructura. Generalmente la cantidad de carriles que se encuentran en una ciudad varía entre un solo carril hasta más de cuatro carriles existentes, esto se debe a que existen diferentes tipos de vialidades como son las avenidas, ejes viales, vialidades secundarias y calles. Mientras en carreteras la cantidad de carriles por lo general son dos, un carril por sentido.

### 1.8.2. Anchura del carril vehicular[19]

Es de gran importancia al momento del diseño de una vialidad contemplar la anchura de los carriles previniendo a los conductores y sus vehículos de accidentes, dependiendo de la anchura del carril podrá haber una mayor fluidez en el tránsito si el carril es más ancho, esto es porque el conductor es más propenso a realizar basculaciones laterales y para el caso de un carril de anchura menor será para vialidades de menor velocidad y donde la demanda de circulación sea menor.

Las dimensiones del ancho del carril son muy variadas por las cuestiones de velocidad, demanda de circulación y diseño de la vía por lo que las dimensiones varían desde 2.50 m. hasta 3.75 m. aunque el carril normalizado es de 3.50 m. , pasarse o reducir el límite de anchura de un carril podría ser contraproducente ya que si el carril es menor de 2.50 m. impediría la correcta circulación de los vehículos y algunos vehículos de mayores dimensiones no podrían circular y en el caso de un carril con un mayor ancho al de los 3.75 m quitaría espacio fundamental para la existencia de otros carriles. En

ocasiones es necesario otorgar al carril con una anchura suplementaria para facilitar el giro de los vehículos, el tamaño influye por la geometría de las curvas de la vía.

### 1.8.3. Color de las marcas del carril[20][21]

Los colores de las marcas viales que se utilizan para dividir la calzada o carretera por carriles son el blanco y el amarillo. En el caso de estas marcas viales son líneas continuas o discontinuas donde las líneas blancas son las más utilizadas y son reflejantes y mientras las líneas amarillas generalmente no son reflejantes. Dentro del carril puede haber la posibilidad de que exista otro tipo de marcas viales diferentes a las líneas como son las flechas que indican el sentido de circulación, también puede haber indicadores de un cruce de un ferrocarril, etc.

### 1.8.4. Detección de carriles por visión por computadora[22][23]

La detección y reconocimiento de carriles es parte esencial para un sistema de advertencia de colisión vehicular. El propósito de la detección de carriles es encontrar y rastrear las marcas viales en forma de líneas, ya sea si son continuas o discontinuas y de color blanco o amarillo, dentro del asfalto negro de una carretera o vialidad. Al momento de diseñar estos sistemas se requiere que sean lo más exactos, que sean en tiempo real y robustos para asegurar al conductor de que el sistema será útil al momento de usarlo.

La detección y rastreo de carriles es considerado uno de los más desafiantes problemas que hay en el área de visión por computadora. La investigación sobre la detección de carriles ha abordado el tema con diferentes técnicas o soluciones para cumplir con esta tarea.

Por lo general un sistema de visión por computadora aplicado para la detección de carriles utiliza una video cámara la cual es colocada atrás del parabrisas de un vehículo con la finalidad de captar imágenes del camino en donde viaja el conductor. Las líneas marcadas dentro del camino son interpretadas y son detectadas. Una advertencia visual o auditiva aparecerá cuando el conductor salga de su carril donde viaja intencionalmente. Algunos sistemas deshabilitan la advertencia cuando el conductor activa las intermitentes.

La detección de las marcas viales es considerada el paso inicial de un sistema de advertencia de salida de carril. Existe dos metodologías usadas para la detección de carriles: los enfoques basados en características y modelos; en el enfoque basado en características usa características de bajo nivel como son los bordes mientras que el enfoque en modelos usa parámetros geométricos para detectar carriles.

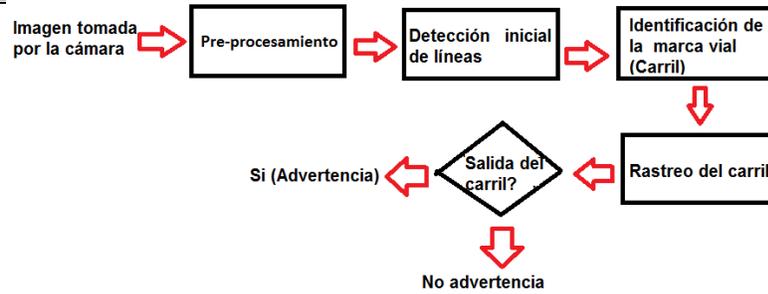


Figura 1.10. Diagrama en bloques de sistema de advertencia de salida de carril.

La figura 1.10 es un diagrama básico para un sistema de advertencia de salida de carril, este diagrama sirve como referencia para el desarrollo de un sistema que tenga el mismo propósito, aunque no significa que sea obligatorio seguir por completo el diagrama. Al momento de diseñar un sistema de advertencia de salida de carril se puede encontrar con problemas y desafíos como la diversidad del aspecto del carril, variación en la claridad de la imagen, cambios en las condiciones de visibilidad.

Hay una gran diversidad en cuanto al aspecto del carril debido a las normativas de tránsito estipuladas en un país esto puede considerarse como un problema para este tipo de sistemas ya que es un desafío diseñar un sistema que pueda detectar y rastrear diferentes carriles. Debido que las vialidades y carreteras se encuentran en espacios abiertos habrá problemas en la claridad de la imagen debido a la presencia de sombras dentro de la imagen. La presencia de objetos dentro de lo carriles, en especial otros vehículos, puede obstruir la detección del carril. Otro problema que se encuentra en las vialidades o carreteras es cuando un vehículo sale de un túnel y ocurren variaciones en la iluminación muy abruptas. La visibilidad de los carriles es afectada por las diferentes condiciones de clima como es la lluvia, así como variaciones de iluminación debido a los diferentes momentos del día. En la figura 1.11 se presentan algunos escenarios a afrontar al momento de detectar los carriles en una vialidad:

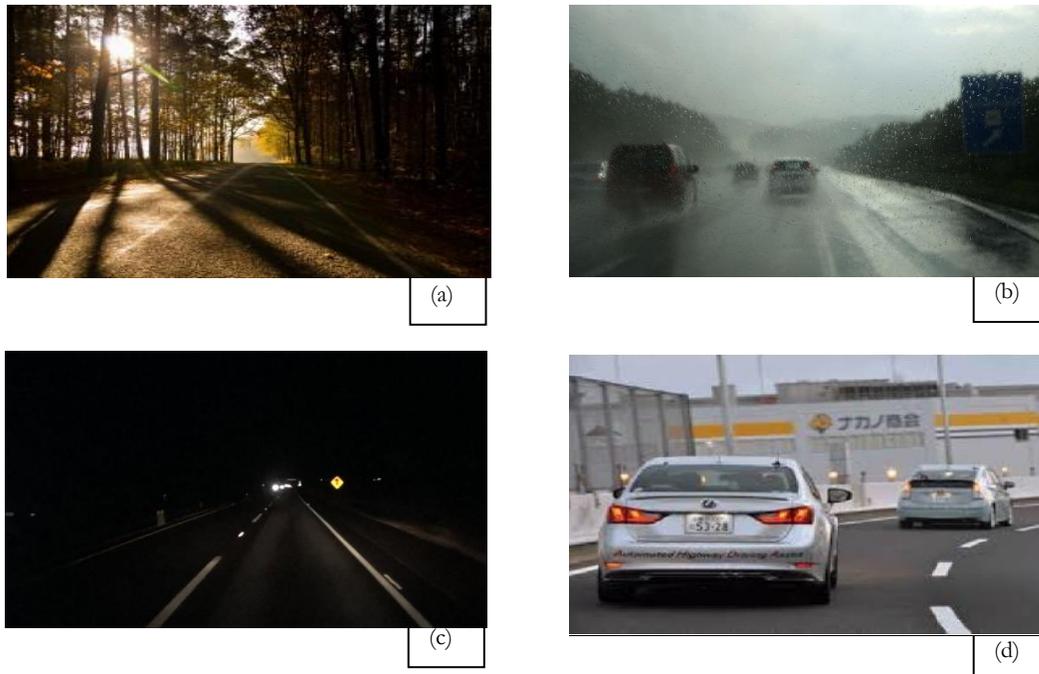


Figura 1.11. Algunos ejemplos de problemas a afrontar en la detección de carriles (a) Variaciones de iluminación con sombras (b) Presencia de lluvia (c) Carretera en la noche (d) Obstrucción de las líneas de carril debido a la presencia de un vehículo cercano.

A continuación, se muestra una tabla con diferentes métodos para la detección y rastreo de carriles como antecedentes.

Tabla 1.1. Métodos para detección y rastreo de carriles.

Método	Preprocesamiento	Detección	Rastreo
Lane Detection and Tracking in Challenging Environments based on a Weighted Graph and Integrated Cues. <b>C. Guo et al. (2010) [23]</b>	Clasificadores en cascada basados en características.	Spline de Catmull-Rom.	Filtro de partículas.
Combining Statistical Hough Transform and Particle Filter for Robust Lane Detection and Tracking.	Transformación perspectiva inversa. Umbral adaptativo.	Transformada de Hough probabilística.	Filtro de partículas.

<b>G. Liu et al. (2010) [24]</b>			
Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization. <b>P. Daigavane and P. Bajaj (2010) [25]</b>	Detector de bordes Canny. Optimación por colonia de hormigas.	Transformada Hough.	
Vision-Based Lane Departure Detection System in Urban Traffic Scenes. <b>Y.-C. Leng and C.-L. Chen (2010) [26]</b>	Operador Sobel.	Transformada Hough.	
Detección de Líneas y Sistema de Estabilidad de Carril basado en cámara frontal. <b>Álvaro Medina Ballester y Xavier Leal Meseguer (2010)</b>	Transformación perspectiva inversa. Detector de bordes Canny.	Transformada Hough.	
RESEARCH ON LANE DETECTION TECHNOLOGY BASED ON OPENCV. <b>Xu Yang and Zhang Ling. (2012) [27]</b>	Método Otsu.	Método LMedSquare.	
A Much Advanced and Efficient Lane Detection Algorithm for Intelligent Highway Safety.	Transformada Top- Hat. Umbral dinámico.	Transformada Hough.	

<b>Prof. Sachin Sharma<sup>1</sup> and Dr. D. J. Shah (2013) [28]</b>			
ROBUST LANE DETECTION & TRACKING BASED ON NOVEL FEATURE EXTRACTION AND LANE CATEGORIZATION. <b>U. Ozgunalp, and N. Dahnoun, (2014) [29]</b>	Umbral local simétrico.	Transformada Hough.	Filtro Kalman.
Multiple Lane Boundary Detection Using A Combination of Low-Level Image Features. <b>Y.Li et al. (2014) [30]</b>	Detector Canny.	Transformada Hough.	Filtro Kalman.

A continuación, se presenta una tabla comparativa entre las referencias y el proyecto de tesis.

*Tabla 1.2. Ventajas y desventajas del proyecto de tesis con respecto a las referencias consultadas.*

Título	Ventajas	Desventajas	Pruebas
Lane Detection and Tracking in Challenging Environments based on a Weighted Graph and Integrated Cues.	El tiempo en procesar un cuadro del proyecto es de 42ms y en este artículo es 200ms.	No existe un criterio de evaluación para comparar este artículo con el proyecto de tesis. Aunque se expresa en el artículo que trabaja sin	Cinco videos en diferentes carreteras y en diferentes condiciones (como es clima, iluminación, etc.)

		problemas la detección con sombras, obstrucciones de objetos y variaciones de iluminación.	
Combining Statistical Hough Transform and Particle Filter for Robust Lane Detection and Tracking.	El tiempo en procesar un cuadro del proyecto es de 42ms y en este artículo es 1.1s.	La precisión para detectar los carriles en el proyecto es de 87.79% en 1008 cuadros, mientras que en este artículo es de 100% en 837 imágenes.	Las pruebas en el artículo fueron realizadas con 837 imágenes de un mismo escenario.
Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization.	No existe un criterio de evaluación para comparar este artículo con el proyecto de tesis.	No existe un criterio de evaluación para comparar este artículo con el proyecto de tesis.	No existen pruebas dentro de este trabajo (solo se probó el programa en un escenario simulado).
Vision-Based Lane Departure Detection System in Urban Traffic Scenes.	Se menciona en el artículo una gran variedad de pruebas, pero no hay una evaluación del programa.	Se menciona en el artículo una gran variedad de pruebas, pero no hay una evaluación del programa la cual pueda demostrar que el artículo es mejor que el proyecto.	Existen pruebas, con escenarios con variaciones de iluminación, con obstrucciones de vehículos y presencia de otras marcas viales.
Detección de Líneas y Sistema de Estabilidad de Carril basado en cámara frontal.	No existe un criterio de evaluación para comparar este	No existe un criterio de evaluación para comparar este	No existen pruebas dentro de este trabajo.

	artículo con el proyecto de tesis.	artículo con el proyecto de tesis.	
RESEARCH ON LANE DETECTION TECHNOLOGY BASED ON OPENCV.	El tiempo de ejecución del programa es de 42 seg(s) en 1008 cuadros y en este artículo es de 68 seg(s) en 1000 cuadros.	La precisión para detectar los carriles en el proyecto es de 87.79% en 1008 cuadros, mientras que en este artículo es de 97.2% en 1000 cuadros.	Pruebas en diferentes escenarios: con variaciones de clima y estados del día.
A Much Advanced and Efficient Lane Detection Algorithm for Intelligent Highway Safety.	El tiempo en procesar un cuadro del proyecto es de 42ms y en este artículo es 53.6ms.	No existe un criterio de evaluación para comparar este artículo con el proyecto de tesis.	Se mencionan una variedad de escenarios a probar con el programa.
ROBUST LANE DETECTION & TRACKING BASED ON NOVEL FEATURE EXTRACTION AND LANE CATEGORIZATION.		La precisión para detectar los carriles en el proyecto es de 87.79% en 1008 cuadros, mientras que en este artículo es de 96.2%.	Variedad de pruebas en diferentes escenarios.
Multiple Lane Boundary Detection Using A Combination of Low-Level Image Features.		La precisión para detectar los carriles en el proyecto es de 87.79% en 1008 cuadros, mientras que en este artículo es de 91.4%. El tiempo en procesar un cuadro del proyecto es de	Variedad de pruebas en diferentes escenarios.

		42ms y en este artículo es 13ms.	
--	--	----------------------------------	--

### 1.9. Estado del conocimiento en el reconocimiento de las placas vehiculares

En este apartado se explicará que son las placas vehiculares y sus características, así como las metodologías usadas para lograr el reconocimiento de matrículas por visión por computadora en previos trabajos.

#### 1.9.1. Placas Vehiculares[31]

Una placa vehicular o matricula vehicular sirve para identificar al vehículo y a su conductor y permite la circulación del vehículo en la vialidad.

La estructura de la placa varía dentro de la República Mexicana, la entidad gubernamental encarga de regular las placas es la secretaria de transportes, que establece la **Norma oficial mexicana NOM-001-SCT-2-2000**.

La norma clasifica las placas de acuerdo con el uso que se destine:

- Automóviles.
- Autobuses.
- Camiones.
- Midibuses.
- Remolques.
- Motocicletas.

Las especificaciones de las placas son:

Las placas para automóviles, autobuses, camiones, midibuses y remolques son rectangulares con las siguientes dimensiones:

- Largo 300mm.
- Ancho 150mm.

Para placas de siete caracteres:

- Dígitos: largo 70mm y ancho 30mm.
- Letras: largo 75mm y ancho 30mm.

Separación entre:

- Dígitos: 7.0mm.
- Letras: 7.0mm.
- Letras y dígitos: 7.0mm.

Para placas con seis o menos caracteres:

- Dígitos: largo 75mm y ancho 36mm.
- Letras: largo 80mm y ancho 36mm.

Separación entre:

- Dígitos: 7.0mm.
- Letras: 7.0mm.
- Letras y dígitos: 7.0mm.

#### 1.11.2. Reconocimiento Óptico de Caracteres (OCR)[32]

La necesidad de realizar tareas y trabajos en el menor tiempo posible requiere de tecnologías que logren este objetivo; en el caso de requerir información de un documento impreso para procesarlo dentro de una computadora puede tomar algo de tiempo pasar la información a la computadora, para evitar perder el tiempo en escribir en la computadora se puede automatizar esta tarea por medio del reconocimiento óptico de caracteres.

La tecnología de reconocimiento de caracteres constituye un conjunto de técnicas basadas en estadísticas, en las formas de los caracteres, transformadas y en comparaciones, juntas se emplean para distinguir de forma automática entre los caracteres alfanuméricos. En realidad, no se reconocen exactamente los caracteres de un alfabeto, sino diferenciar entre cualquier tipo de formas y símbolos. La precisión del OCR no llega al 100%, esto se debe que en la práctica en el mundo real la forma de un carácter puede variar.

En un sistema de reconocimiento óptico de caracteres, por lo menos se constituyen de cuatro etapas:

- Adecuación de la imagen (Preprocesamiento).
- Selección del área de interés (Segmentación).
- Representación digital de la imagen (Extracción de características).
- Distinción del carácter contenido en la imagen (Reconocimiento).

De estas etapas se considera que la extracción de características ya que es muy difícil elegir un conjunto óptimo de características. En general para que una característica sea buena se considera:

- Discriminación: Cada clase debe contener características diferentes para poder diferenciarse de otras clases.
- Mismo valor para mismas clases.
- Independencia: Las características no deben estar relacionadas unas de otras.
- Pequeño espacio para características: El número de características debe ser pequeño para la rapidez y facilidad de clasificación.

Las características deben contener un bajo gasto computacional tanto en tiempo y complejidad, considerando todos estos puntos puede ser un reto desarrollar un OCR.

### 1.11.3. Reconocimiento automático de matrícula (ANPR)[33][34][35]

El Reconocimiento automático de matrícula es un método de vigilancia que usa el reconocimiento óptico de caracteres en las imágenes para poder leer las matrículas en los vehículos.

ANPR puede guardar las imágenes capturadas tomadas por una cámara, así como el texto de la matrícula. Estos sistemas ANPR tiene una cámara infrarroja para poder captar las imágenes dentro de cualquier hora del día. También dentro de la cámara hay un flash potente cuyas funciones son para iluminar la imagen, así como indicador de que el individuo que cometió una infracción. El sistema ANPR varía su configuración dependiendo la región donde se implemente ya que cada país tiene su diseño de la placa.

En un sistema ANPR, se constituyen de cuatro etapas como se muestra en la figura 1.12:

- Captura de imagen: Obtener la imagen por medio de una video cámara ya sea por una foto o por un video, en el caso del video se requiere de más trabajo para obtener el objeto de interés.
- Preprocesamiento: La imagen debe estar claramente visible y debe adecuarse para su procesamiento final.
- Segmentación de caracteres: La región donde se encuentra la placa dentro de la imagen es extraída.
- OCR: Con la técnica de reconocimiento de caracteres se obtiene el número de la placa a partir de la imagen.

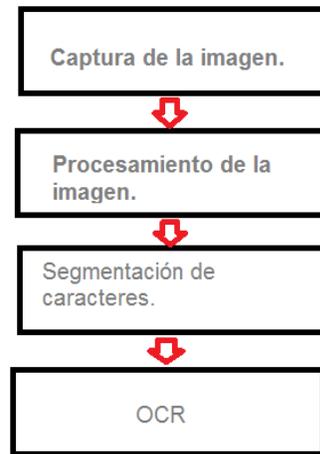


Figura 1.12. Diagrama en bloques de sistema Reconocimiento automático de matrícula.

Al igual que otros sistemas de visión por computadora el reconocimiento automático de matrículas tiene problemas de efectividad de ejecución cuando hay variaciones de iluminación, lluvia, nieblas, sombras, si se usa cámaras de baja calidad, etc.

A continuación, se muestra una tabla con diferentes métodos para el reconocimiento de placas vehiculares como antecedentes.

Tabla 1.3. Métodos para reconocimiento de placas vehiculares.

Método	Descripción	Técnicas usadas
Number Plate Recognition Using an Improved Segmentation. <b>Mr.G.T. Sutar et al. (IJIRSET 2014) [36]</b>	Implementa el reconocimiento automático de placas vehiculares, para sistemas de seguridad.	-Segmentación de la imagen para extraer la placa. -OCR.
License Plate Recognition System based on Image Processing Using LabVIEW. <b>Kuldeepak et al. (IJECCCT 2012) [37]</b>	Implementa el reconocimiento automático de placas vehiculares, incluyendo además del reconocimiento, ofrece una alerta para el robo de vehículos.	-Segmentación de la imagen para extraer la placa. -Uso de software: vision assistant 8.2.1 y LabVIEW 11.0.
Impact of character spacing on the performance of Automatic Number Plate Recognition (ANPR) systems through	Determina el impacto en la ejecución del ANRP causado a las variaciones de espacio entre los caracteres del número de	-Simulación de placas vehiculares para pruebas. -OCR.

Simulation. <b>S. Ramalingam et al.</b> <b>(IEEE 2014) [38]</b>	la placa. Se examinan los errores producidos para mejorar el ANPR.	
Automatic License Plate Recognition using Python and OpenCV. <b>K.M. Sajjad et.al</b> <b>(Department of Computer Science and Engineering 2012) [39]</b>	Implementa el reconocimiento automático de placas vehiculares, por medio de las librerías OpenCV con Python y la librería Tesseract-OCR.	-Técnicas de procesamiento de imágenes mediante la librería OpenCV.  -OCR mediante la librería Tesseract-OCR.

A continuación, se presenta una tabla comparativa entre las referencias y el proyecto de tesis.

*Tabla 1.4. Ventajas y desventajas del proyecto de tesis con respecto a las referencias consultadas.*

Título	Ventajas	Desventajas	Pruebas
Number Plate Recognition Using an Improved Segmentation.	El proyecto trabaja con matrículas de mayor dificultad para reconocer el número de estas a comparación del artículo.	Para la detección de la matrícula el proyecto tubo un 85.71% de precisión mientras que el artículo tiene un 96%. Y para el reconocimiento del número de matrícula es del 80.7% y el del artículo es del 93%.	Fotos de diferentes vehículos (no se especifica a detalle la distancia y la posición de las fotos, ni la cantidad de fotos tomadas).
License Plate Recognition System based on Image Processing Using LabVIEW.	El proyecto trabaja con matrículas de mayor dificultad para reconocer el número de estas a comparación del artículo.	El Proyecto posee una precisión para reconocer el número de matrícula del 65.71%, mientras en este artículo posee el 98% de precisión.	Fotos de diferentes vehículos (no se especifica a detalle la distancia y la posición de las fotos, ni la cantidad de fotos tomadas).

<p>Impact of character spacing on the performance of Automatic Number Plate Recognition (ANPR) systems through Simulation.</p>	<p>El proyecto trabaja con matrículas de mayor dificultad para reconocer el número de estas a comparación del artículo.</p>	<p>No da un porcentaje en general de la precisión para reconocer los caracteres de la matrícula en el artículo, por lo que no se puede hacer una comparación entre los dos trabajos.</p>	<p>Fotos de diferentes vehículos (no se especifica a detalle la distancia y la posición de las fotos, ni la cantidad de fotos tomadas).</p>
<p>Automatic License Plate Recognition using Python and OpenCV.</p>	<p>El proyecto trabaja con matrículas de mayor dificultad para reconocer el número de estas a comparación del artículo.</p>	<p>Para la detección de la matricula el proyecto tubo un 85.71% de precisión mientras que el artículo tiene un 92%. Para la extracción de los caracteres de la placa en el proyecto fue de un 95% un 95.7%. Y para el reconocimiento del número de matrícula es del 80.7% y el del artículo es del 94.3%.</p>	<p>Fotos de diferentes vehículos (no se especifica a detalle la distancia y la posición de las fotos, ni la cantidad de fotos tomadas).</p>

## CAPÍTULO 2

### DETECCIÓN DE CARRILES

#### Introducción

En el proyecto se desarrolla un programa computacional que utiliza las librerías de visión por computadora (OpenCV) para lograr la detección de carriles vehiculares. La visión por computadora será el tema central de investigación para poder desarrollar tal programa, no obstante, la visión por computadora se apoya del procesamiento digital de imágenes que cuenta con variedad de técnicas permitiendo mejorar una imagen o adecuarla dentro de aplicaciones de visión por computadora.

En este capítulo se explica el programa computacional y las técnicas usadas dentro del programa.

#### 2.1. Algoritmo desarrollado para la detección de carriles

Con base en la figura 2.1 se explicará el algoritmo de detección de carriles:

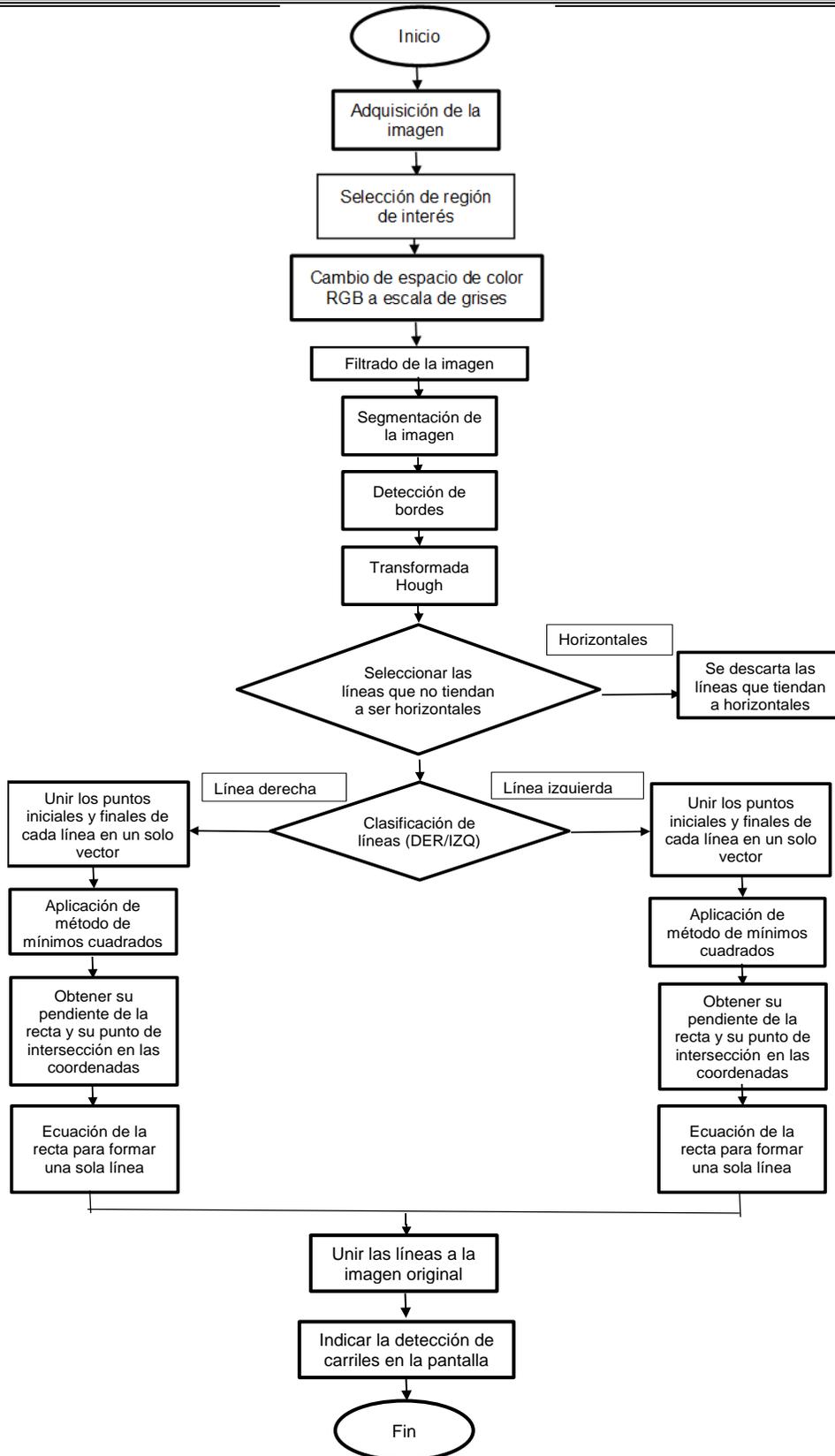
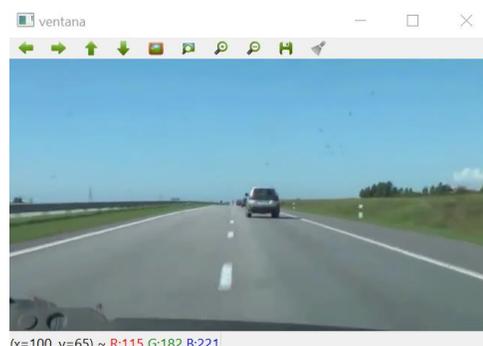


Figura 2.1. Diagrama en bloques de la detección de carriles.

### 2.1.1. Adquisición de la imagen[40]

Desde un video o de una video cámara digital, se obtiene una imagen de un escenario a capturar. En el caso de la detección de carriles el escenario a capturar será una autopista o una vialidad de tránsito, por medio de una video cámara montada dentro de un carro o usando videos ya previamente grabados se obtiene el escenario deseado para poder desarrollar la detección de carriles.

Tomando un video de una grabación de una carretera como se observa en la figura 2.2, se utiliza como escenario de experimentación para desarrollar el algoritmo.



**Figura 2.2.** Adquisición de la imagen (video tomado de la plataforma YouTube).

El video original presenta características como la resolución, el alto y el ancho del fotograma y la velocidad de fotogramas por segundo cuyas características son generalizadas en el programa computacional para poder realizar pruebas con otros videos.

Establecer estas características también sirve para obtener un video más fluido al momento de procesarlo para la detección de carriles en el caso de la resolución del video se establece una resolución fija de 384x216 pixeles para todos los videos.

De igual manera establecer la velocidad de fotogramas por segundo a 24 cps/seg, la selección de esta velocidad se realizó para que el procesamiento del video fuera fluido, este tipo de velocidad era muy común en el cine y televisión por tal razón no habrá problemas para el ojo humano visualizar el video.

En el caso de establecer el ancho y alto del fotograma se establecieron las dimensiones de 320 pixeles para el ancho y para la altura de 234 pixeles. Gracias a

estos parámetros se tendrá un menor costo computacional en el programa y el video se aproxima a una transmisión en tiempo real ya que al procesar el video habrá la tendencia a que pueda atrasarse su transmisión de la imagen al momento de ser mostrada en pantalla y de igual manera el dispositivo de procesamiento a usar afecta la fluidez de la imagen.

### 2.1.2. Selección de región de interés[13][40]

Para detectar las líneas de carril no es necesario analizar y procesar toda la imagen ya que al usar toda la imagen se produce más problemas al momento de detectar el objeto de interés esto debido a que en toda la imagen se encuentran más objetos que complican la detección. También al seleccionar una región de interés el gasto computacional es menor, considerando que solo se toma una parte de la imagen.

Para poder seleccionar la región de interés se hace mediante transformaciones geométricas de la cual se puede clasificar sus operaciones de la siguiente manera:

- Transformaciones afines predefinidas.
- Transformaciones afines genéricas.
- Transformaciones perspectivas.
- Transformaciones de mapeo arbitrario.

Para seleccionar la región de interés se recurre a la operación de las transformaciones de perspectivas con el fin de obtener una vista de arriba hacia abajo del camino, a diferencia de las anteriores formas de seleccionar la región de interés donde no se utilizaban transformaciones geométricas, en este caso la transformación de perspectiva es: dado un plano (la imagen de entrada) colocarlo en una posición cualquiera del espacio 3D y después proyectarlo sobre el plano de imagen, en la figura 2.3 se muestra el proceso anteriormente mencionado en una imagen. En otras palabras, se obtiene una vista del asfalto donde se encuentran el carril de adelante del vehículo, dando la ilusión óptica de que la imagen es tomada desde una video cámara colocada desde arriba del vehículo.

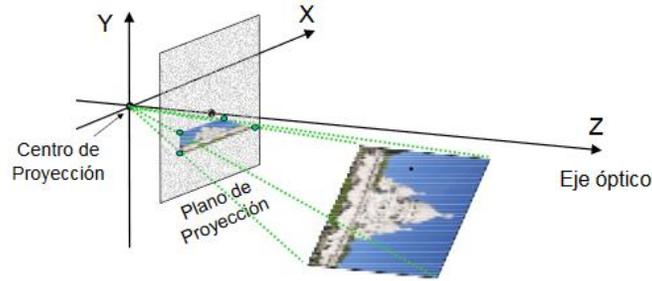


Figura 2.3. Proceso de la transformación perspectiva de una imagen: colocar la imagen plana en el espacio 3D y la proyección perspectiva de la imagen en el espacio.

En la figura 2.4 se muestra la vista de arriba hacia abajo del asfalto, después de aplicar la transformación geométrica:

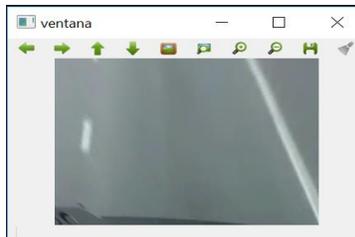


Figura 2.4. Aplicando la transformada perspectiva a la imagen para obtener la región de interés donde se encuentra el carril frontal.

También se aplicó el proceso inverso (transformación inversa de perspectiva). Se llama transformación inversa debido a que cambia la perspectiva tomada por la cámara como se ilustra en la figura 2.5. En este caso la figura 2.4 donde se muestra el resultado de aplicar la transformación de perspectiva se aplicará este proceso inverso para regresar la escena de los carriles a su perspectiva original, no obstante, este proceso será hecho hasta el final, hasta que se logre detectar los carriles. Teniendo la imagen original y la imagen que se aplicó la transformación inversa (considerando que solo regresa la región donde se encuentran los carriles) se unirán para presentar la detección de carriles dentro de toda la escena de la carretera o vialidad como se muestra en la sección 2.1.12.

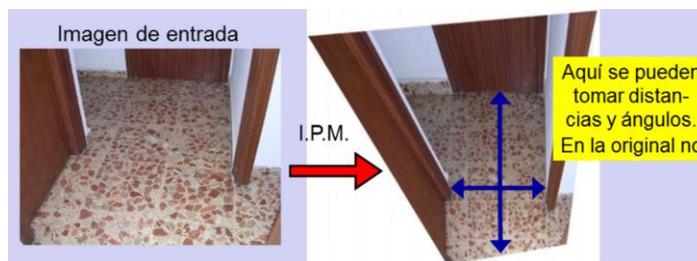


Figura 2.5. Resultado de aplicar la transformación inversa a la perspectiva producida por la cámara, utilizando la nueva perspectiva de la imagen se puede realizar mediciones u obtener parámetros que en el caso de la imagen original podrían ser más difíciles de obtener.

No solamente se obtiene la región de interés de igual manera como los anteriores métodos, sino que también se obtienen beneficios adicionales que ayudan a la detección de carriles: se selecciona de mejor forma los carriles ya que los carriles son más visibles que en los anteriores métodos y tienden a verse rectos los carriles (paralelos entre sí) mejorando la detección de carril, también gracias a la vista de ojo de pájaro se puede medir distancias y ángulos de las líneas detectadas para la detección del carril de mejor forma que en los anteriores casos. Dentro de las pruebas realizadas este método también presenta falsos positivos al detectar carriles de igual manera como los anteriores casos, pero se presentan en menor cantidad.

Definitivamente el último método es el que se considera para tomar la región de interés debido a las ventajas que presentan en comparación a los anteriores métodos mencionados, aun así, se presenta un problema que es la velocidad de ejecución de las funciones de transformación perspectiva, haciendo que el programa sea menos rápido en comparación de los métodos anteriores. No obstante, en la experimentación este último método se aproxima a un procesamiento en tiempo real esperado en la detección de carriles.

Matemáticamente la transformación de perspectiva se expresa como:

- 1) Colocar la imagen plana en el espacio 3D:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.1)$$

- 2) Proyección perspectiva de la imagen en el espacio:

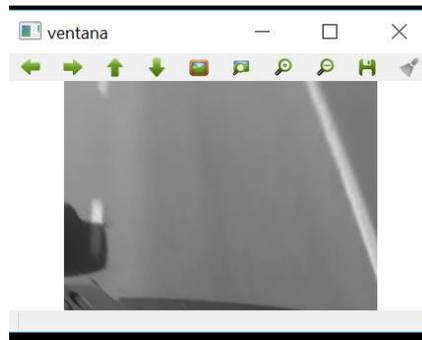
$$R(x, y) = A \left( x' / z', y' / z' \right) \quad (2.2)$$

### 2.1.3. Cambio de espacio de color RGB a escala de grises[13]

Con el fin de reducir la carga computacional y el tiempo de procesamiento se opta por trabajar con imágenes en escala de grises. Trabajar con imágenes en escala de grises es mucho más fácil ya que solo se utiliza una sola matriz en vez de utilizar tres matrices de los tres colores primarios (rojo, verde y azul), contiendo una menor información que las imágenes a color facilitando y agilizando el procesamiento en la imagen. El cambio de espacio de color de RGB a escala de grises matemáticamente es:

$$G(x, y) = \frac{R(x, y) + A(x, y) + V(x, y)}{3} \quad (2.3)$$

En la figura 2.6 se muestra el resultado del cambio de espacio de color.



---

Figura 2.6. Cambio de espacio de color RGB a escala de grises en la imagen donde se encuentra los carriles.

#### 2.1.4. Filtrado de la imagen[1][41]

Después de tener la imagen en escala de grises se requiere filtrar la misma para mejorarla (reducir el ruido). El filtrado es una operación de vecindario, en el cual el valor de un píxel dado en la imagen procesada se calcula mediante un algoritmo que toma en cuenta los valores de los píxeles de la vecindad de la imagen original. Existen diferentes operaciones, filtros y algoritmos dentro del procesamiento digital de imágenes que permiten resaltar elementos, detectar cambios, reducir el ruido existente y otras características.

Durante el desarrollo del programa se eligió el filtro gaussiano debido a que es uno de los más rápidos; el filtrado gaussiano se realiza al convolucionar cada punto en la matriz de entrada con un kernel gaussiano y sumando todos los elementos para producir la matriz de salida.

El resultado del filtrado se ilustra en la figura 2.7:



---

Figura 2.7. Aplicando filtro gaussiano en la imagen para reducir el ruido.

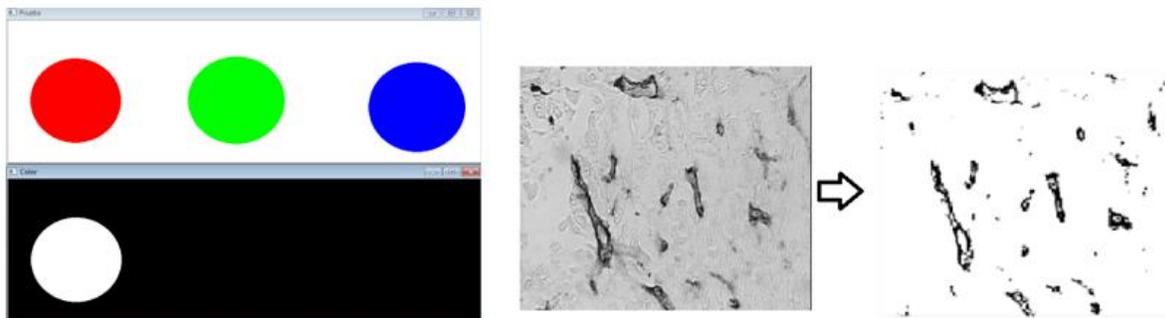
---

El filtro gaussiano se usa para suavizar y eliminar ruido. Su función matemática es:

$$G(x, y) = \frac{e^{-(x+y)^2/2\sigma^2}}{2\pi\sigma^2} \quad (2.4)$$

### 2.1.5. Segmentación de la imagen[41][42]

La segmentación es una etapa de gran importancia para los sistemas de análisis de imágenes dado a su uso. Básicamente, la segmentación puede considerarse la partición de la imagen, es decir separar el objeto u objetos de interés del resto de elementos no relevantes dentro de una imagen (considerado como fondo). A pesar de ser muy utilizada la segmentación, no es una tarea trivial y tampoco existe un método de segmentación perfecto y que pueda segmentar lo que se requiera. Esto debido al nivel de problema que se desee resolver, segmentar un objeto dependerá del objetivo a cumplir. En la figura 2.8 se ilustra dos ejemplos de segmentación:



*Figura 2.8. A pesar de utilizar en las dos imágenes la segmentación en el círculo rojo y en los vasos sanguíneos en sus respectivas imágenes no utilizan el mismo método: en el primer caso se utiliza la segmentación por color y el otro caso se utiliza umbralización a partir del histograma. Además de que el segmentar el círculo rojo es más fácil que segmentar vasos sanguíneos ya que estos tienen mayor complejidad de forma y se encuentran en un fondo complejo.*

Debido a las diferentes necesidades de extraer un objeto de interés de una imagen han surgido diferentes técnicas de segmentación. Existen métodos de segmentación por color y por imágenes monocromáticas. Actualmente existen muchos métodos para segmentar imágenes, para el caso de la detección de los carriles se utiliza un método muy simple y básico el cual es la selección de un umbral específico dentro de un rango (0 a 255) de una imagen en escala de grises para extraer el objeto de interés que se desea analizar (los carriles vehiculares). La segmentación por umbral se usa porque es un método que no tiene un alto gasto computacional y debido a que las marcas viales que forman los carriles son líneas de color blanco y amarillo por lo general y se encuentran en el asfalto de color negro facilitando la extracción de los carriles para realizar la detección de ellos. En este proceso de umbralización se van comparando

los niveles de intensidad de los píxeles de la imagen en escala de grises con el umbral seleccionado con el fin de seleccionar solo los píxeles que pertenezcan a las líneas de carril.

Se realizará la umbralización a la imagen, en la que se selecciona un rango de umbral para poder separar las líneas del carril dentro del asfalto a partir de la experimentación el umbral entre (180-255), se selecciona este intervalo ya que el valor 255 es del blanco y dentro del círculo cromático, el amarillo es el color de mayor luminosidad (más cercano al blanco, se observa en la figura 2.9).



Figura 2.9. Escala de luminosidad donde el color amarillo se aproxima a los niveles de intensidad del color blanco.

Se utiliza el umbralizado binario, el cual consiste en transformar la imagen de escala de grises a una imagen binaria, donde todos los tonos de grises sus niveles están dentro del intervalo definido por el umbral seleccionado, son transformados a 255 y todos los valores fuera de este intervalo a 0. Como se muestra matemáticamente a continuación:

$$is(x,y) = \begin{cases} 0; & \text{para } ie(x,y) \leq u1 \text{ o } u2 \leq ie(x,y) \\ 255; & \text{para } u1 < ie(x,y) < u2 \end{cases} \quad (2.5)$$

Donde:

$ie(x,y)$ : Imagen en escala de grises.

$is(x,y)$ : Imagen binaria.

$u1$ : Límite inferior del umbral seleccionado.

u2: Límite superior del umbral seleccionado.

A continuación, se muestra el resultado de la binarización de la imagen en la figura 2.10:



---

**Figura 2.10.** Resultado de aplicar la umbralización binaria inversa donde los carriles (objetos de interés) son de valor 255 mientras lo demás (el fondo) es cero.

### 2.1.6. Detección de bordes[40][41]

Los bordes de una imagen digital son definidos como las transacciones entre dos regiones de niveles de grises significativamente distintos. Proporcionan una valiosa información sobre las fronteras de los objetos, sirven para segmentar las imágenes, reconocer los objetos, etc.

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen. En la detección de carriles se utilizó el operador de bordes de Canny, uno de los más usados en la investigación y en aplicaciones de visión por computadora.

El operador de bordes de Canny es la primera derivada del Gaussiano y basa su operación en los siguientes criterios:

- Buena detección: El detector debe distinguir con suficiente margen entre bordes verdaderos y bordes falsos.
- Buena localización: Los puntos marcados como bordes deben corresponder en lo posible con los centros de los bordes verdaderos.
- Respuesta única: El detector debe responder con un solo borde, esto debido a que, en el primer criterio de distinguir bordes verdaderos de falsos, en el caso de que hubiera dos respuestas para un mismo borde una de ellas se debe considerar falsa.

La detección de bordes mediante Canny tiene como principal ventaja su gran adaptabilidad para poder ser aplicado a diversos tipos de imágenes, además de no disminuir su performance ante la presencia de ruido en la imagen original.

El método Canny usa el gradiente de la imagen para la detección de bordes. El gradiente de una imagen mide cómo esta cambia en términos de color o intensidad. La magnitud del gradiente nos indica la rapidez con la que la imagen está cambiando, mientras que la dirección del gradiente nos indica la dirección en la que la imagen está cambiando más rápidamente.

Al igual que al aplicar la umbralización se debe elegir un rango entre dos umbrales, para un borde para detectar los bordes. Si un gradiente del píxel es mayor al umbral seleccionado, se mantiene el píxel, de otra manera si es menor que el umbral seleccionado se rechaza el píxel y en el caso de que los píxeles se encuentren dentro del intervalo permanecerán, si una vecindad de píxel es considerada parte de un borde. En la experimentación el umbral seleccionado es de 180 a 255. La figura 2.11 muestra la detección de bordes en el asfalto con la presencia carriles:



Figura 2.11. Resultado de aplicar el método Canny para detección de bordes a la imagen previamente segmentada.

Gracias a la detección de bordes se podrá determinar cuál de estos bordes dentro de la imagen son líneas o no. Si no se tuviera una imagen con bordes no se podría tener una buena detección de líneas eficiente y que proporcione resultados significativos. En efecto una línea dentro de la imagen es un borde.

Pasos:

1. Filtrar el ruido existente. Utilizando un filtro Gaussiano. El Kernel utilizado es de tamaño 3.

$$k = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.6)$$

2. Encontrar la intensidad del gradiente de la imagen:
-

a) Aplicar un par de máscaras de convolución (en las direcciones “x” y “y”):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2.7)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (2.8)$$

b) Encontrar la longitud y la dirección del gradiente:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.9)$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (2.10)$$

La dirección es rodeada a uno de los cuatro posibles ángulos (0, 45, 90 o 135).

3. Supresión no máxima es aplicada. Esto remueve los pixeles que no son considerados parte de un borde. Solo las líneas candidatas (Candidatos Canny) permanecerán.
4. Histéresis: El último paso; Canny utiliza dos umbrales (superior e inferior):
  - a) Si un píxel gradiente es mayor al umbral superior. El píxel es aceptado como un borde.
  - b) Si el valor de un píxel gradiente esta debajo del umbral menor, entonces será rechazado.
  - c) Si el píxel gradiente está dentro los umbrales, entonces será aceptado solo si esta junto a un píxel que se encuentra arriba de un umbral superior.

#### 2.1.7. Transformada Hough[43]

La siguiente técnica usada dentro del programa computacional, siendo una de las más importantes, ya que la transformada Hough es uno de los métodos más utilizados y clásicos para poder detectar las líneas. En efecto, los carriles geoméricamente son líneas por lo que utilizar este método es ideal para poder detectar los carriles.

La transformada extrae todas las líneas que pasan por cada uno de los puntos de los bordes y agrupando los puntos que se localicen en la misma línea. Dentro de la librería OpenCV, se encuentran dos funciones que realizan dos tipos de transformada Hough:

- La transformada Hough estándar.
- La transformada Hough probabilística.

Para la detección de carriles se usó la transformada Hough probabilística debido a que es más eficiente, rápida y minimiza el consumo computacional que la transformada Hough estándar. No toma todos los puntos en consideración, en su lugar toma solo un subconjunto de puntos al azar y eso es suficiente para la detección de línea. Sólo se necesita disminuir el umbral. En la salida de aplicar este tipo de transformada se obtiene los extremos de las líneas detectadas.

Otra desventaja de la transformada Hough estándar es que simplemente busca una alineación de píxeles de borde en la imagen. Esto puede crear algunas falsas detecciones, debido que a alineaciones incidentales de píxeles o detecciones múltiples cuando hay varias líneas con valores de parámetros ligeramente diferentes que pasan por la misma alineación de píxeles.

El sistema de votación es el mismo que en el estándar, cada punto de los bordes vota por una línea compatible; es decir, vota por cada línea que pueda pasar a través de él. Estos votos se almacenan en un acumulador matricial, de dos dimensiones, relacionado con los parámetros que definen una recta en coordenadas polares. Las rectas principales o reales son aquellas que reciben un mayor número de votos, ya que hay más puntos en la imagen alineados con ella.

Otra ventaja por la que se utilizó la transformada Hough probabilística en vez de la transformada Hough estándar es porque esta transformada puede detectar segmentos discontinuos de líneas como son las líneas de carril discontinuas debido a que el otro tipo de transformada no detecta estas líneas.

A continuación, se muestra el resultado de aplicar la transformada Hough probabilística en el programa con la figura 2.12:



---

**Figura 2.12.** Resultado de aplicar la transformada Hough probabilística para detectar las líneas del carril.

### 2.1.8. Selección de las líneas que no tiendan a ser horizontales

Después de realizar la detección de líneas mediante la transformada Hough se obtiene una detección de carriles con resultados aún no deseados esto debido a que aparecen muchas líneas detectadas dentro de la imagen. El objetivo del programa es detectar las líneas de un carril (una línea por cada dirección del carril izquierda y derecha), en este caso si se detectan las líneas de carril, pero en gran cantidad dentro de los carriles, afectando la velocidad de ejecución del programa generando que al mostrar el resultado de la detección se vea lento y no cumpla con una transmisión en tiempo real. Otra desventaja es que al haber muchas líneas es muy difícil medir o verificar si el vehículo permanece dentro del carril. Es por lo que se requiere filtrar todas esas líneas y solo obtener un par de líneas que identifiquen el sentido de las dos marcas viales que conforman un carril (línea izquierda y derecha).

Primeramente, se filtrarán las líneas horizontales:

-Se recorre la matriz donde se encuentra los puntos iniciales y finales de las líneas (que es un vector que almacena cuatro puntos), mediante una estructura de control (ciclo *for*) se establece un intervalo, esto se debe a que se usa como un equivalente más legible del bucle *for* tradicional que opera en un rango de valores, como todos los elementos en un contenedor.

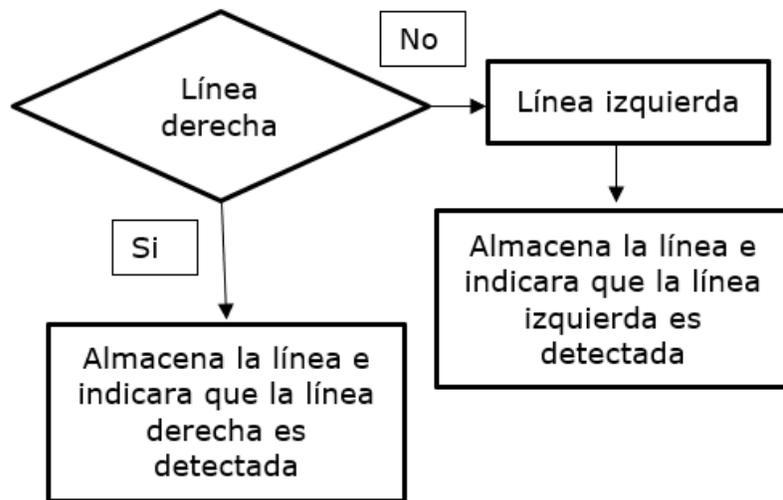
Dentro de este ciclo *for* se encuentra un condicional *if* este permite seleccionar que líneas no son o que tienden a ser horizontales con la ayuda de un umbral que indique el valor de una recta horizontal ("valor 0") y haciendo que la pendiente de cada línea tuviera valores positivos para no tener problemas con las líneas con pendiente negativa. La ayuda de utilizar el absoluto a cada valor de pendiente es para el propósito de no tomar líneas horizontales, pero en el caso para separar las líneas por sentido (DER-IZQ) se requiere de también del valor de pendientes negativas.

### 2.1.9. Clasificación de líneas (DER/IZQ)

Una vez que se obtienen las líneas que no sean horizontales, se procede a separar las por dirección o sentido de las líneas del carril, sirviendo en un futuro para indicar al conductor que se encuentra confinado en su carril al saber de la existencia de las dos líneas que conforman un carril.

Para separar las líneas se utiliza las dimensiones de la imagen y la pendiente de las líneas. Para cada sentido se utiliza un condicional *if*, para separarlas por líneas de dirección derecha e izquierda.

En la figura 2.13 se ilustra la clasificación de las líneas:



---

Figura 2.13. Diagrama de flujo de la clasificación de líneas.

Conociendo los puntos iniciales y finales de las líneas, el centro de la imagen y la pendiente de las líneas, se distribuirán las líneas en las dos mitades de la imagen (considerando que existen dos espacios dentro de la imagen divididos por la mitad de la imagen). En el caso para el sentido derecho se considera que la pendiente de las líneas debe ser positiva y sus puntos deben ser mayores en el eje "x" a la coordenada de la mitad de la imagen en el eje "x". Mientras para el sentido izquierdo la pendiente es negativa y sus puntos deben ser menores en el eje "x" a la coordenada de la mitad de la imagen en el eje "x".

En la figura 2.14 se muestra gráficamente como hacer la división. No obstante, el resultado obtenido todavía no es tan satisfactorio, ahora se requiere solo que se muestre dos líneas dentro de la imagen.

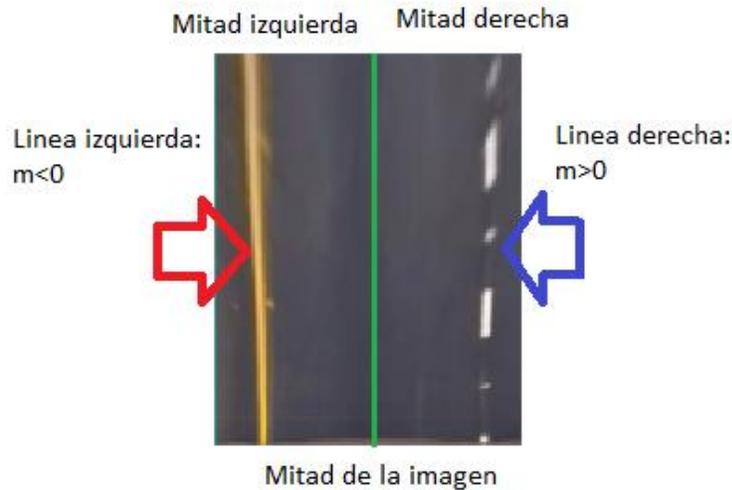


Figura 2.14. División de las líneas para determinar el sentido de las líneas del carril.

#### 2.1.10. Aplicación de método de mínimos cuadrados[13][43]

Teniendo las líneas separadas por su sentido o dirección, se requiere reducir la cantidad de líneas que aparezcan dentro de la imagen ya que la transformada Hough detecta la continuidad de un borde y lo considera una línea y por lo tanto la cantidad de líneas puede ser bastante ya que puede considerar a una grieta como línea así como la presencia de algún objeto en la imagen y también por las líneas de un carril vecino, en el caso de líneas continuas aparecerán dos líneas debido a sus extremos y para las líneas discontinuas se detectará una mayor cantidad de líneas.

Adicionalmente, la gran cantidad de líneas que aparezcan dentro de la imagen afectan en dos aspectos muy importantes: la velocidad de ejecución del programa y la visualización del carril detectado dentro de la imagen. En el caso de la velocidad de ejecución en el video procesado se debe a que a mayor cantidad de líneas que aparezcan en cada cuadro del video reduciendo la velocidad con la que se muestra el mismo video ya que las funciones de dibujo de OpenCV tienden a reducir la velocidad de ejecución del programa si se utilizan mucho dentro del mismo programa.

Y para la visualización del carril dentro de la imagen, se afecta al tener muchas líneas debido a que puede confundir al conductor si se encuentra dentro del carril o no, esto debido a que una línea que aparezca surge por una grieta u otra línea de otro carril o una señal de tránsito pintada en la vialidad.

Debido a que al tener muchas líneas dentro de la imagen causa problemas, es necesario la reducción de líneas dentro del video, colocando solo dos líneas que representen al carril y sirvan para indicar la estabilidad o posición del conductor dentro

de la vialidad. Se aplicará el método de mínimos cuadrados ponderados permitiendo formar una línea con base a un conjunto de puntos.

Primeramente, almacenar los puntos iniciales y finales de cada línea en sus respectivos sentidos en un solo vector de puntos con la finalidad de aplicar el método de mínimos cuadrados en los respectivos conjuntos de puntos.

Es un método de análisis de regresión el cual consiste en encontrar una función lineal la cual se aproxime a un conjunto de datos, la aproximación lineal obtenida proporcionara información de la línea obtenida, como es un vector colinial normalizado (o también conocido como la dirección de la línea) y un punto que se encuentre en la línea.

Después se obtendrá la pendiente de la línea por medio del vector colonial y con el punto que se encuentra en la línea para los dos sentidos (DER-IZQ) permitiendo obtener las dos líneas a mostrar en la pantalla con el fin de indicar la presencia del carril.

#### 2.1.11. Ecuación de la recta para formar una sola línea

Con la información obtenida, se utilizarán para la ecuación de la recta para poder formar las líneas que describan el carril. Para formar estas líneas se requiere conocer su pendiente y sus coordenadas de los puntos que formen a la línea.

Cada línea contiene dos coordenadas que permiten delimitar el tamaño de la línea (sus extremos). Por ejemplo: la línea derecha contiene en sus extremos las coordenadas  $P_D(x1, y1)$  "inicio de la línea" y  $P_D(x2, y2)$  "final de la línea", para facilitar la creación de las líneas, las coordenadas en el eje "y" se establecen por constantes ya definidas con el fin de delimitar el tamaño en longitud de las líneas. Para las coordenadas en el eje "x" se obtienen por medio de la ecuación de la recta. A continuación, se muestra el proceso para obtener las coordenadas para la línea:

**El proceso es el mismo para las dos líneas.**

**Donde:**

$x_1$ : coordenada en el eje "x" del punto inicial de la recta.

$x_2$ : coordenada en el eje "x" del punto final de la recta.

$y_1$ : coordenada en el eje "y" del punto inicial de la recta.

$y_2$ : coordenada en el eje "y" del punto final de la recta.

$x_3$ : coordenada en el eje "x" del punto obtenido del método de mínimos cuadrados.

$y_3$ : coordenada en el eje "y" del punto obtenido del método de mínimos cuadrados.  
 $m$ : pendiente obtenida por el vector colineal.

Ecuación de la recta:

$$(y_1 - y_3) = m(x_1 - x_3) \quad (2.11)$$

$$\frac{(y_1 - y_3)}{m} = x_1 - x_3 \quad (2.12)$$

$$x_1 = \frac{(y_1 - y_3)}{m} + x_3 \quad (2.13)$$

$$(y_2 - y_3) = m(x_2 - x_3) \quad (2.14)$$

$$\frac{(y_2 - y_3)}{m} = x_2 - x_3 \quad (2.15)$$

$$x_2 = \frac{(y_2 - y_3)}{m} + x_3 \quad (2.16)$$

En el caso del eje "y" se establece con los valores  $y_1=234$  y  $y_2=100$ . Con la posibilidad de poder variar estos valores, recordando que el eje "y" empieza su origen arriba de la imagen y termina en la parte inferior de la imagen.

El resultado es el siguiente en la figura 2.15:

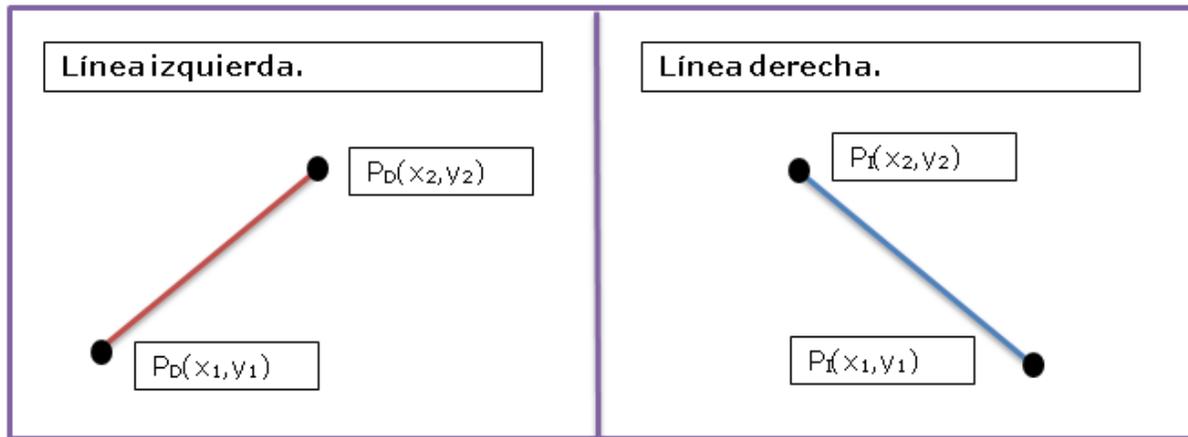


Figura 2.15. Obtención de las líneas con sus respectivas coordenadas.

### 2.1.12. Unir las líneas a la imagen original[13]

Teniendo las dos líneas que indicarán la presencia de un carril, se requiere mostrar la detección al conductor o usuario será necesario colocar estas líneas dentro de la imagen original, ya que la imagen con la que se realizó el procesamiento para detectar los carriles es solo una parte de la imagen (toma de región de interés). Por lo que se unen las dos imágenes para mostrar la detección del carril. Adicionalmente, para dar una mejor presentación al detector de carriles, dentro de las líneas se coloca un polígono convexo, con la ayuda de los puntos de las dos líneas se forma el polígono de cuatro vértices.

Con esto se presenta de mejor forma la detección del carril incluyendo dos indicadores visuales de texto que indiquen la presencia de la detección de las dos líneas con el fin de indicar al conductor que se encuentra estable dentro de su carril como se muestra en la figura 2.16, si alguna línea no es indicada, servirá para advertir al conductor que no se encuentra en su carril establemente o se encuentra cambiando del carril.



Figura 2.16. Resultado final de la detección de carriles.

A pesar de lograr la detección de carriles, en algunos escenarios los resultados para la detección no son los mejores. Debido a que el escenario es un entorno no controlado, esto se debe a las variaciones de iluminación existentes en el entorno, la presencia de sombras, lluvia, asfalto con grietas y cuando los carriles están desvanecidos. No obstante, no se descarta para trabajos futuros, el mejoramiento del programa computacional.

## CAPÍTULO 3

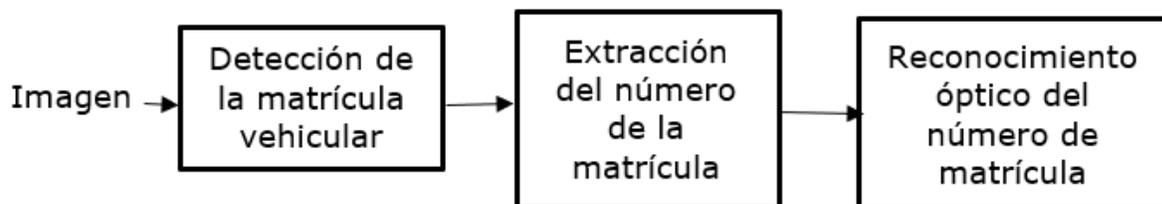
### RECONOCIMIENTO DEL NÚMERO DE MATRÍCULAS

#### Introducción

En el proyecto se desarrolla un programa computacional que utiliza las librerías de visión por computadora (OpenCV) y la API Tesseract-OCR para lograr el reconocimiento del número de matrículas. En dicha detección se utilizarán técnicas de procesamiento de imágenes para obtener el objeto de interés (número de matrícula) dentro de la imagen tomada a un vehículo. Segmentado el número de matrícula se utiliza el reconocimiento óptico de caracteres para reconocer el número de matrícula pasando de una imagen a caracteres UTF-8. En este capítulo se explica el programa computacional y las técnicas usadas dentro del programa.

#### 3.1. Algoritmo desarrollado para el reconocimiento del número de matrícula

El algoritmo se divide en tres etapas como se muestra en el diagrama de la figura 3.1:



---

Figura 3.1. Diagrama del reconocimiento del número de matrícula.

A continuación, se explicarán en detalle las tres etapas.

#### 3.2. Detección de la matrícula vehicular

Esta etapa consta en localizar la placa vehicular dentro de toda la escena de la imagen para que posteriormente sea segmentada, con el objetivo para extraer su número que posee dicha matrícula. Dentro de esta etapa se encuentran diferentes procesos que permiten realizar la detección. En la figura 3.2 se presenta un diagrama del proceso de la detección de la matrícula vehicular:

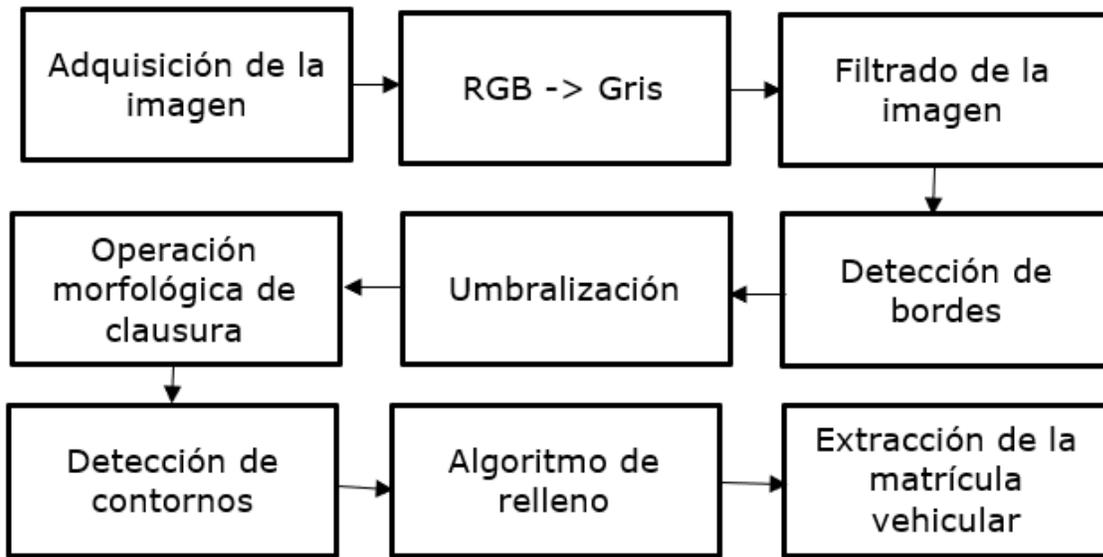


Figura 3.2. Diagrama de bloques del procedimiento para poder detectar la matrícula.

### 3.2.1 Adquisición de la imagen[13][40]

Como cualquier aplicación de visión por computadora, el objeto de estudio son las imágenes las cuales son obtenidas por una video cámara o que sean generadas por una computadora. En el caso de las matrículas vehiculares el escenario a capturar será un automóvil que contenga su respectiva matrícula. Simulando un ambiente real se tomaron fotos utilizando cuatro diferentes celulares con ángulos y tomas distintas.

Tomando una de las fotografías como se muestra en la figura 3.3, se utiliza como escenario de experimentación para desarrollar el algoritmo.



Figura 3.3. Fotografía tomada desde un celular Motorola G.

En la tabla 3.1 se presenta las resoluciones de las cámaras, donde se tomaron las fotografías:

Tabla 3.1. Resoluciones de las cámaras de celulares.

Celular	Marca	Resolución
1	Motorola G	2592x1456
2	IPhone 5	1136 x 640
3	LG-G6	2880 x 1440
4	Moto G <sup>5</sup> Plus	1920x1080

Como se muestra en la tabla 3.1. existen cuatro tipos de resoluciones en las fotos capturadas, por lo que será necesario establecer una resolución fija para todas las imágenes.

Antes de cambiar la resolución de la imagen, se debe primero leer una imagen del banco de fotos tomadas.

Ya leída la imagen se procede a cambiar el tamaño de la imagen, estableciendo las dimensiones de 700x300.

### 3.2.2. Procesamiento de la imagen[40][41]

Con el fin de reducir la complejidad de programa y el tiempo de procesamiento, se opta por trabajar con imágenes en escala de grises.

Anteriormente en el capítulo 2 se mencionaba en qué consistía el filtrado de una imagen. En la experimentación al momento de desarrollar el programa se eligió un filtro de desenfoque Gaussiano de igual manera que para la detección de carriles.

Se reduce el ruido existente, suavizando la imagen. Al aplicar el filtrado se debe elegir el tamaño del kernel el cual fue de 5x5, seleccionado por experimentación realizada al programar.

Como el objetivo de esta etapa es extraer la matrícula del vehículo, los bordes de una imagen determinan que es una placa dentro de una imagen. Con la ayuda del operador de Sobel se detectarán los bordes de la imagen. Los operadores de vecindad, como es el caso del operador Sobel, tienen la propiedad de añadir un pequeño suavizado a la imagen, lo que ayuda en cierta medida a eliminar parte del ruido y, por tanto, a minimizar la aparición de falsos bordes. El operador de Sobel tiene la función de poder detectar dos tipos de bordes en la imagen: bordes verticales y bordes horizontales.

Las matrículas vehiculares contienen una gran cantidad de bordes verticales, considerando que la imagen tomada donde se encuentra la matrícula no está rotada o contenga alguna distorsión de perspectiva. Con esta característica se podrá descartar regiones que no contengan la matrícula vehicular para poder segmentar la placa de la imagen. Con el operador de Sobel, detectaremos los bordes verticales, en la figura 3.4 se muestra la detección de bordes, con el cálculo de la primera derivada horizontal.



---

Figura 3.4. Detección de bordes mediante el operador de Sobel utilizando la primera deriva horizontal.

A continuación, se expresa matemáticamente el cálculo de las derivadas verticales, horizontales en una imagen y el gradiente:

Asumiendo que la imagen es I:

1. Calcular las dos derivadas:

- a) Cambios Horizontales: Será calculado convolucionando I con el kernel  $G_X$  con un tamaño impar. Tomando como un ejemplo un kernel de tamaño 3,  $G_X$  puede ser calculada así:

$$G_X = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad (3.1)$$

- b) Cambios Verticales: Será calculado convolucionando I con el kernel  $G_Y$  con un tamaño impar. Tomando como un ejemplo un kernel de tamaño 3,  $G_Y$  puede ser calculada así:

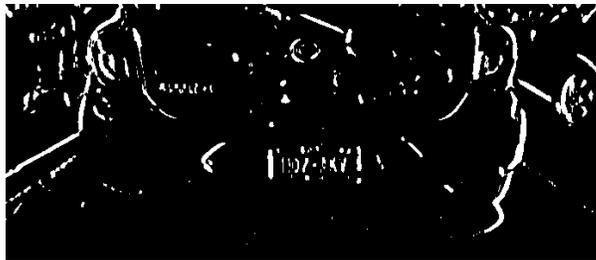
$$G_Y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \quad (3.2)$$

2. En cada punto de la imagen se calcula una aproximación del gradiente a partir de combinar ambos resultados:

$$G = \sqrt{G_X^2 + G_Y^2} \quad (3.3)$$

### 3.2.3. Umbralización de la imagen[40][41]

El siguiente proceso será la binarización de la imagen mediante el método de Otsu, determinando de forma automática el umbral para la segmentación de la imagen. Particionando la imagen en dos clases: objeto y fondo. A diferencia de la manera en que se seleccionó un umbral mediante límites para la segmentación de carriles, en el caso de Otsu automáticamente busca el umbral. En la figura 3.5 se muestra el resultado de aplicar el método de Otsu:



---

Figura 3.5. Resultado de umbralizar la imagen.

El método Otsu tiene sus ventajas y desventajas:

-Ventajas:

- Buena respuesta ante situaciones del mundo real.
- No requiere de la supervisión del humano.

-Desventajas:

- El método será más preciso al dividir la imagen por lo que causaría más gasto computacional el método.

Descripción del método:

-De una imagen en niveles de grises con N pixeles y L posibles niveles diferentes.

---

-Probabilidad de ocurrencia del nivel de gris  $i$  en la imagen:

$$P_i = \frac{f_i}{N} \quad (3.4)$$

$f_i$  -> Frecuencia de repetición del nivel de gris  $i$ -ésimo con  $i = 1, 2, \dots, L$ .

Para el caso particular de umbralización en dos niveles, los pixeles se dividen en dos clases ->  $C_1$  y  $C_2$ , con niveles de gris  $[1, 2, \dots, t]$  y  $[t + 1, t + 2, \dots, L]$  respectivamente, donde sus distribuciones de probabilidad de ambas clases son:

$$C_1: \frac{P_1}{w_1(t)}, \dots, \frac{P_t}{w_1(t)} \quad (3.5)$$

$$C_2: \frac{P_{t+1}}{w_2(t)}, \frac{P_{t+2}}{w_2(t)}, \dots, \frac{P_L}{w_2(t)} \quad (3.6)$$

Donde:

$$w_1(t) = \sum_{i=1}^t P_i \quad (3.7)$$

$$w_2(t) = \sum_{i=t+1}^L P_i \quad (3.8)$$

-Las medidas para cada una de las clases se define como:

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot p_i}{w_1(t)} \quad (3.9)$$

$$\mu_2 = \sum_{i=t+1}^L \frac{i \cdot p_i}{w_2(t)} \quad (3.10)$$

-La intensidad media total de la imagen se define:

$$w_1 \cdot \mu_1 + w_2 \cdot \mu_2 = \mu_T \quad (3.11)$$

$$w_1 + w_2 = 1 \quad (3.12)$$

-Utilizando un análisis discriminante, Otsu definió la varianza entre clases de una imagen umbralizada como:

$$\sigma_B^2 = W_1 \cdot (\mu_1 - \mu_T)^2 + W_2 \cdot (\mu_2 - \mu_T)^2 \quad (3.13)$$

-Finalmente se encuentra el umbral,  $t$ , que maximice la varianza (umbral optimo):

$$t^* = \underset{t}{\text{Max}} \{ \sigma_B^2(t) \} \quad (3.14)$$

Donde:  $1 \leq t \leq L$

#### 3.2.4. Operación morfológica de clausura[44]

El resultado obtenido en la figura 3.6 no permite aun diferenciar la matrícula de los demás objetos que aparecen dentro de la imagen, se requiere de otra herramienta de procesamiento de imágenes. Dicha herramienta será la operación morfológica de cierre o clausura. Las operaciones morfológicas tienen la función de realzar la geometría, simplifican las imágenes y la forma de los objetos.

La operación de cierre enlaza o conecta las regiones más pequeñas y llena sus huecos. Esta operación se realiza mediante la dilatación de la imagen seguido de la erosión, combinando dos técnicas morfológicas. El motivo por el cual se utiliza esta operación morfológica es para remover los espacios en blanco entre los bordes verticales y conectar todas las regiones que contengan una gran cantidad de bordes. Con ayuda de esta técnica se obtienen las posibles regiones candidatas que contengan las matrículas.

El resultado de utilizar la operación morfológica de clausura dentro de la imagen se muestra en la figura 3.6:



Figura 3.6. Resultado de aplicar la operación de cierre.

Este resultado obtenido proporciona los posibles candidatos a ser una placa, no obstante, la mayoría de estas regiones no se encuentran las matrículas.

Matemáticamente:

La operación de clausura o cierre de A por un elemento estructural K se define como:

$$A \bullet K = (A \oplus K) \ominus K \quad (3.15)$$

En otras palabras, esta expresión matemática establece que la clausura de A por K es la dilatación de A por K, seguido de la erosión del resultado por K.

### 3.2.5. Detección de contornos[44]

Todas las regiones candidatas dentro de la imagen serán detectadas, buscando cada contorno de la imagen de la imagen binaria. Al tener un contorno se puede realizar la detección de formas, colores u objetos.

Encontrando los contornos externos de la imagen, almacenando los contornos dentro de un vector.

A partir de los contornos obtenidos, extraer el cuadrado delimitador mínimo o más pequeño de cada contorno. Este cuadrado o caja se define para un conjunto de puntos en “N” dimensiones, con las más pequeñas medidas en sus proporciones (área y volumen), donde todos estos puntos se encuentran dentro de esta caja. Estos puntos serán de cada uno de los contornos.

El proceso de recorrer cada contorno se realiza mediante un vector iterador, además de obtener el cuadrado más pequeño para cada contorno también se realizan algunas validaciones de cada cuadrado, esto con el fin de ir descartando aquellas regiones que no contengan a la matrícula. Estas validaciones se realizan basándose en el área y la relación de aspecto de cada región. Tomando como una relación de aspecto de 2 tomando como referencia las medidas de las matrículas de México  $300/150=2$  (el ancho de la placa dividido entre la altura de la placa) y calculando un área aproximada donde se encuentra la matrícula mediante sus dimensiones en pixeles (altura y anchura). A pesar de tener estos parámetros de validación se tendrán que declarar tanto para la relación de aspecto como del área, seleccionar límites (mínimo y máximo) para que dentro de estos límites validen que región si es una matrícula, esto se debe a que las imágenes tomadas provienen de diferentes cámaras de celulares y algunas imágenes varían en posición de captura, por tal razón se consideran estos límites debido a la variación de las fotos tomadas.

Descartando una gran cantidad de regiones que no contenga la placa, las posibilidades de encontrar la placa serán más, aun así, se considerará utilizar otra técnica de procesamiento de imágenes para dar más robustez al momento de encontrar la matrícula.

### 3.2.6. Algoritmo de relleno[13][44]

Una de las características de las matrículas es tener un fondo blanco, aunque en las matrículas de México poseen una variedad de adornos o diseños, en la mayoría ellas tienen un fondo blanco. Esta característica permitirá reforzar la búsqueda de la matrícula. Utilizando el algoritmo de relleno se obtendrá la región precisa de la matrícula. El algoritmo de relleno por difusión, también llamado algoritmo de relleno determina el área formada por elementos contiguos en una matriz (imagen).

El primer paso para seleccionar el área donde se encuentra la matricula es obtener algunos puntos cerca del centro de cada rectángulo. Mediante el mínimo tamaño de la matricula entre su ancho y alto, para generar puntos aleatorios cercanos al centro del rectángulo. El objetivo de generar estos puntos es para que alguno de ellos toque al menos algún píxel blanco para seleccionar la región en blanco de la placa vehicular.

Para cada punto generado, donde la mayoría de ellos coincidan dentro de la región se dibujará una nueva mascara de imagen donde se almacenará la región más cercana (en este caso será donde se encuentra la placa).

Se llenará una componente conectada con un color dentro de esta máscara empezando desde un punto. Seleccionando un límite inferior y superior de diferencia en cuanto brillo/color entre el píxel a llenar y sus pixeles vecinos.

Una vez que se tenga la máscara donde se encuentra la región seleccionada, en la figura 3.7 se observa el resultado de la máscara. Se volverá a obtener el cuadrado delimitador mínimo para cada mascara, a partir de los puntos que contengan cada mascaré (las regiones seleccionadas) y aplicando otra vez la validación en área y relación de aspecto, como se hizo en los contornos. Se indica su detección al encerrar la región seleccionada en un rectángulo de color rojo como se muestra en la figura 3.8.

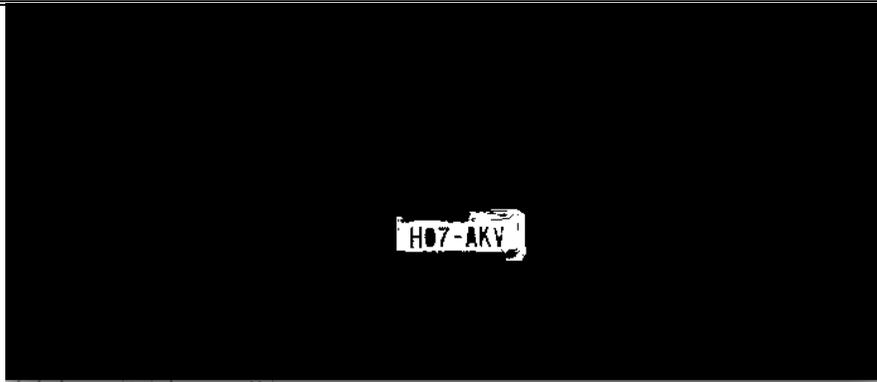


Figura 3.7. Máscara de la imagen donde solo se selecciona la región de la matrícula vehicular.



Figura 3.8. Resultado de la detección de la matrícula dentro de la imagen.

Después de haber obtenido el mejor candidato a ser la matrícula mediante la detección de contornos, el algoritmo de rellenado y hacer las respectivas validaciones en tamaño para descartar regiones. Se cortará dicha región, se removerá alguna posible rotación, redimensionar y se ecualizará la iluminación de la región.

Dentro de las consideraciones a tomar es que en ocasiones existirán más de una región seleccionada a pesar de utilizar las anteriores técnicas mencionadas, este se debe a que puede que estas regiones poseen similares características al momento de seleccionarlas y sean consideradas como candidatos a ser una matrícula vehicular.

### 3.2.7. Extracción de la matrícula vehicular[40][43][44]

Dentro de las correcciones a ser a la región seleccionada será corregir su ángulo de esta.

Se generará una matriz de transformación que servirá para remover posibles rotaciones en la región seleccionada. En ocasiones al aplicar dicha transformación a la región seleccionada, surge el inconveniente de que esta región será rotada 90°, así

que se deberá checar de nueva forma la relación de aspecto de la región y si es menor a 1 será rotada 90°. Con la matriz de transformación, se podrá rotar dicha región mediante una transformación afines (consiste en mapear un rombo en otro rombo como se muestra en la figura 3.9.).



---

Figura 3.9. Ejemplo de una transformación afines.

Matemáticamente se expresa la transformación afine así:

$$R(x, y) = A \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.16)$$

Después de corregir la rotación de la región, se procede a cortar la región y copiar la porción de la imagen dado a un punto central de la región a cortar. Cabe destacar que si la región fue rotada será necesario cambiar las dimensiones de la región (altura y ancho).

Después de ser cortada la región se requiere para el siguiente proceso, que es la detección del texto de la matrícula, normalizar el histograma de esta región cortada. La ecualización del histograma consiste en mejorar el contraste de la imagen ampliando el rango de intensidad. Para finalmente, redimensionar todas las regiones que contengan las matrículas de un mismo tamaño (100x250). A continuación, se muestra el resultado final en la figura 3.10 del reconocimiento de la matrícula.



---

Figura 3.10. Extracción de la matrícula vehicular.

### 3.3. Extracción del número de la matrícula

Además de usar la librería OpenCV para realizar el procesamiento de la imagen, la librería Tesseract-OCR servirá para reconocer los caracteres de la matrícula desde la imagen. Esta librería tiene un buen desempeño al reconocer los caracteres, aunque presenta limitaciones al momento de leer o escanear la imagen.

Existen algunos escenarios donde la librería Tesseract no reconoce de manera correcta:

- El texto contenga ruido o manchas.
- El texto este rotado y no esté alineado aproximadamente a 90°.
- La fuente del texto no sea conocida para la librería.

Teniendo en cuenta estos puntos y tomando en cuenta que las placas de México contienen una gran variedad de fondos y algunas tienen marcos por lo que al binarizar la placa, aparecerán manchas u objetos que afecten el reconocimiento. Considerando todo lo anterior solo se selecciona una parte de la imagen de la placa donde se encuentra el texto.

Esta etapa consta en localizar el texto dentro de toda la matrícula, para mejorar el reconocimiento del número de matrícula realizado por la librería Tesseract-OCR. Dentro de esta etapa se encuentran diferentes procesos que permiten realizar la detección del texto. En la figura 3.11 se muestra el diagrama de bloques de la extracción de la matrícula:

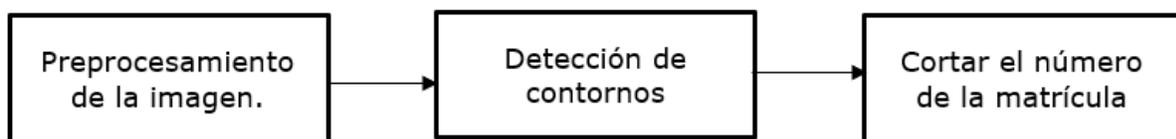


Figura 3.11. Diagrama de bloques de la extracción del número de la placa vehicular.

#### 3.3.1. Detección de contornos[13][43][44]

A la matrícula obtenida se le aplicará un preprocesamiento para que pueda ser más fácil la segmentación o extracción del texto. Se le cambiará su espacio de color a escala de grises, después se binarizará (aplicando un umbral duro) y se detectaran los bordes por medio de Canny.

Teniendo solamente los bordes de la imagen se procederá a detectar sus contornos de la misma manera que en la sección 3.2.7. Se vuelven a detectar los contornos, como se muestra en la figura 3.12. (a) y a partir de un vector iterativo se recorrerán los contornos con el objetivo de ir seleccionando a cada contorno en regiones, estas regiones serán rectángulos que contenga a cada contorno externo de la imagen. Dentro de la figura 3.12. (b) se ilustran las regiones seleccionadas con rectángulos de color verde.

No todos los rectángulos obtenidos serán necesarios ya que no contienen los caracteres del número de la matrícula, por lo que se procede a realizar algunas validaciones a cada uno para ir descartando todos aquellos que no cumplan con las medidas en área y relación de aspecto de los rectángulos que contienen los contornos de los caracteres de la matrícula. Las medidas en relación de aspecto y área se realizaron desde las imágenes de las matrículas recortadas, haciendo un estimado mediante todas las dimensiones de los caracteres de todas las matrículas (ya que, aunque todas las matrículas obtenidas por el anterior proceso 3.2. se redimensionaron con dimensiones fijas, pareciera que algunas matriculas se hubieran alargado debido a que las imágenes donde se encontraban las placas fueron tomadas en diferentes distancias, al momento de extraer las matrículas se tengan diferentes dimensiones) se establecerán límites entre el área y la relación aspecto donde se descartarán aquellos rectángulos que no se encuentren dentro de estos límites y solo se obtendrán los que se encuentren dentro de estos límites (serán aquellos que contengan a los caracteres del número de matrícula, como se muestra en la figura 3.12. (c)) a pesar de eso en ocasiones, algún rectángulo que no contenga un carácter y se encontraran dentro de estos dos límites y será considerado como un contorno valido, pero esto no afectará mucho en la detección del texto de la matrícula.



*Figura 3.12. Pasos para detectar los caracteres de la matrícula: a) detección de contornos (en azul). b) encerrando los contornos dentro de rectángulos (color verde).c) selección de los rectángulos que contengan a los caracteres (color naranja).*

### 3.3.2. Cortar el número de la matrícula[13]

Teniendo los rectángulos que encierran los caracteres de la matrícula, se tiene sus puntos (vértices) de cada uno, todos estos puntos se unirán para formar una nueva región de interés.

Uniéndolos todos estos puntos para generar una nueva área rectangular la cual contenga a todos los caracteres de la matrícula, se tendrán los cuatro vértices del rectángulo. Con estos vértices se dibujará la nueva región. En la figura 3.13 se muestra la región dibujada dentro de la imagen de la matrícula vehicular:



---

Figura 3.13. Detección del número de la matrícula.

Se ha logrado la detección del número de matrícula, ahora se extraerá la región seleccionada de la imagen, aunque en algunas ocasiones las regiones seleccionadas superan las dimensiones de las imágenes de las matrículas, por lo que se considera que cuando suceda esto se generará un error al momento de usar la imagen cortada ya que puede ser que no se corte correctamente los números, es por eso que se consideran dos casos al momento de cortar el número: cuando la región sobrepasa la imagen y cuando se encuentra dentro de la misma. Si la región está afuera de la imagen, se tendrá que cambiar las dimensiones de la imagen. En la figura 3.14 se muestra la extracción del número de la matrícula vehicular:



---

Figura 3.14. Imagen del número de la matrícula.

Teniendo el número de matrícula segmentado, se procederá al último paso para el reconocimiento del número de matrícula.

#### 3.4. Reconocimiento óptico del número de matrícula[43]

La librería Tesseract-OCR proporcionará el reconocimiento óptico de los caracteres del número de placa. Como tal Tesseract es un motor para el reconocimiento óptico de caracteres con la capacidad de reconocer más de 100 idiomas y con capacidad de entrenar nuevos idiomas o un nuevo texto con un nuevo tipo de fuente.

El reconocimiento óptico de caracteres se divide en los siguientes pasos:

### Capítulo 3

#### Reconocimiento del número de matrículas

---

---

-Preprocesamiento y segmentación del texto: se deberá eliminar el posible ruido existente y las rotaciones que tenga el texto e identificar en qué áreas de la imagen se encuentra el texto.

-Identificación del texto: identifica a cada letra en el texto.

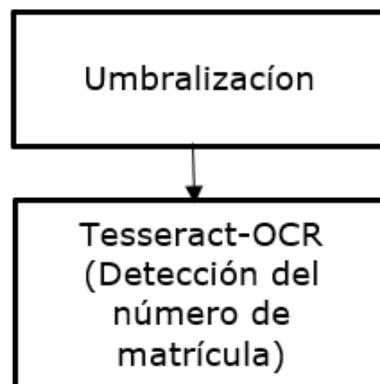
El preprocesamiento y segmentación dependerá del estado del texto, ya que existen situaciones en donde no es necesario aplicarlas:

-Aquel texto que se encuentre en documentos, libros, apuntes etc. Constan de un fondo blanco y el texto está de color negro. Además, su texto está alineado. De tal manera que no será necesario realizar estos procesos, no obstante, existe texto con ruido, el cual será necesario reducir.

-Texto generado desde una computadora, este caso es el más ideal para el reconocimiento de texto, al igual que el otro caso tiene un fondo blanco, en general letras en negro y se encuentra alineado. La posibilidad de ruido es casi nula.

En el caso de las matrículas, será necesario realizar el preprocesamiento y la segmentación del texto como se vio en la sección 3.3. Teniendo el texto segmentado se procede a identificarlo.

En la figura 3.15 se presenta un diagrama de bloques del proceso para reconocer el número de matrícula vehicular.



---

Figura 3.15. Diagrama de bloques de la detección del número de matrícula.

### 3.4.1. Umbralización del texto[40]

La imagen cortada se encuentra en escala de grises, se procederá a segmentar el texto del fondo de la imagen. Facilitando el reconocimiento óptico del texto. En la figura 3.16 se muestra la binarización:



---

Figura 3.16. Imagen binarizada del número de matrícula para facilitar la detección y reconocimientos de los caracteres.

Se vuelve a utilizar una umbralización de la imagen, a diferencia de las otras ocasiones donde se usó la umbralización, la selección del umbral será seleccionada desde el mismo programa (ventana donde se muestra la imagen segmentada) con el uso de barras de seguimiento, debido a que, en las pruebas realizadas en las imágenes, la variación de iluminación dentro de las imágenes es variante y no será el mismo umbral para todas las imágenes. Variando el umbral de la imagen se seleccionará el mejor umbral para que la librería Tesseract pueda reconocer los caracteres.

### 3.4.2. Aplicando Tesseract-OCR para el reconocimiento del número de matrícula [31][43][45]

Como tal la librería Tesseract está escrita en c++, por lo que incorporarla dentro del programa no será un problema, esta permitirá el reconocimiento óptico de un texto. A continuación, se describe las funciones utilizadas para el reconocimiento óptico de caracteres.

En primer lugar, se debe indicar el lenguaje español, también es necesario indicar que tipo de caracteres a identificar, se selecciona los siguientes caracteres: "ABCDEFGHJKLMNPRSTUVWXYZ0123456789", se descartan los caracteres en minúsculas y las letras Q y O, con base en la **Norma oficial mexicana NOM-001-SCT-2-2000**, se menciona que las letras Q y O, se pueden confundir con el número 0 de tal manera que no existen dentro de los caracteres que puedan poseer una matrícula.

Para poder leer y reconocer una imagen dentro de Tesseract-OCR, se pasará toda la imagen al motor Tesseract con el fin de que dicho motor conozca las características de la imagen a reconocer el texto que contenga la imagen. El texto reconocido es devuelto y codificado con UTF-8.

### Capítulo 3

#### Reconocimiento del número de matrículas

UTF-8 (Unicode Transformación Format-8) es un formato de codificación de caracteres, en el que cada carácter es representado por un octeto (8 bits). También existen UTF-16 y UTF-32. La ventaja de UTF-8 respecto a estos otros es que es compatible con versiones anteriores de ASCII.

Finalmente se muestra dentro de la imagen original, el resultado final del reconocimiento del número de matrícula vehicular desplegando el número en letras de color azul como se muestra en la figura 3.17.



Figura 3.17. Resultado final del reconocimiento del número de matrícula.

## CAPÍTULO 4

### PRUEBAS Y RESULTADOS

#### Introducción

El presente capítulo contendrá todas las pruebas realizadas a los dos programas.

Primero se presentarán las pruebas realizadas al detector de carriles, las cuales son videos tomados desde la plataforma YouTube y también videos capturados desde una video cámara montada en un vehículo. Todas las pruebas realizadas tienen la finalidad de evaluar la precisión al momento de detectar las líneas del carril y el tiempo de ejecución de las pruebas (si el procesamiento está en tiempo real o en vivo).

Seguido de las pruebas realizadas al detector de carriles, se presentan las pruebas realizadas al programa de reconocimiento del número de la matrícula vehicular con fotos tomadas desde diferentes celulares. Con la finalidad de evaluar la precisión del mismo programa para reconocer los números de las matrículas.

#### 4.1. Pruebas al detector de carriles[46][47]

A continuación, se describen todas aquellas pruebas hechas a los videos tomados:

-El primer video donde se probó para la detección de carriles fue tomado desde la plataforma de videos YouTube, por lo que se desconoce el lugar donde fue tomado y la posición de la cámara, pero sus dimensiones se conocen: 640x320. Estas características son importantes para el desarrollo del programa, en el caso de la posición de la cámara se debe tener una posición fija, ya que al seleccionar la región de interés es fundamental para detectar las líneas del carril. Tener una variedad de videos con diferentes posiciones, afecta al seleccionar la región de interés será diferente y en cada video se debería cambiar la posición de la toma de interés.

En este caso el propósito este video fue para probar el algoritmo, mientras en el caso de los videos propios (tomados desde una video cámara), tienen la finalidad de probar el programa desde un entorno real.

En la figura 4.1. se muestra la primera prueba:



Figura 4.1. Primera prueba realizada.

Anteriormente en el capítulo 2, se estableció el número de cuadros por segundo con una tasa de 24 cps y las dimensiones del video (320x234 pixeles), además del beneficio de tener a los videos con las mismas dimensiones para tener la misma posición de la toma de región de interés, se tiene el beneficio de tener una mejor velocidad al momento de reproducir el video de la misma manera para el caso de tener menos cuadros por segundo proporcionará una mejor fluidez al momento de procesar el video. OpenCV tiene un diferente rendimiento para procesar un video desde la memoria (almacenado) y un video grabado directo desde una video cámara ya que será más fluido al momento de ser procesado.

Para evaluar su eficiencia de detectar las líneas del carril, se consideró evaluar el video de manera cualitativa, esta evaluación se realizó visualmente ya que el ir contando todas las líneas que se detecten afectaría el rendimiento del video en cuestiones de velocidad. Visualmente se comprobó que su eficiencia al momento de detectar las líneas de carril es buena, en pocas ocasiones se altera la localización de la línea debido a la presencia de líneas vecinas. Un gran factor por lo que la detección fue buena, era que el escenario era una carretera recta, los carriles estaban bien definidos, no hay presencia de grietas y hay una buena iluminación permitiendo la posibilidad de utilizar la segmentación por selección de umbral.

En la figura 4.2 se presenta una gráfica se dé la evaluación de la detección de carriles en un minuto del video, revisando si en cada segundo los 24 cuadros se visualizaba una detección correcta de carriles, o si solo se detectaba una línea o ninguna de las dos líneas.

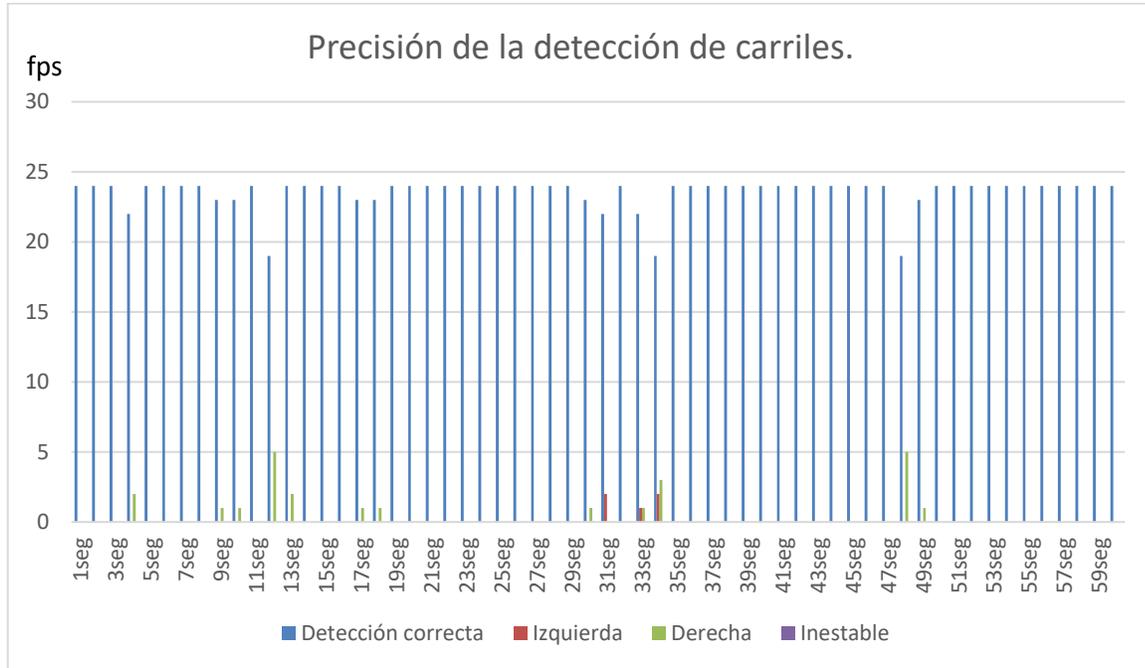


Figura 4.2. Precisión de la detección de carriles.

Para la evaluación de la velocidad de ejecución, se mide el tiempo en que tarda el programa en realizar el procesamiento y se compara con el tiempo de reproducción del video original.

El tiempo de duración del video es de 2.56 minutos y el tiempo que tarda el programa con el video es de 1.911 minutos. La diferencia que existe entre los dos es de 0.649 minutos. Al comparar tiempos de reproducción sin procesamiento y con procesamiento, se nota que el video ejecutado con el procesamiento es más rápido. Este resultado indica que al realizar la detección de carriles en vivo se tendrá la posibilidad de tener un video fluido (en tiempo real).

Esta diferencia demuestra que los cambios en resolución y cps al video permitieron que fuera más rápido con el procesamiento.

-La siguiente prueba también fue tomada desde la plataforma de YouTube, en esta ocasión el video es tomado desde una video cámara en una carretera de México en el día. También se desconoce en donde se colocó la video cámara, pero se conoce las dimensiones del video (640x360 pixeles).

En la figura 4.3. se muestra la segunda prueba:



Figura 4.3. Segunda prueba realizada.

El escenario del video es en la tarde y en la carretera se presenta curvas las cuales en ocasiones la región de interés no aparece las líneas del carril debido a que la selección de región de interés tiene forma de un romboide y es muy difícil tomar las dos líneas del carril al mismo tiempo por momentos ya que puede que su curvatura no permite visualizar a la línea dentro de esta área. En la figura 4.4. se muestra un ejemplo de cómo en ocasiones una de las líneas de carril no es detectada ya que la curvatura del carril no permite visualizarla la línea del carril dentro de la región de interés seleccionada.



Figura 4.4. Dificultad de detectar la línea del carril izquierdo debido a la curvatura de la carretera.

Otro inconveniente que existió para la detección de los carriles dentro de esta prueba fue debido a las condiciones de iluminación que, al momento de segmentar los carriles de la imagen, no hubiera una posibilidad para aplicar la técnica de segmentación dentro de este video. Por lo que solo se utiliza únicamente la detección de bordes, con esta técnica si se segmentan los carriles, pero también aparecen bordes de otros objetos no deseados (como grietas o manchas en el asfalto), alterando la detección de los carriles cuando aparezcan estos objetos dentro de la región de interés. No obstante,

se obtiene una detección de carriles dentro del video aceptable a pesar de tener estos problemas, la detección permanece constante dentro de la escena.

En la figura 4.5 se presenta una gráfica se dé la evaluación de la detección de carriles en un minuto del video, revisando si en cada segundo los 24 cuadros se visualizaba una detección correcta de carriles, o si solo se detectaba una línea o ninguna de las dos líneas.

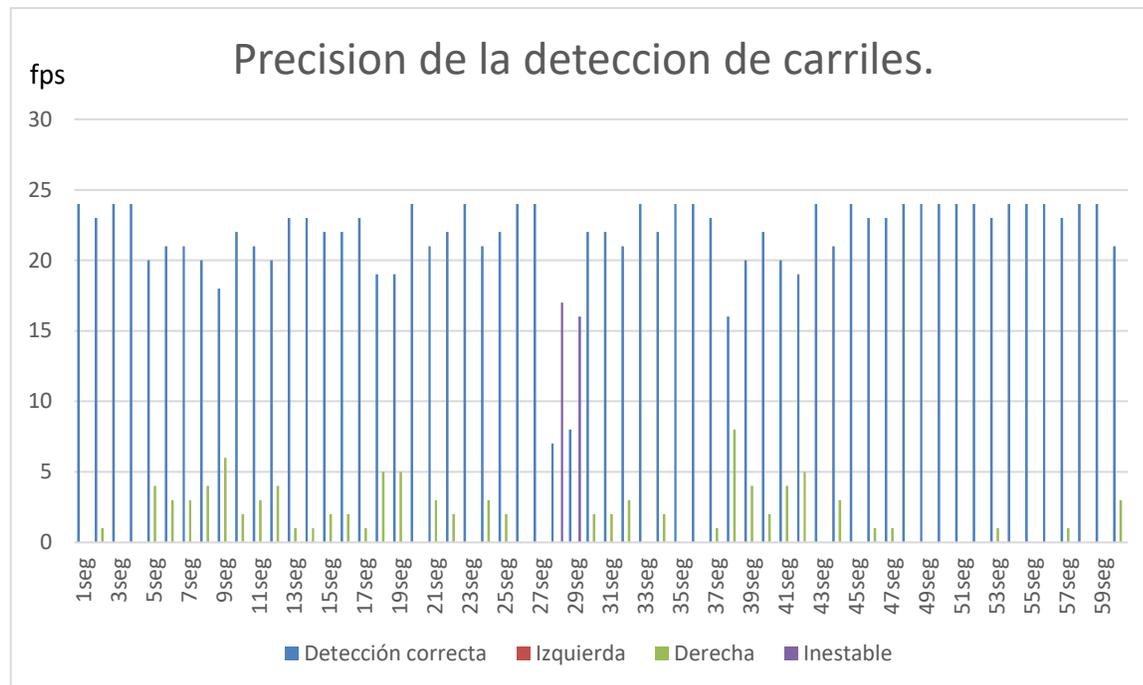


Figura 4.5. Precisión de la detección de carriles.

Para la evaluación de la velocidad de ejecución, se procede a hacer lo mismo que en la anterior prueba, comparar la duración del video sin procesar con la del video procesado.

El tiempo de duración del video es de 3.59 minutos y el tiempo que tarda el programa con el video es de 2.0293 minutos. La diferencia que existe entre los dos es de 1.5607 minutos. Al comparar tiempos de reproducción sin procesamiento y con procesamiento, se nota que el video ejecutado con el procesamiento es más rápido. Este resultado indica que al realizar la detección de carriles en vivo se tendrá la posibilidad de tener un video fluido (en tiempo real).

-La tercera prueba fue realizada desde una grabación tomada desde una cámara web montada dentro de un vehículo. En este caso se conoce la posición donde se encuentra la cámara y la resolución se fija desde la grabación (320x234 pixeles). Siendo la primera prueba realizada desde un video propio.

Para montar la video cámara dentro del vehículo se considera que se encuentre la cámara posicionada arriba y al centro del tablero del vehículo como se observa en la figura 4.6.



*Figura 4.6. Fotografías de la colocación de la video cámara.*

En la figura 4.7. se muestra la video cámara a usar:



*Figura 4.7. Cámara web modelo: Logitech B525 HD WEBCAM.*

Después de colocar la video cámara se procede a grabar un escenario en la ciudad de México.

La grabación fue tomada en la avenida Insurgentes norte, en la tarde. La vialidad posee carriles, pero en esta ocasión los carriles no están tan bien marcados (desvanecidos) y la misma vialidad posee grietas y baches que alteran al escenario, teniendo la presencia de objetos no deseados al momento de realizar la detección de carriles. Al igual que la anterior prueba el uso de la función de umbralización no fue efectiva para segmentar los carriles, de tal forma que solo se usa la detección de bordes Canny para segmentar los carriles, aunque la presencia de estos objetos dentro de la escena, aparecerán en la detección de bordes, desestabilizando en ocasiones la detección de carriles. La detección de carriles dentro del video a pesar de tener estos problemas

tiende a detectar casi siempre los carriles dentro de la escena. En la figura 4.8. se muestra la tercera prueba:



Figura 4.8. Tercera prueba realizada.

En la figura 4.9 se presenta una gráfica se dé la evaluación de la detección de carriles en un minuto del video, revisando si en cada segundo los 24 cuadros se visualizaba una detección correcta de carriles, o si solo se detectaba una línea o ninguna de las dos líneas.

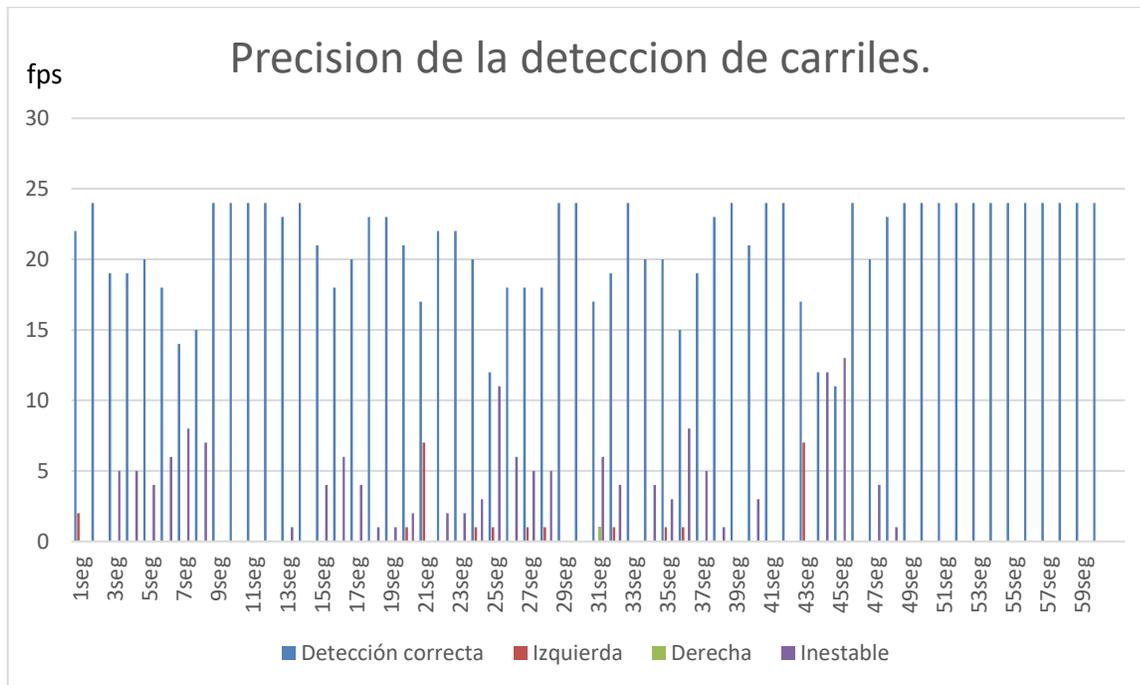


Figura 4.9. Precisión de la detección de carriles.

Para la evaluación de la velocidad de ejecución, comparando la duración del video sin procesar con la del video procesado se tiene un tiempo de duración del video sin procesar de 2.37 minutos y el tiempo que tarda el programa con el video es de 0.5556 minutos. La diferencia que existe entre los dos es de 1.8144 minutos.

El tiempo que tarda en procesar el video se encuentra en condiciones para transmitir en vivo, teniendo otro resultado positivo que demuestra que el programa es rápido al momento de detectar los carriles, demostrando que el programa puede ser utilizado ya en un entorno en vivo evitando accidentes debido a que el programa no sea tan rápido, ya que al tener un retraso en video podría generar problemas al manejar ya que no coincidiría lo que está en pantalla y lo que está pasando en ese momento en el carro. Esto también se comprueba en la comparación del estado de arte con la detección de carriles, mostrada en el capítulo 1. Donde en la mayoría de los artículos el tiempo para ejecutar la detección de carriles es mayor que en este trabajo (con la excepción del último artículo en el que su tiempo de ejecución es menor al de este proyecto), no obstante, esta comparación reafirma que el programa puede ejecutarse desde una video cámara en vivo.

Se considera para trabajos futuros reforzar más la detección de carriles para evitar la presencia de desestabilizaciones y en las comparaciones realizadas la detección en otros artículos es mejor, no obstante, entre más tenga carga computacional el tiempo de ejecución será afectado y también la detección en este proyecto es menor pero no se aleja tanto la precisión de este proyecto de tesis con las referencias comparadas. Considerando lo anterior, el programa se encuentra en condiciones para ser utilizado desde un vehículo en tiempo real.

#### 4.2. Pruebas al reconocedor del número de matrícula[39]

A continuación, se muestran todas las pruebas hechas a las fotografías tomadas:

En el capítulo 3 se mencionó que las fotografías fueron tomadas por varios celulares, dentro del mismo programa se estableció una dimensión fija a las fotos de 700x300 pixeles. Las fotos fueron tomadas desde la parte trasera y delantera de los vehículos, con una distancia entre la cámara y el vehículo aproximadamente de 1 a 2 metros procurando que en las imágenes las placas se encuentren no rotadas como se observa en la figura 4.10.



Figura 4.10. Ejemplo de captura de una matrícula.

Todas estas fotos fueron probadas con el programa computacional, evaluando la precisión del programa al momento de detectar la placa vehicular, extraer el número de matrícula y reconocer sus caracteres. Se tomaron 70 fotos de vehículos con placas vehiculares, dentro de una tabla que se encuentra en la sección de anexos se mostrará la foto original, su detección de la matrícula, con su extracción del número de matrícula y el reconocimiento óptico del número de matrícula.

La siguiente tabla 4.1. presenta la evaluación del programa con las 70 fotografías.

Tabla 4.1. Evaluación del programa computacional.

Operación	Muestras	Éxito	Fallo	%Éxito
Detección de la matrícula vehicular	70	60	10	85.71%
Extracción del número de la matrícula	60	57	3	95%
Reconocimiento óptico del número de matrícula	57	46	11	80.70%

Los resultados obtenidos demuestran que el programa computacional logra tener un desempeño del 65.71% al momento de reconocer los números de las matrículas, ya que las fotos usadas en el programa fueron tomadas en diferentes posiciones y con diferente iluminación, el programa trabaja de manera satisfactoria. También al momento de realizar las pruebas se observó que uno de los problemas para reconocer el número de matrículas es cuando las matrículas poseen un marco que en ocasiones cubre los caracteres obstruyendo el reconocimiento. El desempeño de la librería Tesseract-OCR en el reconocimiento de matrículas, en la mayoría de las ocasiones fue del 80.70% de precisión, no obstante, en algunos casos cuando las matrículas poseían los caracteres “W” y “M”, debido al tipo de fuente de letra que utilizan las matrículas, Tesseract confunde que caracteres son estas dos letras. Para mejorar su precisión ante

el reconocimiento del número de placas, será necesario entrenar a esta librería con el tipo de fuente de letra que poseen las matrículas para un mejor desempeño.

#### 4.3. Limitaciones del trabajo

Teniendo como referencia las pruebas y las comparaciones realizadas se tiene una referencia para mencionar las limitantes que poseen los dos programas computacionales:

-Detección de carriles:

- El programa en ocasiones es inestable al momento de detectar un carril si el escenario contiene carriles desvanecidos, ya que el objeto de estudio son los carriles.
- El programa en escenarios donde exista una gran variación de iluminación presenta por momentos inestabilidad.
- La detección está diseñada para un modelo geométrico de carriles rectangular, por lo que, en carriles curvados, le cuesta trabajo detectar las líneas entre más tiendan a ser más curvadas.

-Reconocimiento del número de matrícula vehicular:

- El programa al tener un 85.71% en la detección de la matrícula, presenta dificultades para detectar cuando hay variaciones de iluminación, por ejemplo: si se presenta un brillo cercano dentro de la matrícula puede afectar el contorno de esta ya que lo altera y será más difícil detectar el posible candidato.
- El desempeño de la librería Tesseract-OCR no era tan preciso, cuando las matrículas poseían los caracteres “W” y “M”, debido al tipo de fuente de letra que utilizan las matrículas, Tesseract confunde que caracteres son estas dos letras.
- Las fotografías tomadas de la experimentación están limitadas en distancia entre la cámara y el vehículo aproximadamente de 1 a 2 metros, si se cambia la distancia podría no haber una detección de la matrícula.
- Con el anterior comentario, este programa está diseñado para trabajar solo con imágenes y no con un video.

## CAPÍTULO 5

### CONCLUSIONES

#### Introducción

El presente capítulo contiene las conclusiones del trabajo. Dando un final al trabajo presente, comentando si se cumplieron los objetivos particulares, así como el objetivo general. A partir de esto también se podrá dar algún comentario para el mejoramiento del trabajo futuro con base a los resultados obtenidos.

Se logró desarrollar los respectivos programas que conforman al sistema para la asistencia vehicular utilizando visión por computadora, los cuales fueron el detector de carriles y el programa para el reconocimiento del número de placas vehiculares.

En el caso del detector de carriles, se logró que desde un video grabado o desde una video cámara en tiempo real, una detección de los mismos carriles, teniendo en ocasiones en escenarios reales, problemas con la detección debido a variaciones de iluminación o por las condiciones del asfalto de la cartería (carriles desvanecidos). No obstante, el tiempo de ejecución del programa funciona para que se utilice desde una video cámara montada en un vehículo grabando en vivo. También a pesar de tener algunas veces inconvenientes en la detección la mayoría de las veces el programa demuestra detectar los carriles para indicar al conductor que se encuentra dentro del mismo carril. Se considera mejorar el programa computacional para que sea más robusto en el momento que se utilice en aplicaciones reales donde se requiera un sistema lo más confiable en respuesta y en tiempo como es en el caso de los autos autónomos. Otra consideración importante para el desarrollo y aplicación de sistemas para la asistencia vehicular en el territorio nacional es que este tipo de sistemas trabajan de mejor forma en escenarios donde la calidad de las carreteras o vialidades es buena, teniendo en respuesta mejores resultados para la detección de carriles. Los escenarios especialmente en la ciudad de México son más difíciles debido a las inconsistencias viales (como es el desvanecimiento de carriles) dentro las vialidades, a pesar de esto se puede también considerar al detector de carriles para utilizarse en los metrobuses los cuales contienen de mejor forma su carril.

Para el caso del reconocedor de número de matrículas se obtuvieron resultados no tan buenos como el caso del detector de carriles, aunque a pesar de tener un porcentaje del 65.71% de precisión para reconocer los números de las matrículas, se considera que este resultado es alentador para dar mejoras al programa, ya que si reconocieron

muchas matrículas por lo que se considera trabajar con un clasificador diferente al proporcionado por Tesseract-OCR, ya que este trabaja de mejor manera con un texto mejor segmentado como es en los documentos de texto. De igual manera al trabajar en un entorno no controlado el cual fue en estacionamiento al aire libre, se tiene diferentes capturas de las imágenes con diferentes iluminaciones, así como es en el caso de las matrículas que se encuentran desgastadas o con marcos los cuales afecten el reconocimiento.

Teniendo los dos programas que conforman el prototipo para la asistencia vehicular en el que se puede agregar más programas computacionales que proporcionen otro tipo de asistencia como es el detector de peatones, el detector de cansancio o de señales viales, etc. Ya que el desarrollar un sistema de esta magnitud conlleva un mayor tiempo para incluir más aplicaciones y que puedan ser probadas dentro de un vehículo, utilizando una microcomputadora o un ordenador de placa reducida. Por lo que el trabajo aún está en una fase de prototipo el cual en trabajos futuros podrá estar en una fase de desarrollo para su uso en vehículos. Siendo este tipo de sistemas parte de la 4° revolución industrial, donde el desarrollo de nuevas aplicaciones facilita las vidas de las personas, como son las tecnologías desarrolladas por la visión por computadora.

---

---

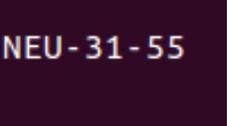
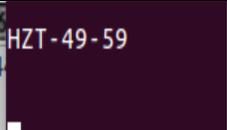
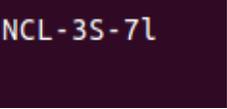
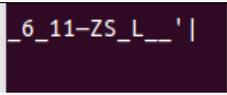
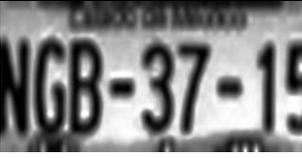
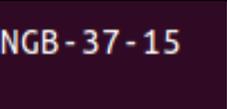
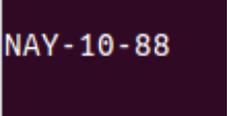
## REFERENCIAS

- [1] Carlos Platero, "Introducción a la Visión Artificial," in Apuntes de Visión Artificial, 2009, pp. 27–32.
- [2] R. Gonzalez and R. Woods, Digital image processing, 3°. 2002.
- [3] "Robot Vision Lighting: Why There's No Perfect Setup." [Online]. Available: <https://blog.robotiq.com/robot-vision-lighting-why-theres-no-perfect-setup>. [Accessed: 30-Jan-2018].
- [4] K. Pulli, N. Research, A. Baksheev, I. K. Korniyakov, and V. Eruhimov, "HCI 1 Realtime Computer Vision with OpenCV."
- [5] W. Haberecht, "Digital Signal Processing,." pp. 1–4, 1990.
- [6] Carlos Platero, "Introducción al Procesamiento digital de Señales," Madrid, p. 69.
- [7] "2. Imagen," in Conceptos básicos de imagen digital Diseño de materiales multimedia, p. 8.
- [8] S. Annadurai and R. Shanmugalakshmi, Fundamentals of digital image processing. Pearson, 2007.
- [9] C. Teórico Práctica N° and R. Wainschenker Mg Ing José María Massa Mg Ing Paula Tristan, "Procesamiento Digital de Imágenes Optativa Área Procesamiento de señales Primer cuatrimestre de 2011."
- [10] A. Bartolomé, "Video Digital," Comunicar, no. 021, pp. 39–48, 2003.
- [11] J. James, "Frame Rate Conversion Simplified An example of a soccer ball in motion."
- [12] "OPENCV." [Online]. Available: <https://opencv.org/>.
- [13] A. Kaehler and G. R. Bradski, Learning OpenCV 3 : computer vision in C++ with the OpenCV library. .
- [14] "Qt." [Online]. Available: <https://www.qt.io/qt-features-libraries-apis-tools-and-ide/>.
- [15] "El Sistema ADAS ayuda a prevenir accidentes de tráfico." [Online]. Available: <https://www.seguridad-vial.net/vehiculo/seguridad-pasiva/156-el-sistema-adas-ayuda-a-prevenir-accidentes-de-trafico-a-los-conductores>. [Accessed: 30-Jan-2018].
- [16] "El control de cruce adaptativo ¿es práctico o engorroso?" [Online]. Available: <https://www.motorpasion.com/tecnologia/control-cruce-adaptativo-acc>. [Accessed: 30-Jan-2018].
- [17] "Honda Sensing: tecnologías de seguridad activa imprescindibles para el nuevo Civic." [Online]. Available: <http://www.lavanguardia.com/motor/innovacion/20161212/412566346029/honda-civic-honda-sensing-tecnologia-seguridad-activa.html>. [Accessed: 30-Jan-2018].
- [18] "About ALPR." [Online]. Available: <http://www.theiacp.org/ALPR-About>. [Accessed: 30-Jan-2018].
- [19] A. M. Díaz, F. Luis, Q. Pereda, E. Francisco, and M. Grajeda, "ALGUNAS CONSIDERACIONES DE SEGURIDAD PARA EL PROYECTO GEOMÉTRICO DE CARRETERAS."
- [20] "Marcas viales," 2014. .

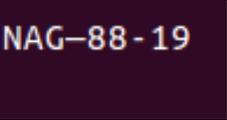
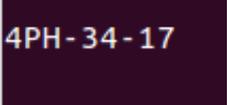
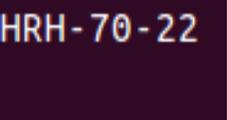
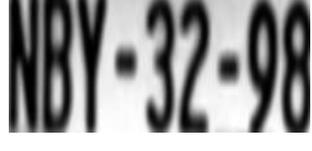
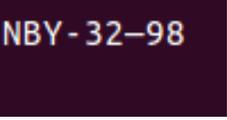
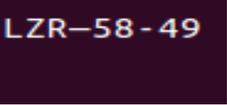
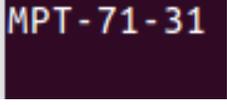
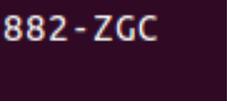
- 
- 
- [21] "MANUAL DEL CONDUCTOR DE VIRGINIA."
- [22] Á. M. Ballester and X. L. Meseguer, "Detección de Líneas y Sistema de Estabilidad de Carril basado en cámara frontal," *Enginy*, no. 2, pp. 33–36, 2010.
- [23] C. Guo, S. Mita, and D. Mcallester, "Lane Detection and Tracking in Challenging Environments based on a Weighted Graph and Integrated Cues," 2010 IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 5543–5550, 2010.
- [24] G. Liu, F. Wörgötter, and I. Markelić, "Combining statistical hough transform and particle filter for robust lane detection and tracking," *IEEE Intell. Veh. Symp. Proc.*, pp. 993–997, 2010.
- [25] P. M. Daigavane and P. R. Bajaj, "Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization," 2010 3rd Int. Conf. Emerg. Trends Eng. Technol., pp. 76–80, 2010.
- [26] Y. C. Leng and C. L. Chen, "Vision-based lane departure detection system in urban traffic scenes," 11th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2010, no. December, pp. 1875–1880, 2010.
- [27] X. Yang and Z. Ling, "RESEARCH ON LANE DETECTION TECHNOLOGY BASED ON OPENCV Xu Yang 1," 3rd Int. Conf. Mech. Eng. Intell. Syst., no. Icmeis, pp. 994–997, 2015.
- [28] S. Sharma and D. J. Shah, "A Much Advanced and Efficient Lane Detection Algorithm for Intelligent Highway Safety," *Comput. Sci. Inf. Technol. ( CS IT )*, pp. 51–59, 2013.
- [29] U. Ozgunalp and N. Dahnoun, "Robust lane detection & tracking based on novel feature extraction and lane categorization," *Acoust. Speech Signal ...*, pp. 8129–8133, 2014.
- [30] Y. Li, A. Iqbal, and N. R. Gans, "Multiple lane boundary detection using a combination of low-level image features," 2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014, no. October, pp. 1682–1687, 2014.
- [31] S. de C. y Transportes, "Norma oficial mexicana NOM-001-SCT-2-2000," 2001.
- [32] C. J. Sánchez Fernández and V. Sadonís Consuegra, "Reconocimiento Óptico de Caracteres (OCR)," *Univ. Carlos III*, p. 7, 2009.
- [33] A. Goyal and R. Bhatia, "Various Techniques for Number Plate Recognition-A Review," *Int. J. Comput. Appl.*, vol. 143, no. 11, pp. 975–8887, 2016.
- [34] A. Badr, M. M. Abdelwahab, A. M. Thabet, and A. M. Abdelsadek, "Automatic Number Plate Recognition System," *Ann. Univ. Craiova, Math. Comput. Sci. Ser.*, vol. 38, no. 1, pp. 62–71, 2011.
- [35] D. Bhardwaj and S. Mahajan, "Review Paper on Automated Number Plate Recognition Techniques," *Int. J. Emerg. Res. Manag. &Technology*, vol. 4, no. 5, pp. 319–324, 2015.
- [36] G. T. Sutar, A. V Shah, and P. G. Student, "Number Plate Recognition Using an Improved Segmentation," *Int. J. Innov. Res. Sci. Eng. Technol. (An ISO Certif. Organ.)*, vol. 3297, no. 5, pp. 12360–12368, 2007.
- [37] M. Kuldeepak and M. Vashishath, "License Plate Recognition System based on Image Processing Using LabVIEW," ... *J. Electron. Commun. Comput. ...*, vol. 2, no. 4, pp. 183–188, 2012.
- [38] S. Ramalingam, M. Rhead, and R. Gurney, "Impact of character spacing on the performance of Automatic Number Plate Recognition (ANPR) systems through simulation," 2014 Int. Carnahan Conf. Secur. Technol., pp. 1–6, 2014.
- [39] K. M. Sajjad, "Automatic License Plate Recognition using Python and OpenCV,"
-

- 
- 
- Dep. Comput. Sci. Eng. MES Coll. Eng., pp. 1–5, 2012.
- [40] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook*. BIRMINGHAM, 2011.
- [41] R. Rodríguez Morales and J. H. Sossa Azuela, *Procesamiento y análisis digital de imágenes*, 1ª ed. México Distrito Federal: Alfaomega, 2011.
- [42] “Análisis de imágenes digitales.”
- [43] P. Joshi, D. M. Escrivá, and V. Godoy, *OpenCV by example : enhance your understanding of computer vision and image processing by developing real-world projects in OpenCV 3*, 1st ed. Packt Publishing, 2016.
- [44] L. B. Daniel, *Mastering OpenCV with Practical Computer Vision Projects*. 2012.
- [45] “UTF-8: codificación para una comunicación digital global - 1&1.” [Online]. Available: <https://www.1and1.mx/digitalguide/paginas-web/creacion-de-paginas-web/utf-8-codificacion-para-una-comunicacion-digital-global/>. [Accessed: 09-Oct-2018].
- [46] “YouTube.” [Online]. Available: <https://www.youtube.com/?gl=MX&hl=es-419>. [Accessed: 02-Oct-2017].
- [47] “C++ ctime() - C++ Standard Library.” [Online]. Available: <https://www.programiz.com/cpp-programming/library-function/ctime/ctime>. [Accessed: 09-Oct-2018].

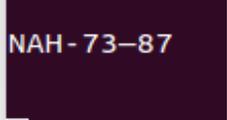
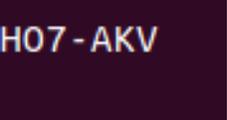
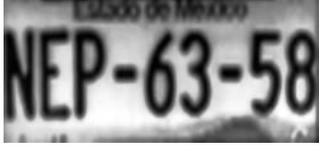
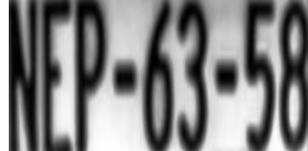
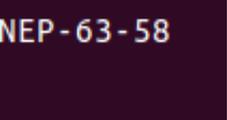
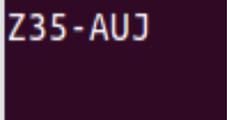
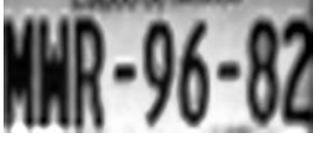
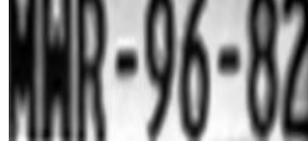
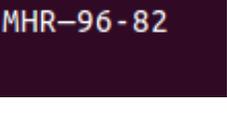
Tabla. Pruebas realizadas al reconocedor del número de matrícula.

Imagen original	Deteccion de la matrícula	Matrícula vehicular	OCR
			
			
			
			
			
			
			

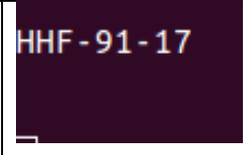
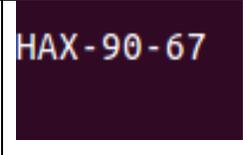
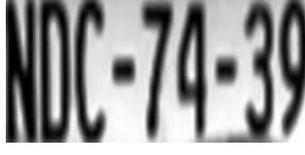
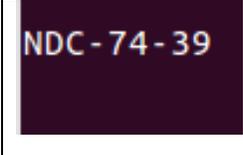
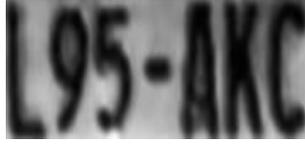
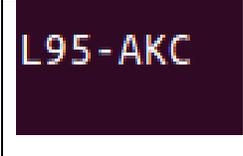
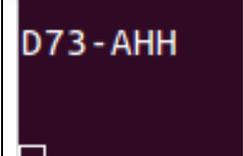
Anexos

			
			
			
			
	<p>No se detectó la matrícula</p>	<p>No hubo detección del número de matrícula</p>	<p>No hubo reconocimiento del número de placa</p>
			
			
			

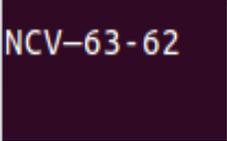
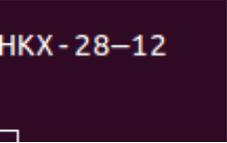
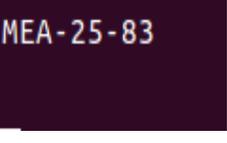
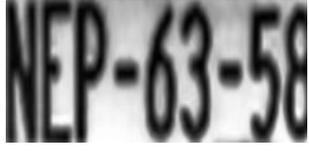
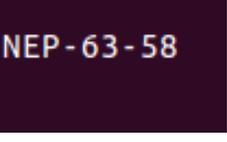
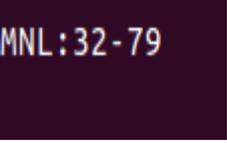
Anexos

			
			
			
			
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
			

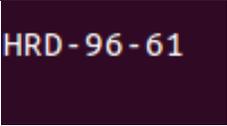
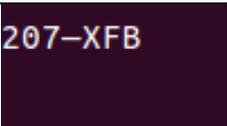
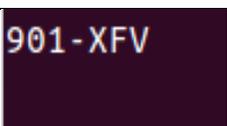
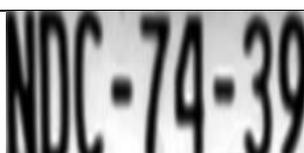
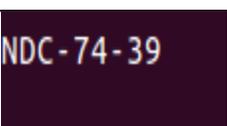
Anexos

			
			
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
			

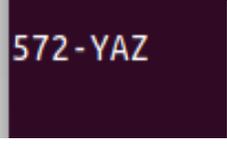
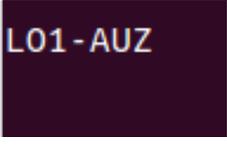
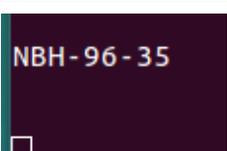
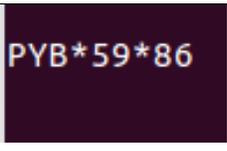
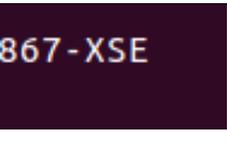
Anexos

			
			
			
			
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
			
			

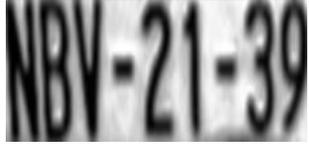
Anexos

	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
			
			
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			
			
			

Anexos

Anexos

			X92-ASD
			649-ZKH
			NBV-21-39
			681-YCL
	No se detectó la matrícula	No hubo detección del número de matrícula	No hubo reconocimiento del número de placa
			?cxf?)
			NEY-48-51
			J29-AVB

Anexos