



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

# Tecnológico Nacional de México

**Centro Nacional de Investigación  
y Desarrollo Tecnológico**

## Tesis de Maestría

**Análisis de Algoritmos de Visión por Computadora  
para un Asistente de Mantenimiento de Carril**

presentada por  
**Ing. Sergio Álvarez Silva**

como requisito para la obtención del grado de  
**Maestro en Ciencias de la Computación**

Director de tesis  
**Dr. Dante Mújica Vargas**

Codirector de tesis  
**Dr. Nimrod González Franco**

**Cuernavaca, Morelos, México. Febrero de 2025.**



Cuernavaca, Mor., 18/febrero/2025

OFICIO No. DCC/048/2025

Asunto: Aceptación de documento de tesis  
CENIDET-AC-004-M14-OFICIO

**CARLOS MANUEL ASTORGA ZARAGOZA**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial de **Sergio Álvarez Silva** con número de control **M23CE001**, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"Análisis de algoritmos de visión por computadora para un asistente de mantenimiento de carril"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

**ATENTAMENTE**  
*Excelencia en Educación Tecnológica®*  
*"Conocimiento y Tecnología al Servicio de México"*

**Dr. Dante Mújica Vargas**  
Director de tesis

**Dr. Nimrod González Franco**  
Codirector de tesis

**Dr. Noé Alejandro Castro Sánchez**  
Revisor 1

**Dr. Juan Gabriel González Serna**  
Revisor 2

C.c.p. Depto. Servicios Escolares.  
Expediente / Estudiante



**2025**  
Año de  
**La Mujer  
Indígena**

Interior Internado Palmira S/N, Col. Palmira,  
C. P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 3201,  
e-mail: dcc\_cenidet@tecnm.mx tecnm.mx | cenidet.tecnm.mx





Centro Nacional de Investigación y Desarrollo tecnológico  
Subdirección Académica

Cuernavaca Mor, 19/febrero/2025

Oficio No. SAC/069/2025

Asunto: Autorización de impresión de tesis

**SERGIO ÁLVAREZ SILVA**  
**CANDIDATO AL GRADO DE MAESTRO**  
**EN CIENCIAS DE LA COMPUTACIÓN**  
**P R E S E N T E**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“Análisis de algoritmos de visión por computadora para un asistente de mantenimiento de carril”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**A T E N T A M E N T E**

*Excelencia en Educación Tecnológica®*  
*“Conocimiento y Tecnología al Servicio de México”*

**CARLOS MANUEL ASTORGA ZARAGOZA**  
**SUBDIRECTOR ACADÉMICO**



c.c.p. Departamento de Ciencias Computacionales  
Departamento de Servicios Escolares

CMAZ/lmz



**2025**  
Año de  
**La Mujer**  
**Indígena**

Interior Internado Palmira S/N, Col. Palmira,  
C. P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 4104,  
e-mail: acad\_cenidet@tecnm.mx tecnm.mx | cenidet.tecnm.mx



## **Dedicatoria**

Dedico esta tesis  
A mis padres, Alberto y Susana  
A mi hermana Alexia  
Y a mis abuelos

## **Agradecimientos**

Agradezco a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) por el apoyo económico brindado durante mis estudios de maestría. Al Tecnológico Nacional de México / Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET por brindar las instalaciones y permitirme realizar los estudios de maestría.

A mi director de tesis, el Dr. Dante Mújica Vargas, por su apoyo, dedicación y guía brindada durante este proyecto de tesis. A mi codirector, el Dr. Nimrod González Franco, por su invaluable apoyo y orientación a lo largo del proceso. A mi comité revisor, el Dr. Noé Alejandro Castro Sánchez y el Dr. Juan Gabriel González Serna, por su asesoramiento para el desarrollo de esta investigación.

¡Gracias!

## Resumen

Esta tesis aborda el análisis y comparación de algoritmos de visión por computadora aplicados a un Asistente de Mantenimiento de Carril (SAMC). El estudio se centra en evaluar el desempeño de métodos tradicionales y técnicas de aprendizaje profundo basadas en redes neuronales convolucionales (CNN), bajo diversas condiciones de conducción. Para ello, se utilizaron tanto bases de datos de referencia ampliamente reconocidas en el estado del arte como pruebas en entornos reales y simulados.

En los experimentos realizados, los algoritmos fueron sometidos a escenarios adversos que incluyeron lluvia de distintas intensidades (leve, moderada y fuerte), exposición a luz solar directa, deslumbramiento solar, sombras, curvas de diferentes grados (suaves y pronunciadas), y conducción nocturna, tanto con iluminación artificial como en su ausencia, empleando únicamente los faros del vehículo. Asimismo, en los entornos simulados se evaluaron condiciones de neblina recreando distintos escenarios: neblina uniforme, heterogénea, nublada y combinaciones de estas.

Los resultados obtenidos revelan que los métodos tradicionales son efectivos en condiciones de conducción controladas, pero su rendimiento disminuye significativamente en situaciones más complejas, como lluvia intensa, curvas pronunciadas, sombras y condiciones de iluminación adversa. Por otro lado, los algoritmos basados en redes neuronales convolucionales, como SCNN y VGG16, demostraron una mayor adaptabilidad y robustez en estos escenarios desafiantes, destacando especialmente en situaciones de deslumbramiento solar, sombras en la carretera y conducción nocturna.

**Palabras clave:** ADAS, visión por computadora, detección de carril, asistencia en mantenimiento de carril.

## **Abstract**

This thesis addresses the analysis and comparison of computer vision algorithms applied to a Lane Keeping Assistant System (LKAS). The study focuses on evaluating the performance of traditional methods and deep learning techniques based on convolutional neural networks (CNNs) under various driving conditions. To this end, both widely recognized benchmark datasets from the state of the art and tests in real and simulated environments were utilized.

In the experiments conducted, the algorithms were subjected to adverse scenarios, including rain of varying intensities (light, moderate, and heavy), direct sunlight exposure, solar glare, shadows, curves of different degrees (gentle and sharp), and nighttime driving, both with artificial road lighting and in its absence, relying solely on vehicle headlights. Additionally, simulated environments were used to evaluate fog conditions, recreating various scenarios: uniform fog, heterogeneous fog, cloudy fog, and combinations of these.

The results obtained reveal that traditional methods are effective under controlled driving conditions but their performance decreases significantly in more complex scenarios, such as heavy rain, sharp curves, shadows, and challenging lighting conditions. In contrast, CNN-based algorithms, such as SCNN and VGG16, demonstrated greater adaptability and robustness in these challenging environments, particularly excelling in situations involving solar glare, road shadows, and nighttime driving.

**Keywords:** ADAS, computer vision, lane detection, lane keeping assistance.

# Índice General

<b>Lista de Figuras</b>	<b>viii</b>
<b>Lista de Tablas</b>	<b>x</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Planteamiento del problema . . . . .	1
1.1.1 Delimitación del problema . . . . .	2
1.1.2 Complejidad del problema . . . . .	2
1.2 Objetivos . . . . .	4
1.2.1 Objetivo general . . . . .	4
1.2.2 Objetivos específicos . . . . .	4
1.3 Alcances y limitaciones . . . . .	5
1.3.1 Alcances . . . . .	5
1.3.2 Limitaciones . . . . .	5
1.4 Justificación . . . . .	6
1.5 Organización de la tesis . . . . .	6
<b>2 Marco Teórico</b>	<b>7</b>
2.1 Sistemas avanzados de asistencia al conductor . . . . .	7
2.2 Sistemas de asistencia de conducción basados en visión . . . . .	8
2.3 Visión por computadora . . . . .	9
2.4 Algoritmos de visión por computadora para un AMC . . . . .	12
2.5 Detección de carril . . . . .	14
<b>3 Estado del Arte</b>	<b>27</b>
3.1 Antecedentes . . . . .	27
3.2 Trabajos relacionados . . . . .	28
3.3 Discusión del Estado del Arte . . . . .	44
<b>4 Metodología de Solución</b>	<b>56</b>
4.1 Propuesta de solución . . . . .	56
4.1.1 Adquisición y Preprocesamiento . . . . .	57
4.1.2 Implementación de Algoritmos . . . . .	60
4.1.3 Evaluación y Análisis . . . . .	66
<b>5 Experimentación y Resultados</b>	<b>68</b>
5.1 Diseño Experimental . . . . .	68
5.2 Entorno de desarrollo . . . . .	68
5.3 Base de Datos . . . . .	68
5.4 Métricas . . . . .	70

---

5.5	Experimentos . . . . .	72
5.5.1	Técnicas tradicionales en carriles rectos de la base de datos de Frida y Frida 2 . . . . .	72
5.5.2	Técnicas tradicionales en carriles rectos de la base de datos de TuSimple . . . . .	75
5.5.3	Técnicas tradicionales en carriles rectos de mi propia base de datos . . . . .	78
5.5.4	Extensión de técnicas tradicionales para la detección en curvas . . . . .	82
5.5.5	Técnicas de modelos de aprendizaje profundo . . . . .	86
<b>6</b>	<b>Conclusiones</b>	<b>95</b>
6.1	Objetivos y alcances logrados . . . . .	95
6.2	Resultados del trabajo . . . . .	96
6.3	Conclusiones . . . . .	97
6.4	Trabajo futuro . . . . .	98
	<b>Referencias</b>	<b>99</b>
	<b>Apéndice A Producción académica</b>	<b>106</b>
	<b>Apéndice B Retribución social</b>	<b>108</b>

# Lista de Figuras

2.1	Linea del tiempo de progreso desde el vehículo autónomo de nivel 1 hasta el de nivel 5 [46]. . . . .	7
2.2	Obtención de información [42]. . . . .	8
2.3	Aplicaciones con Sistemas de Asistencia de Conducción Basados en Visión [34]. . . . .	9
2.4	Etapas de sistema de visión por computadora. . . . .	10
2.5	Simulador para detección de carril para vehículos mediante visión por computadora [5]. . . . .	11
2.6	Detección de carril con método <i>Canny</i> [27]. . . . .	12
2.7	Detección de carril con método de transformada de Hough [27]. . . . .	12
2.8	Detección de carril con método de segmentación [53]. . . . .	13
2.9	Detección de carril con método de clasificación de características [2]. . . . .	13
2.10	Detección de carril con método de Kalman [72]. . . . .	13
2.11	Detección de carril con CNN[91]. . . . .	14
2.12	Detección de carril [82]. . . . .	14
2.13	Etapas de detección de carril [12]. . . . .	15
2.14	Supresión del máximo local de la magnitud del gradiente [9]. . . . .	22
2.15	Un conjunto de líneas que pasan por un punto de imagen. Para todas las posibles líneas $L_j$ que pasan por el punto $p_0 = (x_0, y_0)$ , se cumple la ecuación $y_0 = k_j x_0 + d_j$ para valores apropiados de los parámetros $k_j$ y $d_j$ [9]. . . . .	23
2.16	Representación gráfica de la relación entre el espacio de imagen y el espacio de parámetros. [9] . . . . .	24
2.17	Marcas de carril identificadas [84]. . . . .	25
2.18	Ajuste de línea y región de interés. . . . .	26
3.1	Representación de la estrategia propuesta [1]. . . . .	28
3.2	Diagrama de flujo de método desarrollado [98]. . . . .	29
3.3	Estructura del sistema propuesto [6]. . . . .	29
3.4	Diagrama del método propuesto [50]. . . . .	30
3.5	Diagrama de flujo propuesto [43]. . . . .	31
3.6	Esquema de flujo del sistema desarrollado [83]. . . . .	32
3.7	<i>Framework</i> del método propuesto. [89]. . . . .	32
3.8	Diagrama de principio de transformación de similitud [86]. . . . .	33
3.9	Diagrama del método desarrollado [47]. . . . .	34
3.10	Diagrama de flujo de los algoritmos de detección de carriles [18]. . . . .	34
3.11	Esquema de la arquitectura propuesta [66]. . . . .	35
3.12	Comparaciones visuales entre el método propuesto y otros métodos del estado del arte [80]. . . . .	40

---

3.13	Resultados de métodos de eliminación de lluvia en conjuntos de datos sintéticos [45]. . . . .	41
3.14	Comparación de calidad visual de modelos PReNet entrenados con diferentes funciones de pérdida, incluyendo la pérdida única de MSE, la pérdida única de SSIM negativo y la supervisión recursiva de SSIM negativo [64]. . . . .	42
4.1	Diagrama de metodología de solución propuesta. . . . .	56
4.2	Módulo de Adquisición y Preprocesamiento de Datos . . . . .	57
4.3	Módulo de Implementación de Algoritmos de Detección de Carriles por Visión por Computadora . . . . .	60
4.4	Arquitectura del modelo de la red VGG-16 [49]. . . . .	65
4.5	Arquitectura del modelo de la red SCNN [55]. . . . .	66
5.1	Imágenes aleatorias de la base de datos Frida y Frida 2 con variedad de escenas viales y condiciones climáticas. . . . .	69
5.2	Imágenes aleatorias de la base de datos TuSimple con variedad de escenas viales. . . . .	69
5.3	Imágenes aleatorias en base de datos propia con variedad de escenas viales y condiciones climáticas. . . . .	70
5.4	Resultados cuantitativos de la detección en la base de datos Frida. . . . .	73
5.5	Resultados cualitativos en base de datos Frida. . . . .	75
5.6	Resultados cuantitativos de la detección en la base de datos TuSimple. . . . .	76
5.7	Resultados cualitativos en base de datos TuSimple. . . . .	78
5.8	Resultados cuantitativos de la detección en base de datos propia. . . . .	80
5.9	Resultados cualitativos en base de datos propia. . . . .	81
5.10	Resultados cuantitativos de la detección de carril curvo en diversas bases de datos. . . . .	82
5.11	Imágenes que muestran áreas seleccionadas con zoom en presencia de curvas, etiquetadas como a, b, c y d. . . . .	83
5.12	Resultados cualitativos en diversas base de datos. . . . .	85
5.13	Resultados cuantitativos de la detección de carril con aprendizaje profundo. . . . .	87
5.14	Resultados cualitativos con arquitecturas SCNN & VGG16 en base de datos propia. . . . .	90
5.15	Resultados cualitativos con arquitecturas SCNN & VGG16 en base de Frida & Frida 2. . . . .	91
5.16	Resultados cualitativos con arquitecturas SCNN & VGG16 en base de TuSimple. . . . .	92
A.1	Constancia de la presentación del póster. . . . .	106
A.2	Portada del artículo. . . . .	107
B.1	Constancia de la presentación del póster. . . . .	108

# Lista de Tablas

2.1	Relación entre el espacio de imagen y el espacio de parámetros. [9]	25
3.1	Resumen del estado del arte.	46
4.1	Métodos y algoritmos de técnicas clásicas.	61
5.1	Resumen de bases de datos utilizadas para la detección de carriles.	70
5.2	Elementos de la matriz de confusión para la Figura 5.5.	73
5.3	Comparación de algoritmos basada en diversas métricas.	74
5.4	Elementos de la matriz de confusión para la Figura 5.7.	77
5.5	Comparación de algoritmos basada en diversas métricas.	77
5.6	Condiciones adversas para detección de carriles.	79
5.7	Elementos de la matriz de confusión para la Figura 5.9.	80
5.8	Comparación de algoritmos basada en diversas métricas.	80
5.9	Elementos de la matriz de confusión para la Figura 5.12.	84
5.10	Comparación de algoritmos basada en diversas métricas.	84
5.11	Matriz de confusión para el detector con arquitectura SCNN	88
5.12	Matriz de confusión para el detector con arquitectura VGG16	88
5.13	Comparación de métricas entre las arquitecturas SCNN y VGG16.	89
5.14	Enfoques destacados por experimento.	93
5.15	Algoritmos destacados por experimento	94
6.1	Objetivos y Actividades	95
6.2	Alcances y Limitaciones	96

# Capítulo 1

## Introducción

A lo largo del día, los seres humanos toman decisiones constantemente, muchas de las cuales se basan en nuestras percepciones del entorno. En el ámbito de la conducción, la percepción visual juega un papel esencial. Los vehículos autónomos deben ser capaces de detectar objetos en su entorno y reaccionar de manera adecuada. Este proceso presenta varios desafíos relacionados con la visión por computadora, tales como la identificación de señales de tráfico, semáforos, peatones y, especialmente, las líneas de carril. La presente investigación se centra en la detección de carriles, un componente crucial para la planificación del movimiento de los vehículos.

La planificación del movimiento es una tecnología clave en los vehículos autónomos, pero enfrenta desafíos significativos. Este proceso comienza con la detección de carriles, seguida del seguimiento de su trayectoria, la identificación de señales y la detección de obstáculos que puedan indicar carriles cerrados. El objetivo de esta investigación es identificar los métodos de detección más efectivos, que se agrupan en dos categorías principales: enfoques tradicionales basados en visión y técnicas de aprendizaje profundo.

En este contexto, los sistemas de percepción en los vehículos juegan un papel fundamental, ya que son responsables de capturar, analizar y procesar información del entorno mediante cámaras y sensores ubicados estratégicamente. Esta información es crucial para la toma de decisiones en tiempo real, lo que contribuye a mejorar tanto la seguridad como la comodidad durante la conducción. Los sistemas avanzados de asistencia al conductor (ADAS), como el asistente de mantenimiento de carril y el frenado automático de emergencia, dependen de esta capacidad para interpretar las condiciones de la carretera.

La integración de datos provenientes de múltiples sensores permite obtener una visión integral del entorno, lo que resulta esencial para el funcionamiento adecuado de los sistemas ADAS y la seguridad de los vehículos autónomos. La detección precisa de carriles no solo es crucial para los sistemas de asistencia al conductor, sino también para garantizar que el vehículo autónomo mantenga su trayectoria y responda de manera adecuada ante obstáculos.

### 1.1 Planteamiento del problema

A pesar de la existencia de sistemas avanzados de asistencia para la conservación de carril, que son capaces de detectar las desviaciones del vehículo y ajustar automáticamente la dirección para restablecer su posición dentro del carril, estos sistemas todavía enfrentan desafíos significativos en entornos adversos. Condiciones como curvas pronunciadas, clima adverso (lluvia, niebla) y oscuridad en la carretera pueden dificultar la detección precisa de los límites del carril y la curvatura del camino. Aunque se ha logrado un

avance considerable en su desarrollo, es necesario continuar con su mejora para asegurar su funcionamiento óptimo bajo todas las condiciones de manejo.

Además, los sistemas de asistencia deben ser capaces de manejar la falta de mediciones o mediciones inexactas provenientes de los detectores de carril, para garantizar que el vehículo se mantenga dentro de los límites del carril. En este sentido, se han propuesto soluciones basadas en el control predictivo, pero también es fundamental explorar alternativas algorítmicas adicionales y la aceleración algorítmica mediante hardware, con el fin de mejorar la calidad de la información visual utilizada como entrada en estos sistemas. Por lo tanto, el análisis de algoritmos de visión por computadora se convierte en un componente esencial para mejorar la precisión y eficacia de los sistemas de asistencia de mantenimiento de carril en condiciones de conducción adversas.

Hipótesis:

La hipótesis planteada es que la implementación de técnicas robustas de procesamiento de imágenes, junto con un procesamiento especializado, puede ofrecer una solución efectiva para superar los desafíos presentes en los sistemas de asistencia de conservación de carril. Se espera que, mediante la aplicación de estas técnicas, se logren mejoras significativas en la capacidad de estos sistemas para operar de manera eficiente en entornos adversos.

### **1.1.1 Delimitación del problema**

En este trabajo de investigación se desarrolló un método compuesto por tres módulos para el análisis de algoritmos de visión por computadora aplicados a un sistema de asistencia de mantenimiento de carril. El primer módulo se centró en la adquisición y preprocesamiento de datos, garantizando que la información recopilada fuera adecuada y estuviera debidamente preparada para las etapas posteriores. El segundo módulo abarcó la implementación de algoritmos de detección de carriles mediante técnicas de visión por computadora, combinando enfoques tradicionales con métodos basados en aprendizaje profundo. Finalmente, el tercer módulo se dedicó a la evaluación y análisis del desempeño, midiendo métricas clave para determinar la eficacia de los algoritmos en diversos escenarios de conducción. Este enfoque contempló tanto algoritmos basados en técnicas clásicas como en aprendizaje profundo para la detección de carriles.

### **1.1.2 Complejidad del problema**

Aunado a las condiciones adversas inherentes que puede existir durante el desplazamiento producidas por reflexión de iluminación, neblina, lluvia, oscuridad, entre otras. El mayor desafío tiene que ver con el tiempo de procesamiento de los algoritmos, restringido a la velocidad del desplazamiento vehicular, con objeto de no perder o distorsionar la información. El problema es complejo debido a los desafíos que implica el análisis de algoritmos de visión por computadora en un asistente de mantenimiento de carril. Entre estos desafíos se encuentran:

- Variabilidad en las condiciones de iluminación: la calidad de las imágenes capturadas por las cámaras del sistema puede variar significativamente debido a cambios en las condiciones de iluminación, lo que puede dificultar la detección de los carriles.

- 
- Detección de carriles en diferentes condiciones climáticas: las condiciones climáticas, como la lluvia, la nieve o la niebla, pueden afectar la visibilidad de los carriles y dificultar su detección.
  - Diversidad en los diseños y marcas viales de los carriles: los carriles pueden tener diferentes diseños y marcas viales en diferentes partes del mundo, lo que puede dificultar la detección de los mismos mediante un algoritmo de visión por computadora.
  - Tiempo de procesamiento: el tiempo de procesamiento de los algoritmos debe ser rápido para evitar la pérdida o distorsión de la información, ya que está limitado por la velocidad del vehículo en movimiento.
  - Interferencia de objetos en la carretera: la presencia de otros vehículos, señales de tráfico u objetos en la carretera puede interferir con la detección de los carriles y reducir la precisión del sistema.

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

Analizar algoritmos de visión por computadora utilizados para la asistencia de mantenimiento de carril.

### **1.2.2 Objetivos específicos**

- Comprender los conceptos relacionados con sistema de asistencia de mantenimiento de carril.
- Seleccionar y detallar algoritmos de visión por computadora para detección de marcas y fronteras en carriles, también bajo condiciones adversas curvas, curvas muy pronunciadas, presencia de lluvia, neblina, pocas condiciones de iluminación o bien la oscuridad y presencia de sombras en la carretera.
- Operacionalizar los algoritmos seleccionados en diversas condiciones reales de tráfico y clima para probar la robustez de los algoritmos seleccionados.
- Estimar el rendimiento los algoritmos seleccionados, considerando métricas.
- Evaluar el desempeño de los algoritmos seleccionados.

## **1.3 Alcances y limitaciones**

### **1.3.1 Alcances**

- Se realizará un estudio detallado de los diferentes algoritmos de visión por computadora utilizados en sistemas de asistencia de mantenimiento de carril.
- Se evaluarán las capacidades de detección y seguimiento de los carriles bajo condiciones normales y adversas.
- El análisis se enfocará en la detección y análisis de características del camino, como las líneas de carril y los bordes de la carretera.
- Se contemplarán entornos simulados y reales que representen vías rectas y curvas.
- Se considerarán diferentes estilos de carriles y condiciones de iluminación.

### **1.3.2 Limitaciones**

- El análisis no contemplará la trayectoria ni el control del vehículo.
- La selección de los algoritmos a analizar estará limitada a aquellos que estén disponibles públicamente o que puedan ser obtenidos a través de acuerdos de colaboración.
- No se contemplarán condiciones ambientales totalmente adversas.
- La variabilidad en las marcas de los carriles en diferentes partes del mundo puede presentar un desafío para la detección precisa de las mismas mediante un algoritmo de visión por computadora.
- La velocidad del vehículo puede limitar el tiempo de procesamiento de los algoritmos y afectar la precisión del sistema.

## 1.4 Justificación

La justificación para el análisis de algoritmos de visión por computadora para un asistente de mantenimiento de carril radica en la necesidad de analizar algoritmos robustos y precisos que puedan detectar con precisión la posición de los carriles y alertar al conductor en caso de una posible desviación. Sin embargo, este tipo de tecnología también presenta limitaciones que deben ser consideradas, como la variabilidad en las condiciones de iluminación, la presencia de objetos en la carretera que pueden interferir con la detección de los carriles y la variabilidad en las marcas de los carriles. Por lo tanto, es importante analizar y evaluar diferentes algoritmos de visión por computadora para determinar su efectividad y adecuación para su implementación en un sistema de asistencia de mantenimiento de carril. La industria automotriz ha entrado en un período de transformación que verá una evolución sin precedentes en las capacidades tecnológicas de los vehículos. Los avances en nuevas técnicas de fabricación, sensores de bajo costo, alto poder de procesamiento computacional, acceso en tiempo real a la información y la inclusión de la inteligencia computacional, han hecho que los vehículos estén cambiando rápidamente y creciendo en complejidad. Estas nuevas tecnologías permitirán la inevitable evolución hacia los vehículos autónomos, aunque de manera inmediata brindarán beneficios sustanciales a los conductores, los pasajeros y el medio ambiente.

## 1.5 Organización de la tesis

Este documento está compuesto por seis capítulos, anexos y referencias. El Capítulo 2 abarca el marco teórico, donde se describen los conceptos clave relacionados con el tema de investigación, como los sistemas avanzados de asistencia al conductor (ADAS), sistemas de asistencia basados en visión, visión por computadora, algoritmos aplicados a un sistema de asistencia de mantenimiento de carril y detección de carriles. Estos fundamentos resultaron esenciales para comprender y desarrollar esta investigación. En el Capítulo 3 se presentan los antecedentes y el estado del arte, destacando los trabajos más relevantes que sirvieron de base para este estudio. El Capítulo 4 detalla la metodología propuesta, explicando sus componentes y su importancia dentro del desarrollo de la investigación. El Capítulo 5 describe los experimentos realizados, junto con los resultados obtenidos y su análisis. Finalmente, el Capítulo 6 describe las conclusiones derivadas de la investigación, los productos desarrollados y las posibles líneas de investigación futura.

# Capítulo 2

## Marco Teórico

En este capítulo, se proporcionan los conceptos y métodos fundamentales necesarios para entender de manera adecuada los temas relacionados con la presente investigación. Estos conceptos se presentan de forma específica y complementaria para apoyar la comprensión.

### 2.1 Sistemas avanzados de asistencia al conductor

Los sistemas avanzados de asistencia al conductor son tecnologías diseñadas para ayudar a los conductores humanos a evitar accidentes de tráfico al detectar riesgos, proporcionar alertas y acciones correctivas en tiempo real. Estos sistemas incluyen diferentes características, como el control longitudinal y lateral, el monitoreo de la vigilancia del conductor y la asistencia para el estacionamiento, entre otros. Los sistemas avanzados de asistencia al conductor utilizan una variedad de sensores como radar, GPS, Lidar, cámaras, ultrasónico, etc, como se muestra en la Figura 2.1, estos sistemas buscan mejorar la seguridad en la carretera, su principal objetivo es evitar accidentes de esta manera, se centran en la prevención y son una parte esencial de los esfuerzos de seguridad en la industria automotriz [46].

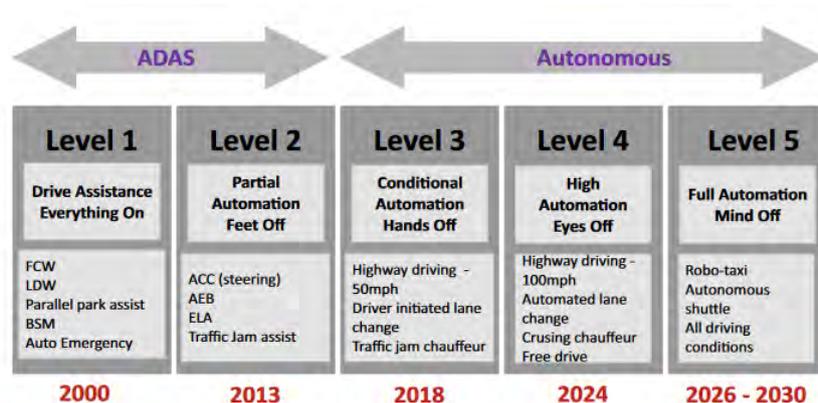


Figura 2.1 Línea del tiempo de progreso desde el vehículo autónomo de nivel 1 hasta el de nivel 5 [46].

Los Sistemas Avanzados de Asistencia al Conductor y los Sistemas de Asistencia de Conducción Basados en Visión están intrínsecamente vinculados en la evolución de la tecnología vehicular. Los ADAS abarcan un conjunto de sistemas que apoyan al conductor durante la conducción, utilizando una combinación de sensores, cámaras y algoritmos para detectar y responder a situaciones de riesgo en tiempo real.

Los Sistemas de Asistencia de Conducción Basados en Visión, dentro de los ADAS, se centran en el uso de cámaras y técnicas de visión por computadora para identificar elementos del entorno del vehículo, como carriles, señales de tráfico, peatones y otros vehículos. Estos sistemas emplean algoritmos avanzados para interpretar la información visual capturada y proporcionar asistencia al conductor en la toma de decisiones y en la ejecución segura de maniobras.

La integración de la visión por computadora en los ADAS ha permitido avances significativos en la detección de objetos, reconocimiento de señales de tráfico, sistemas de mantenimiento de carril, asistencia en la frenada de emergencia, entre otros.

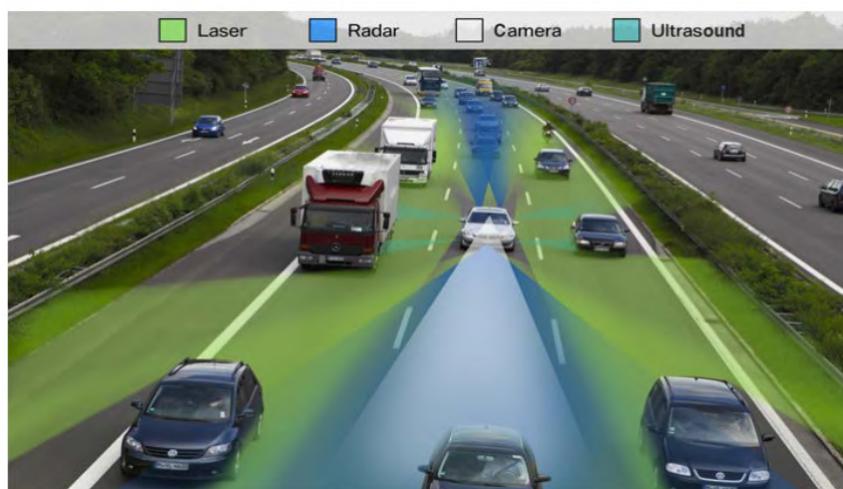


Figura 2.2 Obtención de información [42].

## 2.2 Sistemas de asistencia de conducción basados en visión

Los sistemas de asistencia al conductor basados en visión utilizan datos de imagen tomados por cámaras en el vehículo, estos son parte de la clase de sistemas de asistencia al conductor, estos sistemas analizan la información capturada de imagen para detección de carril, salidas de carril, aviso de colisión por alcance, detección de ángulo muerto, entre otros peligros potenciales, estos se comunican con el conductor a través de interfaces como pantallas de visualización frontal. La visión por computadora es la disciplina que diseña soluciones para comprender datos de imagen, las cámaras de el vehículo capturan imágenes en tiempo real en diferentes momentos del trayecto, que se utilizan para analizar múltiples tareas relacionadas con la asistencia al conductor [34].

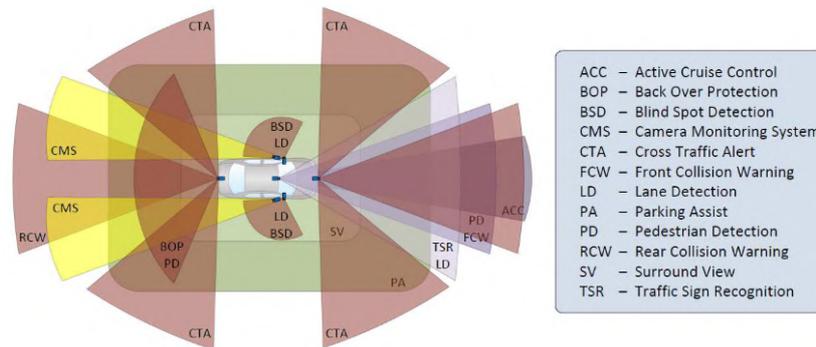


Figura 2.3 Aplicaciones con Sistemas de Asistencia de Conducción Basados en Visión [34].

Los Sistemas de Asistencia de Conducción Basados en Visión están estrechamente relacionados con la visión por computadora, siendo un componente fundamental en el desarrollo de tecnologías automotrices avanzadas. Estos sistemas utilizan cámaras y sensores para capturar datos visuales del entorno del vehículo y emplean algoritmos de visión por computadora para procesar esta información.

La visión por computadora se encarga del análisis de imágenes y videos capturados por las cámaras del vehículo, permitiendo identificar y comprender el contexto del entorno circundante. Este análisis incluye la detección y seguimiento de objetos, reconocimiento de señales de tráfico, identificación de peatones, vehículos y obstáculos, así como la detección de carriles para asistentes de mantenimiento de carril.

## 2.3 Visión por computadora

La Visión por computadora abarca un conjunto de técnicas que involucran la adquisición, procesamiento, análisis y comprensión de imágenes del entorno real con el propósito de extraer información significativa. Este proceso implica la aplicación de diversas transformaciones a las imágenes para resaltar y enfatizar los detalles y características relevantes, al tiempo que reduce o elimina la información que resulta insignificante o no relevante para el análisis deseado [39].

Estas transformaciones pueden incluir técnicas de filtrado, segmentación, detección de bordes y reconocimiento de patrones, entre otras, con el objetivo de obtener una representación más precisa y comprensible de los elementos presentes en las imágenes capturadas del mundo real.

El proceso de Visión por computadora se compone de múltiples etapas interconectadas que permiten el análisis y comprensión de imágenes. Estas etapas se presentan como una secuencia de procesos interdependientes que se encargan de transformar la información visual en datos comprensibles por el sistema computacional. En la Figura 2.4, se representan visualmente estas fases clave del proceso.

Las etapas principales de un sistema de visión por computadora incluyen la adquisición de la imagen, donde se captura la información visual del entorno; el preprocesamiento, que se enfoca en ajustar y mejorar la calidad de la imagen mediante técnicas como

corrección de iluminación, reducción de ruido, y normalización; la segmentación, que divide la imagen en regiones o componentes para facilitar el análisis; la extracción de características, que identifica y resalta elementos específicos como bordes, formas o texturas relevantes; y finalmente, el procesamiento y análisis de la información extraída para realizar tareas específicas como reconocimiento, seguimiento o toma de decisiones basadas en la imagen capturada. Cada una de estas etapas juega un papel crucial en la interpretación y comprensión de la información visual, como se muestra la Figura 2.4.

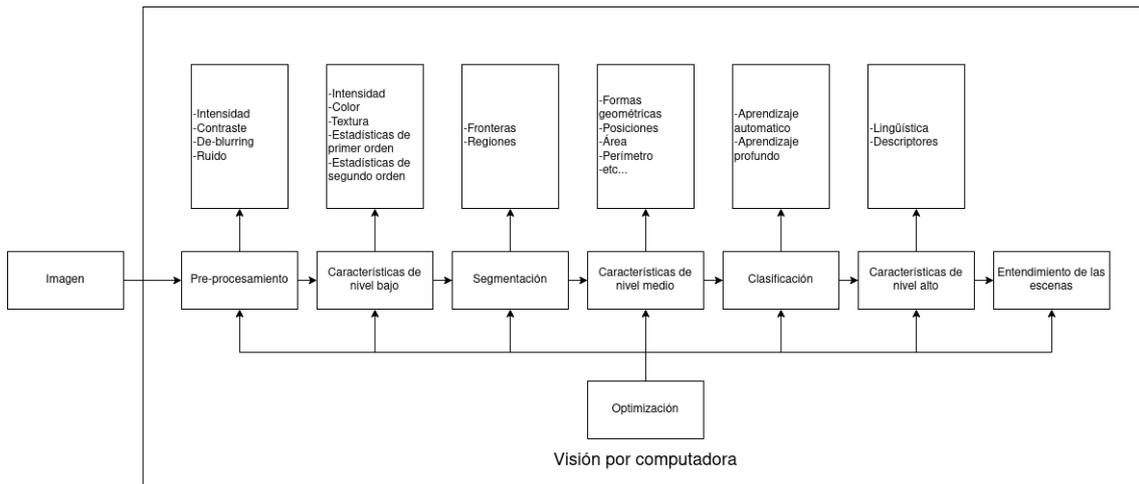


Figura 2.4 Etapas de sistema de visión por computadora.

- **Adquisición de imágenes:** Proceso de captura de imágenes o videos utilizando cámaras.
- **Pre-procesamiento:** Se aplican técnicas como el filtrado, la corrección de distorsiones, el ajuste del brillo y el contraste para mejorar la calidad de las imágenes adquiridas, en esta etapa se incluye la eliminación de ruido y la corrección de errores.
- **Segmentación:** Se identifican y separan las regiones de interés de la imagen, como objetos y áreas de fondo. Para llevar a cabo este proceso se utilizan algoritmos basados en diferentes características de la imagen, tales como intensidad, textura, forma y color.
- **Descripción:** Proceso en el cual se clasifican las regiones de interés identificadas y las características únicas que permiten distinguir un tipo de objeto de otro.
- **Reconocimiento:** Proceso mediante el cual se identifican y clasifican patrones, objetos o eventos en una imagen o secuencia de imágenes.
- **Interpretación:** Proceso de asignar una etiqueta o significado a un objeto específico presente en una imagen.

Los sistemas de visión por computadora se clasifican según su nivel de procesamiento, que van desde un nivel bajo hasta uno alto. [23] Cada nivel de procesamiento tiene como finalidad la comprensión y análisis de la información visual para poder tomar decisiones y

ejecutar acciones específicas, y se apoya en el nivel anterior para lograrlo, a continuación se describen los tres niveles mencionados de la siguiente manera: Los sistemas de visión por computadora se dividen en diferentes niveles, cada uno enfocado en distintas técnicas para el procesamiento de imágenes:

En el nivel bajo, se aplican técnicas fundamentales como la captura y preprocesamiento de imágenes. A nivel medio, se utilizan técnicas más avanzadas que incluyen la segmentación, detección y seguimiento de objetos en las imágenes. En el nivel alto, se implementan técnicas complejas como el reconocimiento de patrones, clasificación y la interpretación avanzada de la información visual.

Estos niveles representan una progresión desde las técnicas básicas de adquisición de imágenes hasta el análisis más sofisticado y la comprensión avanzada de la información visual.

La Visión por Computadora y los Algoritmos asociados a esta disciplina son elementos clave en el desarrollo de sistemas de asistencia al mantenimiento de carril en vehículos, como se muestra la Figura 2.5. La Visión por Computadora implica el uso de cámaras y sensores para capturar información visual del entorno circundante del vehículo, mientras que los Algoritmos de Visión por Computadora procesan esta información para detectar, identificar y rastrear los carriles en la carretera.

Los Algoritmos de Visión por Computadora para el asistente de mantenimiento de carril se encargan de analizar la información visual capturada para reconocer y delinear los carriles en la carretera. Estos algoritmos utilizan técnicas como la segmentación de imágenes, detección de bordes, transformada de Hough, redes neuronales convolucionales (CNN), entre otras, para identificar y seguir con precisión la posición de los carriles.

La fusión de la Visión por Computadora con Algoritmos específicos de detección de carriles ha permitido el desarrollo de sistemas avanzados de asistencia al conductor, garantizando una mayor precisión y fiabilidad en la detección de los carriles, lo que contribuye a una conducción más segura y estable. Esta combinación tecnológica es esencial en la evolución hacia vehículos autónomos y en la mejora continua de la seguridad vial.



Figura 2.5 Simulador para detección de carril para vehículos mediante visión por computadora [5].

## 2.4 Algoritmos de visión por computadora para un AMC

Los algoritmos de visión por computadora utilizados en un asistente de mantenimiento de carril emplean técnicas de procesamiento de imágenes para detectar las líneas del carril y la posición del vehículo en relación con ellas. Para detectar las marcas del carril se han empleado diversos métodos, incluyendo la modelización geométrica y enfoques tradicionales. La mayoría de las estrategias convencionales siguen una serie de procesos, que incluyen preprocesamiento de imágenes, extracción de características, ajuste del modelo de carril y seguimiento de línea, el preprocesamiento de imágenes reduce la cantidad de ruido en la imagen, mientras que la extracción de características utiliza atributos de los carriles para identificar las áreas correspondientes, posteriormente el modelo de carril se ajusta y sigue mediante diversos métodos seleccionados [93].

Los asistentes de mantenimiento de carril utilizan información proporcionada por estos algoritmos para alertar al conductor si el vehículo se está desviando del carril, lo que puede ayudar a prevenir accidentes y mejorar la seguridad en la carretera. A continuación se presentan algunas técnicas que pueden ser incluidas en estos algoritmos:

Clásicas:

-Detección de bordes: Esta técnica identifica los bordes de las líneas del carril utilizando métodos clásicos como Sobel o Canny, como se muestra la Figura 2.6.

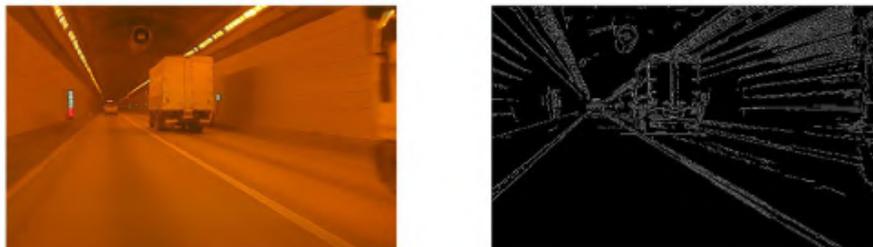


Figura 2.6 Detección de carril con método *Canny* [27].

-Transformada de Hough: Emplea la detección de bordes para encontrar las líneas del carril utilizando la transformada de Hough, una técnica tradicional en este ámbito, como se muestra la Figura 2.7.

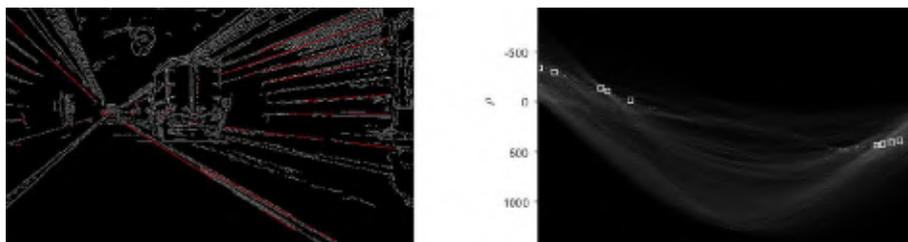


Figura 2.7 Detección de carril con método de transformada de Hough [27].

-Segmentación de imágenes: Separa la imagen en diferentes regiones para identificar las áreas correspondientes al carril y su entorno, un enfoque convencional en el procesamiento de imágenes para esta tarea, como se muestra la Figura 2.8.



Figura 2.8 Detección de carril con método de segmentación [53].

Técnicas actuales:

-Clasificación de características: Utiliza técnicas de aprendizaje automático para clasificar las características de la imagen, como líneas del carril, objetos y la carretera misma, como se muestra la Figura 2.9.

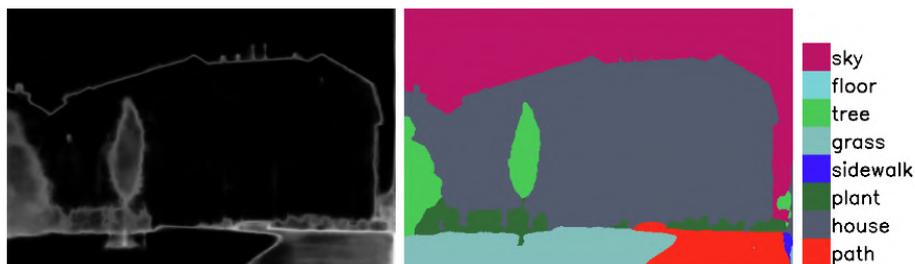


Figura 2.9 Detección de carril con método de clasificación de características [2].

-Filtro de Kalman: Implementa una técnica de filtrado que permite predecir la posición futura del vehículo y del carril, un enfoque más avanzado para mantener la trayectoria, como se muestra la Figura 2.10.

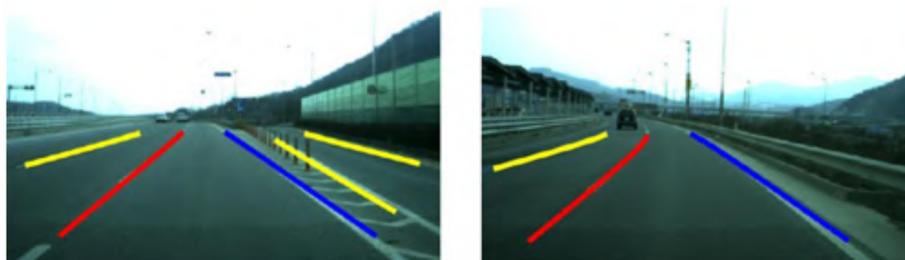


Figura 2.10 Detección de carril con método de Kalman [72].

-Redes neuronales convolucionales (CNN): Emplea estas redes para aprender directamente características de la imagen, detectando líneas del carril y otros elementos en la carretera con métodos más modernos y eficientes, como se muestra la Figura 2.11.



Figura 2.11 Detección de carril con CNN[91].

Los algoritmos de visión por computadora se integran para crear un sistema de asistencia de mantenimiento de carril, cuyo propósito es detectar la posición del vehículo en relación con las líneas del carril y alertar al conductor en caso de una desviación peligrosa. Estos algoritmos son esenciales para la detección de carriles, un componente crucial del asistente de mantenimiento de carril. La detección de carriles se realiza a través del análisis de información visual capturada por cámaras y sensores, empleando diversas técnicas. Entre estas técnicas se incluyen la detección de bordes, la segmentación de imágenes y la aplicación de transformadas, como la Transformada de Hough. Estas metodologías permiten identificar y seguir con precisión los límites de los carriles, asegurando que el vehículo permanezca en su trayectoria de manera segura y efectiva.

## 2.5 Detección de carril

La detección de carril es una técnica que muestra información específica de la geometría y características de las estructuras de línea de carril al sistema inteligente del vehículo, con el fin de detectar automáticamente los marcadores de la carretera y mostrar la posición de las marcas de carril, como se muestra la Figura 2.12. asegurándose de que los autos permanezcan en su carril asignado y no colisionen con vehículos en otros carriles [93].

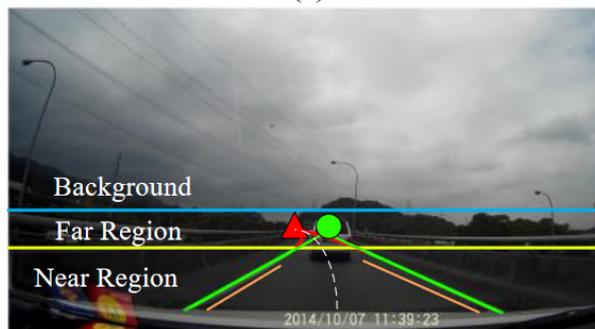


Figura 2.12 Detección de carril [82].

La detección de carril es un proceso automatizado utilizado en sistemas avanzados de asistencia al conductor, que utiliza sensores de visión y algoritmos de visión por computadora para detectar las líneas del carril en una carretera y estimar la posición del vehículo en relación con ellas, esta detección se realiza con cámaras montadas en el vehículo que procesan las imágenes del camino y proporcionan información útil al conductor, como alertas de desviación de carril, para prevenir accidentes y mejorar la

seguridad en la carretera. Cada uno de estos procesos en la detección de carril tiene un papel importante en el proceso de detección como se muestra en la Figura 2.13.

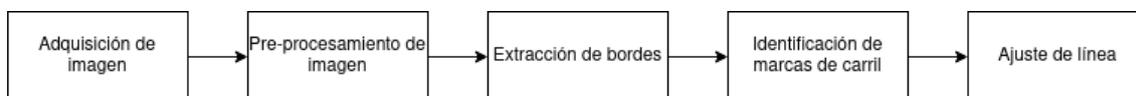


Figura 2.13 Etapas de detección de carril [12].

**Adquisición de imagen:** Es el proceso de capturar una imagen del carril de la carretera utilizando una cámara o cualquier otro dispositivo de imagen. La adquisición de imagen es la fase inicial y fundamental en el proceso de obtener una representación visual del carril de la carretera. Este proceso implica la utilización de una cámara u otro dispositivo de captura de imágenes para registrar y obtener una instantánea del entorno vial y, específicamente, del carril en cuestión.

La cámara, en este contexto, actúa como el medio principal para capturar datos visuales del camino. Puede estar ubicada en el vehículo, ya sea en el parabrisas, el espejo retrovisor o en otras ubicaciones estratégicas para obtener una vista clara del carril. Esta captura inicial es crucial para proporcionar información visual que será procesada por algoritmos y técnicas de visión por computadora en etapas posteriores, con el fin de identificar y analizar el carril de la carretera.

Durante la adquisición de imagen, es esencial considerar aspectos como la resolución, el ángulo de visión y la calidad de la cámara o dispositivo utilizado. La calidad de la imagen capturada influirá directamente en la precisión y eficacia de los algoritmos de detección de carriles que se aplicarán más adelante en el proceso. Además, la ubicación estratégica de la cámara y su orientación jugarán un papel importante en la obtención de una representación visual precisa del carril en el entorno vial.

**Pre-procesamiento de imagen:** Este proceso se enfoca en realizar ajustes y correcciones específicas para optimizar la claridad, el contraste y reducir el ruido presente en la imagen, lo que facilita su análisis y procesamiento posterior. La calidad de las imágenes es de suma importancia para la detección precisa de carriles en un entorno vial. Por ello, se ha llevado a cabo una extensa investigación en el desarrollo de métodos y técnicas de mejora de imágenes. Estos métodos están diseñados para abordar problemas comunes que afectan la calidad visual, tales como el ruido, la presencia de lluvia, neblina y el movimiento, que pueden distorsionar la información capturada por las cámaras. Dentro de las técnicas de procesamiento de imágenes se incluyen métodos de denoising (eliminación de ruido), deraining (eliminación de efectos causados por la lluvia), corrección de neblina y ajustes de brillo y contraste. Además, se han empleado redes neuronales, como las redes de aprendizaje profundo, para realizar restauraciones y mejoras específicas en la claridad visual, contribuyendo de manera significativa a mejorar la calidad de las imágenes capturadas. La aplicación de estos avances en el procesamiento de imágenes ha sido fundamental para el progreso de los sistemas de asistencia de mantenimiento de carril. Han mejorado la precisión, la confiabilidad y la capacidad de adaptación de estos sistemas en diversas condiciones y entornos de conducción, permitiendo una detección más eficaz y robusta de los carriles en situaciones adversas, lo que en última instancia, contribuye a la seguridad vial.

**Extracción de bordes:** La extracción de bordes es un proceso fundamental dentro del análisis de imágenes que tiene como objetivo detectar y resaltar las transiciones más significativas en la intensidad de los píxeles de una imagen. Estos cambios abruptos indican los límites entre diferentes objetos o regiones presentes en la escena, lo que resulta crucial en la identificación de elementos como las líneas del carril en una carretera. Este proceso se realiza mediante algoritmos específicos, como los operadores de detección de bordes Sobel, Prewitt, Canny, entre otros. Estos operadores calculan gradientes de intensidad en la imagen, identificando las áreas donde ocurren cambios bruscos en los niveles de gris o color. Como resultado, se generan mapas de bordes que destacan las regiones con mayor variación de intensidad, es decir, los contornos de los objetos presentes en la imagen. En el contexto de la detección de carriles, la extracción de bordes juega un papel crucial al identificar las transiciones entre el carril y el resto del entorno vial. Las líneas del carril suelen tener contrastes visuales definidos con respecto al asfalto u otros elementos circundantes, lo que se traduce en bordes detectables en la imagen. Al resaltar estos bordes, se proporciona información valiosa para proceder con pasos posteriores, como la identificación y seguimiento preciso de las líneas del carril. En resumen, la extracción de bordes es una etapa esencial en el procesamiento de imágenes, ya que permite la identificación de los contornos y límites presentes en la escena, siendo un paso inicial y crucial en la detección de elementos relevantes, como las líneas del carril en el contexto vial.

Operadores de Detección de Bordes Sobel y Prewitt:

Estos operadores se utilizan para calcular aproximaciones del gradiente de una imagen en direcciones específicas (horizontal y vertical) para detectar los bordes, son dos construcciones clásicas que difieren solo marginalmente en los filtros derivativos que emplean. Ambos operadores utilizan filtros lineales que se extienden sobre tres líneas y columnas adyacentes, respectivamente, para contrarrestar la sensibilidad al ruido de los operadores de gradiente simples de una sola línea/columna, el operador de Prewitt utiliza los núcleos del filtro.

HP: Definición: Representa los filtros utilizados en el operador de Prewitt.

Notación:  $HP_x^P$  y  $HP_y^P$ .

Descripción:

$HP_x^P$ : Filtro que calcula el gradiente en la dirección horizontal.

$HP_y^P$ : Filtro que calcula el gradiente en la dirección vertical.

Fórmulas:

$$HP_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$HP_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.2)$$

Que calculan los componentes promedio del gradiente a través de tres líneas o columnas adyacentes, respectivamente. Cuando los filtros están escritos en forma separada,

$$HP_x^P = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [-1 \ 0 \ 1] \quad (2.3)$$

$$HP_y^P = [1 \ 1 \ 1] * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad (2.4)$$

Respectivamente, se hace evidente que  $HP_x^P$  realiza un suavizado simple (de caja) sobre tres líneas antes de calcular el gradiente en x, y de manera análoga,  $HP_y^P$  suaviza sobre tres columnas antes de calcular el gradiente en y. Debido a la propiedad conmutativa de la convolución lineal, esto podría describirse de igual manera, es decir, con el suavizado aplicado después del cálculo de los gradientes. Los núcleos de filtro para el operador Sobel son casi idénticos; sin embargo, la parte de suavizado asigna un peso mayor a la línea y columna central actual, respectivamente:

HS:

Definición: Representa los filtros utilizados en el operador de Sobel.

Notación:  $HS_x^S$  y  $HS_y^S$ .

Descripción:

$HS_x^S$ : Filtro que calcula el gradiente en la dirección horizontal con un mayor peso en el centro.

$HS_y^S$ : Filtro que calcula el gradiente en la dirección vertical con un mayor peso en el centro.

Fórmulas:

$$HS_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$HS_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.6)$$

Las estimaciones de los componentes locales del gradiente se obtienen a partir de los resultados del filtro mediante una escala adecuada, es decir,

$$\nabla I(u, v) \approx \frac{1}{6} ((I * HP_x)(u, v) + (I * HP_y)(u, v)) \quad (2.7)$$

para el operador Prewitt y

$$\nabla I(u, v) \approx \frac{1}{8} ((I * HS_x)(u, v) + (I * HS_y)(u, v)) \quad (2.8)$$

para el operador Sobel.

Operador Sobel extendido

HE:

Definición: Representa los filtros del operador Sobel extendido.

Notación:  $HE_i^{ES}$ , donde  $i$  denota la orientación del filtro (desde 0 hasta 7).

Descripción: Responde a una orientación diferente, en intervalos de 45 grados.

Ejemplos de Filtros:

$$HE_0^{ES} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$HE_1^{ES} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (2.10)$$

$$HE_2^{ES} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.11)$$

$$HE_3^{ES} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad (2.12)$$

$$HE_4^{ES} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.13)$$

$$HE_5^{ES} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad (2.14)$$

$$HE_6^{ES} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.15)$$

$$HE_7^{ES} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad (2.16)$$

Solamente los resultados de cuatro de estos ocho filtros ( $HES_0^{ES}, HES_1^{ES}, \dots, HES_7^{ES}$ ) deben ser calculados ya que los otros cuatro son idénticos excepto por el signo invertido. Por ejemplo, debido a que  $HES_4^{ES} = -HES_0^{ES}$  y la convolución es lineal, se sigue que

$$I * HES_4 = I * -HES_0 = -(I * HES_0) \quad (2.17)$$

Es decir, el resultado para el filtro  $HS_4$  es simplemente el resultado negativo para el filtro  $HS_0$ . Las salidas direccionales  $D_0, D_1, \dots, D_7$  para los ocho filtros Sobel pueden calcularse de la siguiente manera:

D:

Definición: Representa las salidas direccionales de los filtros del operador Sobel extendido.

Notación:  $D_i$ , donde  $i$  denota la orientación de la salida (desde 0 hasta 7).

Descripción: Cada  $D_i$  representa la respuesta del filtro correspondiente a la orientación  $i$ .

Cálculo:

$$\begin{aligned}
D0 &\leftarrow I * H_0^{ES} \\
D1 &\leftarrow I * H_1^{ES} \\
D2 &\leftarrow I * H_2^{ES} \\
D3 &\leftarrow I * H_3^{ES} \\
D4 &\leftarrow -D0 \\
D5 &\leftarrow -D1 \\
D6 &\leftarrow -D2 \\
D7 &\leftarrow -D3.
\end{aligned} \tag{2.18}$$

La fuerza del borde  $E^S$  en la posición  $(u, v)$  se define como el máximo de las salidas de los ocho filtros; es decir,...

$$\begin{aligned}
E^{ES}(u, v) &= \max(D0(u, v), D1(u, v), \dots, D7(u, v)) \\
&= \max(|D0(u, v)|, |D1(u, v)|, |D2(u, v)|, |D3(u, v)|),
\end{aligned} \tag{2.19}$$

y el filtro que responde con mayor fuerza también determina la orientación local del borde como:

$$\Phi_{ES}(u, v) = \frac{\pi}{4} j, \quad \text{con } j = \arg \max_{0 \leq i \leq 7} D_i(u, v). \tag{2.20}$$

La fuerza del borde  $E^S$  en la posición  $(u, v)$  se define como el máximo de las salidas de los ocho filtros; es decir,...

$$\begin{aligned}
E^{ES}(u, v) &= \max(D0(u, v), D1(u, v), \dots, D7(u, v)) \\
&= \max(|D0(u, v)|, |D1(u, v)|, |D2(u, v)|, |D3(u, v)|),
\end{aligned} \tag{2.21}$$

y el filtro que responde con mayor fuerza también determina la orientación local del borde como:

$$\Phi_{ES}(u, v) = \frac{\pi}{4} j, \quad \text{con } j = \arg \max_{0 \leq i \leq 7} D_i(u, v). \tag{2.22}$$

### Parámetros y Literales

- $E^{ES}$  (Fuerza del borde):
  - Definición: La fuerza del borde en la posición  $(u, v)$  es una medida de la intensidad del borde en esa ubicación.
  - Fórmula:

$$E^{ES}(u, v) = \max(D0(u, v), D1(u, v), \dots, D7(u, v)) \tag{2.23}$$

o

$$E^{ES}(u, v) = \max(|D0(u, v)|, |D1(u, v)|, |D2(u, v)|, |D3(u, v)|) \tag{2.24}$$

- Descripción: Esto significa que la fuerza del borde en la posición  $(u, v)$  se determina como el valor máximo de las salidas absolutas de los ocho filtros direccionales aplicados en esa posición. Los filtros direccionales corresponden a diferentes orientaciones.
- $u, v$ :
  - Definición: Estas son las coordenadas en el plano de la imagen.
  - Descripción: Representan la posición de un píxel en la imagen donde se está calculando la fuerza del borde y su orientación.
- $D_i$  (Salida del filtro):
  - Definición: La salida del filtro  $i$ -ésimo en la posición  $(u, v)$ .
  - Notación:  $D_i(u, v)$  donde  $i$  puede ser un valor entre 0 y 7, representando las ocho posibles orientaciones de los filtros Sobel extendidos.
  - Descripción: Cada  $D_i$  representa la respuesta del filtro en la orientación  $i$  cuando se aplica en la posición  $(u, v)$ .
- max (Máximo):
  - Definición: Función que devuelve el valor máximo entre los argumentos que se le pasan.
  - Descripción: En el contexto de las ecuaciones, ‘max’ se utiliza para encontrar la mayor respuesta de borde entre las diferentes orientaciones del filtro.
- $|D_i(u, v)|$  (Valor absoluto):
  - Definición: El valor absoluto de la salida del filtro  $i$  en la posición  $(u, v)$ .
  - Descripción: Se usa para considerar solo la magnitud de la respuesta del filtro, independientemente de la dirección del gradiente.
- $\Phi_{ES}$  (Orientación del borde):
  - Definición: La orientación local del borde en la posición  $(u, v)$ .
  - Fórmula:
 
$$\Phi_{ES}(u, v) = \frac{\pi}{4} j \quad (2.25)$$

donde  $j = \arg \max_{0 \leq i \leq 7} D_i(u, v)$ .
  - Descripción: Esto indica la dirección del borde en radianes. El valor  $j$  determina cuál de las ocho orientaciones (espaciadas en intervalos de  $\frac{\pi}{4}$ ) tiene la máxima respuesta de borde. La función  $\arg \max$  encuentra el índice  $i$  para el cual  $D_i(u, v)$  es máximo.

La fuerza del borde  $E^S$  en la posición  $(u, v)$  se define como el máximo de las salidas de los ocho filtros; es decir,...

$$\begin{aligned}
 E^{ES}(u, v) &= \max(D0(u, v), D1(u, v), \dots, D7(u, v)) \\
 &= \max(|D0(u, v)|, |D1(u, v)|, |D2(u, v)|, |D3(u, v)|),
 \end{aligned}
 \tag{2.26}$$

y el filtro que responde con mayor fuerza también determina la orientación local del borde como:

$$\Phi_{ES}(u, v) = \frac{\pi}{4}j, \quad \text{con } j = \arg \max_{0 \leq i \leq 7} D_i(u, v).
 \tag{2.27}$$

Operador de Canny:

El operador propuesto por Canny es ampliamente reconocido y sigue siendo considerado como la vanguardia en la detección de bordes. Este método se esfuerza por alcanzar tres metas fundamentales: (a) reducir al mínimo el número de puntos de borde falsos, (b) lograr una precisa localización de los bordes y (c) proporcionar únicamente una marcación singular en cada borde. Estas características normalmente no se consiguen con operadores de borde simples, que en su mayoría se basan en primeras derivadas y umbralización posterior.

En su esencia, el "filtro" de Canny se basa en un método de gradiente que utiliza primeras derivadas, pero se vale de los cruces por cero de las segundas derivadas para una localización precisa de los bordes. En este sentido, el enfoque del método es similar a detectores de bordes que se fundamentan en las segundas derivadas de la función de la imagen.

En su implementación completa, el detector Canny emplea un conjunto de filtros orientados de tamaño relativamente grande a diversas resoluciones de imagen, y fusiona los resultados individuales en un único mapa de bordes. No obstante, es común usar solamente una versión de escala única del algoritmo, con un radio de filtro ajustable (parámetro de suavizado  $\sigma$ ), que aún así supera en rendimiento a la mayoría de los operadores de borde simples.

Además, el algoritmo no solo produce un mapa de bordes binario, sino que también identifica cadenas conectadas de píxeles de borde, simplificando en gran medida los pasos de procesamiento posteriores. Por lo tanto, incluso en su forma más básica (escala única), el operador de Canny suele ser preferido sobre otros métodos de detección de bordes.

En su forma básica (escala única), el operador de Canny realiza los siguientes pasos:

1. Preprocesamiento: Suaviza la imagen con un filtro gaussiano de ancho  $\sigma$ , que especifica el nivel de escala del detector de bordes. Calcula el vector de gradiente  $x/y$  en cada posición de la imagen filtrada y determina la magnitud y orientación del gradiente local.
2. Localización de bordes: Aísla máximos locales de la magnitud del gradiente mediante la supresión de no máximos a lo largo de la dirección del gradiente local.
3. Trazado de bordes y umbralización por histéresis: Recolecta conjuntos de píxeles de borde conectados a partir de los máximos locales aplicando umbralización por histéresis, como se muestra la Figura 2.14.

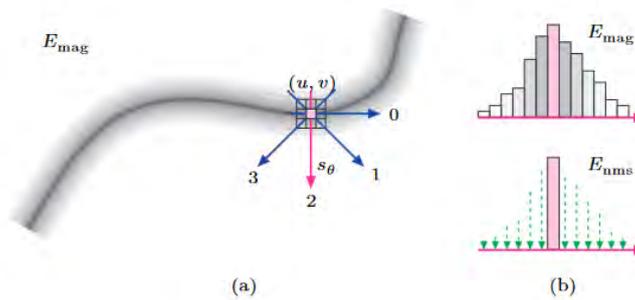


Figura 2.14 Supresión del máximo local de la magnitud del gradiente [9].

La dirección del gradiente en la posición  $(u, v)$  se cuantifica de manera gruesa en cuatro orientaciones discretas  $s\theta \in \{0, 1, 2, 3\}$  (a). Solo se consideran como puntos candidatos de borde aquellos píxeles donde la magnitud del gradiente  $E_{\text{mag}}(u, v)$  es un máximo local en la dirección del gradiente (es decir, perpendicular a la tangente del borde) (b). La magnitud del gradiente en todos los demás puntos se establece (se suprime) a cero.

Preprocesamiento: La imagen original de intensidad, representada por  $I$ , se somete a un proceso de suavizado mediante un filtro Gaussiano con un kernel  $H_{G,\sigma}$ . El parámetro  $\sigma$  determina la escala espacial para la detección de bordes. Posteriormente, se aplican filtros de primera derivada a la imagen suavizada,  $\bar{I}$ , para obtener las componentes  $\bar{I}_x$  y  $\bar{I}_y$  de los vectores de gradiente local. A continuación, se calcula la magnitud local  $E_{\text{mag}}$  como la norma del vector de gradiente correspondiente. Esto puede resultar útil para normalizar los valores de magnitud de borde dentro de un rango estándar en vista de los pasos posteriores de umbralización.

Localización de los Bordes: Se identifican los píxeles candidatos a bordes mediante una supresión local de los máximos de la magnitud del borde  $E_{\text{mag}}$ . Durante esta etapa, se conservan únicamente aquellos píxeles que representan un máximo local a lo largo de un perfil 1D en la dirección del gradiente, es decir, perpendicular al borde. Aunque el gradiente puede apuntar en diversas direcciones continuas, por lo general se emplean solo cuatro direcciones discretas para optimizar el proceso. Para que un píxel en la posición  $(u, v)$  sea considerado como candidato a borde, su magnitud de gradiente debe ser mayor que la de ambos vecinos inmediatos en la dirección indicada por el vector de gradiente  $(dx, dy)$  en esa posición. Si un píxel no es un máximo local, su valor de magnitud de borde se establece en cero (es decir, se suprime). Los valores de borde suprimidos por no ser máximos se almacenan en el mapa  $E_{\text{nms}}$ .

Rastreo de bordes y umbralización por histéresis: En el paso final, se recopilan conjuntos de puntos de borde conectados a partir de los valores de magnitud que permanecieron sin supresión en la operación anterior. Esto se realiza con una técnica llamada umbralización por histéresis usando dos valores de umbral diferentes,  $t_{lo}$  (con  $t_{hi} > t_{lo}$ ). Se escanea la imagen en busca de píxeles con magnitud de borde  $E_{\text{nms}}(u, v) \geq t_{hi}$ . Cada vez que se encuentra una ubicación (previamente no visitada), se inicia un nuevo rastreo de borde y se agregan todos los píxeles de borde conectados  $(u', v')$  a este mientras  $E_{\text{nms}}(u', v') \geq t_{lo}$ .

Transformada de Hough: La técnica conocida como Transformada de Hough constituye un enfoque versátil para ubicar cualquier forma que se pueda definir paramétricamente

dentro de una disposición de puntos. Es comúnmente empleada para detectar segmentos de líneas rectas en mapas de bordes. En un plano 2D, un segmento de línea puede describirse mediante dos parámetros de valores reales utilizando la conocida forma de pendiente-intersección.

$k$  y  $d$  son parámetros por lo cual se tienen que escribir como ecuaciones:

$$y = k \cdot x + d, \quad (2.28)$$

donde  $k$  es la pendiente y  $d$  es la intersección, es decir, la altura en la que la línea interceptaría el eje  $y$ .

Un segmento de línea que pasa por dos puntos de borde dados,  $p_1 = (x_1, y_1)$  y  $p_2 = (x_2, y_2)$ , debe satisfacer las condiciones, como se muestra en la Figura 2.15.

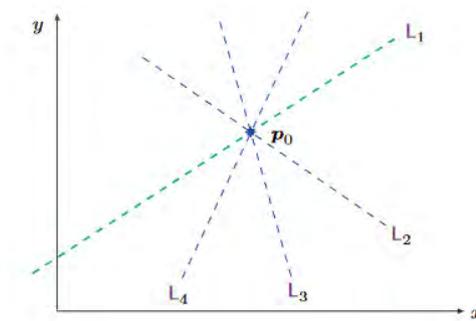


Figura 2.15 Un conjunto de líneas que pasan por un punto de imagen. Para todas las posibles líneas  $L_j$  que pasan por el punto  $p_0 = (x_0, y_0)$ , se cumple la ecuación  $y_0 = k_j x_0 + d_j$  para valores apropiados de los parámetros  $k_j$  y  $d_j$  [9].

$$y_1 = k \cdot x_1 + d \quad (2.29)$$

$$y_2 = k \cdot x_2 + d \quad (2.30)$$

Explicación de literales y parámetros:

1.  $k$ : Pendiente de la línea en el espacio de imagen.
2.  $d$ : Intersección de la línea con el eje  $y$  en el espacio de imagen.
3.  $L_j$ : Línea que pasa por un punto específico  $p_0$ , representada por sus parámetros  $(k_j, d_j)$ .
4.  $p_0 = (x_0, y_0)$ : Punto específico en la imagen por el cual pasan múltiples líneas.
5.  $k_j$ : Pendiente específica para una de las líneas que pasan por  $p_0$ .
6.  $d_j$ : Intersección específica para una de las líneas que pasan por  $p_0$ .
7.  $M_i$ : Línea en el espacio de parámetros correspondiente a un punto de imagen arbitrario  $p_i$ .

8.  $p_i = (x_i, y_i)$ : Punto arbitrario en la imagen.
9. Espacio de parámetros o espacio de Hough: Espacio donde se representan las líneas en términos de sus parámetros  $k$  y  $d$ .
10.  $(x, y)$ : Coordenadas en el espacio de imagen.
11.  $(k, d)$ : Coordenadas en el espacio de parámetros o espacio de Hough.

El propósito es determinar valores de  $k$  y  $d$  de modo que la mayor cantidad posible de puntos de borde se ajusten a la línea que representan.

La transformada de Hough aborda el problema desde otra perspectiva. Examina todos los posibles segmentos de línea que pasan por un único punto dado en la imagen. Cada línea  $L_j = \langle k_j, d_j \rangle$  que pasa por un punto  $p_0 = (x_0, y_0)$  debe satisfacer la condición:

$$L_j : y_0 = k_j x_0 + d_j \quad (2.31)$$

para valores adecuados  $k_j, d_j$ . La ecuación (1) está indeterminada y las posibles soluciones para  $k_j, d_j$  corresponden a un conjunto infinito de líneas que pasan por el punto dado  $p_0$ . Para un  $k_j$  dado, la solución para  $d_j$  en la ecuación (2) es

$$d_j = -x_0 \cdot k_j + y_0 \quad (2.32)$$

que es otra ecuación para una línea, donde ahora  $k_j, d_j$  son las variables y  $x_0, y_0$  son los parámetros constantes de la ecuación. El conjunto de soluciones  $\{(k_j, d_j)\}$  de la ecuación (1) describe los parámetros de todas las posibles líneas  $L_j$  que pasan por el punto de la imagen  $p_0 = (x_0, y_0)$ .

Para un punto de imagen arbitrario  $p_i = (x_i, y_i)$ , la ecuación (1) describe la línea

$$M_i : d = -x_i \cdot k + y_i \quad (2.33)$$

con los parámetros  $-x_i, y_i$  en el llamado espacio de parámetros o espacio de Hough, abarcado por las coordenadas  $k, d$ .

La relación entre el espacio de imagen  $(x, y)$  y el espacio de parámetros  $(k, d)$  se puede resumir como se muestra en la Figura 2.16 y en la Tabla 2.1.

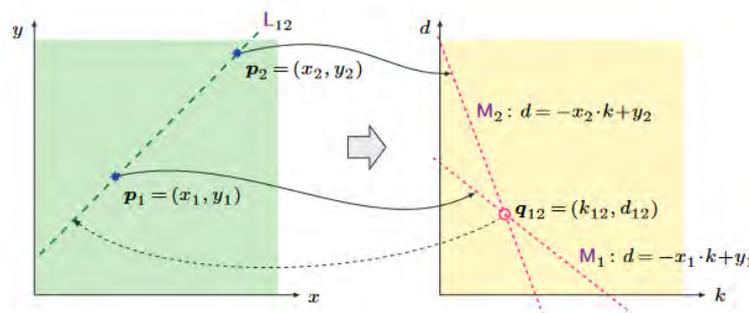


Figura 2.16 Representación gráfica de la relación entre el espacio de imagen y el espacio de parámetros. [9]

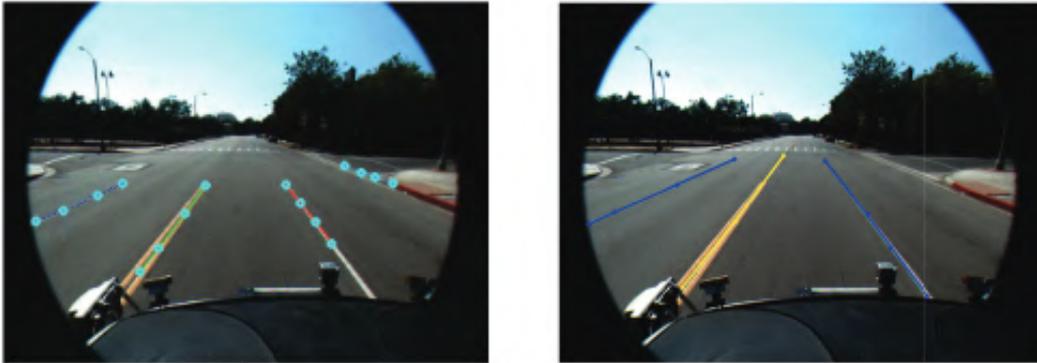


Figura 2.17 Marcas de carril identificadas [84].

Tabla 2.1 Relación entre el espacio de imagen y el espacio de parámetros. [9]

Espacio de la Imagen ( $x, y$ )	Espacio de Parámetros ( $k, d$ )
Punto $p_i = (x_i, y_i)$	$\leftrightarrow$ Línea $M_i : d = -x_i \cdot k + y_i$
Línea $L_j : y = k_j \cdot x + d_j$	$\leftrightarrow$ Punto $q_j = (k_j, d_j)$

Los valores de los parámetros para todas las posibles líneas que atraviesan el punto de imagen  $p_i = (x_i, y_i)$  en el espacio de imagen (a) se encuentran en una única línea  $M_i$  en el espacio de parámetros (b). Esto significa que cada punto  $q_j = (k_j, d_j)$  en el espacio de parámetros corresponde a una única línea  $L_j$  en el espacio de imagen. La intersección de las dos líneas  $M_1, M_2$  en el punto  $q_{12} = (k_{12}, d_{12})$  en el espacio de parámetros indica que una línea  $L_{12}$  a través de los dos puntos  $k_{12}$  y  $d_{12}$  existe en el espacio de imagen.

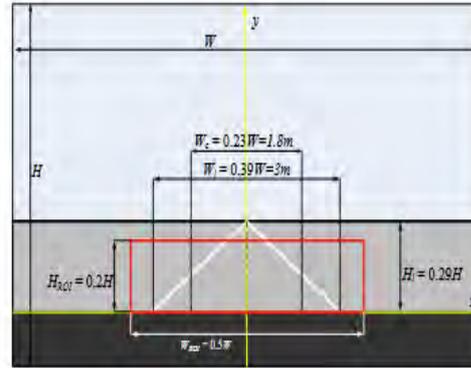
**Identificación de marcas de carril:** Es un proceso dentro del contexto de visión por computadora y detección de carriles en imágenes. Una vez que los bordes relevantes se han detectado mediante operadores como Sobel, Prewitt o Canny, se procede a buscar las líneas que representan los carriles en una imagen. Para realizar esta tarea, se utilizan técnicas de segmentación de imagen. La segmentación de imagen se refiere al proceso de dividir o separar una imagen en regiones o segmentos con características visuales similares. En este caso específico, se buscan las regiones de la imagen que corresponden a las líneas de los carriles. Las líneas de los carriles pueden ser identificadas dentro de estas regiones aplicando técnicas como umbrales de intensidad, transformaciones geométricas, o métodos más avanzados basados en aprendizaje automático. Estos métodos permiten distinguir las líneas del carril de otros elementos presentes en la imagen, como marcas viales, sombras o cambios en el pavimento. El objetivo final de la identificación de marcas de carril es definir con precisión la posición y orientación de los carriles en la imagen. Este paso es crucial en sistemas de asistencia de mantenimiento de carril en vehículos autónomos o sistemas de asistencia al conductor, ya que proporciona información esencial para mantener el vehículo en su carril y prevenir accidentes, como se muestra la Figura 2.17

**Ajuste de línea:** Este proceso implica el ajuste de un modelo de línea a las líneas del carril detectadas, esto se hace utilizando algoritmos de ajuste de línea como mínimos cuadrados para determinar la mejor línea que se ajusta a los datos de la línea del carril, este

proceso es fundamental para la detección precisa del carril y se utiliza para determinar la posición y orientación del carril en la imagen, como se muestra la Figura 2.18.



(a) Ajuste de línea[82].



(b) Ajuste de región de interés [84].

Figura 2.18 Ajuste de línea y región de interés.

## Discusión

En este capítulo se presentó un panorama general de los temas relacionados con esta investigación, abarcando conceptos fundamentales de los sistemas avanzados de asistencia al conductor (ADAS), los sistemas de asistencia basados en visión, la visión por computadora y los algoritmos aplicados a la detección y mantenimiento de carriles. Además, se describieron detalladamente los conceptos implementados en la propuesta de solución, proporcionando una base sólida para comprender su funcionamiento y justificación.

# Capítulo 3

## Estado del Arte

En este capítulo se presentan los trabajos más relevantes relacionados con la detección y mantenimiento de carril como parte de un asistente de mantenimiento de carril se presentan brevemente en esta sección. Esta revisión es un compilado de las aportaciones más relevantes para la investigación en los últimos 5 años. A continuación se presentan algunos trabajos relacionados.

### 3.1 Antecedentes

Algunas de las investigaciones que anteceden a la actual propuesta de tesis son:

#### **Conducción Autónoma de un Vehículo Simulado mediante un Modelo de Red Neuronal Convolutiva Recurrente [48].**

Se planteó un enfoque en esta investigación para lograr la conducción autónoma de un vehículo. Se procesaron secuencias de imágenes utilizando una Red Neuronal Convolutiva Recurrente, que permitió obtener ángulos de dirección del vehículo, necesarios para realizar distintas maniobras según las condiciones de la vía. Además, se incorporó un canal de profundidad a las imágenes para fusionar datos de sensores visuales y espaciales.

#### **Sistema Embebido para Asistencia de Conducción Basado en Lógica Difusa Tipo-2 [67].**

En este estudio se desarrolló un sistema de asistencia a la conducción que utiliza Lógica Difusa Tipo-2 y visión por computadora para detectar niveles de riesgo en tiempo real. Se tuvo en cuenta la percepción del vehículo que se encuentra en frente y la posible distracción del conductor en el camino. El sistema se implementó en una plataforma embebida NVIDIA Jetson TX2, que se probó en un automóvil real. Los resultados demostraron una alta eficacia, con una tasa de acierto superior al 0.90% y una velocidad de procesamiento de 24.63 cuadros por segundo.

#### **Navegación de un robot omnidireccional basada en Lógica Difusa Tipo-2 [75].**

En este trabajo se desarrolló un sistema de navegación para un robot omnidireccional. Se utilizó una cámara estéreo para capturar imágenes y generar un mapa de profundidad, el cual fue utilizado para calcular las trayectorias necesarias para alcanzar un objetivo. Para determinar las velocidades lineales y angulares del robot, se empleó la Lógica Difusa

Tipo-2. El sistema de navegación fue implementado en una plataforma embebida NVIDIA Jetson TX2, en la cual se llevaron a cabo pruebas con obstáculos tanto dinámicos como estáticos. Además, se utilizó el algoritmo *Probabilistic Random Map* como método de planificación de rutas.

### **Detección de obstáculos y planos durante el desplazamiento vehicular [54].**

En este trabajo se empleó el sensor RGB-D, que tenía la capacidad de trabajar en conjunto con LIDAR y disminuir el ruido para obtener una nube de puntos más optimizada. A partir de esta información sensorial, se realizó un análisis en tiempo real, segmentando los planos del suelo e identificando los obstáculos mediante la extracción de características relevantes. Esto permitió obtener una representación precisa del entorno y facilitó la toma de decisiones para la planificación de las maniobras del vehículo autónomo.

## **3.2 Trabajos relacionados**

### **Detección de carriles**

#### ***A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle's Forward Monocular Camera [1]***

En este artículo se presentó la estrategia para la detección y seguimiento de carriles, implementando las características del sistema de asistencia para el conductor (DAS), que incluyen la advertencia de cambio y el asistente de mantenimiento del carril. Para lograr buenos resultados, el procesamiento digital de imagen se dividió en 3 niveles. 1) En el nivel-bajo, la dimensión de la imagen de entrada se redujo de tres capas a una capa, se mejoró la nitidez y se definió la región de interés en función de la distancia mínima de seguridad del vehículo de adelante. 2) El nivel-intermedio correspondió a la característica del extractor para el diseño de la detección de los bordes del carril. 3) El nivel-alto correspondió al desarrollo de la estrategia de seguimiento de los carriles. Se utilizó la transformada de Hough y una interpolación *spline* que preserve la forma para lograr un ajuste de carril suave. En la Figura 3.1, se puede observar la representación de la estrategia propuesta.

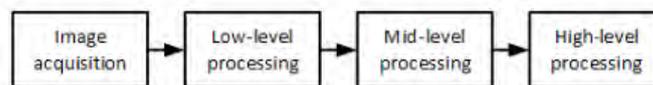


Figura 3.1 Representación de la estrategia propuesta [1].

#### ***RS-Lane: A Robust Lane Detection Method Based on ResNeSt and Self-Attention Distillation for Challenging Traffic Situations [98]***

En este artículo se presentó *RS-Lane*, un algoritmo de detección de carriles basado en la segmentación de instancias y *LaneNet*. Se utilizó *Split Attention*, *Self-Attention Distillation* (SAD) para mejorar la representación de características de las marcas de carril. *RS-Lane*

pudo detectar carriles sin límites de número, incluso en ausencia de las líneas de carril, occlusión de sombras y condiciones de iluminación complejas. Se lograron resultados comparables con SOTA en los conjuntos de datos *TuSimple* y *CULane*. En la Figura 3.2, se puede observar el diagrama de flujo de método desarrollado.

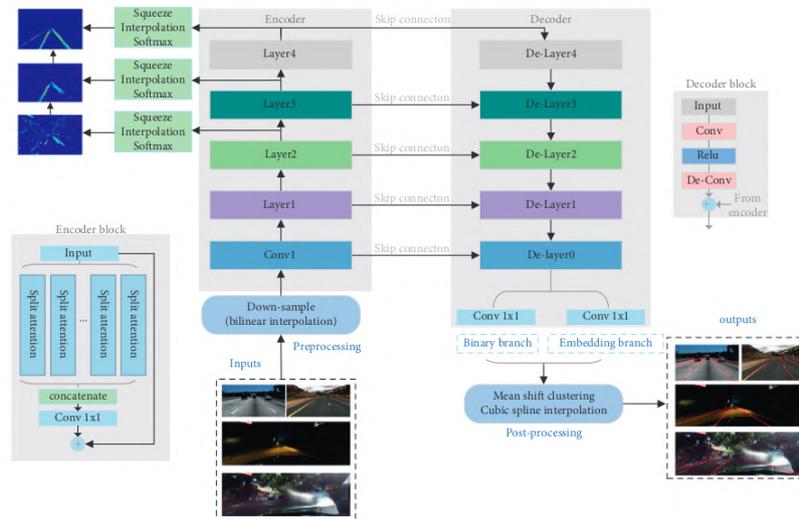


Figura 3.2 Diagrama de flujo de método desarrollado [98].

**An Advanced Lane-Keeping Assistance System With Switchable Assistance Modes [6]**

En este artículo se propuso un sistema avanzado de asistencia para mantenerse en el carril (LKAS), con dos modos de asistencia: el modo de prevención de cambio de carril y el modo de copiloto de mantenimiento de carril. Se construyó la estructura del sistema y se definieron las funciones y estrategias de control para los dos modos de asistencia. Los experimentos de conductor en el circuito demostraron la efectividad de ambos modos de asistencia. El enfoque propuesto presentó las ventajas de los modos conmutables, pero su aplicación en vehículos reales planteó muchos desafíos. En la Figura 3.3, se puede observar la estructura del sistema propuesto.

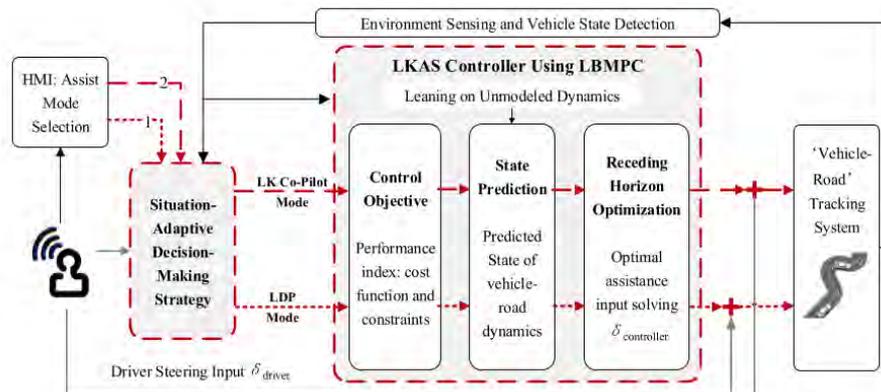


Figura 3.3 Estructura del sistema propuesto [6].

### *Unsupervised Labeled Lane Markers Using Maps [4]*

En este artículo se presentó la anotación automática de carriles acotados en imágenes y la asignación de atributos a cada carril, como posiciones 3D, mediante el uso de datos de mapas. El método proyectó los carriles acotados del mapa al espacio de la imagen para distancias lejanas y se basó en una optimización basada en muestras para refinar las proyecciones y aumentar la precisión de las etiquetas. Se publicó el conjunto de datos no supervisado LLAMAS, que consta de 100,042 imágenes con carriles acotados etiquetados. Este conjunto de datos se utiliza como punto de referencia y métodos de referencia para la detección de carriles acotados. El conjunto de datos contiene anotaciones a nivel de píxel de marcadores de carril discontinuos, puntos finales 2D y 3D para cada carril acotado, y asociaciones para vincular los carriles acotados.

### *LDNet: End-to-End Lane Marking Detection Approach Using a Dynamic Vision Sensor [50]*

En este artículo se presentó un codificador-decodificador basado en redes neuronales convolucionales para la detección de carriles utilizando una cámara de eventos. La arquitectura incluyó un codificador, un bloque de agrupación de pirámide espacial atrous (ASPP) y un decodificador guiado por la atención. El método propuesto se evaluó en el conjunto de datos DVS para la extracción de carriles (DET) y se logró una mejora significativa en las puntuaciones de F1 en tareas de detección de señalización de carriles multiclase y clase binaria. En la Figura 3.4, se puede observar el diagrama del método propuesto.

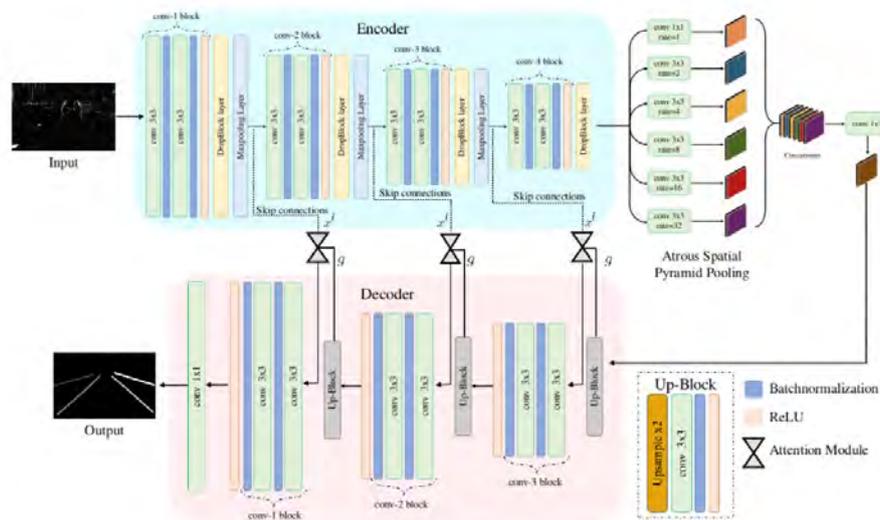


Figura 3.4 Diagrama del método propuesto [50].

### ***Lane Detection and Trajectory Tracking Control of Autonomous Vehicle Based on Model Predictive Control [36]***

En este artículo se presentó una aproximación que combinó la detección de carriles y el control predictivo de modelos para obtener de manera eficiente información de las líneas de carriles y realizar el seguimiento de trayectorias en vehículos autónomos. Se emplearon tecnologías de reconocimiento de imágenes, como la extracción dinámica de regiones de interés, la detección de bordes y la identificación de líneas rectas utilizando la transformada de Hough. Además, se desarrolló un controlador difuso adaptado a situaciones de inestabilidad del vehículo en condiciones de alta velocidad y baja adherencia en la superficie de la carretera.

### ***Robust Lane Detection and Tracking for Real-Time Applications [43]***

En este artículo se presentó un algoritmo de detección de carril basado en visión en tiempo real, el cual incorporó una región de interés eficiente para reducir el alto nivel de ruido y el tiempo de cálculo. El algoritmo propuesto procesó señales de gradiente y color, además de realizar un agrupamiento de líneas con pruebas de exploración para verificar las características de acotamiento del carril. Se implementó un mecanismo para eliminar las marcas de carril falsas y realizar un seguimiento de las marcas de carril reales utilizando datos estadísticos acumulados. Los resultados experimentales demostraron que el algoritmo propuesto proporciona resultados precisos y cumple con los requisitos de operación en tiempo real en sistemas integrados con baja potencia de cómputo. Se puede observar el diagrama de flujo propuesto en la Figura 3.5.

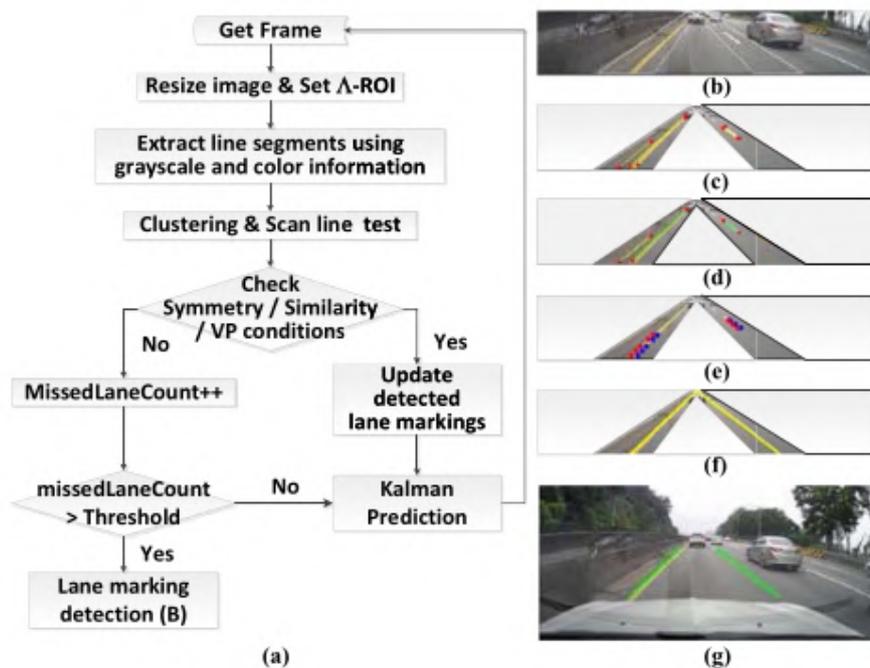


Figura 3.5 Diagrama de flujo propuesto [43].

***Ultra-low Complexity Block-based Lane Detection and Departure Warning System [83]***

En este artículo se presentó un sistema de alerta de salida y detección de carril basado en bloques de complejidad ultrabaja. Se utilizó la distribución de las marcas de carril en la región cercana al vehículo para determinar una región de interés (ROI) parametrizada. En la Figura 3.6 se puede observar el esquema de flujo del sistema desarrollado.

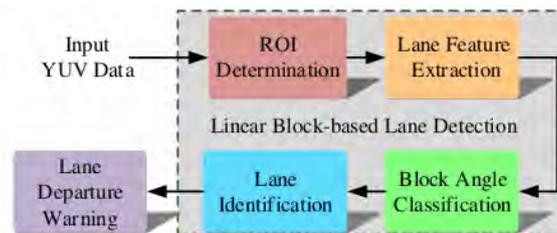


Figura 3.6 Esquema de flujo del sistema desarrollado [83].

***Lane Detection Based on Inverse Perspective Transformation and Kalman Filter [89]***

En este artículo se presentó un algoritmo para la detección de carriles basado en la transformación de perspectiva inversa y el filtro de Kalman. Se propuso un método de transformación de perspectiva inversa para generar una imagen de vista superior sin necesidad de obtener los parámetros internos y externos de la cámara. En la Figura 3.7 se puede observar el *framework* del método propuesto.

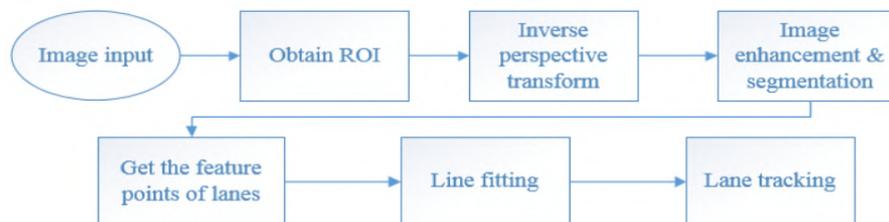


Figura 3.7 *Framework* del método propuesto. [89].

***Dynamic integration and online evaluation of vision-based lane detection algorithms [85]***

En este artículo se presentó un sistema de detección de carril basado en visión con integración dinámica. Para el sistema de detección de carriles, se implementaron dos módulos de detección. En primer lugar, se diseñó un módulo de detección de carril principal utilizando el filtro orientable y el algoritmo de la transformada de Hough. Posteriormente, se activó un algoritmo secundario que combinó el modelo de mezcla gaussiana para la segmentación de imágenes y el consenso de muestra aleatoria para el ajuste del modelo de carril cuando el algoritmo principal encontraba una confianza de detección baja.

### ***A 3D Estimation of Structural Road Surface Based on Lane-Line Information [86]***

En este artículo se presentó un método de estimación 3D de la superficie de la carretera estructural basado en la información de la línea del carril de tráfico. En primer lugar, se seleccionó el modelo semiparamétrico *B-spline* para describir la línea del carril en la carretera. Durante el proceso de ajuste de *B-spline*, se utilizó la extracción de regiones de interés (ROI) mediante la técnica de *Image District Extraction*, dividiendo así los píxeles de la carretera en partes más pequeñas. Posteriormente, se aplicaron la transformada de *Hough* y el método de *Least Squares* para ajustar líneas cortas en cada ROI. En la Figura 3.8 se puede observar el diagrama de principio de transformación de similitud.

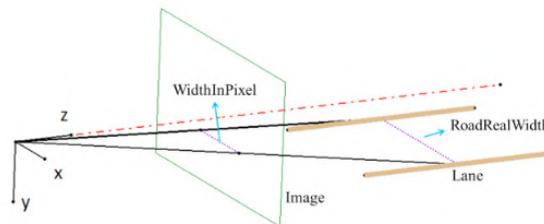


Figura 3.8 Diagrama de principio de transformación de similitud [86].

### ***Implementing and Parallelizing Real-time Lane Detection on Heterogeneous Platforms [78]***

En este artículo se presentó un algoritmo de detección de carril mejorado basado en la transformación de la imagen a vista superior y el ajuste del modelo *RANdom SAMple Consensus* (RANSAC) en dos etapas. Mediante la adaptación de una matriz de homografía afín fuera de línea, se delimitó una *region of interest* (ROI) adaptativa para la transformación subsiguiente en línea utilizando *Warp Perspective Mapping* (WPM). Este enfoque permitió analizar videos en carretera de manera eficiente y generar una ROI adaptativa sin la necesidad de conocer previamente los parámetros de la cámara.

### ***A Robust Method for Lane Detection under Adverse Weather and Illumination Conditions Using Convolutional Neural Network [92]***

En este artículo se presentó un método de detección de carriles mediante aprendizaje profundo que fue capaz de detectar carriles en diversas situaciones meteorológicas y de iluminación. Además, este método tenía como objetivo ayudar al conductor a tener una visión clara del carril, generando alertas en caso de cambios involuntarios o incorrectos de carril para prevenir accidentes.

### ***Learning full-reference quality-guided discriminative gradient cues for lane detection based on neural networks [47]***

En este artículo se presentó un marco de trabajo (*Framework*) para la detección de carriles en vehículos autónomos utilizando un modelo de aprendizaje profundo llamado gradiente discriminativo consciente de la calidad de referencia completa. Se propusieron dos tipos

de redes profundas y se diseñó una red convolucional profunda guiada por gradiente para detectar la presencia de carriles. Además, se utilizó el método de evaluación de calidad de imagen de referencia completa (FR-IQA) para identificar señales de gradiente más discriminativas. Se implementó una capa neuronal recurrente para representar la distribución espacial de los carriles detectados, cuyas señales visuales son difíciles de definir explícitamente. En la Figura 3.9 se puede observar el diagrama del método desarrollado.

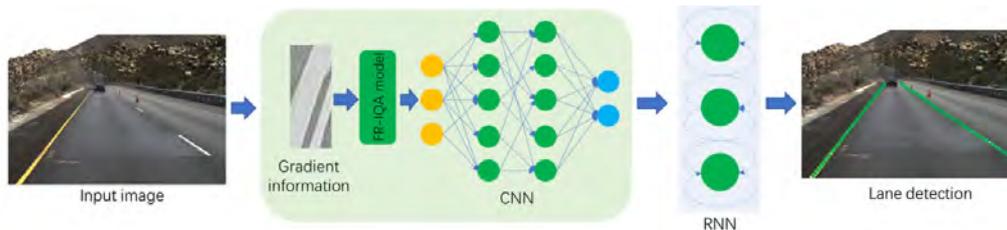


Figura 3.9 Diagrama del método desarrollado [47].

### ***Highly Curved Lane Detection Algorithms Based on Kalman Filter [18]***

En este artículo se presentó un algoritmo de detección de carril en curva muy pronunciada basado en el filtro de Kalman. Se implementaron modelos de ecuación de parábola y de ecuación de círculo dentro del filtro de Kalman para estimar los parámetros de un carril con curva pronunciada. Esto permitió mejorar la precisión y robustez de la detección de carriles en este tipo de condiciones. En la Figura 3.10 se puede observar el diagrama de flujo de los algoritmos de detección de carriles.

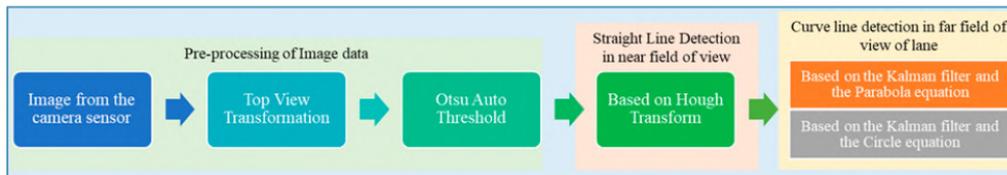


Figura 3.10 Diagrama de flujo de los algoritmos de detección de carriles [18].

### ***LaneDraw: Cascaded lane and its bifurcation detection with nested fusion [66]***

En este artículo se presentó una segmentación precisa de carriles y la identificación de cada carril. Se desarrolló un modelo de detección de carril de instancia en cascada y un método de fusión anidado de incrustación de píxeles de bifurcación basado en la incrustación de píxeles de segmentación binaria completa con un clúster de autoagrupación, denominado *LaneDraw*. Este enfoque permitió una detección y segmentación más precisa de los carriles, mejorando la capacidad de identificación de cada carril individualmente. En la Figura 3.11 se puede observar el esquema de la arquitectura propuesta.

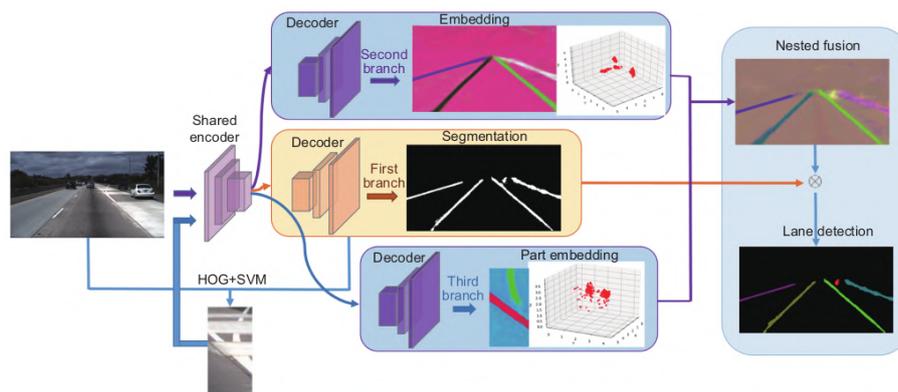


Figura 3.11 Esquema de la arquitectura propuesta [66].

### ***Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure [27]***

En este artículo se presentó un método de detección y seguimiento de carril para vehículos autónomos utilizando un marco de trabajo basado en *Integrated Behavioral Framework* (IBF). El marco de trabajo IBF constaba de tres módulos: la placa del vehículo (VB), el módulo de nube (CM) y el control remoto del vehículo. Inicialmente, la detección y el seguimiento del carril eran realizados por el VB. En caso de cualquier falla, los datos eran transferidos al CM para su procesamiento, y los resultados se enviaban de vuelta al VB para que tomara la acción correspondiente. Si el CM detectaba un cambio de carril, el vehículo autónomo era controlado de forma remota y se reiniciaba el VB.

### ***Lane Detection Based on Connection of Various Feature Extraction Methods [44]***

En este artículo se presentó un método de preprocesamiento y selección de ROI. El objetivo principal fue utilizar la transformación de color HSV para extraer las características blancas y agregar la detección preliminar de características de borde en la etapa de preprocesamiento. A continuación, se seleccionó el ROI basado en el preprocesamiento propuesto.

### ***Robust multi-lane detection and tracking using adaptive threshold and lane classification [72]***

En este artículo se presentó un método de umbral adaptativo para extraer características de carril fuertes de imágenes con obstáculos y carriles apenas visibles. Además, se propuso un algoritmo de consenso de muestreo aleatorio (RANSAC) que utilizó la retroalimentación de los ángulos de los bordes de los carriles y la curvatura del historial del carril para evitar la detección de carriles falsos. Por último, se mejoró significativamente el rendimiento de la detección de carriles al seleccionar solo los carriles verificados a través del algoritmo de clasificación de carriles.

***Development of an embedded road boundary detection system based on deep learning [60]***

En este artículo se presentó un sistema integrado de detección de límites de carreteras basado en aprendizaje profundo. El sistema fue capaz de detectar tanto caminos estructurados como no estructurados en diversas situaciones. Para mejorar la visibilidad de las marcas de carril en las imágenes, se empleó un codificador automático de convolución que redujo el ruido y eliminó otros objetos presentes en las imágenes, dejando únicamente las marcas de carril. A continuación, se extrajeron los puntos característicos de la línea del carril y se ajustó una línea hiperbólica para representarla. Por último, se utilizó un filtro de partículas para realizar el seguimiento de los carriles a lo largo del tiempo.

***Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments [11]***

El artículo propone un algoritmo de detección de carril para vehículos inteligentes en condiciones de carreteras complejas y entornos dinámicos. El enfoque se desarrolla para abordar problemas como la baja precisión de detección de métodos tradicionales y el rendimiento deficiente en tiempo real de las metodologías basadas en aprendizaje profundo. En primer lugar, se utiliza el algoritmo de umbral de superposición para la detección de bordes en imágenes distorsionadas, obteniendo una vista aérea del carril mediante la extracción de regiones de interés y la transformación inversa de perspectiva. Luego, se emplea el algoritmo de consenso de muestras aleatorias para ajustar las curvas de los carriles basándose en un modelo de curva de B-spline de tercer orden, seguido por la evaluación del ajuste y el cálculo del radio de curvatura de la curva.

***Lane Detection of Curving Road for Structural High-way with Straight-curve Model on Vision [76]***

El artículo propone un algoritmo de detección de curvas para identificar con precisión las líneas de carril en carreteras curvas, un área propensa a accidentes de tráfico. La detección de carriles en estas condiciones con los métodos tradicionales es difícil. El algoritmo se basa en un modelo de curva recta-curva que resulta aplicable para la mayoría de las condiciones de carreteras curvas. El método divide la imagen de la carretera en regiones de interés y antecedentes, luego distingue entre regiones de curva y recta. Establece un modelo matemático que incluye ecuaciones para modelos de curvas y rectas usando la transformada de Hough mejorada.

***Lane detection technique based on perspective transformation and histogram analysis for self-driving cars [51]***

El artículo propone un algoritmo de percepción basado exclusivamente en datos de visión o cámaras. Se enfoca en demostrar un método potente de detección de carriles de extremo a extremo utilizando técnicas contemporáneas de visión por computadora para vehículos autónomos. Inicialmente, se presenta un enfoque minimalista basado en detección de bordes y regresión polinómica, que es el enfoque base para detectar solo las líneas de

carril rectas. Luego se propone una técnica mejorada de detección de carriles basada en transformaciones de perspectiva y análisis de histogramas. En esta última técnica, se pueden detectar tanto líneas de carril rectas como curvas. Para demostrar la superioridad del enfoque propuesto de detección de carriles sobre el enfoque convencional, se presentan resultados de simulación en diferentes entornos.

***In Defense of Pre-trained ImageNet Architectures for Real-time Semantic Segmentation of Road-driving Images [52]***

El artículo se centra en el desafío de implementar segmentación semántica en tiempo real en plataformas móviles como vehículos, drones y robots, considerando la alta complejidad computacional involucrada. A diferencia de enfoques anteriores que utilizan arquitecturas personalizadas más livianas, proponen una estrategia alternativa para lograr un rendimiento superior en una amplia gama de presupuestos computacionales. El método propuesto se basa en una arquitectura liviana de propósito general como el motor principal de reconocimiento. Luego, emplea un proceso de upsampling ligero con conexiones laterales para restaurar la resolución de la predicción de manera más rentable. Además, plantea aumentar el campo receptivo fusionando características compartidas en múltiples resoluciones de manera novedosa.

***FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network [61]***

En este artículo, se aborda la búsqueda de modelos predictivos para la detección de carriles en vehículos autónomos que se generalicen a una amplia gama de entradas de sensores. Se introduce un novedoso modelo completamente convolucional para la detección de carriles que aprende a decodificar las estructuras de carril sin depender del procesamiento posterior. Este modelo permite representar un número arbitrario de carriles por imagen, conserva la representación de polilíneas de los carriles sin reducirlos a polinomios y dibuja los carriles de manera iterativa sin necesidad de redes neuronales recurrentes. Además, al incluir una estimación de la distribución conjunta de píxeles vecinos pertenecientes al mismo carril, ofrece una definición natural y económica computacionalmente de la incertidumbre. En cuanto al entrenamiento, se demuestra un método simple y efectivo para adaptar el modelo a nuevos entornos mediante transferencia de estilo no supervisada. Al entrenar el modelo para realizar predicciones invariantes a diferencias estilísticas entre imágenes, logran un rendimiento sólido en condiciones climáticas y de iluminación distintas a las de los conjuntos de datos anotados disponibles públicamente.

***DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios [87]***

El artículo presenta un avance significativo en la estimación de mapas de disparidad a partir de imágenes estéreo para aplicaciones de conducción autónoma. A pesar de los avances actuales en este campo, el uso de conjuntos de datos estéreo limitados y la precisión de rango inestable en los métodos estéreo existentes siguen siendo desafíos para la correspondencia estéreo a nivel industrial en el ámbito de la conducción autónoma. En

respuesta a esta limitación, los autores han desarrollado un nuevo conjunto de datos estéreo llamado DrivingStereo. Este conjunto de datos es considerablemente más grande que los conjuntos de datos previos, con más de 180,000 imágenes que abarcan una amplia gama de escenarios de conducción, siendo cientos de veces más grande que el conjunto de datos KITTI Stereo.

#### ***Learning Lightweight Lane Detection CNNs by Self Attention Distillation [35]***

El artículo introduce SAD (Self Attention Distillation) para mejorar la detección de carriles en modelos de redes neuronales. SAD permite que un modelo aprenda de sí mismo, utilizando mapas de atención internos para mejorar el rendimiento sin necesitar más datos de entrenamiento. Funciona bien en modelos ligeros como ENet, siendo más rápido y preciso que los enfoques actuales.

#### ***3D-LaneNet: End-to-End 3D Multiple Lane Detection [24]***

En este artículo, se presenta una red neuronal que predice directamente la disposición tridimensional de carriles en una escena de carretera a partir de una única imagen. Este trabajo aborda esta tarea utilizando sensores a bordo sin asumir un ancho de carril constante conocido ni depender de entornos previamente mapeados. La arquitectura de red, denominada 3D-LaneNet, emplea dos conceptos nuevos: mapeo inverso de perspectiva (IPM) intra-red y representación de carriles basada en anclas. La proyección IPM intra-red facilita un flujo de información de doble representación tanto en vista de imagen regular como en vista superior. Una representación de anclaje por columna permite un enfoque de extremo a extremo que reemplaza las heurísticas comunes como la agrupación y el rechazo de valores atípicos, convirtiendo la estimación de carriles en un problema de detección de objetos.

#### ***Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks [99]***

El artículo se centra en la detección de carriles en escenas de conducción, un componente fundamental para los sistemas avanzados de asistencia al conductor y los vehículos autónomos. Aunque se han propuesto muchos métodos sofisticados de detección de carriles, la mayoría se enfoca en detectar los carriles a partir de una sola imagen, lo que a menudo conduce a un rendimiento insatisfactorio al enfrentar situaciones extremadamente difíciles, como sombras intensas, degradación severa de las marcas viales u obstrucciones vehiculares. Propone abordar esta limitación investigando la detección de carriles mediante el uso de múltiples fotogramas continuos de una escena de conducción. Para ello, se propone una arquitectura híbrida de redes neuronales convolucionales (CNN) y recurrentes (RNN). Cada fotograma se procesa utilizando bloques de CNN, y las características de múltiples fotogramas continuos, que mantienen la propiedad de series temporales, se introducen en bloques de RNN para el aprendizaje de características y la predicción de carriles.

## Mejoramiento de imagen

### *Semantic Segmentation: A Zoology of Deep Architectures [3]*

En este artículo se presentó la evolución de arquitecturas profundas para la segmentación semántica, destacando el éxito inicial de FCN en 2015 y el surgimiento de métodos posteriores como PSPnet y DeepLabV3, que mejoran la segmentación. Se señala que, aunque las arquitecturas convolucionales han estancado su avance, los transformadores de NLP han tomado relevancia, presentando modelos como SETR y SegFormer. Estos modelos son parte de la biblioteca MM-Segmentation basada en PyTorch.

### *An Analysis of Multi-stage Progressive Image Restoration Network (MPRNet) [63]*

En este artículo se presentó un modelo de red neuronal convolucional MPRNet de tres etapas diseñado para la restauración de imágenes. Sus resultados destacados se han observado en múltiples conjuntos de datos al abordar problemas como el ruido, el desenfoque y la eliminación de efectos de lluvia en las imágenes. Su singularidad radica en su capacidad para eliminar estos tres tipos de artefactos utilizando una única arquitectura.

### *LSD: a Line Segment Detector [29]*

En este artículo se presentó un Detector de Segmentos de Línea LSD eficiente que ofrece resultados precisos a nivel de subpíxel. Su particularidad radica en su capacidad para trabajar en imágenes digitales sin requerir ajustes de parámetros.

### *A Sub-Pixel Edge Detector: an Implementation of the Canny/Devernavy Algorithm [30]*

En este artículo se presentó un detector de bordes de imagen que genera puntos de borde encadenados con precisión a nivel de subpíxel. El método incorpora los principales conceptos de los algoritmos clásicos de Canny y Devernavy. El análisis demuestra que una ligera modificación a la formulación original mejora la precisión a nivel de subpíxel de los puntos de borde.

### *Density-aware Single Image De-raining using a Multi-stream Dense Network [96]*

En este artículo se presentó eliminar franjas de lluvia en imágenes individuales, causado por densidades no uniformes de lluvia. Se presenta el método DID-MDN, una innovadora red neuronal convolucional densamente conectada de múltiples flujos, que aborda conjuntamente la estimación de la densidad de lluvia y la eliminación de franjas. Este enfoque permite que la red identifique automáticamente la densidad de lluvia y elimine eficazmente las franjas de lluvia correspondientes, guiada por la información de densidad estimada.

### *Deep Joint Rain Detection and Removal from a Single Image [88]*

En este artículo se presentó eliminar la lluvia de una imagen única, incluso en condiciones de lluvia intensa y acumulación de franjas de lluvia, se baso en una arquitectura de

aprendizaje profundo multi-tarea que aprende el mapa binario de franjas de lluvia, la apariencia de las franjas de lluvia y el fondo limpio.

### ***Total Variation Deconvolution using Split Bregman [26]***

En este artículo se presentó la restauración de imágenes borrosas mediante deconvolución es crucial para revertir el efecto del desenfoque y el ruido en una imagen. La técnica de regularización de variación total (TV) ha demostrado ser efectiva en la eliminación del desenfoque al equilibrar la recuperación de frecuencias y la supresión de ruido. En este artículo, se discute la aplicación de la deconvolución regularizada por TV en problemas de desenfoque con ruido gaussiano, presentando una solución eficiente mediante el algoritmo de Bregman dividido.

### ***Multi-Stage Progressive Image Restoration [95]***

En este artículo, se presentó un enfoque para abordar tareas de restauración de imágenes, enfocado en lograr un equilibrio óptimo entre los detalles espaciales y la información contextualizada durante la recuperación de imágenes. Consiste en una arquitectura multi-etapa denominada MPRNet, la cual se encarga progresivamente de aprender funciones de restauración para imágenes degradadas.

### ***Restormer: Efficient Transformer for High-Resolution Image Restoration [94]***

En este artículo se presenta un enfoque innovador para la restauración de imágenes, centrado en el desarrollo de un modelo eficiente basado en transformadores denominado Restormer. El Restormer logra resultados destacados en diversas tareas de restauración de imágenes, como la eliminación de lluvia, el desenfoque de movimiento en imágenes individuales, el desenfoque por desenfoque (tanto datos de imagen única como de píxel doble), y la reducción de ruido en imágenes (incluyendo ruido gaussiano en escala de grises/color y la reducción de ruido en imágenes reales).

### ***Uformer: A General U-Shaped Transformer for Image Restoration [80]***

En este artículo se presento Uformer, una arquitectura basada en transformadores efectiva y eficiente para la restauración de imágenes. Se construye una red codificador-decodificador jerárquica utilizando bloques Transformer. Se realizan extensos experimentos en varias tareas de restauración de imágenes, incluyendo denoising, desenfoque por movimiento, desenfoque de desenfoque y eliminación de lluvia. En la Figura 3.12 se puede observar los resultados del método desarrollado.



Figura 3.12 Comparaciones visuales entre el método propuesto y otros métodos del estado del arte [80].

### ***Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning [45]***

En este artículo se aborda el desafío de la eliminación de lluvia en imágenes, particularmente en situaciones de lluvia intensa, donde los métodos convencionales no logran resultados satisfactorios. Para superar estas limitaciones, se propone un enfoque novedoso basado en una red de dos etapas: un esquema basado en la física seguido de un refinamiento GAN guiado por la profundidad.

### ***Blind Image Deblurring with Local Maximum Gradient Prior [14]***

En este artículo se presenta el problema de imágenes borrosas de manera ciega, es decir, sin conocer el kernel de desenfoque. Se presenta un método basado en el "Local Maximum Gradient (LMG)" como prioridad de imagen. Este enfoque se inspira en la observación intuitiva de que el valor máximo del gradiente de un parche local disminuirá después del proceso de desenfoque, lo cual se demuestra ser cierto tanto matemáticamente como empíricamente.

### ***Frame-Consistent Recurrent Video Deraining with Dual-Level Flow [87]***

En este artículo, se aborda el problema de la eliminación de la lluvia de videos mediante la propuesta de un marco de trabajo más completo que considera los factores adicionales de degradación en escenas reales. El marco propuesto se basa en una red recurrente. En la primera etapa, se estima el cuadro libre de lluvia a partir de un solo cuadro de lluvia. Este cuadro se utiliza como guía, junto con cuadros limpios previamente recuperados, para obtener un cuadro limpio más preciso en la segunda etapa. En la Figura 3.13 se puede observar los resultados del método desarrollado.

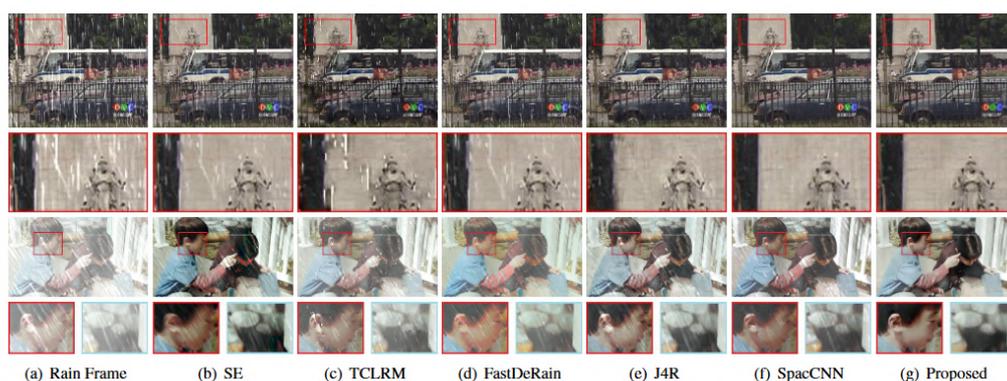


Figura 3.13 Resultados de métodos de eliminación de lluvia en conjuntos de datos sintéticos [45].

### ***Single Image Deraining: A Comprehensive Benchmark Analysis [45]***

El artículo aborda la evaluación de varios algoritmos de eliminación de lluvia en imágenes individuales, que hasta ahora han sido principalmente evaluados utilizando imágenes sintéticas específicas. Este conjunto de datos abarca diversos modelos de lluvia (lluvia

en forma de rayas, gotas de lluvia, lluvia y niebla), además de una amplia variedad de criterios de evaluación (objetivos de referencia completos y no, subjetivos y específicos para tareas). Los resultados obtenidos de esta evaluación y análisis permiten identificar la brecha de rendimiento entre imágenes lluviosas sintéticas e imágenes reales, así como comprender mejor las fortalezas y limitaciones de cada método.

### ***Progressive Image Deraining Networks: A Better and Simpler Baseline [64]***

Este artículo aborda la creciente complejidad y diversidad en las estructuras y aprendizaje de las redes neuronales profundas aplicadas a la eliminación de lluvia en imágenes. Ante esta complejidad, el documento propone un enfoque más simple para servir como línea base en el desarrollo de nuevas redes de eliminación de lluvia. Se introduce una red denominada Progressive ResNet (PRN), que despliega repetidamente una arquitectura ResNet poco profunda para beneficiarse de la computación recursiva. Se añade una capa recurrente para explotar las dependencias entre características profundas en diferentes etapas, formando la Progressive Recurrent Network (PReNet). En la Figura 3.14 se puede observar los resultados del método desarrollado.



Figura 3.14 Comparación de calidad visual de modelos PReNet entrenados con diferentes funciones de pérdida, incluyendo la pérdida única de MSE, la pérdida única de SSIM negativo y la supervisión recursiva de SSIM negativo [64].

### ***Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset [77]***

El artículo aborda la dificultad de eliminar manchas de lluvia de una sola imagen. Propone un método semiautomático para generar imágenes limpias de alta calidad a partir de secuencias de imágenes con lluvia. Este proceso da lugar a un conjunto de datos grande y realista. Además, presenta una nueva red, SPatial Attentive Network (SPANet), que elimina las manchas de lluvia de forma progresiva y global. Los experimentos demuestran que esta red tiene un desempeño superior a otros métodos existentes en la eliminación de manchas de lluvia en imágenes reales.

### ***Semi-supervised Transfer Learning for Image Rain Removal [81]***

El artículo propone un enfoque semi-supervisado para eliminar la lluvia en imágenes individuales. Tradicionalmente, los métodos de aprendizaje profundo necesitaban imágenes con y sin lluvia sintética para entrenar, lo que dificultaba generalizar a diferentes tipos de lluvia real. Esta nueva técnica incorpora imágenes reales de lluvia al entrenamiento de la red sin requerir sus contrapartes limpias. Esto permite que la red se adapte a diversos tipos de lluvia real y supera limitaciones previas. Los experimentos muestran que este método supera a las técnicas existentes.

***Blind Visual Motif Removal from a Single Image [33]***

El artículo propone una técnica basada en el aprendizaje profundo para la eliminación ciega de objetos superpuestos en imágenes, como texto decorativo o símbolos. Estos elementos visuales añadidos a las imágenes a menudo difieren en ubicación, estilo y contenido, y pueden ser desconocidos en su ubicación y geometría exactas. El enfoque desarrollado estima simultáneamente los píxeles que contienen el objeto visual y sintetiza la imagen subyacente latente, logrando resultados avanzados para la eliminación ciega de objetos tanto opacos como semitransparentes, sin requerir la ubicación precisa del motivo visual por parte del usuario.

***Depth-attentional Features for Single-image Rain Removal [37]***

El artículo aborda la falta de consideración de propiedades físicas en métodos de eliminación de lluvia existentes. Se analiza cómo la lluvia y la niebla afectan la visibilidad de objetos en profundidad desde la cámara. Se crea un nuevo conjunto de datos, RainCityscapes, con imágenes reales que contienen lluvia y niebla. Además, se propone una red neuronal que aprende de manera selectiva atendiendo a la profundidad para eliminar la lluvia en imágenes. Experimentos muestran que este método supera a otros en la eliminación de la lluvia.

***Single Image Reflection Removal Exploiting Misaligned Training Data and Network Enhancements [81]***

El artículo se centra en la eliminación de reflejos no deseados de imágenes capturadas a través de una ventana de vidrio, una tarea importante en sistemas de computación visual. Aunque los métodos actuales logran resultados aceptables en ciertas situaciones, su rendimiento decae en casos más generales del mundo real debido a la complejidad inherente del problema. Para abordar estas limitaciones, proponen mejoras en redes neuronales y utilizan datos desalineados. Las mejoras en la red incluyen módulos de codificación de contexto que reducen la incertidumbre en áreas con reflejos fuertes. Además, introducen una función de pérdida invariante a la alineación para aprovechar datos del mundo real desalineados, lo que resulta en una mejora significativa en comparación con los métodos actuales, especialmente al utilizar datos desalineados adicionales.

***PMS-Net: Robust Haze Removal Based on Patch Map for Single Image [14]***

En este artículo se propone un novedoso algoritmo de eliminación de niebla basado en una nueva característica llamada mapa de parches. Los algoritmos convencionales de eliminación de niebla basados en parches (por ejemplo, el prior del canal oscuro) suelen realizar la deshaze con un tamaño de parche fijo. Sin embargo, esto puede llevar a varios problemas como la sobre saturación y la distorsión del color. Por lo tanto, en este trabajo, se diseñó un modelo de selección automática y adaptativa del tamaño de parche llamado Red de Selección de Mapa de Parches (PMS-Net) para seleccionar el tamaño de parche correspondiente a cada píxel. Esta red está diseñada basada en la red neuronal convolucional (CNN), que puede generar el mapa de parches a partir de la imagen de

entrada. Los resultados experimentales en imágenes con niebla tanto sintéticas como del mundo real muestran que, con la combinación del propuesto PMS-Net, el rendimiento en la eliminación de niebla es mucho mejor que el de los algoritmos del estado del arte y podemos abordar los problemas causados por el tamaño de parche fijo.

### 3.3 Discusión del Estado del Arte

Se abordaron diversos temas relevantes sobre los sistemas de asistencia de mantenimiento de carril en los trabajos y artículos previamente mencionados. Entre ellos se incluyen la recopilación de información, los algoritmos utilizados y también se exploraron técnicas de mejora de imagen. La investigación sobre sistemas de asistencia de mantenimiento de carril ha sido extensa y multifacética, abordando una serie de aspectos cruciales. En particular, se han analizado en profundidad tres áreas fundamentales:

- **Recopilación de información:** Los avances en este campo se centran en la captura y procesamiento de datos precisos y diversos para garantizar la fiabilidad del sistema de mantenimiento de carril. Se han explorado diversas fuentes de información, desde cámaras y sensores LiDAR hasta sistemas de mapeo y sistemas de posicionamiento global (GPS). La fusión de múltiples fuentes de datos ha emergido como una estrategia efectiva para proporcionar información completa y actualizada sobre el entorno vial, permitiendo una detección de carril más robusta y precisa.

El uso de sensores de visión permite la adquisición de datos en tiempo real, lo que es esencial para el funcionamiento efectivo de los sistemas de asistencia de mantenimiento de carril. Además, la implementación de algoritmos de visión por computadora en los dispositivos de adquisición de información ha demostrado ser muy efectiva para la detección de carriles.

- **Algoritmos y técnicas de detección de carril:** La literatura revisada destaca el uso de algoritmos avanzados de visión por computadora, como redes neuronales convolucionales (CNN) y modelos basados en redes recurrentes (RNN), para identificar y seguir los carriles en imágenes o secuencias de video. Estos enfoques han evolucionado desde métodos basados en características geométricas hacia modelos más sofisticados que aprovechan la capacidad de aprendizaje profundo para analizar patrones complejos en los carriles y su entorno.

Se observa una prevalencia de la transformada de Hough, debido a su alto desempeño en tiempo real. Se han utilizado combinaciones de técnicas para mejorar la precisión y la robustez del sistema, como el modelo de control predictivo, el filtro Kalman extendido, la selección de una región de interés, la extracción de características y la detección de bordes, entre otros.

Cabe resaltar que cada enfoque de algoritmos tiene sus propias fortalezas y debilidades. La selección del algoritmo adecuado para un sistema de asistencia de mantenimiento de carril dependerá de las necesidades específicas del mismo y de las condiciones del entorno en el que se utilizará. Por lo tanto, es importante realizar una evaluación rigurosa de los diferentes enfoques para seleccionar la mejor opción para un sistema de asistencia de mantenimiento de carril específico.

- **Técnicas de mejora de imagen:** La calidad de las imágenes es esencial para una detección precisa de carriles. Por tanto, se ha investigado ampliamente el desarrollo de técnicas de mejora de imágenes que aborden problemas como el ruido, lluvia, neblina y movimiento. Métodos de procesamiento de imágenes, así como el uso de redes neuronales para restaurar y mejorar la claridad visual, han sido áreas de interés, ya que contribuyen significativamente a mejorar la eficacia de los algoritmos de detección de carriles.

En conjunto, estos avances han sido fundamentales para el progreso de los sistemas de asistencia de mantenimiento de carril, ya que han mejorado la precisión, la robustez y la adaptabilidad de estos sistemas en una amplia gama de condiciones y entornos de conducción.

La información presentada en las investigaciones anteriores resulta altamente relevante para el desarrollo de la tesis en cuestión. Estos estudios proporcionan una base sólida de conocimiento e análisis de sistemas de asistencia de mantenimiento de carril. En la Tabla 3.1 se presenta una comparación detallada de cada trabajo de los artículos, que incluye información relevante sobre los enfoques obtenidos en cada estudio considerando su objetivo, algoritmos o técnicas empleadas y validación. Esta comparación permite identificar las similitudes y diferencias entre los diferentes enfoques utilizados para el desarrollo de sistemas de asistencia de mantenimiento de carril y proporciona información valiosa para futuras investigaciones en este campo. Además, la tablas proporcionan una visión general de utilidad de cada estudio en términos de la precisión y la robustez del sistemas de asistencia de mantenimiento de carril implementado. Esta información es fundamental para la evaluación y mejora continua de los sistemas de asistencia de conducción existentes y para la implementación de nuevos sistemas en el futuro.

Tabla 3.1 Resumen del estado del arte.

Titulo	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle's Forward Monocular Camera</i> [1]	Transformada de Hough.	Estrategia para detectar y mantener el carril, implementando funciones DAS (sistemas de asistencia al conductor).	Los resultados se evalúan cualitativa y cuantitativamente comparando contra una base de datos verídica.	Detección y mantenimiento de carril en carreteras incluyendo escenarios con sombras, curvas y pendientes en la carretera.
<i>RS-Lane: A Robust Lane Detection Method Based on ResNeSt and Self-Attention Distillation for Challenging Traffic Situations</i> [98]	RS-Lane basado en LaneNet, Split Attention, Self-Attention Distillation (SAD), transformada de Hough, RANSAC y YOLO.	Detección de carriles basado en un modelo de aprendizaje profundo que puede detectar carriles sin límites de número.	La validación del algoritmo se realiza sobre datasets, TuSimple y CULane.	Detección de carriles con ausencia de las líneas de carril, oclusión de sombras y condiciones de iluminación complejas.
<i>An Advanced Lane-Keeping Assistance System With Switchable Assistance Modes</i> [6]	Aprendizaje basado en modelo de control predictivo (LBMPC) y marco de trabajo usando filtro de Kalman extendido.	Sistema avanzado de asistencia de mantenimiento de carril (LKAS) con dos modos de asistencia conmutables: el modo de prevención de salida de carril y el modo de copiloto de mantenimiento de carril.	En el futuro este trabajo se centrará en la validación experimental mediante la participación de conductores humanos.	Intercambiar entre los dos modos de asistencia conmutables.
<i>Unsupervised Labeled Lane Markers Using Maps</i> [4]	Segmentación de marcadores binarios, la segmentación de nivel de píxel dependiente del carril y la regresión de borde de carril.	Presentar el conjunto de datos LLAMAS no supervisado, un conjunto de datos (100,042 imágenes) de carriles acotados etiquetados, y utilizarlo para crear un punto de referencia y métodos de referencia para la detección de carriles acotados.	Usa una heurística de imagen para afinarla, para validar la mejor corrección. Comparando etiquetas para cada píxel.	Ajustar con precisión las proyecciones del mapa de carril.
<i>LDNet: End-to-End Lane Marking Detection Approach Using a Dynamic Vision Sensor</i> [50]	Un codificador-decodificador de arquitectura red neuronal convolucional (CNN) para la detección de carriles.	Presentar la detección de carriles con una cámara de eventos.	La validación a través de una extensa experimentación y evaluación para determinar los mejores parámetros para el método propuesto.	Extraer características de mayor dimensión de una imagen, refinando las detecciones de resolución completa.

## Resumen del estado del arte (continuación).

Titulo	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>Lane Detection and Trajectory Tracking Control of Autonomous Vehicle Based on Model Predictive Control</i> [36]	Transformada de Hough, MPC, controlador difuso.	Detección de carril y control de seguimiento de trayectoria de vehículo autónomo basado en control predictivo de modelos.	Para comprobar la eficacia del controlador diseñado se establece un modelo de simulación basado en Matlab/Simulink-Carsim.	Mejorar la adaptabilidad del modelo mediante el uso de linealización aproximada y puede rastrear la trayectoria deseada al minimizar el sesgo entre el sistema de referencia y el sistema actual.
<i>Robust Lane Detection and Tracking for Real-Time Applications</i> [43]	Metodo de agrupamiento jerárquico aglomerado (HAC) con predicción de Kalman.	Detección de carril y seguimiento de carril en tiempo real.	Se comparo el algoritmo propuesto con otros trabajos, creando su propia métrica.	Reduce el nivel de ruido y el tiempo de cálculo y elimina las marcas de carril falsas realizando un seguimiento de las marcas de carril reales utilizando los datos estadísticos acumulados.
<i>Ultra-low Complexity Block-based Lane Detection and Departure Warning System</i> [83]	Region de interés (ROI) parametrizada, extracción de características, clasificación de ángulo de bloque.	Asistente de mantenimiento y detección de carril basado en bloques de complejidad ultrabaja.	No mencionó validación.	Partición de las regiones de interés (ROI) en bloques que no se sobreponen, clasificación de las condiciones de conducción.
<i>Lane Detection Based on Inverse Perspective Transformation and Kalman Filter</i> [89]	ROI, transformación de perspectiva inversa, segmentación, filtro de Kalman.	Detección de carriles basada en transformación de perspectiva inversa.	Se comparan resultados con la base de datos estándar de Caltech.	Es sólido frente a líneas rotas o faltantes, sombras, malas condiciones de iluminación, y puede detectar carriles en diversas condiciones de la carretera.
<i>Dynamic integration and online evaluation of vision-based lane detection algorithms</i> [85]	Detección de bordes, transformada de Hough, RANSAC, filtro de Kalman.	Sistema de detección de carril basado en visión con integración dinámica y evaluación en línea.	Se compara con el conjunto de datos Caltech Road.	Combinación de sistemas de muestreo y votación con conocimiento previo de geometría de carril.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>A 3D Estimation of Structural Road Surface Based on Lane-Line Information</i> [86]	ROI, ajuste de línea, generación de curvas de carril B-spline, transformada de Hough, nube de puntos 3D.	Estimación 3D de la superficie de la carretera estructural basado en la información de la línea del carril de tráfico.	Recopilación de video de la escena de conducción real mediante una cámara frontal.	Modelo B-spline para describir cualquier forma arbitraria, principio de transformación de similitud para estimar la profundidad, nube de puntos 3D.
<i>Implementing and Parallelizing Real-time Lane Detection on Heterogeneous Platforms</i> [78]	ROI, matriz de homografía, transformada de Hough, RANSAC.	Detección de carril basado en la transformación de la imagen de la vista superior.	Se comparó el algoritmo propuesto con otros trabajos, creando su propia métrica.	Detectar múltiples carriles en la imagen BEV transformada de vista superior.
<i>A Robust Method for Lane Detection under Adverse Weather and Illumination Conditions Using Convolutional Neural Network</i> [92]	Red neuronal convolucional (CNN), aprendizaje profundo, filtro gaussiano, región de interés (ROI).	Detección de carriles mediante aprendizaje profundo.	Se compara el método propuesto con otros tres métodos de detección de carril.	Detectar carriles en situaciones meteorológicas adversas y de escasa iluminación.
<i>Learning full-reference quality-guided discriminative gradient cues for lane detection based on neural networks</i> [47]	Red neuronal convolucional (CNN).	Detección de carriles mediante el modelo de aprendizaje profundo.	Se comparó el algoritmo propuesto con otros trabajos.	Utilizar redes neuronales para detectar señales visuales de marcas de carriles.
<i>Highly Curved Lane Detection Algorithms Based on Kalman Filter</i> [18]	Filtro Kalman, transformada de Hough.	Detección de carril en curvas muy pronunciadas.	Se probó con la medición de datos con ruido.	Detección de curvas pronunciadas en un carril de curva, en un entorno muy ruidoso.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>LaneDraw: Cascaded lane and its bifurcation detection with nested fusion</i> [66]	Fusión anidada de incrustación basada en la incrustación de píxeles de segmentación binaria completa con un clúster de autoagrupación (LaneDraw).	Segmentación precisa de los carriles.	LaneDraw se compara con LaneNet y SCNN.	Detección de carriles y sus bifurcaciones.
<i>Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure</i> [27]	Marco de trabajo basado en IoT (IBF).	Detección y seguimiento de carril.	Se comparan los resultados de la simulación con datos de la vida real.	Detectar marcadores de carril bajo diferentes condiciones de luz.
<i>Lane Detection Based on Connection of Various Feature Extraction Methods</i> [44]	ROI, transformada de Hough, detección de bordes Canny, filtro de Kalman.	Preprocesamiento y selección de ROI para detección de carril.	Se comparó el algoritmo propuesto con otros trabajos.	Realización de doble detección de bordes para mejorar la tasa de reconocimiento de carril.
<i>Robust multi-lane detection and tracking using adaptive threshold and lane classification</i> [72]	ROI, RANSAC, filtro Kalman.	Detección y seguimiento de carriles múltiples para varios ambientes en la carretera.	Se comparó el algoritmo propuesto con otros trabajos, usando conjuntos de datos de Caltech Lane.	Extracción de características de carril fuertes y eliminación de características erróneas; solo se verifican los carriles válidos.
<i>Development of an embedded road boundary detection system based on deep learning</i> [60]	CNN, CAE, modelo hiperbólico.	Extracción de límites de carril.	Se validó en diferentes climas, caminos rectos, con curvas, estructurados y no estructurados.	Detectar carreteras estructuradas o no estructuradas en diferentes climas.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>Semantic Segmentation: A Zoology of Deep Architectures</i> [3]	Arquitecturas profundas.	Revisión sobre la evolución de las arquitecturas profundas utilizadas en la segmentación semántica.	Se compara con otras arquitecturas profundas.	Segmentación semántica de carriles.
<i>An Analysis of Multi-stage Progressive Image Restoration Network (MPRNet)</i> [63]	Red neuronal convolucional.	Proporcionar una solución integral para la restauración de imágenes, abordando problemas como el ruido, el desenfoco y la eliminación de efectos de lluvia.	Se prueba y analiza su rendimiento en la restauración de imágenes utilizando datos que no se usaron durante su entrenamiento.	Restauración de imágenes.
<i>LSD: a Line Segment Detector</i> [29]	Basado en el método de Burns, Hanson y Riseman.	Detector de segmentos de línea eficiente y preciso a nivel de subpíxel.	Se basa en un enfoque a contrario, según la teoría desarrollada por Desolneux, Moisan y Morel.	Detectar con precisión segmentos de líneas en imágenes digitales de una manera eficiente y sin la necesidad de ajustes manuales de parámetros.
<i>A Sub-Pixel Edge Detector: an Implementation of the Canny/Devernay Algorithm</i> [30]	Algoritmos clásicos de detección de bordes, como los algoritmos de Canny y Devernay.	Detector de bordes de imagen que genere puntos de borde encadenados con precisión a nivel de subpíxel.	Pruebas comparativas entre los resultados obtenidos con la modificación propuesta y los resultados originales.	Generación de puntos de borde encadenados con precisión a nivel de subpíxel.
<i>Density-aware Single Image De-raining using a Multi-stream Dense Network</i> [96]	Red neuronal convolucional y algoritmo DID-MDN.	Eliminar las franjas de lluvia en imágenes individuales, considerando densidades de lluvia no uniformes.	Comparación de los resultados obtenidos por el método propuesto con otros enfoques existentes, utilizando diferentes conjuntos de datos sintéticos y reales.	Eliminación de franjas de lluvia en imágenes individuales.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>Deep Joint Rain Detection and Removal from a Single Image</i> [88]	Redes neuronales convolucionales y técnicas de aprendizaje profundo.	Eliminar la lluvia de una sola imagen, incluso en condiciones adversas como la presencia de lluvia intensa y la acumulación de franjas de lluvia.	Se evalúa con conjuntos de datos sintéticos y reales para demostrar la efectividad del método en condiciones de lluvia intensa.	Eliminación de la lluvia en imágenes individuales.
<i>Total Variation Deconvolution using Split Bregman</i> [26]	Algoritmo de Bregman dividido.	Restaurar imágenes borrosas causadas por el desenfocado y posiblemente corrompidas por el ruido.	Se valida con métricas como SSIM, SNR y MSE.	Restauración de imágenes afectadas por el desenfocado y el ruido.
<i>Multi-Stage Progressive Image Restoration</i> [95]	Red neuronal.	Restauración de imágenes.	Evaluación con métricas estándar de calidad de imagen como SSIM, MSE y SNR.	Restauración de imágenes degradadas por desenfocado, ruido o lluvia.
<i>Restormer: Efficient Transformer for High-Resolution Image Restoration</i> [94]	Modelo de restauración basado en transformadores (Restormer).	Modelo de restauración de imágenes de alta resolución.	Evaluado en conjuntos de datos de referencia con diferentes configuraciones experimentales.	Eliminación de lluvia, reducción de ruido y corrección de desenfocado en imágenes.
<i>Uformer: A General U-Shaped Transformer for Image Restoration</i> [80]	Bloques Transformer.	Abordar tareas de restauración como eliminación de ruido, desenfocado por movimiento y eliminación de lluvia.	Evaluado en múltiples conjuntos de datos, métricas de calidad de imagen y comparaciones con otros métodos.	Restauración de imágenes con diversas degradaciones visuales.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning</i> [45]	Redes GAN guiadas por la profundidad.	Abordar los desafíos asociados con la eliminación de la lluvia en imágenes, específicamente en escenarios de lluvia intensa.	Se utilizan métricas de calidad de imagen como PSNR y SSIM para evaluar el rendimiento en la restauración de imágenes bajo condiciones de lluvia intensa.	Mejorar la calidad visual de las imágenes afectadas por la lluvia, especialmente en condiciones de lluvia intensa.
<i>Blind Image Deblurring with Local Maximum Gradient Prior</i> [14]	Local Maximum Gradient (LMG).	Restaurar imágenes borrosas de manera ciega, sin conocer el kernel de desenfoque.	Se valida con conjuntos de datos sintéticos y reales.	Restauración de imágenes afectadas por el desenfoque y la borrosidad.
<i>Frame-Consistent Recurrent Video Deraining with Dual-Level Flow</i> [87]	Red recurrente.	Eliminación de la lluvia en videos.	Se realizan experimentos exhaustivos utilizando tanto videos sintéticos como videos reales.	Eliminación de lluvia en videos, especialmente en escenarios de lluvia intensa y condiciones climáticas adversas.
<i>Single Image Deraining: A Comprehensive Benchmark Analysis</i> [45]	Métodos de eliminación de lluvia.	Proporcionar una evaluación exhaustiva y comparativa de múltiples algoritmos de eliminación de lluvia en imágenes individuales.	Evaluación con métricas PSNR y SSIM.	Proporciona una evaluación comparativa de métodos de eliminación de lluvia en imágenes.
<i>Progressive Image Deraining Networks: A Better and Simpler Baseline</i> [64]	Red neuronal Progressive Recurrent Network (PReNet).	Abordar la dificultad derivada de la creciente complejidad de las redes neuronales profundas utilizadas para mejorar el rendimiento en la eliminación de lluvia.	Evaluación con funciones de pérdida MSE y SSIM en imágenes sintéticas y reales.	Elimina eficazmente la lluvia en imágenes sintéticas y reales.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset</i> [77]	Método semi-automático y SPANet, una red de atención espacial.	Resolver el problema de eliminación de gotas de lluvia en imágenes individuales.	Comparaciones con otros métodos del estado del arte.	Eliminación de rachas de lluvia en imágenes reales.
<i>Semi-supervised Transfer Learning for Image Rain Removal</i> [81]	Se emplea un método semi-supervisado.	Eliminar la lluvia de imágenes individuales.	Experimentos con datos sintéticos y reales.	Eliminar la lluvia en imágenes.
<i>Blind Visual Motif Removal from a Single Image</i> [33]	Aprendizaje profundo y aborda la eliminación de objetos visuales superpuestos sin necesidad de conocer su ubicación exacta o geometría.	Eliminación ciega de objetos superpuestos o motivos visuales.	Experimentos exhaustivos.	Capacidad para eliminar objetos visuales superpuestos de imágenes.
<i>Depth-attentional Features for Single-image Rain Removal</i> [37]	Red neuronal.	Abordar las limitaciones de los métodos existentes de eliminación de lluvia al no considerar propiedades físicas como la profundidad de la escena.	Experimentos para comparar visual y cuantitativamente el método propuesto con otros métodos del estado del arte en eliminación de lluvia.	Eliminación de la lluvia en imágenes reales, abordando las limitaciones de los métodos existentes.
<i>Single Image Reflection Removal Exploiting Misaligned Training Data and Network Enhancements</i> [81]	Red neuronal convolucional.	Abordar el desafío inherente de eliminar reflejos no deseados en imágenes tomadas a través de ventanas de vidrio, un problema complejo debido a la naturaleza de una sola imagen y la falta de datos de entrenamiento densamente etiquetados.	Experimentos con datos reales.	Eliminar reflejos no deseados en imágenes capturadas a través de ventanas de vidrio, especialmente en casos del mundo real.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>PMS-Net: Robust Haze Removal Based on Patch Map for Single Image</i> [14]	Patch Map Selection Network (PMS-Net), modelo de selección de tamaño de parche adaptativo basado en CNN.	Eliminar niebla en imágenes.	Experimentos con imágenes sintéticas y reales con niebla.	Mejora de la calidad visual eliminando niebla sin sobresaturación ni distorsión de color.
<i>Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments</i> [11]	Transformación de perspectiva inversa con algoritmo de consenso de muestras aleatorias y modelo de curva B-spline de tercer orden.	Detección de carriles en condiciones viales complejas.	Pruebas con videos de conducción y el conjunto de datos Tusimple.	Alta precisión de detección y buen rendimiento en tiempo real, mejorando la seguridad vial.
<i>Lane Detection of Curving Road for Structural High-way with Straight-curve Model on Vision</i> [76]	Algoritmo basado en modelo recto-curvo para detección de carriles en carreteras curvas.	Identificación precisa de las líneas de carril en carreteras curvas.	Pruebas con imágenes de carreteras reales.	Identificación precisa de las líneas de carril curvas.
<i>Lane detection technique based on perspective transformation and histogram analysis for self-driving cars</i> [51]	Algoritmo que emplea detección de bordes y regresión polinómica.	Detección de carriles para vehículos autónomos.	Resultados de simulación en varios entornos.	Detectar líneas de carril en entornos de conducción.
<i>In Defense of Pre-trained ImageNet Architectures for Real-time Semantic Segmentation of Road-driving Images</i> [52]	Red neuronal para mejorar la segmentación semántica en tiempo real.	Mejorar la segmentación semántica en plataformas móviles.	Experimentos en conjuntos de datos de conducción en carretera.	Segmentación semántica para vehículos en tiempo real.

## Resumen del estado del arte (continuación).

Título	Algoritmos o Técnica	Objetivo	Resultados / Base de datos	Utilidad
<i>FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network</i> [61]	Modelo completamente convolucional.	Mejorar la detección de carriles en vehículos autónomos.	Pruebas en el desafío de marcado de carriles TuSimple CVPR 2017, conjuntos de datos desafiantes como CULane, y un conjunto de datos propio.	Capacidad para detectar carriles en condiciones del mundo real.
<i>DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios</i> [87]	Aprendizaje profundo.	Abordar las limitaciones existentes en la correspondencia estéreo para aplicaciones de conducción autónoma.	Métrica sensible a la distancia y otra sensible a la semántica.	Proporcionar un conjunto de datos estéreo más grande y métricas de evaluación más efectivas.
<i>Learning Lightweight Lane Detection CNNs by Self Attention Distillation</i> [35]	Redes neuronales.	Mejorar la detección de carriles.	Se valida en tres conjuntos de datos de detección de carriles (TuSimple, CULane y BDD100K) con modelos ligeros como ENet, ResNet-18 y ResNet-34.	Mejorar el rendimiento de detección sin aumentar la complejidad del modelo.
<i>3D-LaneNet: End-to-End 3D Multiple Lane Detection</i> [24]	3D-LaneNet.	Predecir la disposición tridimensional de los carriles en una escena vial a partir de una sola imagen.	Se realizó en dos nuevos conjuntos de datos tridimensionales de carriles, uno sintético y otro real, así como en el conjunto de datos TuSimple para detección de carriles.	Predecir la disposición tridimensional de los carriles en entornos viales complejos.
<i>Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks</i> [99]	Redes neuronales convolucionales (CNN) y recurrentes (RNN).	Mejorar la detección de carriles en situaciones desafiantes.	Experimentos exhaustivos en dos conjuntos de datos a gran escala.	Mejorar significativamente la precisión de detección de carriles en condiciones adversas y difíciles en entornos de conducción.

# Capítulo 4

## Metodología de Solución

### 4.1 Propuesta de solución

De acuerdo con el objetivo principal de esta investigación, el método propuesto para el análisis de algoritmos de visión por computadora aplicados a un sistema de asistencia de mantenimiento de carril se estructura en tres módulos: adquisición y preprocesamiento de datos, implementación de los algoritmos de detección de carriles, y evaluación del desempeño. El diagrama correspondiente se muestra en la Figura 4.1.

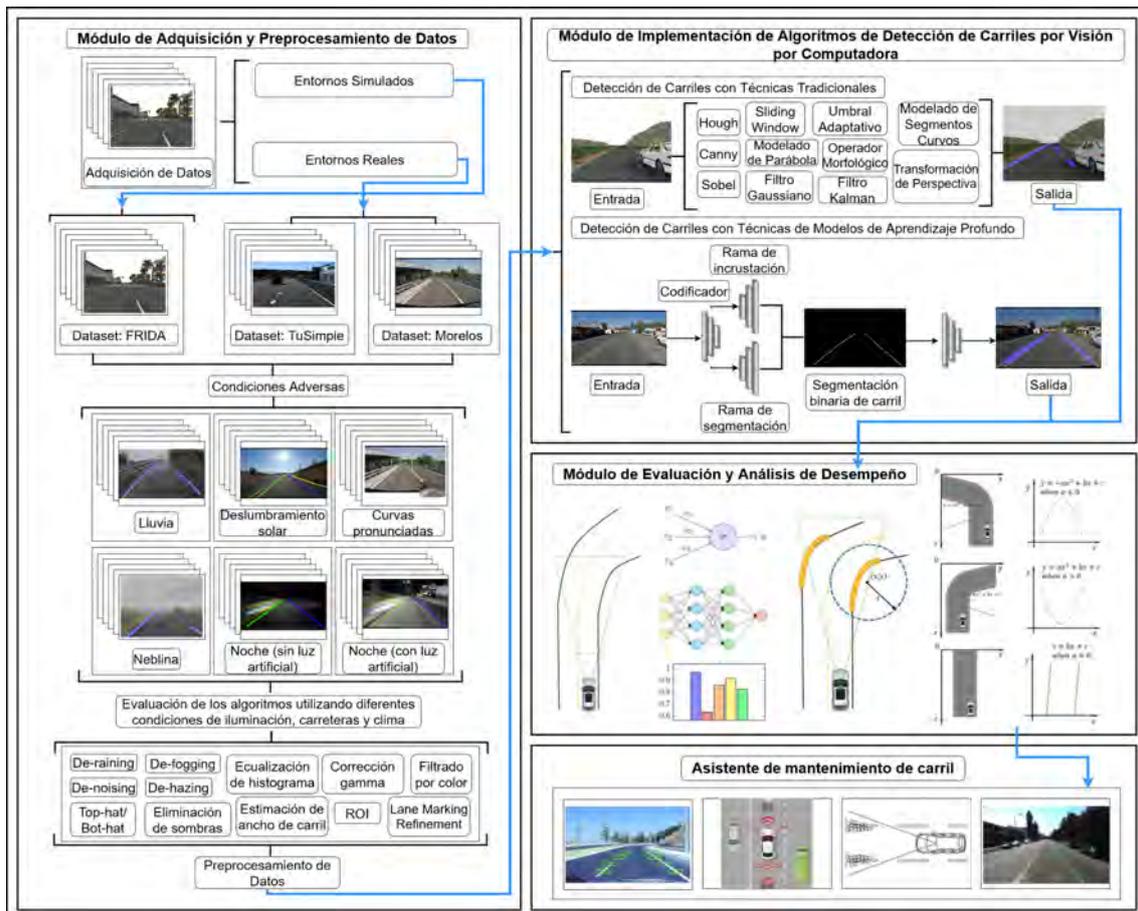


Figura 4.1 Diagrama de metodología de solución propuesta.

A continuación se describe cada módulo de manera detallada, así como su función en la metodología.

### 4.1.1 Adquisición y Preprocesamiento

En primer lugar, el proceso inicia con la adquisición y preprocesamiento de datos visuales, utilizando tanto bases de datos del estado del arte como nuevos datos capturados. Imágenes y videos son recolectados a través de cámaras montadas en vehículos, obteniendo información bajo diversas condiciones de tráfico y clima. Estos datos se someten a un preprocesamiento que incluye reducción de ruido, corrección de distorsiones y filtrado de bordes, optimizando la calidad de las imágenes para una detección más precisa de los carriles, como se ilustra en la Figura 4.2.

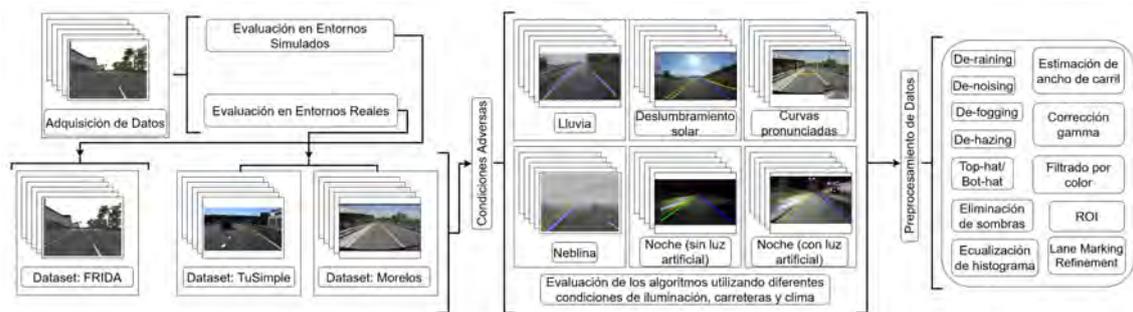


Figura 4.2 Módulo de Adquisición y Preprocesamiento de Datos

#### Bloque 1: Adquisición de Datos

Este primer bloque se enfoca en la recolección de imágenes y videos mediante cámaras montadas en vehículos, cubriendo diversas condiciones de tráfico, clima y entornos viales. Los datos se obtienen tanto de bases de datos preprocesadas como de nuevos escenarios capturados en situaciones adversas, como lluvia, neblina o poca iluminación.

#### Bloque 2: Preprocesamiento de Datos

El preprocesamiento es crucial para optimizar los datos visuales antes de aplicarlos a los algoritmos de detección de carriles. A continuación, se describen las técnicas preprocesamiento de datos utilizadas:

- **De-raining:** Elimina los efectos de gotas de lluvia en las imágenes para mejorar la visibilidad. Matemáticamente, la Imagen de entrada  $I$  puede descomponerse en dos componentes:

$$I = L + R \quad (4.1)$$

donde  $L$  es la imagen limpia y  $R$  representa las gotas de lluvia. [65].

- **De-noising:** Reduce el ruido generado por baja iluminación o interferencias de cámara mediante filtros como el Gaussiano. Este filtro suaviza la imagen convolucionando un kernel Gaussiano  $G(x,y)$  con la imagen original  $I$ :

$$I'(x,y) = \sum_{i=-k}^k \sum_{j=-k}^k G(i,j)I(x-i,y-j) \quad (4.2)$$

El filtro Gaussiano se define como:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4.3)$$

donde  $\sigma$  es la desviación estándar que controla el nivel de suavizado [79].

- **Región de Interés (ROI):** Delimita las zonas de la imagen más propensas a contener carriles, optimizando el procesamiento. Esto se logra mediante una máscara binaria  $\mathbf{M}$  aplicada a la imagen  $\mathbf{I}$ , de la forma:

$$\mathbf{I}_{ROI} = \mathbf{I} \circ \mathbf{M} \quad (4.4)$$

donde  $\circ$  es la multiplicación elemento a elemento.

- **De-fogging:** Mejora la visibilidad en condiciones de niebla utilizando la ecuación de transmisión atmosférica:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (4.5)$$

donde  $I(x)$  es la imagen observada,  $J(x)$  es la imagen libre de niebla,  $t(x)$  es el mapa de transmisión, y  $A$  es la luz ambiental. El mapa de transmisión  $t(x)$  se estima para recuperar la imagen  $J(x)$  [32].

- **De-hazing:** Similar al de-fogging, el de-hazing utiliza el "Dark Channel Prior", que estima las zonas menos afectadas por la neblina basándose en la oscuridad de los canales de color [32].
- **Ecuación de Histograma:** Aumenta la distinción entre los carriles y el fondo de la carretera. El histograma acumulativo  $f(i)$  se usa para ajustar los niveles de intensidad de la imagen:

$$x' = \frac{f(i) - \min(f)}{(M \times N) - \min(f)} \times (L - 1) \quad (4.6)$$

donde  $M \times N$  es el tamaño de la imagen, y  $L$  es el número de niveles de gris [62].

- **Corrección de Gamma:** Ajusta la luminosidad mediante la transformación:

$$I'(x, y) = I(x, y)^\gamma \quad (4.7)$$

donde  $\gamma$  controla el nivel de corrección. Valores  $\gamma < 1$  aclaran la imagen, mientras que  $\gamma > 1$  la oscurecen.

- **Filtrado por Color:** Aísla las líneas de carril según su color mediante umbrales aplicados a los canales  $R$ ,  $G$ , y  $B$  para seleccionar colores típicos de carriles (blanco o amarillo).
- **Eliminación de Sombras:** Identifica y elimina sombras que interfieren en la detección de carriles. Se separan las componentes de iluminación y reflectancia de

la imagen. Esta descomposición se puede expresar mediante la ecuación:

$$I(x,y) = L(x,y)R(x,y) \quad (4.8)$$

donde  $I(x,y)$  es la imagen observada,  $L(x,y)$  es la componente de iluminación y  $R(x,y)$  es la reflectancia (la parte sin sombras). Los métodos para estimar  $L(x,y)$  suelen basarse en la normalización de colores mediante la razón de cromaticidad, de forma que las sombras se detecten como regiones de baja intensidad luminosa:

$$S(x,y) = 1 - \frac{\min(I_R(x,y), I_G(x,y), I_B(x,y))}{I(x,y)} \quad (4.9)$$

donde  $I_R, I_G, I_B$  son las intensidades de los canales de color [68].

- **Estimación del Ancho de Carril:** Verifica la detección de carriles en función de su ancho típico. La estimación del ancho de carril se basa en medir la distancia entre las líneas de carril detectadas en píxeles y convertir esa medida a unidades físicas (como metros) utilizando una transformación de perspectiva. Dado un modelo de cámara y su calibración, la transformación de la imagen a un plano de vista superior se puede describir mediante una matriz de homografía  $\mathbf{H}$ . Si  $(u, v)$  son las coordenadas de los puntos en la imagen y  $(x, y)$  son las coordenadas en el plano real, la relación es:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4.10)$$

Luego, el ancho del carril se estima midiendo la distancia entre las líneas de carril en el plano real [41].

- **Transformaciones Top-hat/Bot-hat:** Resaltan características de brillo o oscuridad mediante las diferencias entre la imagen original y sus operaciones morfológicas. La transformación Top-hat se define como:

$$I_{\text{tophat}} = I - I_o \quad (4.11)$$

donde  $I_o$  es la apertura morfológica. La transformación Bot-hat se usa para resaltar zonas oscuras:

$$I_{\text{bothat}} = I_c - I \quad (4.12)$$

donde  $I_c$  es la clausura morfológica.

- **Refinamiento de Marcas de Carril:** Se ajusta la posición y forma de las líneas de carril detectadas, utilizando algoritmos de refinamiento como el de Canny, que combina gradientes y supresión de no-máximos para detectar bordes [13].

Estas técnicas de preprocesamiento aseguran que las imágenes procesadas estén en condiciones óptimas para una detección precisa y confiable de carriles, independientemente de las condiciones meteorológicas o de iluminación.

### 4.1.2 Implementación de Algoritmos

El segundo módulo se enfoca en la implementación de los algoritmos tradicionales de visión por computadora y de aprendizaje profundo. Se implementan técnicas clásicas que permiten identificar las líneas de los carriles y analizar su geometría en condiciones estándar y complejas. Además, se utilizan redes neuronales convolucionales para mejorar la precisión en la identificación de carriles. Estos modelos se entrenan y prueban en diversos conjuntos de datos, abordando desafíos como condiciones climáticas adversas o carreteras sin marcas visibles, como se ilustra en la Figura 4.3.

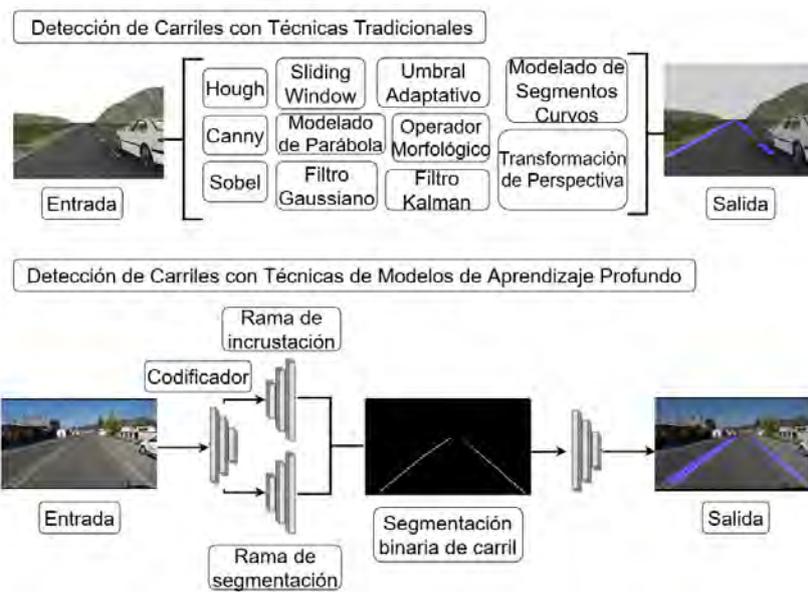


Figura 4.3 Módulo de Implementación de Algoritmos de Detección de Carriles por Visión por Computadora

#### Bloque 3: Detección de Carriles con Algoritmos Clásicos

Este bloque aplica técnicas tradicionales de visión por computadora para la detección de líneas de carriles. Los métodos clásicos, como la Transformada de Hough y el detector de bordes Canny, son eficaces en condiciones ideales, donde las líneas de los carriles son claramente visibles. Aunque estos algoritmos son menos exigentes en términos computacionales, pueden tener limitaciones en escenarios más complejos.

A continuación, se presenta la Tabla 4.1 donde se revisan a detalle los métodos y algoritmos empleados, incluyendo técnicas, mejoras y diferenciadores:

Tabla 4.1 Métodos y algoritmos de técnicas clásicas.

Método/Algoritmo	Técnicas	Mejoramiento	Diferenciador	Referencia
Canny, Hough	Canny + Hough	Mejora en la detección de líneas rectas a partir de bordes.	Utiliza Canny para detección de bordes y Hough para líneas rectas.	[90]
Sobel, Hough	Sobel + Hough	Utiliza el operador Sobel para detectar bordes antes de aplicar Hough.	Sobel resalta bordes basados en gradientes, Hough extrae líneas.	[22]
Sobel, Adaptive	Sobel + Umbral Adaptativo	Mejora la adaptabilidad a diferentes condiciones de iluminación.	Umbral adaptativo ajusta la detección según la iluminación.	[73]
Canny, Morphological	Canny + Operador Morfológico	Mejora la continuidad de los bordes detectados.	Los operadores morfológicos corrigen discontinuidades en los bordes.	[15]
Canny, Hough, Perspective Transformation	Canny + Hough + Transformación de Perspectiva	Mejora la representación visual de los carriles detectados.	La transformación corrige la perspectiva para mejor visualización.	[16]
Sobel, Perspective Transformation, Sliding Window	Sobel + Transformación de Perspectiva + Sliding Window	Permite un análisis más detallado de los carriles en diferentes secciones.	Combinación de diferentes enfoques mejora la detección.	[21]
Canny, Perspective Transformation, Kalman	Canny + Transformación de Perspectiva + Filtro de Kalman	Combina detección de bordes y seguimiento de movimiento.	Filtro de Kalman predice trayectorias después de la detección de bordes.	[59]
Sliding Window, Canny, Hough	Sliding Window + Canny + Hough	Ajusta la detección de carriles en función de la posición.	Análisis dinámico mejora la detección de carriles en movimiento.	[58]
Canny, Hough, Parable	Canny + Hough + Modelado de Parábola	Mejora la detección de carriles curvos.	Modelado de parábola ajusta líneas a curvas suaves.	[7]
Gaussian, Sobel, Curved	Filtro Gaussiano + Sobel + Modelado de Segmentos Curvos	Permite la detección precisa de carriles curvos a partir de bordes suavizados.	Suavizado previo mejora la detección de bordes curvos.	[20]

Las técnicas tradicionales para la detección de carriles incluyen:

- **Transformada de Hough:** La transformada de Hough es una técnica utilizada para detectar líneas rectas en una imagen. Las líneas en el espacio de la imagen se representan mediante la ecuación paramétrica:

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (4.13)$$

donde  $r$  es la distancia perpendicular de la línea al origen y  $\theta$  es el ángulo de la normal a la línea. Se realiza una transformación al espacio de parámetros  $(r, \theta)$ , donde las intersecciones indican la presencia de una línea [19].

La idea detrás de la transformada de Hough es convertir la tarea de detectar líneas en una imagen en un problema de detección de picos en un espacio de parámetros. Esto se logra mediante la parametrización de las líneas rectas utilizando la siguiente ecuación:

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

Donde: -  $r$  es la distancia perpendicular desde el origen hasta la línea. -  $\theta$  es el ángulo de inclinación de la perpendicular con respecto al eje horizontal. -  $(x, y)$  son las coordenadas de un punto en la imagen que pertenece a la línea.

En lugar de trabajar directamente con la ecuación en su forma  $y = mx + b$  (pendiente y ordenada al origen), que puede volverse problemática para líneas verticales (pendiente infinita), se utiliza esta forma paramétrica que describe cualquier línea mediante los parámetros  $r$  y  $\theta$ .

- **Detector de Bordes Canny:** El detector de bordes Canny se basa en detectar cambios abruptos en la intensidad de los píxeles. El gradiente de la imagen se calcula como:

$$G = \sqrt{G_x^2 + G_y^2} \quad (4.14)$$

donde  $G_x$  y  $G_y$  son las derivadas en las direcciones  $x$  e  $y$ . El algoritmo incluye pasos de suavizado, cálculo de gradientes y supresión de no-máximos [10].

- **Sobel-Laplacian:** El filtro Sobel utiliza máscaras convolucionales para detectar bordes mediante la derivada de primer orden. Para la dirección horizontal:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad (4.15)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.16)$$

La laplaciana, como derivada de segundo orden, se expresa como:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.17)$$

Esta técnica detecta áreas de cambio abrupto en la imagen [71].

- **Sliding Window:** El método de la ventana deslizante (*sliding window*) utiliza una ventana rectangular que se mueve sobre la imagen para buscar características dentro de la región. En la detección de carriles, la ventana busca píxeles pertenecientes a las líneas del carril y ajusta la posición de la ventana conforme avanza en la imagen [38].
- **Modelado de Parábola:** Para modelar carriles curvos, se ajusta una parábola a las líneas del carril usando la ecuación:

$$y = ax^2 + bx + c \quad (4.18)$$

donde los parámetros  $a$ ,  $b$  y  $c$  se ajustan para que la parábola represente la curva del carril [97].

- **Filtro Gaussiano:** El filtro gaussiano se utiliza para suavizar una imagen, reduciendo el ruido mediante la convolución de la imagen con una función gaussiana bidimensional:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.19)$$

donde  $\sigma$  controla el nivel de suavizado [28].

- **Umbral Adaptativo:** El umbral adaptativo ajusta el valor del umbral localmente para cada subregión de la imagen. El umbral en una vecindad de píxeles se calcula como:

$$T(x,y) = \frac{1}{N} \sum_{i,j} I(i,j) \quad (4.20)$$

donde  $N$  es el número de píxeles en la vecindad y  $I(i,j)$  es la intensidad del píxel [8].

- **Operador Morfológico:** Los operadores morfológicos como la dilatación y erosión modifican la estructura de las imágenes binarias. La dilatación de una imagen  $I$  con un elemento estructurante  $S$  se define como:

$$I \oplus S = \{z \mid (S)_z \cap I \neq \emptyset\} \quad (4.21)$$

Estos operadores son útiles para eliminar ruido o cerrar pequeños huecos en las líneas detectadas [69].

- **Filtro Kalman:** El filtro de Kalman es un método recursivo para estimar el estado de un sistema. Las ecuaciones del filtro son:

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (4.22)$$

$$z_k = Hx_k + v_k \quad (4.23)$$

donde  $A$  es la matriz de transición de estado,  $H$  es la matriz de observación, y  $w_k$  y  $v_k$  son ruidos de proceso y observación, respectivamente [40].

- **Modelado de Segmentos Curvos:** Se utilizan curvas paramétricas, como las curvas de Bézier o splines cúbicos, para modelar segmentos curvos de carriles. Estas curvas ajustan mejor la forma de carriles complejos o cambiantes [17].
- **Transformación de Perspectiva:** La transformación de perspectiva proyecta una imagen en una vista cenital (bird's-eye view), útil para alinear las líneas de carriles rectamente. Esto se realiza mediante la transformación de homografía, que mapea puntos  $(x,y)$  a nuevos puntos  $(x',y')$  usando una matriz de homografía  $H$ :

$$p' = H \cdot p \quad (4.24)$$

Esta técnica es ampliamente utilizada en aplicaciones de visión por computadora para geometría múltiple de vistas [31].

#### **Bloque 4: Detección de Carriles con Redes Neuronales Convolucionales (CNN)**

Este bloque implementa CNN para mejorar la precisión y robustez de la detección de carriles en condiciones complejas. Las CNN son entrenadas con bases de datos especializadas que les permiten aprender características específicas de los carriles, incluso en situaciones donde las líneas son poco visibles o están deterioradas. El proceso de entrenamiento incluye optimización de los parámetros de la red mediante funciones de pérdida que guían su mejora durante el proceso.

#### **Entrenamiento de redes neuronales**

Se entrena una CNN utilizando conjuntos de datos con imágenes etiquetadas de carriles, como Culane y TuSimple, aprendiendo a identificar patrones complejos.

Culane es un conjunto de datos diseñado para la detección de carriles en entornos urbanos y carreteras, proporcionando miles de imágenes y anotaciones de carriles en diversas condiciones. Por otro lado, TuSimple se centra en la detección de carriles en autopistas, capturando una amplia gama de escenarios de conducción. Ambas bases de datos son fundamentales en la investigación de detección de carriles, ya que permiten mejorar la precisión de los modelos y establecer comparaciones significativas entre diferentes enfoques de aprendizaje automático.

#### **Implementación de modelos**

Se utilizan redes neuronales como VGG16 y SCNN (Spatial CNN), que han demostrado ser efectivas en condiciones adversas, como carreteras sin marcas claras o malas condiciones climáticas [70].

#### **Modelo VGG16**

El modelo VGG16 es una de las arquitecturas de redes neuronales convolucionales (CNN) más conocidas, propuesta por Simonyan y Zisserman en 2014 [70]. La red VGG16 consta de 16 capas, incluyendo 13 capas convolucionales y 3 capas completamente conectadas. Se destaca por su simplicidad y gran capacidad para aprender características visuales, siendo muy eficiente en tareas de clasificación y segmentación de imágenes, como se ilustra en la Figura 4.4.

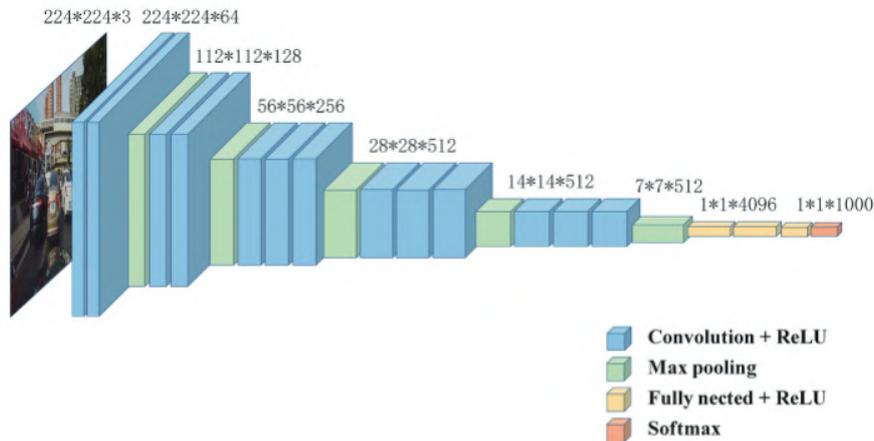


Figura 4.4 Arquitectura del modelo de la red VGG-16 [49].

### Arquitectura de VGG16

La arquitectura VGG16 incluye:

- Capas convolucionales: Se utilizan filtros de tamaño  $3 \times 3$  y se aplican funciones de activación ReLU.
- Capas de pooling: Se aplican capas de max pooling ( $2 \times 2$ ) para reducir la dimensionalidad.
- Capas completamente conectadas: Se utilizan para realizar la clasificación final, adaptadas para la segmentación de carriles.

Para detectar carriles, VGG16 se adapta eliminando las últimas capas de clasificación y utilizando una técnica de segmentación semántica que predice la categoría (carril o no carril) para cada píxel de la imagen.

### Modelo SCNN (Spatial CNN)

El modelo SCNN, propuesto por Pan et al. en 2018, es una arquitectura diseñada específicamente para la detección de carriles [57]. Se diferencia de las CNN tradicionales en su capacidad para manejar la estructura espacial de las líneas de carril de forma más eficiente, como se ilustra en la Figura 4.5.

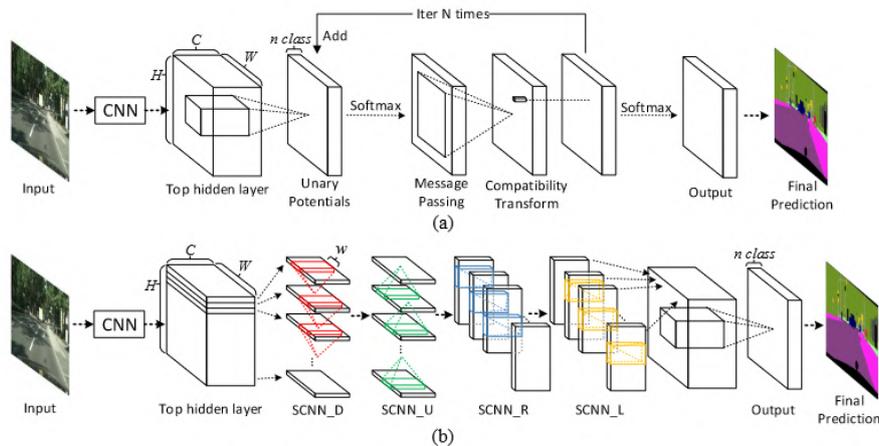


Figura 4.5 Arquitectura del modelo de la red SCNN [55].

### Arquitectura de SCNN

SCNN incluye:

- Capas convolucionales: Comienza con capas convolucionales regulares para extraer características espaciales locales.
- Convoluciones espaciales: Implementa convoluciones que propagan información a lo largo de las dimensiones, mejorando la detección de carriles curvos y discontinuos.
- Segmentación de carriles: Asigna etiquetas de clase (carril/no carril) a cada píxel, permitiendo detectar patrones en condiciones difíciles.

La propagación espacial en SCNN permite reconocer patrones de carriles, incluso cuando están parcialmente ocultos o desgastados, lo que la hace muy efectiva en entornos urbanos o de autopistas con alto tráfico.

#### 4.1.3 Evaluación y Análisis

Posteriormente, el tercer módulo se enfoca en la evaluación y análisis de los algoritmos en entornos reales y simulados en carreteras sintéticas simuladas, bases de datos del estado del arte y un dataset propio de carreteras del estado de Morelos. Se utilizan métricas como sensibilidad, especificidad, IoU, precisión y recall para identificar qué métodos ofrecen el mejor balance entre rendimiento y eficiencia en escenarios de conducción real.

##### **Bloque 5: Evaluación en Entornos Simulados**

Este bloque implica la evaluación inicial de los algoritmos en simulaciones controladas. Estas simulaciones permiten recrear condiciones de conducción específicas, facilitando la medición del rendimiento en un entorno controlado antes de realizar pruebas reales, se ocuparon las bases de datos de imágenes FRIDA y FRIDA 2, diseñadas específicamente para este propósito. La base de datos FRIDA contiene 90 imágenes sintéticas organizadas en 18 escenas de vías urbanas, mientras que FRIDA 2 incluye 330 imágenes sintéticas

que abarcan 66 escenas viales variadas para detección de carriles con condiciones de iluminación y neblina uniforme, heterogénea, y niebla nublada.

#### **Bloque 6: Evaluación en Entornos Reales**

Este bloque es crucial para validar los algoritmos utilizando datos del mundo real. Se realizan pruebas con videos grabados en carreteras, incluyendo el dataset propio del estado de Morelos este conjunto está compuesto por 22,566 imágenes y con sus 41 clips de video, que abarca una variedad de escenas viales y condiciones climáticas adversas como curvas, curvas muy pronunciadas, presencia de lluvia, pocas condiciones de iluminación o bien la oscuridad y presencia de sombras en la carretera se incorporó la base de datos de TuSimple, reconocida como uno de los conjuntos de datos más utilizados para la detección de carriles este conjunto está compuesto por 55,640 imágenes de 2,782 videoclips y cada clip tiene 20 imágenes Base de datos de escenas viales variadas para detección de carriles, abarcando curvas pronunciadas, lluvia, baja iluminación, oscuridad, y sombras en la carretera.

#### **Bloque 7: Métricas de Evaluación**

Se emplearon métricas cuantitativas clave como precisión, *recall*, F1 Score, *accuracy*, IoU, *specificity* y Dice Index [74], que permiten medir de manera precisa la eficacia de los algoritmos bajo distintas condiciones y entornos.

#### **Bloque 8: Análisis Comparativo de Resultados**

En este bloque se realiza un análisis comparativo de los resultados de diferentes algoritmos, tanto tradicionales como basados en redes neuronales.

# Capítulo 5

## Experimentación y Resultados

### 5.1 Diseño Experimental

En este capítulo se describen los experimentos realizados para evaluar el método propuesto, incluyendo las métricas utilizadas y los resultados obtenidos en cada uno de ellos.

### 5.2 Entorno de desarrollo

Para la ejecución de los experimentos, que involucraron tanto imágenes simuladas como de entornos reales, fue fundamental disponer de un equipo con las capacidades adecuadas para procesar la carga de trabajo, además del software especializado y sus librerías correspondientes.

#### Arquitectura de Hardware

- **Procesador:** 13th Gen Intel(R) Core(TM) i7-13650HX 2.60 GHz
- **Memoria RAM:** 16 GB DDR5
- **GPU:** RTX 4060 con 16 GB de VRAM
- **Almacenamiento:** SSD de 1 TB

#### Plataforma de Software

- **Sistema operativo:** Windows 11 Home
- **Sistema operativo:** Ubuntu 22.04 LTS
- **Entornos de desarrollo:** MATLAB, Python

### 5.3 Base de Datos

Para la fase experimental, se emplearon tres bases de datos. Se evaluaron los modelos de detección de carriles utilizando las métricas previamente mencionadas en las bases de datos de imágenes Frida y Frida 2, diseñadas específicamente para este propósito. La base de datos Frida contiene 90 imágenes sintéticas organizadas en 18 escenas de vías

urbanas, mientras que Frida 2 incluye 330 imágenes sintéticas que abarcan 66 escenas viales variadas.

Además, se incorporó la base de datos de TuSimple, reconocida como uno de los conjuntos de datos más utilizados para la detección de carriles, junto con otros como CULane [56] y KITTI [25] Este conjunto está compuesto por 55,640 imágenes de 2,782 videoclips y cada clip tiene 20 imágenes., también se utilizó mi propia base de datos de carreteras del estado de Morelos. Este conjunto está compuesto por 22,566 imágenes y con sus 41 clips de video, que abarca una variedad de escenas viales y condiciones climáticas adversas como curvas, curvas muy pronunciadas, presencia de lluvia, pocas condiciones de iluminación o bien la oscuridad y presencia de sombras en la carretera. En la Tabla 5.1 se presenta un resumen de las bases de datos utilizadas para la detección de carriles.

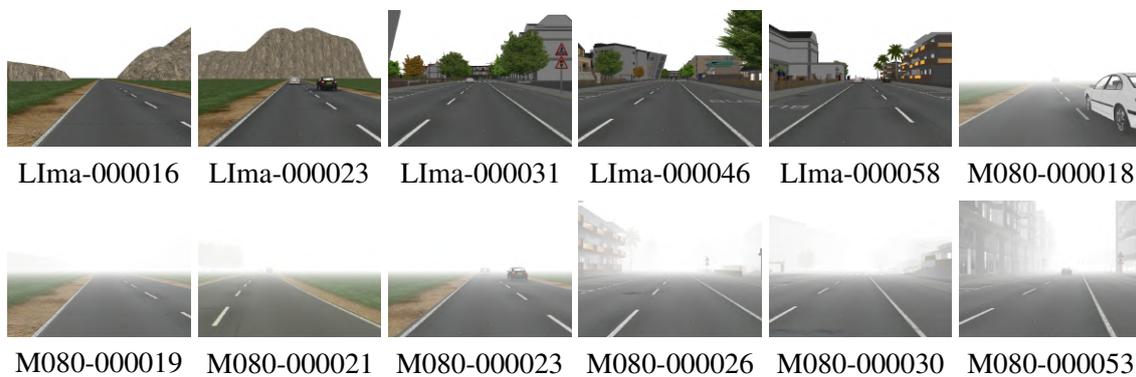


Figura 5.1 Imágenes aleatorias de la base de datos Frida y Frida 2 con variedad de escenas viales y condiciones climáticas.



Figura 5.2 Imágenes aleatorias de la base de datos TuSimple con variedad de escenas viales.



Figura 5.3 Imágenes aleatorias en base de datos propia con variedad de escenas viales y condiciones climáticas.

Tabla 5.1 Resumen de bases de datos utilizadas para la detección de carriles.

Base de datos	Imágenes	Clips de video	Descripción
 Frida y Frida 2	420	N/A	Base de datos de imágenes simuladas con un conjunto de escenas viales variadas para detección de carriles con condiciones de iluminación y neblina uniforme, heterogénea, y niebla nublada.
 TuSimple Dataset	55,640	2,782	Conjunto de datos con variedad de escenas viales para detección de carriles y condiciones climáticas adversas como curvas pronunciadas y presencia de sombras en la carretera.
 Propia	22,566	41	Base de datos propia del estado de Morelos, México de escenas viales variadas para detección de carriles, abarcando curvas pronunciadas, lluvia, baja iluminación, oscuridad, y sombras en la carretera.

## 5.4 Métricas

En esta sección se presentan las métricas y resultados utilizados para evaluar los algoritmos implementados en la detección de carriles. Se emplearon métricas cuantitativas clave como precisión, *recall*, F1 Score, *accuracy*, IoU, *specificity* y Dice Index, que permiten medir de manera precisa la eficacia de los algoritmos bajo distintas condiciones y entornos.

### Precisión

La precisión se define como:

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (5.1)$$

donde  $TP$  son los verdaderos positivos y  $FP$  son los falsos positivos.

## Recall

El *recall* se expresa como:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.2)$$

donde *FN* son los falsos negativos.

## F1 Score

El F1 Score es la media armónica de la precisión y el *recall*:

$$F1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}} \quad (5.3)$$

## Accuracy

La *accuracy* se define como:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.4)$$

donde *TN* son los verdaderos negativos.

## IoU

La Intersección sobre Unión (IoU) se calcula como:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (5.5)$$

donde *A* es el área predicha y *B* es el área real.

## Specificity

La *specificity* se define como:

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5.6)$$

donde *TN* son los verdaderos negativos y *FP* son los falsos positivos.

## Dice Index

El *Dice Index* se calcula como:

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} \quad (5.7)$$

donde *A* es el área predicha y *B* es el área real.

## 5.5 Experimentos

Con el objetivo de evaluar la efectividad de los algoritmos de visión por computadora en la detección de carriles, se llevaron a cabo diversas experimentaciones. Estas pruebas se realizaron utilizando diferentes conjuntos de datos y técnicas, con el fin de comprender mejor el rendimiento de los modelos en distintas condiciones. Los objetivos de la serie de experimentaciones se detallan en los siguientes puntos:

- Experimentación para la detección de carriles con algoritmos de visión por computadora usando técnicas tradicionales en carriles rectos de la base de datos de Frida y Frida 2.
- Experimentación para la detección de carriles con algoritmos de visión por computadora usando técnicas tradicionales en carriles rectos de la base de datos de TuSimple.
- Experimentación para la detección de carriles con algoritmos de visión por computadora usando técnicas tradicionales en carriles rectos de mi propia base de datos.
- Experimentación para la detección de carriles con algoritmos de visión por computadora usando una extensión de las técnicas tradicionales de carriles rectos en carriles curvos con múltiples bases de datos donde se perciban curvas.
- Experimentación para la detección de carriles con algoritmos de visión por computadora usando técnicas de modelos de aprendizaje profundo.

### 5.5.1 Técnicas tradicionales en carriles rectos de la base de datos de Frida y Frida 2

En este experimento, se llevó a cabo una evaluación de los algoritmos de detección de carriles utilizando técnicas tradicionales de visión por computadora en condiciones adversas, estas condiciones incluyeron la presencia de sol directo, reflejos solares (sun glare), curvas normales y diversas etapas de neblina: niebla uniforme, niebla heterogénea, niebla nublada y niebla heterogénea nublada. Para ello, se seleccionaron imágenes de las bases de datos Frida y Frida 2, que ofrecen una variedad de escenarios de conducción, incluyendo aquellos que simulan condiciones de visibilidad reducida. En la Figura 5.5 se pueden apreciar los resultados obtenidos por los diferentes detectores de líneas rectas. En la Table 5.2 se presentan los elementos de la matriz de confusión, que permiten un análisis detallado de las métricas de rendimiento del modelo. Para evaluar el desempeño, se emplearon métricas cuantitativas clave como *precisión*, *recall*, *F1 Score*, *accuracy*, *IoU*, *specificity* y *Dice Index*. Estas se encuentran en la Tabla 5.3 además la Figura 5.4 ilustra gráficamente los resultados obtenidos en relación con estas métricas en la base de datos Frida, la leyenda SA se refiere a Sobel + Adaptativo, CM se refiere a Canny + Morphological, SCH se refiere a Sliding + Canny + Hough, CH se refiere a Canny + Hough y A se refiere a el promedio.

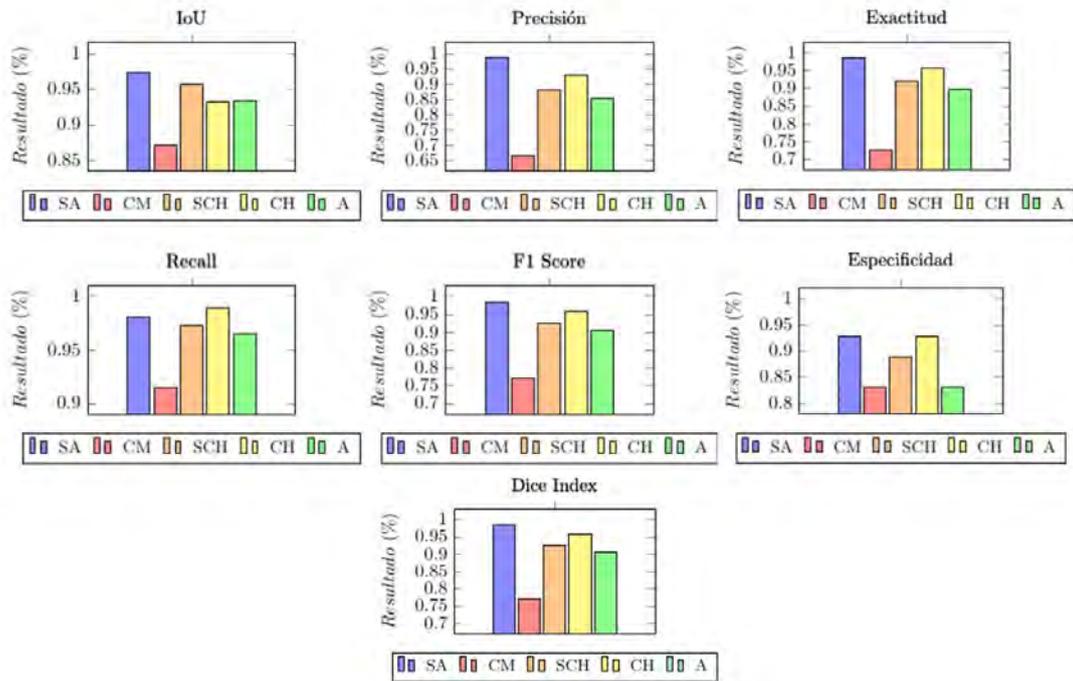


Figura 5.4 Resultados cuantitativos de la detección en la base de datos Frida.

El algoritmo Sobel + Adaptive Thresholding mostró un rendimiento destacado, con una alta tasa de verdaderos positivos (TP) y bajos valores en falsos positivos (FP) y falsos negativos (FN). Esto indica que es muy eficaz para detectar carriles en condiciones de neblina. El rendimiento del algoritmo Canny + Morphological fue menos eficiente, presentando una tasa de falsos positivos considerablemente alta, lo que sugiere que, aunque identifica correctamente la mayoría de los carriles, también genera un número significativo de detecciones incorrectas. El Sliding Canny + Hough obtuvo un rendimiento sólido, con un buen equilibrio entre los verdaderos positivos y los negativos. Su capacidad para minimizar los falsos negativos es notable, lo que lo convierte en una opción confiable para la detección de carriles. El algoritmo Canny + Hough presentó el mejor desempeño entre todos, con una alta tasa de verdaderos positivos y muy pocos falsos negativos, lo que demuestra su efectividad en la detección de carriles, incluso en condiciones adversas. El promedio de los algoritmos muestra un buen desempeño general, con un TP de 0.96395 y un FP de 0.16795, lo que indica que, en general, los algoritmos probados tienen una buena capacidad para detectar carriles bajo las condiciones de neblina.

Tabla 5.2 Elementos de la matriz de confusión para la Figura 5.5.

Algoritmo	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97995	0.01105	0.01905	0.98795
Canny + Morphological	0.91605	0.45905	0.08495	0.54195
Sliding + Canny + Hough	0.97195	0.13205	0.02695	0.86895
Canny + Hough	0.98895	0.07295	0.01105	0.92795

Tabla 5.3 Comparación de algoritmos basada en diversas métricas.

Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
SA	0.97295	0.98885	0.98093	0.98487	0.98493	0.98894	0.98487
CM	0.87195	0.66617	0.91513	0.77105	0.72827	0.54141	0.77105
SCH	0.95705	0.88039	0.97302	0.92439	0.92050	0.86808	0.92439
CH	0.93195	0.93130	0.98895	0.95926	0.95802	0.92712	0.95926

El algoritmo Sobel + Adaptive Thresholding se destaca en todas las métricas, mostrando un rendimiento excepcional. Su alta precisión y Recall indican que es capaz de detectar carriles con gran exactitud y con muy pocos falsos negativos, lo que se traduce en un F1 Score igualmente alto. Este algoritmo muestra un desempeño notablemente inferior, especialmente en precisión y especificidad, lo que sugiere que, aunque identifica correctamente muchos carriles, también produce un número elevado de falsos positivos, reduciendo su efectividad general. Este algoritmo presenta un buen equilibrio entre precisión y recall, lo que resulta en un rendimiento general sólido. Su capacidad para detectar carriles de manera efectiva lo convierte en una alternativa viable. El Canny + Hough demuestra ser uno de los mejores algoritmos, con una alta tasa de recall y buenos resultados en precisión. Esto indica que es capaz de detectar carriles con eficacia, incluso en condiciones adversas. Las métricas promedio indican que, en general, los algoritmos tienen un desempeño aceptable, con un F1 Score de 0.90478 y un IoU de 0.93295. Esto sugiere que, en conjunto, los métodos evaluados pueden ser útiles para la detección de carriles, aunque hay oportunidades para mejorar la precisión de algunos algoritmos.

### Detector de línea recta

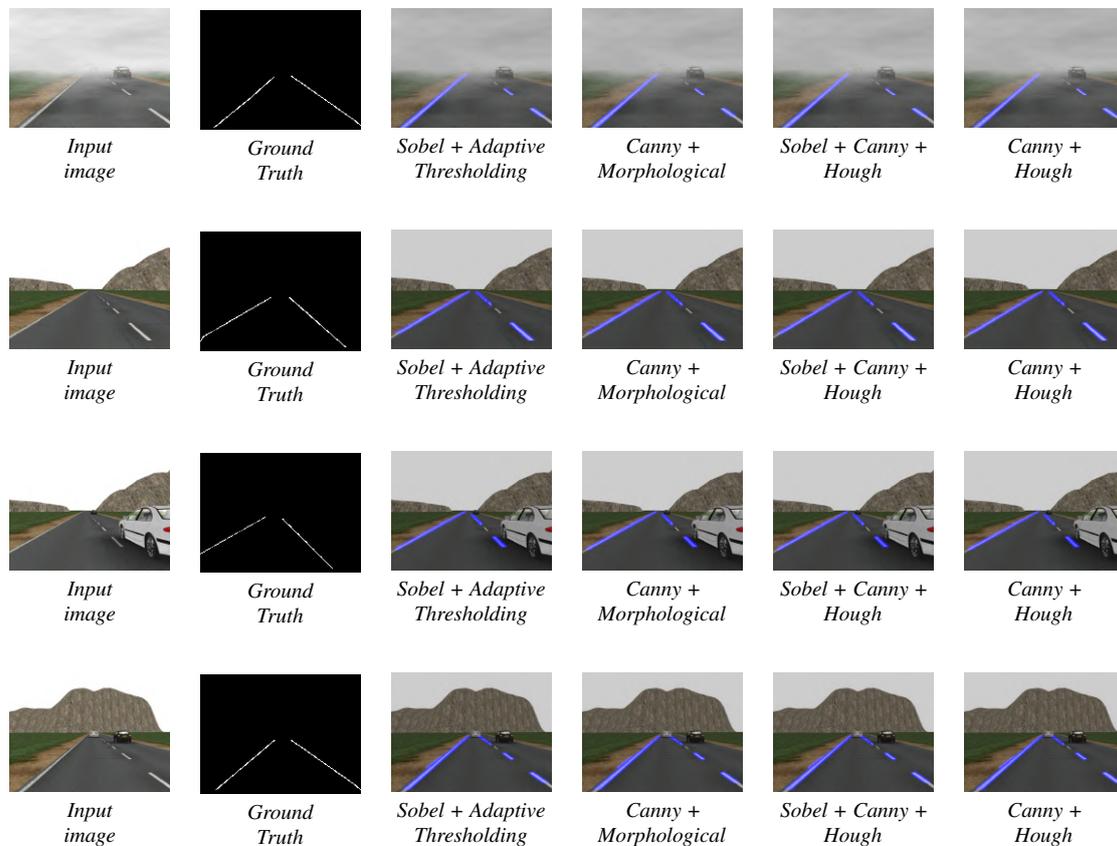


Figura 5.5 Resultados cualitativos en base de datos Frida.

### 5.5.2 Técnicas tradicionales en carriles rectos de la base de datos de TuSimple

En este experimento, se abordó la detección de carriles utilizando algoritmos de visión por computadora aplicados a la base de datos TuSimple. Esta base de datos, conocida por su amplia variedad de condiciones de conducción y configuraciones de carretera, es ampliamente utilizada en el estado del arte para evaluar métodos de detección de carriles. Permite una evaluación robusta de los algoritmos en un contexto realista. En particular, se consideraron diversas condiciones adversas, incluyendo sol directo y reflejos solares (sun glare), así como variaciones en la condición de la carretera, como sombras. Además, se evaluaron diferentes tipos de curvas en la carretera, que variaron desde leves hasta moderadas y muy cerradas. Estas consideraciones son esenciales para analizar el rendimiento de los algoritmos en situaciones desafiantes y reales. En la Figura 5.7 se pueden apreciar los resultados obtenidos por los diferentes detectores de líneas rectas. En la Tabla 5.4 se presentan los elementos de la matriz de confusión, que permiten un análisis detallado de las métricas de rendimiento del modelo. Para evaluar el desempeño, se emplearon métricas cuantitativas clave como *precisión*, *recall*, *F1 Score*, *accuracy*, *IoU*, *specificity* y *Dice Index*. Estas se encuentran en la Tabla 5.5 además la Figura 5.6 ilustra gráficamente los resultados obtenidos en relación con estas métricas en la base de datos

TuSimple, la leyenda SA se refiere a Sobel + Adaptive Thresholding, CM se refiere a Canny + Morphological, SCH se refiere a Sliding + Canny + Hough, CH se refiere a Canny + Hough y A se refiere a el promedio.

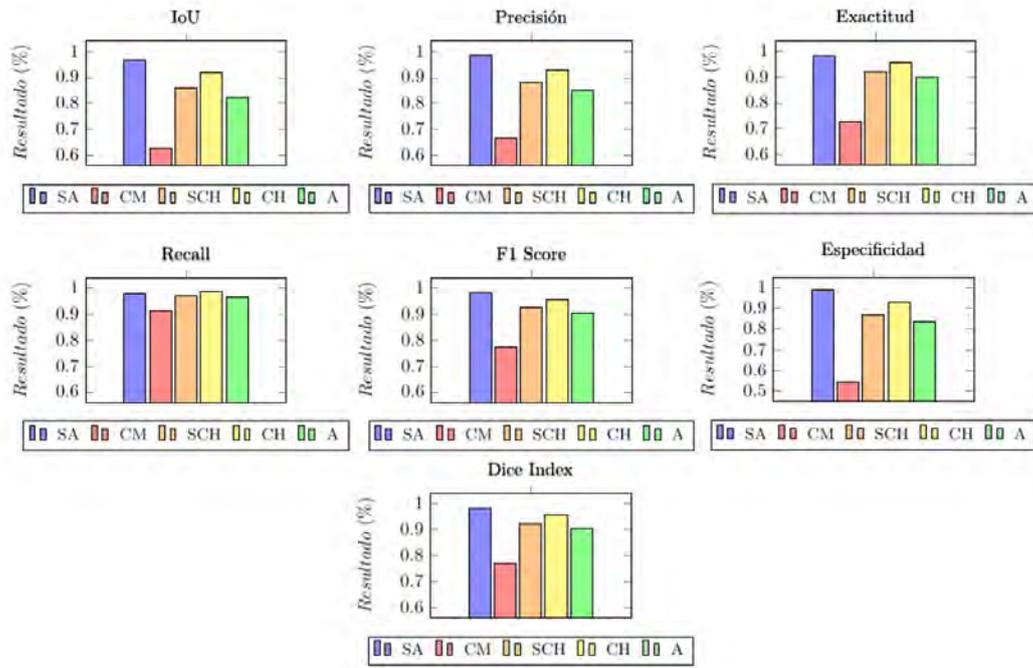


Figura 5.6 Resultados cuantitativos de la detección en la base de datos TuSimple.

El algoritmo Sobel + Adaptive Thresholding presenta un alto porcentaje de verdaderos positivos y un bajo porcentaje de falsos positivos. Esto indica que es altamente efectivo en la detección de carriles, con un rendimiento similar al obtenido en el primer experimento, lo que reafirma su consistencia. A diferencia de otros algoritmos, el Canny + Morphological muestra una alta tasa de falsos positivos, lo que sugiere que, aunque logra detectar una cantidad considerable de carriles, también confunde el fondo y los objetos cercanos como carriles, disminuyendo así su efectividad general. El Sliding + Canny + Hough mantiene un buen equilibrio entre verdaderos positivos y falsos positivos. Su rendimiento es comparable al del Sobel + Adaptive Thresholding, lo que indica que puede ser una opción viable para la detección precisa de carriles. El Canny + Hough se destaca en este experimento con un alto número de verdaderos positivos y un bajo número de falsos negativos. Esto indica que es el algoritmo más efectivo en esta serie de pruebas, mostrando una gran capacidad para detectar carriles correctamente. El análisis promedio de las métricas revela un buen desempeño general, con un 0.96589 de verdaderos positivos y 0.83222 de verdaderos negativos, lo que sugiere que los algoritmos evaluados son efectivos para la detección de carriles en la base de datos TuSimple, aunque hay margen de mejora en términos de falsos positivos.

Tabla 5.4 Elementos de la matriz de confusión para la Figura 5.7.

Algoritmo	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97734	0.01322	0.02156	0.98567
Canny + Morphological	0.91823	0.45734	0.08745	0.54456
Sliding + Canny + Hough	0.97344	0.13122	0.02834	0.87145
Canny + Hough	0.99034	0.07511	0.01089	0.93012

Tabla 5.5 Comparación de algoritmos basada en diversas métricas.

Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
SA	0.96564	0.98665	0.97842	0.98252	0.98259	0.98677	0.98252
CM	0.62763	0.66753	0.91304	0.77122	0.72863	0.54353	0.77122
SCH	0.85917	0.88121	0.97171	0.92425	0.92040	0.86913	0.92425
CH	0.92010	0.92950	0.98912	0.95839	0.95714	0.92528	0.95839

Los resultados revelan que el algoritmo Canny Hough obtuvo un alto desempeño en términos de precisión, recall, y F1 Score, sugiriendo su eficacia en la detección de carriles en esta base de datos. Por otro lado, el Canny Morfológico mostró un rendimiento inferior, indicando que las técnicas empleadas no son tan efectivas en este contexto específico. En general, estos resultados proporcionan información valiosa para la selección de algoritmos en aplicaciones de detección de carriles, así como para futuras mejoras en el desarrollo de sistemas de asistencia al conductor.

### Detector de línea recta

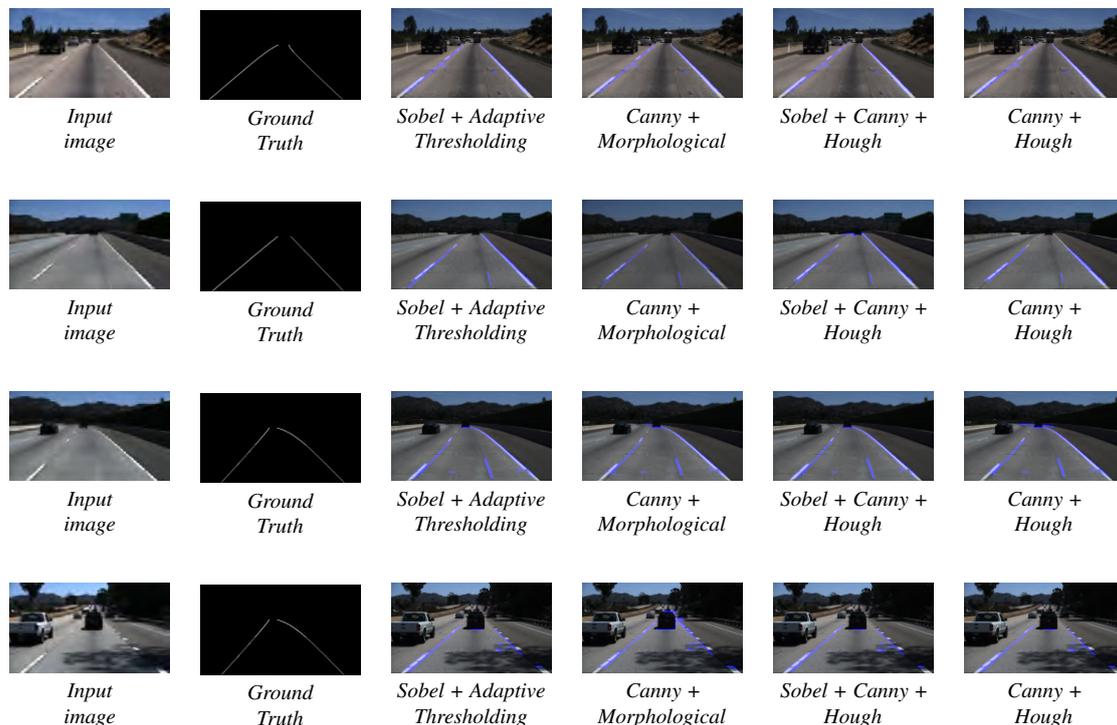


Figura 5.7 Resultados cualitativos en base de datos TuSimple.

### 5.5.3 Técnicas tradicionales en carriles rectos de mi propia base de datos

Este experimento enfocado en la detección de carriles utilizando técnicas tradicionales de visión por computadora aplicadas a mi propia base de datos de Morelos. Este conjunto de datos fue creado específicamente para abordar las características particulares de las carreteras en el estado de Morelos, donde se pueden encontrar diversas condiciones adversas que representan un desafío significativo para los algoritmos de detección. Ver Tabla 5.6. Las condiciones adversas consideradas en este experimento son variadas y buscan cubrir escenarios realistas de conducción. Entre ellas se encuentra la lluvia, que se evalúa en distintos niveles de intensidad: leve, moderada y fuerte. Cada una de estas situaciones afecta la visibilidad de los carriles de manera diferente, ya sea creando reflejos, difuminaciones o dificultando la claridad de las marcas viales. También se incluye el fenómeno del deslumbramiento solar (sun glare), causado por el sol directo, lo que puede saturar la imagen y hacer que los bordes de los carriles sean más difíciles de detectar. Adicionalmente, se analizan las condiciones de neblina en diferentes grados: ligera y moderada las cuales reducen la visibilidad de manera considerable y complican la detección precisa de los carriles. El experimento contempla además la conducción en condiciones normales, sin fenómenos climáticos adversos. Por otro lado, se evalúan situaciones de curvas normales, donde la detección de carriles requiere un seguimiento preciso de trayectorias no lineales, algo que puede afectar el rendimiento de los algoritmos. Las condiciones de conducción nocturna también forman parte de este análisis, considerando

tanto escenarios con ausencia de luz artificial (donde la única fuente de iluminación son los faros del vehículo), como aquellos con iluminación artificial tanto de autos como de infraestructura vial, los cuales ofrecen desafíos únicos para los algoritmos debido a los cambios en el contraste y el ruido visual, también se analizan variaciones en la superficie de la carretera, que incluyen diferentes tipos de asfalto, desgaste, y marcas viales deterioradas, todos factores que pueden influir en el rendimiento de los algoritmos de detección. A través de este conjunto de pruebas, se busca evaluar la robustez y efectividad de los algoritmos tradicionales en un entorno que simula las condiciones adversas más comunes encontradas en las carreteras del estado de Morelos. En la Figura 5.9 se pueden apreciar los resultados obtenidos por los diferentes detectores de líneas rectas. En la Tabla 5.7 se presentan los elementos de la matriz de confusión, que permiten un análisis detallado de las métricas de rendimiento del modelo. Para evaluar el desempeño, se emplearon métricas cuantitativas clave como *precisión*, *recall*, *F1 Score*, *accuracy*, *IoU*, *specificity* y *Dice Index*. Estas se encuentran en la Tabla 5.8 además la Figura 5.8 ilustra gráficamente los resultados obtenidos en relación con estas métricas en la base de datos Morelos, la leyenda SA se refiere a Sobel + Adaptive Thresholding, CM se refiere a Canny + Morphological, SCH se refiere a Sliding + Canny + Hough, CH se refiere a Canny + Hough y A se refiere a el promedio.

Tabla 5.6 Condiciones adversas para detección de carriles.

Condición Adversa	Descripción	Desafíos para la Detección de Carriles
<b>Lluvia</b>	Precipitaciones que pueden empañar la visibilidad y generar reflejos en la carretera.	Pérdida de contraste en las marcas viales, reflejos en la superficie y acumulación de agua sobre las líneas.
<b>Sol directo (Sun Glare)</b>	Luz intensa del sol, especialmente al amanecer o atardecer.	Deslumbramiento que oculta las marcas viales y sombras que confunden al modelo.
<b>Condiciones normales</b>	Escenarios estándar sin obstáculos climáticos o lumínicos extremos.	Menos desafíos, aunque puede haber variaciones menores como desgaste de líneas o suciedad.
<b>Curvas</b>	Cambios bruscos en la dirección del carril.	Dificultad para prever la continuidad del carril, especialmente en curvas cerradas o sin líneas marcadas.
<b>Niebla</b>	Reducción de la visibilidad debido a partículas de agua suspendidas.	Dificultad para detectar líneas a larga distancia o líneas difusas cercanas.
<b>Noche con luz artificial</b>	Iluminación proporcionada por postes o infraestructura vial.	Detección influenciada por sombras proyectadas o patrones de iluminación.
<b>Noche sin luz artificial</b>	Solo iluminación proporcionada por los faros del vehículo.	Campo de visión limitado, sombras intensas, y posibles interferencias por los faros de otros vehículos.
<b>Nieve o fangos de lodo</b>	Cobertura parcial o total de la carretera por nieve o barro.	Ocultación de las marcas viales, cambios en la textura de la superficie y patrones irregulares visibles.

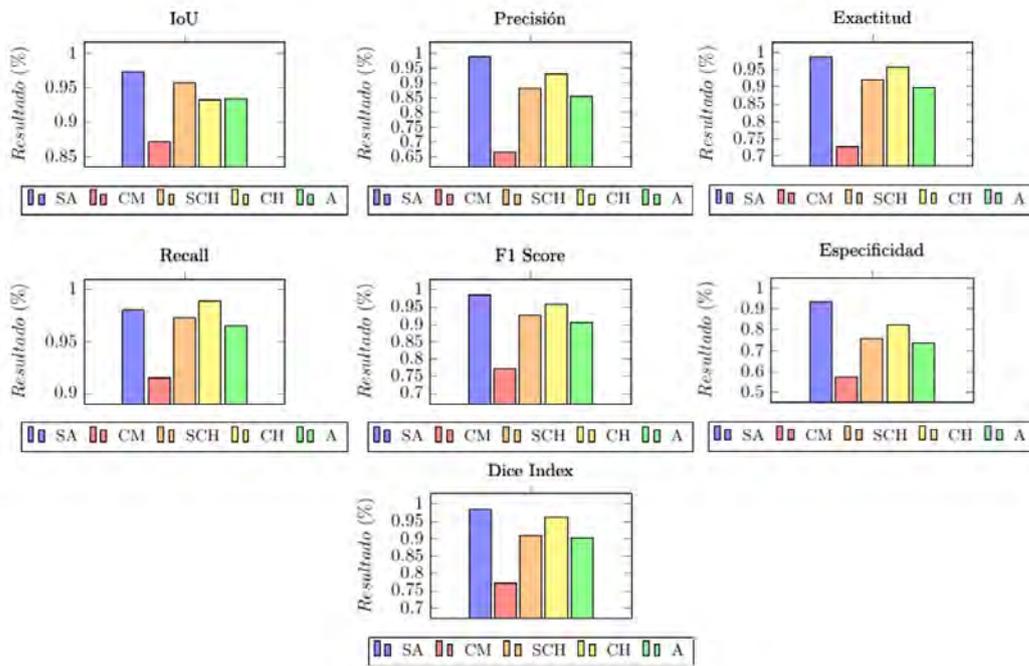


Figura 5.8 Resultados cuantitativos de la detección en base de datos propia.

Tabla 5.7 Elementos de la matriz de confusión para la Figura 5.9.

Algoritmo	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97888	0.01234	0.02001	0.98677
Canny + Morphological	0.91465	0.46012	0.08678	0.54389
Sliding + Canny + Hough	0.97023	0.13347	0.02911	0.87056
Canny + Hough	0.98712	0.07422	0.01345	0.92678

Tabla 5.8 Comparación de algoritmos basada en diversas métricas.

Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
SA	0.96801	0.98755	0.97997	0.98374	0.98381	0.98765	0.98374
CM	0.62581	0.66531	0.91334	0.76984	0.72729	0.54172	0.76984
SCH	0.85648	0.87907	0.97087	0.92269	0.91885	0.86707	0.92269
CH	0.91843	0.93007	0.98656	0.95748	0.95620	0.92585	0.95748

Estos resultados muestran que el algoritmo Canny + Hough obtuvo la mejor precisión en la detección de carriles con un valor de 0.98712 para verdaderos positivos (TP) y un bajo porcentaje de falsos positivos (FP) de 0.07422. El algoritmo Sobel + Adaptive Thresholding también presentó un rendimiento destacado, con una alta tasa de verdaderos

negativos (TN) de 0.98677 y un FP reducido de 0.01234. Por otro lado, el algoritmo Canny + Morphological mostró un menor rendimiento, con un alto valor de falsos positivos (0.46012) y una tasa de verdaderos negativos relativamente baja (0.54389). Los valores promedios indican que la detección de carriles en condiciones adversas es un desafío significativo, donde los algoritmos más robustos tienden a ser aquellos que manejan mejor las variaciones en las condiciones de luz y superficie de la carretera, como lo demuestra el rendimiento consistente de Canny + Hough y Sliding + Canny + Hough. El algoritmo Sobel + Adaptive Thresholding muestra el mejor rendimiento general, obteniendo una IoU de 0.96801 y un F1 Score de 0.98374, lo que refleja su capacidad para detectar carriles con gran precisión en condiciones adversas. Además, su exactitud de 0.98381 y especificidad de 0.98765 confirman su robustez frente a falsos positivos. Por otro lado, el algoritmo Canny + Morphological obtuvo el rendimiento más bajo, con una IoU de 0.62581 y un F1 Score de 0.76984, lo que sugiere que tiene dificultades en escenarios más complejos, probablemente debido a su menor precisión (0.66531) y especificidad (0.54172). El Sliding + Canny + Hough y Canny + Hough también presentan buenos resultados, con F1 Scores de 0.92269 y 0.95748, respectivamente. Esto indica que ambos algoritmos son capaces de manejar mejor las variaciones en las condiciones de luz y la calidad de la superficie vial. En promedio, las métricas reflejan que la precisión global de los algoritmos fue de 0.85064 y el recall de 0.96288, mostrando que la detección de carriles fue generalmente acertada, aunque algunos algoritmos se desempeñaron mejor que otros en condiciones adversas.

### Detector de línea recta

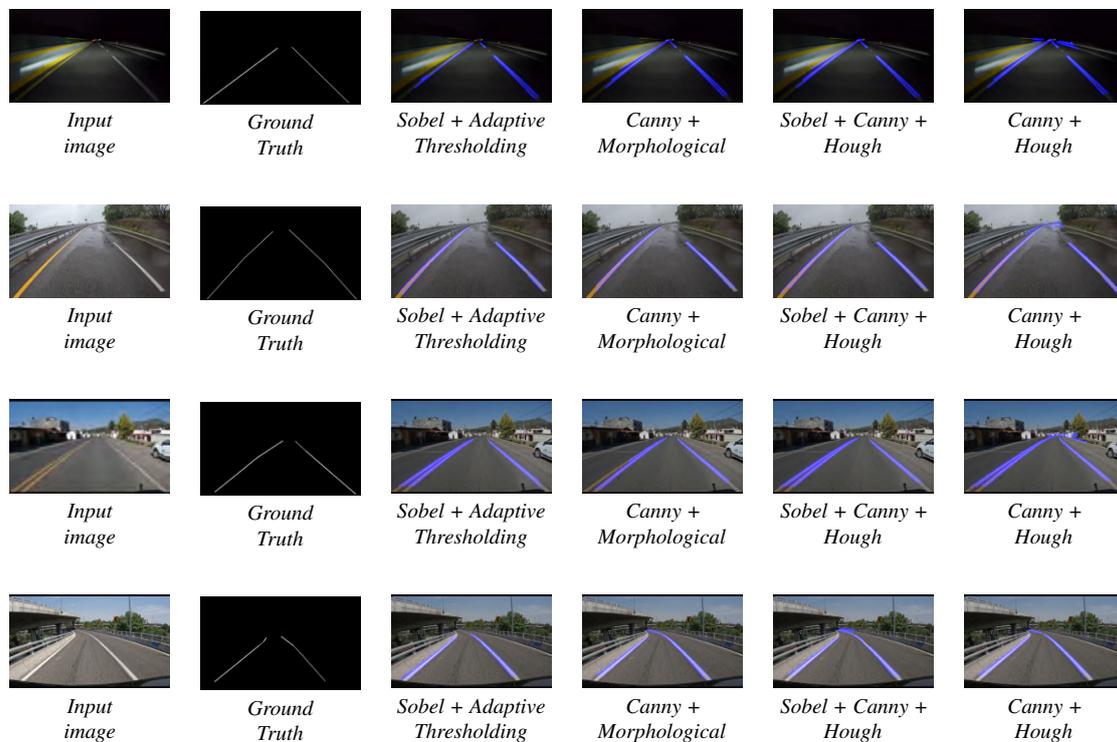


Figura 5.9 Resultados cualitativos en base de datos propia.

### 5.5.4 Extensión de técnicas tradicionales para la detección en curvas

En este experimento, se evaluaron diferentes métodos de detección de carriles curvos mediante algoritmos de visión por computadora. A diferencia de los carriles rectos, los carriles curvos presentan un desafío mayor debido a cómo la percepción de la curva cambia a medida que el vehículo se aproxima. Esta variación de perspectiva puede hacer que las curvas se perciban como más agudas o más suaves, según la distancia y el ángulo de visión. Por ello, se implementaron extensiones a las técnicas tradicionales para garantizar que los algoritmos sean más robustos y adaptativos en la detección de carriles curvos.

Entre los métodos utilizados, el algoritmo CHP, conocido como Canny + Hough + Parábola, emplea el detector de bordes de Canny en combinación con la transformada de Hough, ajustada específicamente para reconocer curvas en forma de parábola. Otro algoritmo destacado es GSC, o Gaussiano + Sobel + Curvo, el cual integra un filtro gaussiano para suavizar la imagen antes de aplicar el detector de bordes Sobel, mejorando la precisión en la detección de líneas curvas. Además, se utilizó CHP2, una variante del algoritmo Canny + Hough + Parábola, que ajusta el análisis de la curva tomando en cuenta la perspectiva del vehículo y las variaciones de la carretera. Finalmente, el algoritmo SPS, o Sobel + Perspectiva + Sliding, combina el detector Sobel con un ajuste dinámico de la perspectiva mediante un método sliding, lo que mejora la capacidad de detectar curvas complejas. En la Figura 5.10 la leyenda CHP se refiere a Canny + Hough + Parábola, GSC se refiere a Gaussiano + Sobel + Curvo, CHP2 se refiere a Canny + Hough + Perspectiva, SPS se refiere a Sobel + Perspectiva + Sliding y A se refiere a el promedio.

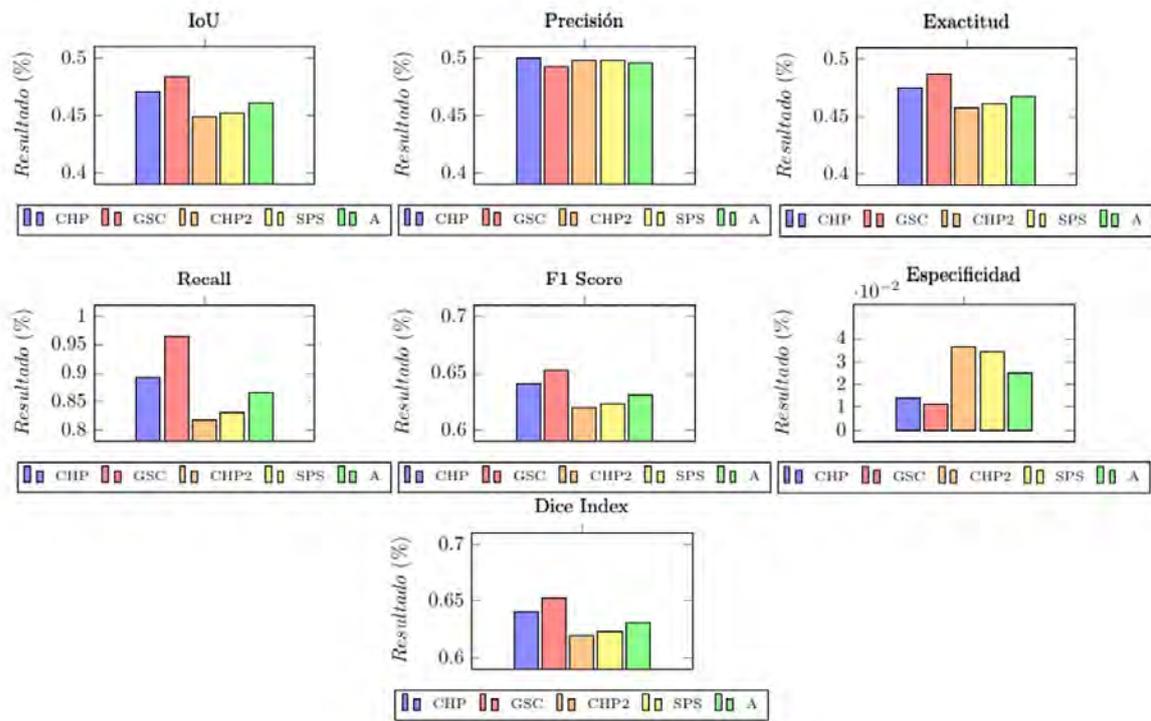
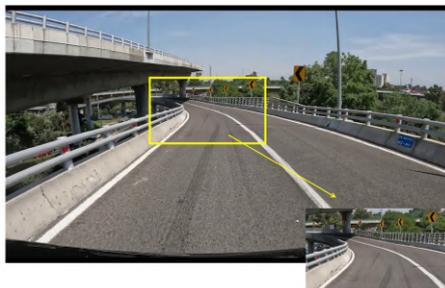
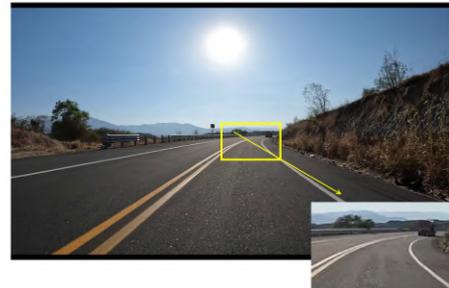


Figura 5.10 Resultados cuantitativos de la detección de carril curvo en diversas bases de datos.

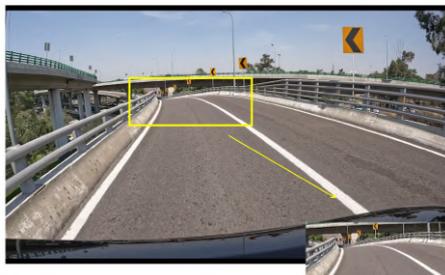
Estos métodos fueron evaluados con diversas bases de datos que incluyen escenarios de conducción en curvas bajo distintas condiciones adversas. Las pruebas incluyeron variaciones en la iluminación, geometría de los carriles y entorno de la carretera, lo que permitió medir el rendimiento de los algoritmos en situaciones reales. La adaptación de estas técnicas resulta crucial para asegurar que los sistemas de asistencia al conductor puedan mantener una detección precisa y confiable de los carriles, incluso cuando las curvas cambian su proyección conforme el vehículo se aproxima. Esto puede observarse en la Figura 5.11.



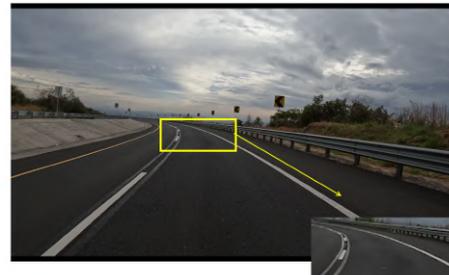
(a) Zoom del área seleccionada



(b) Zoom del área seleccionada



(c) Zoom del área seleccionada



(d) Zoom del área seleccionada

Figura 5.11 Imágenes que muestran áreas seleccionadas con zoom en presencia de curvas, etiquetadas como a, b, c y d.

En la Figura 5.12 se pueden apreciar los resultados obtenidos por los diferentes detectores de líneas curvos. En la Tabla 5.9 se presentan los elementos de la matriz de confusión, que permiten un análisis detallado de las métricas de rendimiento del modelo. Para evaluar el desempeño, se emplearon métricas cuantitativas clave como *precisión*, *recall*, *F1 Score*, *accuracy*, *IoU*, *specificity* y *Dice Index*. Estas se encuentran en la Tabla 5.10 además la Figura 5.10 ilustra gráficamente los resultados obtenidos en relación con estas métricas en mi propia base de datos de el estado Morelos.

Tabla 5.9 Elementos de la matriz de confusión para la Figura 5.12.

Algoritmo	TP	FP	FN	TN
Canny + Hough + Parábola	0.974	0.975	0.119	0.014
Gaussiano + Sobel + Curvo	0.959	0.988	0.035	0.011
Canny + Hough + Perspectiva	0.950	0.956	0.212	0.036
Sobel + Perspectiva + Sliding	0.951	0.957	0.193	0.034

Tabla 5.10 Comparación de algoritmos basada en diversas métricas.

Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
CHP	0.47099	0.49974	0.89113	0.64037	0.47454	0.01416	0.64037
GSC	0.48385	0.49255	0.96479	0.65216	0.48670	0.01101	0.65216
CHP2	0.44854	0.49843	0.81756	0.61930	0.45775	0.03629	0.61930
SPS	0.45264	0.49843	0.83129	0.62320	0.46136	0.03431	0.62320

Los resultados revelan diferencias clave en el rendimiento de los métodos cuando se enfrentan a carriles curvos, los cuales representan un desafío mayor que los carriles rectos debido a la variabilidad en la perspectiva y la complejidad de las curvas. El algoritmo Canny + Hough + Parábola (CHP) presenta una tasa elevada de verdaderos positivos (TP = 0.974), lo que sugiere que es muy eficaz en la detección de carriles curvos en la mayoría de las situaciones. Sin embargo, también tiene una tasa considerable de falsos positivos (FP = 0.975), lo que indica que a menudo puede identificar bordes o estructuras que no forman parte de los carriles como si lo fueran. Por otro lado, el algoritmo Gaussiano + Sobel + Curvo (GSC) muestra el mejor rendimiento en términos de falsos negativos (FN = 0.035), lo que significa que rara vez omite carriles que realmente están presentes. Este algoritmo es muy preciso, con una menor tasa de falsos positivos (FP = 0.988) que el resto, lo cual lo convierte en una opción robusta para escenarios con curvas pronunciadas. Canny + Hough + Perspectiva (CHP2) y Sobel + Perspectiva + Sliding (SPS) tienen tasas de falsos negativos y falsos positivos más altas en comparación con los otros métodos, lo que sugiere que, aunque pueden detectar carriles, son menos precisos en su identificación, ya que confunden más fácilmente estructuras que no son carriles (FP) o no reconocen los carriles correctamente (FN). En términos de promedio, observamos que la tasa de verdaderos positivos es alta (TP = 0.956), lo que indica que los algoritmos, en conjunto, tienden a detectar correctamente la mayor parte de los carriles. Sin embargo, la tasa de falsos negativos promedio (FN = 0.147) muestra que en algunas situaciones los algoritmos tienen dificultades para reconocer los carriles correctamente, lo que puede deberse a factores como cambios de iluminación, sombras o curvas muy pronunciadas. En conclusión, Gaussiano + Sobel + Curvo parece ser la técnica más equilibrada y precisa en la detección de carriles curvos, con bajos niveles de falsos negativos y falsos positivos. Canny + Hough + Parábola, aunque efectivo, tiene un mayor riesgo de detectar erróneamente otros elementos como carriles. En conjunto, estas técnicas tradicionales pueden ser efectivas, pero es necesario seguir mejorando para reducir los errores, especialmente en condiciones adversas y curvas

más pronunciadas. El Índice de IoU (Intersection over Union) revela la superposición entre la predicción y la verdad del terreno, siendo un buen indicador de cuán precisa es la detección general. El algoritmo Gaussiano + Sobel + Curvo muestra el mejor desempeño con un IoU de 0.48385, lo que indica que es el más preciso en términos de superposición entre las áreas detectadas y las reales. Sin embargo, los valores de IoU en general son bajos, lo que resalta la dificultad de detectar carriles curvos de manera efectiva. En cuanto a la precisión, que mide la proporción de verdaderos positivos frente a la suma de verdaderos y falsos positivos, todos los algoritmos presentan valores cercanos entre sí, alrededor de 0.49, lo que significa que aunque identifican carriles, también tienden a identificar erróneamente otros elementos como carriles. El recall es una métrica clave en este tipo de problemas, ya que mide cuántos de los carriles reales fueron correctamente detectados. El algoritmo Gaussiano + Sobel + Curvo se destaca con un recall de 0.96479, lo que indica que detecta correctamente casi todos los carriles presentes. Este alto recall sugiere que es especialmente efectivo para asegurar que los carriles no pasen desapercibidos, aunque con el costo de algunos falsos positivos. El F1 Score, que combina precisión y recall, muestra que Gaussiano + Sobel + Curvo nuevamente tiene el mejor balance entre estas dos métricas con un valor de 0.65216. La exactitud mide la proporción de todas las predicciones correctas, y también es similar para todos los algoritmos, con un promedio de 0.46692, lo que sugiere que, en general, los algoritmos tienen una tasa moderada de éxito al clasificar correctamente los carriles. El algoritmo Gaussiano + Sobel + Curvo se destaca por su alta tasa de recall, lo que lo convierte en la opción más confiable para asegurar que los carriles sean detectados correctamente en curvas.

### Detector de línea curvo

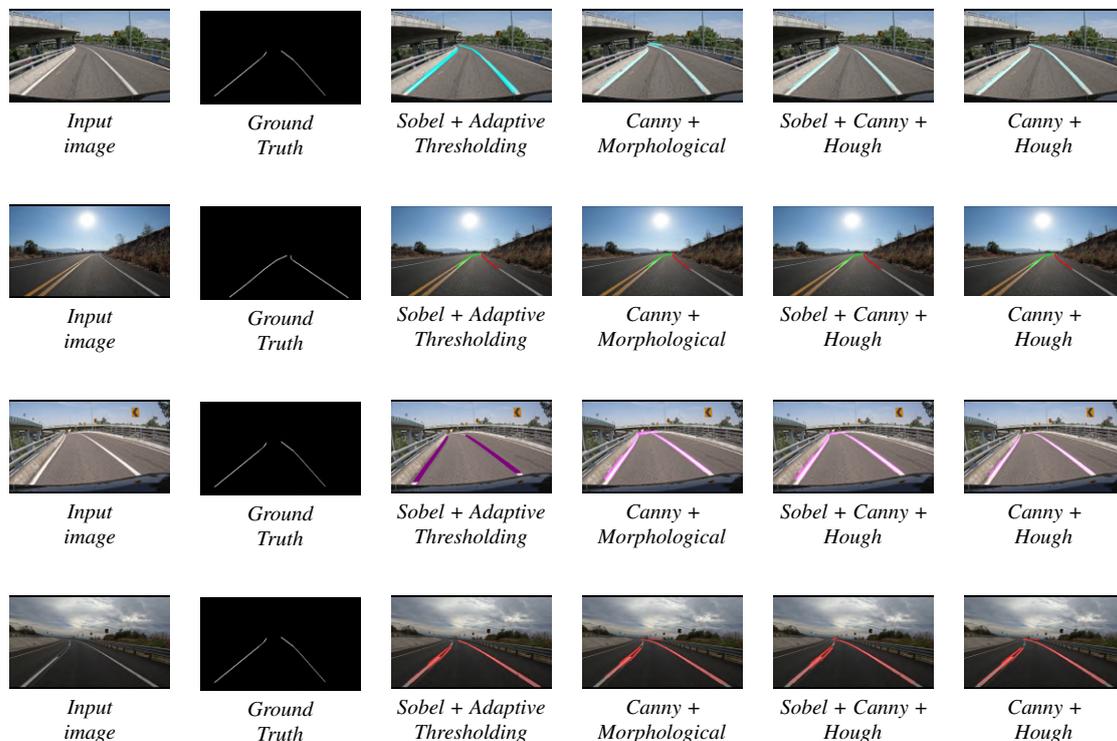


Figura 5.12 Resultados cualitativos en diversas base de datos.

### 5.5.5 Técnicas de modelos de aprendizaje profundo

En este experimento, el enfoque cambia radicalmente al emplear técnicas de aprendizaje profundo para la detección de carriles, dejando de lado los métodos clásicos de visión por computadora que se usaron en los experimentos anteriores. Se utilizaron arquitecturas de redes neuronales avanzadas, específicamente SCNN (Spatial Convolutional Neural Network) y VGG16, que ofrecen una ventaja considerable al aprender características complejas y detalladas a partir de grandes volúmenes de datos etiquetados, adaptándose mejor a variaciones en las condiciones de la carretera, cambios de iluminación y escenarios desafiantes como curvas pronunciadas o intersecciones.

Los modelos basados en SCNN son particularmente efectivos en la detección de carriles, ya que están diseñados para manejar la topología espacial de las carreteras y pueden modelar la relación entre diferentes puntos de carriles. Por otro lado, VGG16, con su arquitectura profunda y convolucional, permite extraer características jerárquicas que ayudan a mejorar la precisión en la identificación de carriles, incluso en condiciones difíciles. Estos modelos han demostrado ser particularmente efectivos en escenarios donde los carriles no son fácilmente distinguibles para los métodos tradicionales, debido a cambios de perspectiva, sombras, o la presencia de obstáculos en la carretera. Las bases de datos contienen imágenes de carreteras en diferentes condiciones ambientales, como luz diurna, nocturna, lluvia, niebla, y distintos tipos de carreteras con curvas y líneas de carril desgastadas. Esto permitió evaluar el rendimiento de los modelos de aprendizaje profundo en situaciones similares a las que enfrentan los conductores en la vida real. En la Figura 5.14, en la Figura 5.15 y en la Figura 5.16 se pueden apreciar los resultados obtenidos de los detectores CNN. En la Tabla 5.11 se presentan los elementos de la matriz de confusión, que permiten un análisis detallado de las métricas de rendimiento del modelo. Para evaluar el desempeño, se emplearon métricas cuantitativas clave como *precisión*, *recall*, *F1 Score*, *accuracy*, *IoU*, *specificity* y *Dice Index*. Estas se encuentran en la Tabla 5.11 y en la Tabla 5.12 además la Figura 5.13 ilustra gráficamente los resultados obtenidos en relación con estas métricas.

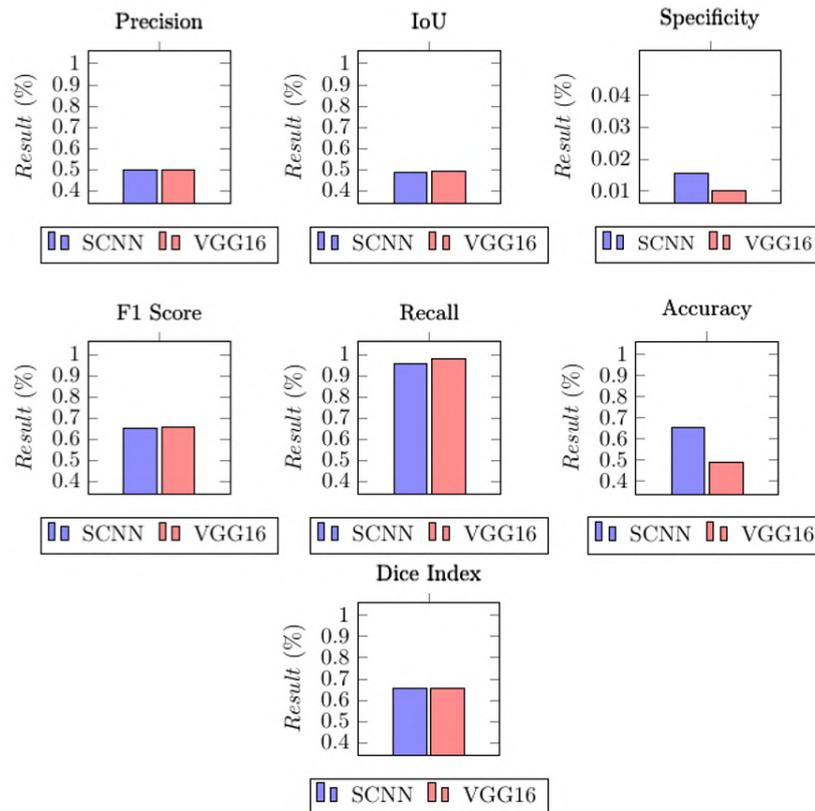


Figura 5.13 Resultados cuantitativos de la detección de carril con aprendizaje profundo.

El modelo de detección de carriles basado en la arquitectura SCNN demostró un desempeño significativo, reflejado en los valores de la matriz de confusión. La tasa de verdaderos positivos (TP) fue de 0.955, lo que significa que el modelo detectó correctamente el 95.5% de los carriles en las imágenes evaluadas. Este alto porcentaje indica que SCNN es eficaz para identificar carriles incluso en condiciones desafiantes.

No obstante, la tasa de falsos positivos (FP) fue de 0.960, lo que evidencia que el modelo clasificó erróneamente un número considerable de áreas no correspondientes a carriles como carriles. Este valor señala un área de mejora, ya que un alto FP puede resultar en detecciones incorrectas que podrían complicar las decisiones en sistemas de asistencia a la conducción.

Por otro lado, la tasa de falsos negativos (FN) se situó en 0.045, lo que implica que solo el 4.5% de los carriles presentes en las imágenes no fueron detectados. Esto refleja que SCNN tiene una alta sensibilidad y puede identificar la mayoría de los carriles con precisión, lo cual es crítico para aplicaciones en tiempo real.

Finalmente, la tasa de verdaderos negativos (TN) fue de 0.015. Aunque este valor es bajo, refleja que el modelo tiene dificultades para identificar correctamente las áreas que no contienen carriles, lo que sugiere que podría beneficiarse de ajustes para mejorar la especificidad.

El modelo basado en la arquitectura VGG16 presentó resultados notables en la detección de carriles. La tasa de verdaderos positivos (TP) alcanzó un valor de 0.960, indicando que el modelo identificó correctamente el 96% de los carriles en las imágenes

evaluadas. Este rendimiento destaca la capacidad de VGG16 para manejar condiciones adversas como sombras, líneas desgastadas o la presencia de obstáculos.

Sin embargo, la tasa de falsos positivos (FP) fue de 0.980, lo que señala un número elevado de áreas no correspondientes a carriles clasificadas erróneamente como carriles. Aunque la tasa de TP es alta, este valor de FP sugiere que hay margen de mejora en la especificidad del modelo para minimizar detecciones incorrectas.

La tasa de falsos negativos (FN) fue de 0.020, lo que demuestra que el modelo omitió detectar solo el 2% de los carriles presentes en las imágenes. Este resultado es muy alentador y refuerza la fiabilidad de VGG16 para aplicaciones prácticas de detección de carriles.

Por último, la tasa de verdaderos negativos (TN) fue de 0.010, lo que sugiere que el modelo también enfrentó dificultades para identificar correctamente las áreas que no contenían carriles. Esto resalta la necesidad de optimización en este aspecto.

Ambas arquitecturas mostraron un desempeño sólido en términos de verdaderos positivos, con VGG16 ligeramente superando a SCNN. Sin embargo, ambas presentan un desafío común: la alta tasa de falsos positivos. Este aspecto podría afectar su desempeño en escenarios prácticos donde es crucial minimizar las detecciones erróneas.

En términos de falsos negativos, VGG16 mostró una ventaja sobre SCNN, detectando un menor porcentaje de carriles omitidos. Esto hace que VGG16 sea particularmente adecuado para aplicaciones que requieren alta sensibilidad.

En conclusión, mientras que ambas arquitecturas tienen sus fortalezas, la elección entre SCNN y VGG16 dependerá del contexto de aplicación. SCNN destaca por su capacidad para capturar la estructura espacial de los carriles, mientras que VGG16 ofrece una detección robusta en condiciones visualmente desafiantes. Las áreas de mejora incluyen la reducción de falsos positivos y la optimización para clasificar correctamente las áreas que no contienen carriles, como se ilustra en la Tabla 5.11, y en la Tabla 5.12.

Tabla 5.11 Matriz de confusión para el detector con arquitectura SCNN

Arquitectura	TP	FP	FN	TN
SCNN	0.955	0.960	0.045	0.015

Tabla 5.12 Matriz de confusión para el detector con arquitectura VGG16

Arquitectura	TP	FP	FN	TN
VGG16	0.960	0.980	0.020	0.010

Para la arquitectura SCNN, el índice de Intersección sobre Unión (IoU) alcanzó un valor de 0.4872, lo que representa una capacidad moderada para identificar correctamente las áreas de carriles en las imágenes evaluadas. La precisión fue de 0.4987, indicando que aproximadamente el 49.87% de las predicciones realizadas por el modelo correspondieron a verdaderos positivos. Sin embargo, el recall fue notablemente alto, con un valor de 0.9550, lo que significa que el modelo logró detectar correctamente el 95.5% de los carriles

presentes, aunque con una tasa considerable de falsos positivos. El F1 Score, que equilibra la precisión y el recall, se situó en 0.65165, lo que refleja un rendimiento aceptable en la combinación de estas métricas. La exactitud del modelo fue de 0.6552, mientras que la especificidad, que mide la capacidad del modelo para identificar correctamente las áreas que no contienen carriles, fue baja, con un valor de 0.0154. Finalmente, el índice de Dice también alcanzó 0.6552, reflejando una coherencia con la métrica F1 Score.

En el caso de la arquitectura VGG16, se observó un rendimiento ligeramente superior en algunas métricas. El IoU fue de 0.4898, lo que representa una mejora marginal en comparación con SCNN. La precisión alcanzó 0.4948, un valor similar al obtenido por SCNN. El recall fue significativamente alto, con un valor de 0.9796, indicando que VGG16 fue capaz de detectar correctamente el 97.96% de los carriles presentes en las imágenes, lo que resalta su eficacia en la identificación de carriles. El F1 Score fue de 0.6575, ligeramente superior al de SCNN, mostrando un mejor balance entre precisión y recall. No obstante, la exactitud de VGG16 fue más baja, con un valor de 0.4924, y la especificidad también se mantuvo reducida, alcanzando apenas 0.0101. El índice de Dice fue igual al F1 Score, con un valor de 0.6575, reforzando la consistencia de estas métricas.

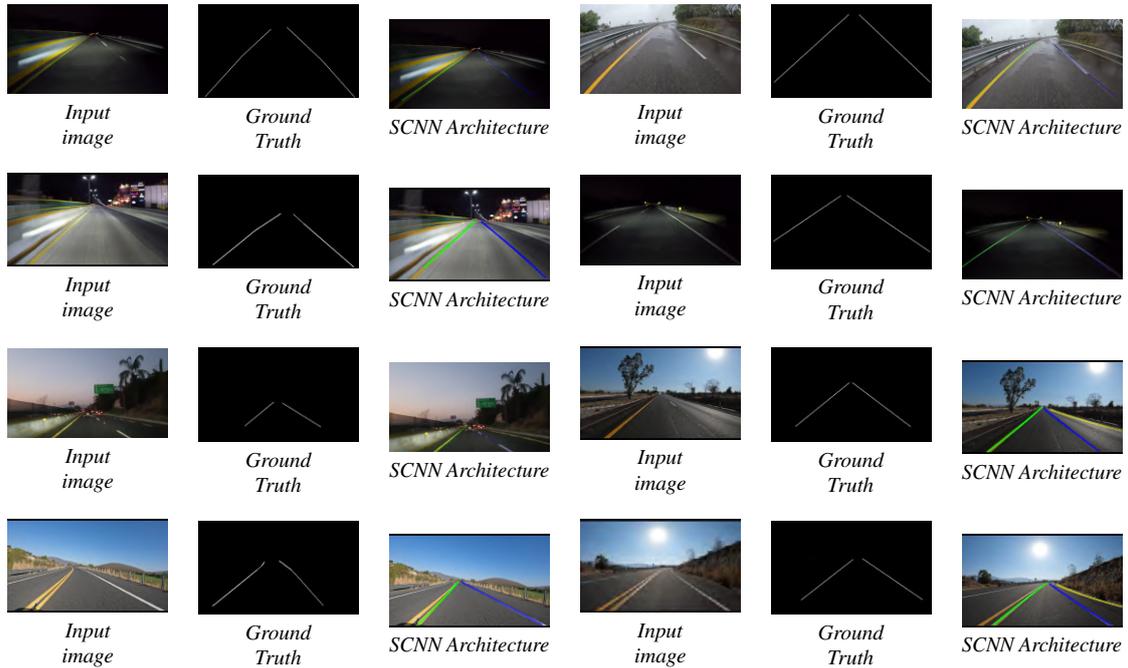
En general, los resultados indican que ambas arquitecturas, SCNN y VGG16, presentan un desempeño competitivo en la detección de carriles. Si bien SCNN demuestra un mejor equilibrio en términos de exactitud general, VGG16 muestra una ventaja en el recall y en la combinación de métricas como el F1 Score e IoU. Esto sugiere que VGG16 podría ser más adecuada en aplicaciones donde la sensibilidad a los carriles detectados es crítica, mientras que SCNN ofrece una opción más equilibrada para escenarios donde la exactitud global sea prioritaria.

Estos resultados subrayan la importancia de evaluar múltiples arquitecturas para optimizar los modelos de detección de carriles en aplicaciones del mundo real, considerando los requisitos específicos del sistema y las condiciones del entorno en el que serán implementados, como se ilustra en la Tabla 5.13.

Tabla 5.13 Comparación de métricas entre las arquitecturas SCNN y VGG16.

Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
SCNN	0.4872	0.4987	0.9550	0.65165	0.6552	0.0154	0.6552
VGG16	0.4898	0.4948	0.9796	0.6575	0.4924	0.0101	0.6575

### Detector de carril con arquitectura SCNN



### Detector de carril con VGG16 Architecture

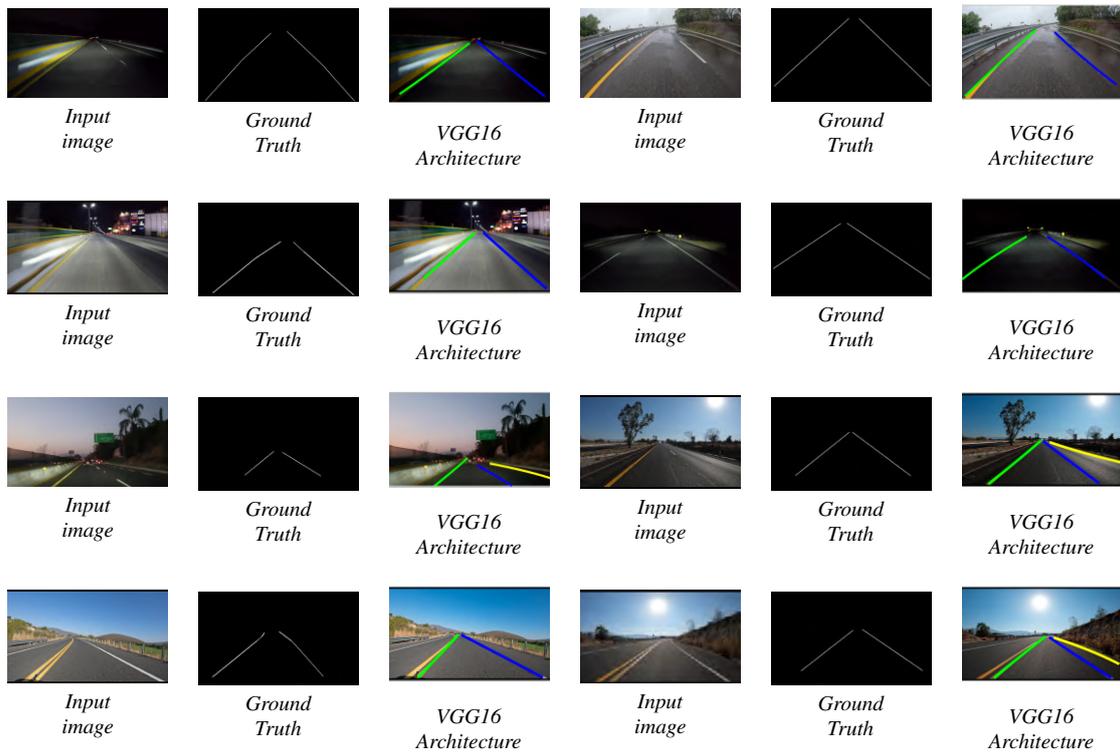
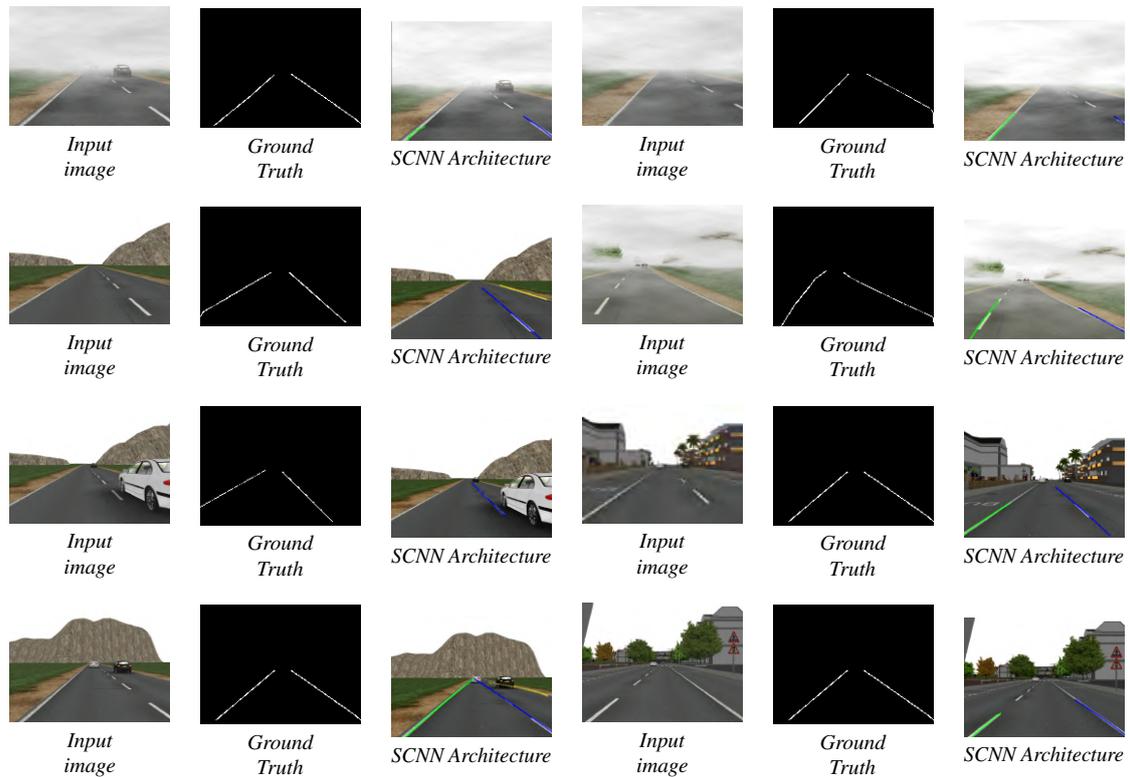


Figura 5.14 Resultados cualitativos con arquitecturas SCNN & VGG16 en base de datos propia.

### Detector de carril con arquitectura SCNN



### Detector de carril con arquitectura VGG16

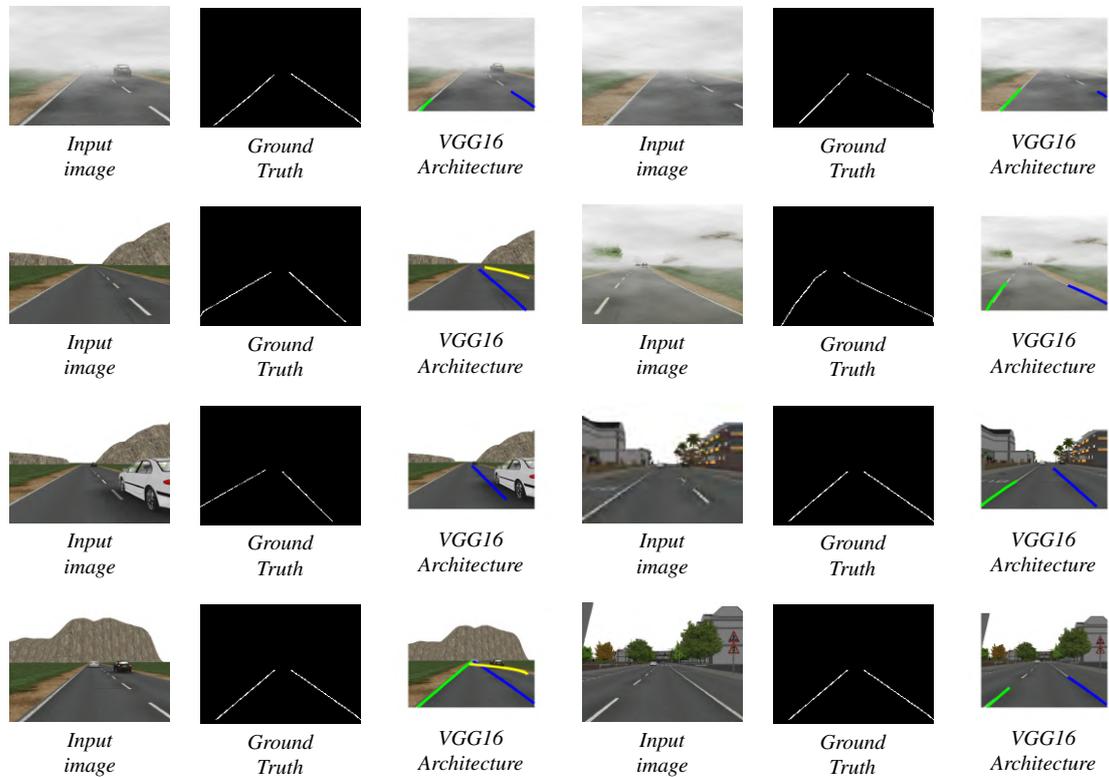
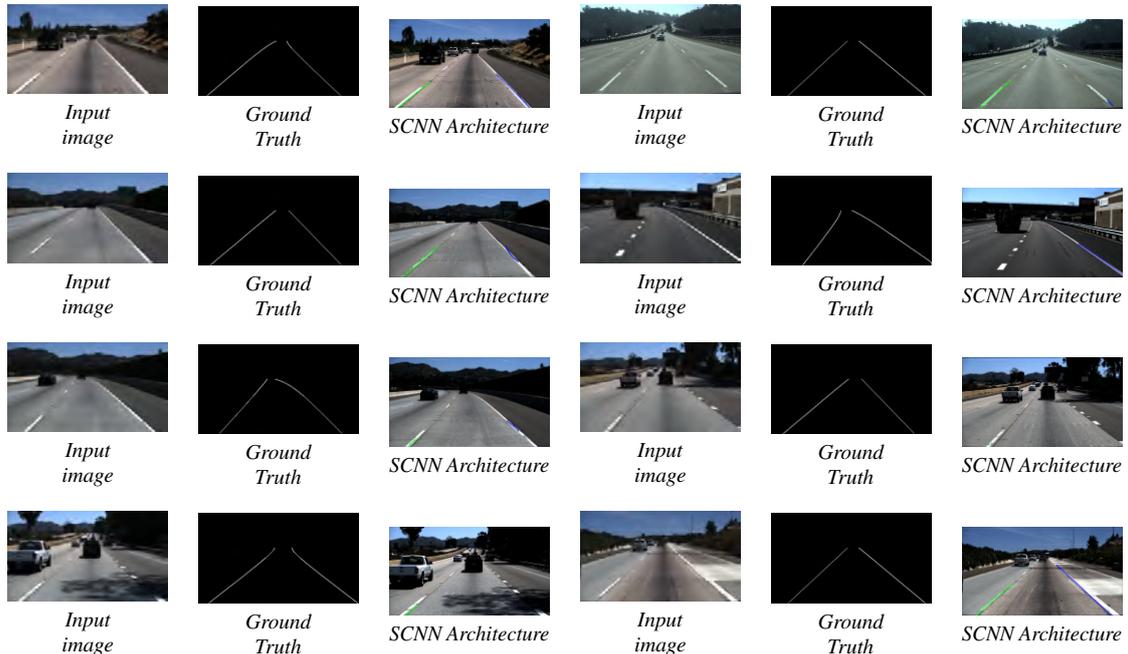


Figura 5.15 Resultados cualitativos con arquitecturas SCNN & VGG16 en base de Frida & Frida 2.

### Detector de carril con arquitectura SCNN



### Detector de carril con arquitectura VGG16

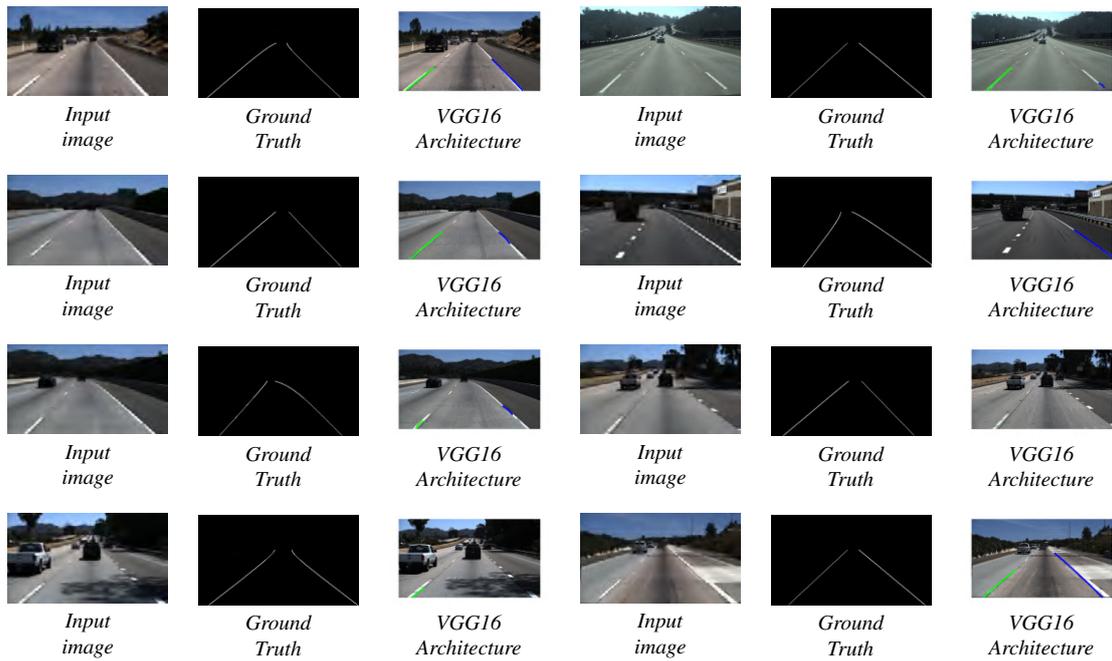


Figura 5.16 Resultados cualitativos con arquitecturas SCNN & VGG16 en base de TuSimple.

Como se muestra en la Tabla 5.14, los enfoques destacados para cada experimento demuestran las configuraciones de mejor rendimiento basadas en las métricas de la matriz de confusión. La tabla resume los Verdaderos Positivos (TP), Falsos Positivos (FP), Falsos Negativos (FN) y Verdaderos Negativos (TN) para cada enfoque seleccionado.

Tabla 5.14 Enfoques destacados por experimento.

Experimento	Enfoque Destacado	TP	FP	FN	TN
1	Canny + Hough	0.988	0.072	0.011	0.927
2	Canny + Hough	0.990	0.075	0.010	0.930
3	Canny + Hough	0.987	0.074	0.013	0.926
4	Gaussian + Sobel + Curve	0.959	0.988	0.035	0.011
5	VGG16	0.960	0.988	0.020	0.010

Los resultados presentados en la tabla destacan el rendimiento de los enfoques seleccionados según las métricas de la matriz de confusión. El método Canny + Hough se posiciona como el más consistente en los primeros tres experimentos, obteniendo altos valores de Verdaderos Positivos (TP) y manteniendo bajas tasas de Falsos Positivos (FP) y Falsos Negativos (FN), lo que evidencia su robustez en condiciones variadas. En el cuarto experimento, el enfoque Gaussian + Sobel + Curve mostró un desempeño competitivo, aunque con un leve incremento en FP y FN. Por último, el modelo basado en VGG16 sobresalió en el quinto experimento, registrando el menor número de FN, lo que sugiere una mayor capacidad para capturar características complejas. En general, los altos valores de Verdaderos Negativos (TN) reflejan la eficacia de los enfoques en la identificación de áreas negativas, destacando su adaptabilidad a distintos escenarios experimentales.

Las métricas de la matriz de confusión permiten evaluar el rendimiento de los modelos en términos de detección de carriles, considerando aciertos (TP, TN) y errores de clasificación (FP, FN). Sin embargo, para una evaluación más completa de la segmentación y efectividad del modelo, se han considerado métricas adicionales como la Intersección sobre Unión (IoU), Precisión, Recall, Puntuación F1, Exactitud, Especificidad e Índice de Dice.

### Comparación de Algoritmos en los Experimentos

La Tabla 5.15 presenta un análisis comparativo de los algoritmos con mejor rendimiento en cada experimento, basado en métricas clave de evaluación, incluyendo Intersección sobre Unión (IoU), Precisión, Recall, Puntuación F1, Exactitud, Especificidad e Índice de Dice. El algoritmo destacado en cada experimento corresponde al que obtuvo el mejor desempeño en la mayoría de las métricas.

Tabla 5.15 Algoritmos destacados por experimento

Exp.	Algoritmo	IoU	Precisión	Recall	F1 Score	Exactitud	Especificidad	Índice de Dice
1	S + A	0.9729	0.9888	0.9809	0.9848	0.9849	0.9889	0.9848
2	S + A	0.9656	0.9866	0.9784	0.9825	0.9825	0.9867	0.9825
3	S + A	0.9680	0.9875	0.9799	0.9837	0.9838	0.9876	0.9837
4	G + S + C	0.4838	0.4925	0.9647	0.6521	0.4867	0.0110	0.6521
5	VGG16	0.4898	0.4948	0.9796	0.6575	0.4924	0.0101	0.6575

Estas métricas permiten evaluar el rendimiento general de los algoritmos de detección de carriles. Incluyen Intersección sobre Unión (IoU), Precisión, Recall, Puntuación F1, Exactitud, Especificidad y el Índice de Dice. En conjunto, proporcionan una evaluación completa de la capacidad del modelo para detectar correctamente las marcas de carril en las imágenes. IoU cuantifica la superposición entre la predicción del modelo y el ground truth, mientras que la Precisión y el Recall evalúan la capacidad del modelo para minimizar falsos positivos e identificar correctamente los verdaderos positivos, respectivamente. La Puntuación F1 proporciona una medida equilibrada entre Precisión y Recall, y la Exactitud refleja la proporción total de predicciones correctas.

# Capítulo 6

## Conclusiones

Dentro de este capítulo se realiza un análisis de las conclusiones del trabajo de investigación a partir de los resultados obtenidos, mediante la experimentación realizada, en relación a los objetivos contemplados. A su vez se hace una descripción de los productos generados y las aportaciones realizadas en el tema de investigación. Para finalizar se abordan ideas y sugerencias para trabajos futuros.

### 6.1 Objetivos y alcances logrados

Como aportaciones al tema de análisis de algoritmos de visión por computadora para un asistente de mantenimiento de carriles, se detallan en la Tabla 6.1, alineadas con los objetivos específicos establecidos, y en la Tabla 6.2, en concordancia con los alcances y limitaciones de la investigación.

Tabla 6.1 Objetivos y Actividades

Objetivos	Actividades
Comprender los conceptos relacionados con el sistema de asistencia de mantenimiento de carril.	Se realizó el estudio del marco teórico y estado del arte de algoritmos de visión por computadora enfocados en la detección de carriles, incluyendo técnicas de preprocesamiento de imágenes, métodos de segmentación y clasificación de características relevantes, así como enfoques de aprendizaje profundo aplicados a la mejora de la precisión y robustez en condiciones adversas.
Seleccionar algoritmos de visión por computadora para la detección de marcas y fronteras en carriles bajo condiciones adversas.	Se evaluarán diferentes algoritmos y se seleccionarán los más adecuados basándose en criterios de efectividad en condiciones adversas.
Probar los diferentes algoritmos en condiciones normales y adversas.	Se realizarán simulaciones y pruebas para evaluar el rendimiento de los algoritmos bajo distintas condiciones.
Evaluar el desempeño de los algoritmos seleccionados.	Se llevarán a cabo análisis de rendimiento utilizando métricas cuantitativas como precisión, recall, F1 Score, accuracy, IoU, specificity y Dice Index, que permiten medir la eficacia de los algoritmos bajo diferentes condiciones y entornos.

Tabla 6.2 Alcances y Limitaciones

<b>Alcances</b>	<b>Actividades</b>
Se realizará un estudio detallado de los diferentes algoritmos de visión por computadora utilizados en sistemas de asistencia de mantenimiento de carril.	Se analizarán y documentarán, en una revisión del estado del arte, las características y capacidades de diversos algoritmos aplicados a la detección de carriles.
Se evaluarán las capacidades de detección y seguimiento de los carriles bajo condiciones normales y adversas.	Se llevarán a cabo pruebas específicas para medir el rendimiento de los algoritmos utilizando métricas cuantitativas como precisión, recall, F1 Score, accuracy, IoU, specificity y Dice Index.
El análisis se enfocará en la detección y análisis de características del camino, como las líneas de carril y los bordes de la carretera.	Se realizarán pruebas para identificar y analizar efectivamente las características del camino mediante diferentes técnicas de visión por computadora.
Se contemplarán entornos simulados y reales que representen vías rectas y curvas.	Se realizarán pruebas con simulaciones y entornos reales que incluyan tanto vías rectas como curvas.
Se considerarán diferentes estilos de carriles y condiciones de iluminación.	Se realizarán pruebas en condiciones variadas de iluminación y diferentes tipos de carriles para evaluar la robustez de los algoritmos.
<b>Limitaciones</b>	<b>Actividades</b>
El análisis no contemplará la trayectoria ni el control del vehículo.	El enfoque estará limitado a la detección y análisis de carriles, sin abordar el control dinámico del vehículo.
La selección de los algoritmos a analizar estará limitada a aquellos que estén disponibles públicamente o que puedan ser obtenidos a través de acuerdos de colaboración.	Se restringirá la investigación a métodos accesibles y documentados en la literatura existente.
No se contemplarán condiciones ambientales totalmente adversas.	Las pruebas se llevarán a cabo en condiciones seleccionadas, excluyendo escenarios extremos.
La variabilidad en las marcas de los carriles en diferentes partes del mundo puede presentar un desafío para la detección precisa de las mismas mediante un algoritmo de visión por computadora.	Se discutirá la influencia de diferentes estilos y condiciones de carriles en el rendimiento de los algoritmos.
La velocidad del vehículo puede limitar el tiempo de procesamiento de los algoritmos y afectar la precisión del sistema.	A mayor velocidad, la efectividad y precisión de los algoritmos se ven afectados.

## 6.2 Resultados del trabajo

### Productos

En esta investigación se ha considerado entregar como producto final lo siguiente:

- **Reporte del estado del arte:** En este documento se presenta un resumen de las publicaciones recientes y con mayor relevancia para el desarrollo de esta investigación, considerando para su inclusión los núcleos temáticos contemplados.

- **Reporte de resultados:** Un documento detallado que incluye los datos experimentales, la metodología empleada, los resultados obtenidos y un análisis de los mismos, con especial énfasis en las métricas de los algoritmos evaluados.
- **Artículo:** "*Lane Detection Using Computer Vision and Convolutional Neural Networks for Autonomous Vehicles*" en la revista IJCOPI International Journal of Combinatorial Optimization Problems and Informatics (Figura A.2).
- **Póster de divulgación científica:** Presentado en el 1er. Coloquio Internacional de Investigación, Tecnología y Sustentabilidad (Figura A.1).
- **Documento de tesis:** Este documento contiene todos los aspectos de la investigación dividido en capítulos, con el objetivo de aportar una contribución al campo de la detección de carriles para vehículos autónomos.

## 6.3 Conclusiones

Basado en los resultados obtenidos de los diversos experimentos, se pueden extraer conclusiones significativas respecto al desempeño de los algoritmos bajo diferentes condiciones. En los experimentos realizados, los algoritmos Sobel + Thresholding Adaptativo y Canny + Hough demostraron un rendimiento sobresaliente, particularmente en condiciones simples o ideales, donde lograron valores altos en métricas clave como IoU (hasta 0.97295 para Sobel + Thresholding Adaptativo en el Experimento 1) y Precisión (hasta 0.98885 para Sobel + Thresholding Adaptativo en el mismo experimento). Estos algoritmos mantuvieron un desempeño superior en términos de Recall, Puntuación F1, Exactitud e Índice de Dice, lo que sugiere una alta capacidad para detectar carriles correctamente en escenarios sencillos.

Sin embargo, a medida que las condiciones de prueba se volvieron más complejas, como se evidenció en el Experimento 4, los algoritmos tradicionales como Canny + Hough + Parábola y Gaussiano + Sobel + Curvo experimentaron una caída significativa en su rendimiento. Estos métodos mostraron bajos valores de IoU (alrededor de 0.47) y una Precisión moderada (alrededor de 0.49), lo que indica que no son tan efectivos para manejar situaciones desafiantes como curvas pronunciadas o marcas de carriles deterioradas. Este comportamiento resalta las limitaciones de los enfoques clásicos para gestionar condiciones más dinámicas y complejas.

Por otro lado, los modelos basados en redes neuronales profundas, como SCNN y VGG16, demostraron un rendimiento superior en términos de Recall y Puntuación F1, particularmente en escenarios más difíciles. En el Experimento 5, VGG16 alcanzó un Recall de 0.9796 y un Índice de Dice de 0.6575, lo que indica una notable capacidad para detectar carriles en condiciones desafiantes con una tasa de falsos negativos más baja. Aunque las métricas como IoU y Precisión fueron comparables entre las redes profundas y los algoritmos tradicionales, el mayor Recall sugiere que las redes neuronales son más sensibles a la detección de carriles, lo cual es crucial en entornos de tráfico dinámico.

Los resultados sugieren que los algoritmos basados en redes neuronales profundas, particularmente VGG16, ofrecen una ventaja significativa sobre los métodos tradicionales cuando se adaptan a condiciones complejas. Si bien Sobel + Thresholding Adaptativo y

Canny + Hough son efectivos en entornos simples, las redes neuronales exhiben mayores capacidades de generalización y un rendimiento más robusto en entornos dinámicos. Esto refuerza la importancia de considerar las redes neuronales como una opción preferida para la detección de carriles en escenarios del mundo real, donde las condiciones pueden variar significativamente.

## **6.4 Trabajo futuro**

Como trabajo futuro derivado de este proyecto de investigación, se propone explorar técnicas de mejora en la detección de líneas curvas mediante la implementación de arquitecturas de redes neuronales más avanzadas, esto podría permitir una mejor adaptación a las variaciones en la geometría de las carreteras y las condiciones de iluminación. Asimismo, se plantea la posibilidad de incorporar datos sintéticos generados a partir de simulaciones de tráfico y condiciones climáticas diversas para enriquecer las bases de datos utilizadas en el entrenamiento, aumentando así la robustez y generalización del modelo.

# Referencias

- [1] Andrade, D. C., Bueno, F., Franco, F. R., Silva, R. A., Neme, J. H. Z., Margraf, E., Omoto, W. T., Farinelli, F. A., Tusset, A. M., Okida, S., Santos, M. M. D., Ventura, A., Carvalho, S., and Amaral, R. d. S. (2019). A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1497–1507.
- [2] Artola, A. (2023a). Semantic Segmentation: A Zoology of Deep Architectures. *Image Processing On Line*, 13:167–182. <https://doi.org/10.5201/ipol.2023.447>.
- [3] Artola, A. (2023b). Semantic Segmentation: A Zoology of Deep Architectures. *Image Processing On Line*, 13:167–182. <https://doi.org/10.5201/ipol.2023.447>.
- [4] Behrendt, K. and Soussan, R. (2019). Unsupervised labeled lane markers using maps. pages 832–839.
- [5] Benloucif, M., Sentouh, C., Floris, J., Simon, P., and Popieul, J.-C. (2019). Online adaptation of the level of haptic authority in a lane keeping system considering the driver's state. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61:107–119. Special TRF issue: Driving simulation.
- [6] Bian, Y., Ding, J., Hu, M., Xu, Q., Wang, J., and Li, K. (2020). An advanced lane-keeping assistance system with switchable assistance modes. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):385–396.
- [7] Bilal, H., Yin, B., Khan, J., Wang, L., Zhang, J., and Kumar, A. (2019). Real-time lane detection and tracking for advanced driver assistance systems. In *2019 Chinese control conference (CCC)*, pages 6772–6777. IEEE.
- [8] Bradley, D. and Roth, G. (2007). Adaptive thresholding using the integral image. *Journal of Graphics Tools*, 12:13–21.
- [9] Burger, W. and Burge, M. J. (2022). Digital image processing: An algorithmic introduction. *Texts in Computer Science*.
- [10] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.
- [11] Cao, Song, Song, Xiao, and Peng (2019). Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments. *Sensors*, 19(14):3166.
- [12] Cario, G., Casavola, A., and Lupia, M. (2017). Lane detection and tracking problems in lane departure warning systems. *Computer Vision and Imaging in Intelligent Transportation Systems*, pages 283–303.
- [13] Chan, T., Sandberg, B., and Xu, J. (2001). Adaptive de-fogging using dark channel prior. In *Proceedings of the IEEE International Conference on Image Processing*, pages 234–238. IEEE.

- [14] Chen, W.-T., Ding, J.-J., and Kuo, S.-Y. (2019). Pms-net: Robust haze removal based on patch map for single images.
- [15] Choi, Y., Park, J. H., and Jung, H.-Y. (2018). Lane detection using labeling based ransac algorithm. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, 12(4):245–248.
- [16] Collado, J. M., Hilario, C., de la Escalera, A., and Armingol, J. M. (2006). Adaptive road lanes detection and classification. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 1151–1162. Springer.
- [17] Cui, Y. e. a. (2017). Curved lane detection using bézier curves. *Journal of Computer Vision*, 67:284–295.
- [18] Dorj, B., Hossain, S., and Lee, D.-J. (2020). Highly curved lane detection algorithms based on kalman filter. *Applied Sciences*, 10(7).
- [19] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15.
- [20] Fakhfakh, M., Chaari, L., and Fakhfakh, N. (2020). Bayesian curved lane estimation for autonomous driving. *Journal of Ambient Intelligence and Humanized Computing*, 11(10):4133–4143.
- [21] Farag, W. (2020). A comprehensive real-time road-lanes tracking technique for autonomous driving. *International Journal of Computing and Digital Systems*, 9(03).
- [22] Farag, W. and Saleh, Z. (2019). An advanced road-lanes finding scheme for self-driving cars.
- [23] García, G. J., Jara, C. A., Pomares, J., Alabdo, A., Poggi, L. M., and Torres, F. (2014). A survey on fpga-based sensor systems: Towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing. *Sensors*, 14(4):6247–6278.
- [24] Garnett, N., Cohen, R., Pe’er, T., Lahav, R., and Levi, D. (2019). 3d-lanenet: End-to-end 3d multiple lane detection.
- [25] Geiger, A., Lenz, P., Stiller, C., and Franke, U. (2012). Vision meets robotics: The kitti dataset. *arXiv preprint arXiv:1204.2004*.
- [26] Getreuer, P. (2012). Total Variation Deconvolution using Split Bregman. *Image Processing On Line*, 2:158–174. <https://doi.org/10.5201/ipol.2012.g-tvdc>.
- [27] Ghanem, S., Kanungo, P., Panda, G., Satapathy, S. C., and Sharma, R. (2021). Lane detection under artificial colored light in tunnels and on highways: an iot-based framework for smart city infrastructure. *Complex amp; Intelligent Systems*, 9(4):3601–3612.
- [28] Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall.
- [29] Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., and Randall, G. (2012). LSD: a Line Segment Detector. *Image Processing On Line*, 2:35–55. <https://doi.org/10.5201/ipol.2012.gjmr-lsd>.

- [30] Grompone von Gioi, R. and Randall, G. (2017). A Sub-Pixel Edge Detector: an Implementation of the Canny/Devernavy Algorithm. *Image Processing On Line*, 7:347–372. <https://doi.org/10.5201/ipol.2017.216>.
- [31] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- [32] He, K., Sun, J., and Tang, X. (2010). Fast single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353.
- [33] Hertz, A., Fogel, S., Hanocka, R., Giryes, R., and Cohen-Or, D. (2019). Blind visual motif removal from a single image.
- [34] Horgan, J., Hughes, C., McDonald, J., and Yogamani, S. (2015). Vision-based driver assistance systems: Survey, taxonomy and advances. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE.
- [35] Hou, Y., Ma, Z., Liu, C., and Loy, C. C. (2019). Learning lightweight lane detection cnns by self attention distillation.
- [36] Hu, J., Xiong, S., Zha, J., and Fu, C. (2020). Lane detection and trajectory tracking control of autonomous vehicle based on model predictive control. *International Journal of Automotive Technology*, 21(2):285–295.
- [37] Hu, X., Fu, C.-W., Zhu, L., and Heng, P.-A. (2019). Depth-attentional features for single-image rain removal.
- [38] Huang, S. e. a. (2019). Efficient sliding window search for lane detection. *Pattern Recognition Letters*, 116:77–85.
- [39] Ikeuchi, K., editor (2014). *Computer Vision*. Springer US.
- [40] Kalman, R. E. (1960). *A New Approach to Linear Filtering and Prediction Problems*, volume 82. Journal of Basic Engineering.
- [41] Kim, Z. et al. (2008). Robust lane detection and tracking in challenging scenarios. In *IEEE Transactions on Intelligent Transportation Systems*, volume 9, pages 16–26.
- [42] Klette, R. and CITR, A. (2015). Vision-based driver assistance systems. *enpeda..(Environment Perception and Driver Assistance) Project CITR, Auckland, New Zealand*, 5:1–31.
- [43] Lee, C. and Moon, J.-H. (2018). Robust lane detection and tracking for real-time applications. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):4043–4048.
- [44] Li, M., Li, Y., and Jiang, M. (2018). Lane detection based on connection of various feature extraction methods. *Advances in Multimedia*, 2018:1–13.
- [45] Li, R., Cheong, L.-F., and Tan, R. T. (2019). Heavy rain image restoration: Integrating physics model and conditional adversarial learning.
- [46] Li, Y. and Shi, H. (2022). *Advanced Driver Assistance Systems and Autonomous Vehicles: From Fundamentals to Applications*. Springer Nature Singapore.

- [47] Liu, J. (2019). Learning full-reference quality-guided discriminative gradient cues for lane detection based on neural networks. *Journal of Visual Communication and Image Representation*, 65:102675.
- [48] Luna, A. J. A. (2020). Conducción autónoma de un vehículo simulado mediante un modelo de red neuronal convolucional recurrente.
- [49] Ma, C., Luo, D., and Huang, H. (2021). Lane line detection based on improved semantic segmentation in complex road environment. *Sensors & Materials*, 33.
- [50] Munir, F., Azam, S., Jeon, M., Lee, B.-G., and Pedrycz, W. (2020). Ldnet: End-to-end lane marking detection approach using a dynamic vision sensor.
- [51] Muthalagu, R., Bolimera, A., and Kalaichelvi, V. (2020). Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Computers and Electrical Engineering*, 85:106653.
- [52] Orsic, M., Kreso, I., Bevandic, P., and Segvic, S. (2019). In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images.
- [53] Oršić, M., Krešo, I., Bevandić, P., and Šegvić, S. (2019). In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images.
- [54] Palacios, C. A. (2023). Detección de obstáculos y planos durante el desplazamiento vehicular.
- [55] Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2017). Spatial as deep: Spatial cnn for traffic scene understanding.
- [56] Pan, Y., Li, Y., Wang, J., Xu, Y., Zhao, J., and Zhang, H. (2018). Culane: A curriculum learning based lane detection dataset. *arXiv preprint arXiv:1710.10370*.
- [57] Pan, X., Z. L. Y. Y. and Yang, Y. (2018). Spatial as deep: A spatial cnn for traffic scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10.
- [58] Panda, L. and Mohanty, B. P. (2020). Recent developments in lane departure warning system: An analysis. *Ethics and Information Technology (ETIT)*.
- [59] Panev, S., Vicente, F., De la Torre, F., and Prinnet, V. (2018). Road curb detection and localization with monocular forward-view vehicle camera. *IEEE Transactions on Intelligent Transportation Systems*, 20(9):3568–3584.
- [60] Perng, J. W., Hsu, Y. W., Yang, Y. Z., Chen, C. Y., and Yin, T. K. (2020). Development of an embedded road boundary detection system based on deep learning. *Image and Vision Computing*, 100:103935.
- [61] Phillion, J. (2019). Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network.
- [62] Pizer, S. M., Amburn, J., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J., and Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368.

- [63] Rajaei, B., Rajaei, S., and Damavandi, H. (2023). An Analysis of Multi-stage Progressive Image Restoration Network (MPRNet). *Image Processing On Line*, 13:140–152. <https://doi.org/10.5201/ipol.2023.446>.
- [64] Ren, D., Zuo, W., Hu, Q., Zhu, P., and Meng, D. (2019a). Progressive image deraining networks: A better and simpler baseline.
- [65] Ren, D., Zuo, W., Hu, Q., Zhu, P., Meng, D., and Wang, L. (2019b). Single image deraining using reinforcement adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3823–3832.
- [66] Ren, K., Hou, H., Li, S., and Yue, T. (2021). LaneDraw: Cascaded lane and its bifurcation detection with nested fusion. *Science China Technological Sciences*, 64(6):1238–1249.
- [67] Rendón, C. A. A. (2020). Sistema embebido para asistencia de conducción basado en lógica difusa tipo-2.
- [68] Sanin, A., Sanderson, C., and Lovell, B. C. (2012). Shadow detection: A survey and comparative evaluation of recent methods. *Pattern Recognition*, 45(4):1684–1695.
- [69] Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Academic Press.
- [70] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.
- [71] Sobel, I. (1990). An isotropic 3x3 image gradient operator. *Machine Vision for Three-Dimensional Scenes*, pages 376–379.
- [72] Son, Y., Lee, E. S., and Kum, D. (2018). Robust multi-lane detection and tracking using adaptive threshold and lane classification. *Machine Vision and Applications*, 30(1):111–124.
- [73] Suddamalla, U., Kundu, S., Farkade, S., and Das, A. (2015). A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 87–92. IEEE.
- [74] Tang, J., Li, S., and Liu, P. (2021). A review of lane detection methods based on deep learning. *Pattern Recognition*, 111:107623.
- [75] Valdepeña, R. F. J. (2022). Navegación de un robot omnidireccional basada en lógica difusa tipo-2.
- [76] Wang, H., Wang, Y., Zhao, X., Wang, G., Huang, H., and Zhang, J. (2019a). Lane detection of curving road for structural highway with straight-curve model on vision. *IEEE Transactions on Vehicular Technology*, 68(6):5321–5330.
- [77] Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., and Lau, R. W. (2019b). Spatial attentive single-image deraining with a high quality real rain dataset.
- [78] Wang, X., Kiwus, C., Wu, C., Hu, B., Huang, K., and Knoll, A. (2018). Implementing and parallelizing real-time lane detection on heterogeneous platforms. pages 1–8.

- [79] Wang, Y. and Wang, S. (2006). Fast de-noising algorithm based on gaussian filter. *Computational Science–ICCS 2006*, pages 1260–1263.
- [80] Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., and Li, H. (2022). Uformer: A general u-shaped transformer for image restoration. pages 17683–17693.
- [81] Wei, W., Meng, D., Zhao, Q., Xu, Z., and Wu, Y. (2019). Semi-supervised transfer learning for image rain removal.
- [82] Wu, C.-B., Wang, L.-H., and Wang, K.-C. (2019a). Ultra-low complexity block-based lane detection and departure warning system. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):582–593.
- [83] Wu, C.-B., Wang, L.-H., and Wang, K.-C. (2019b). Ultra-low complexity block-based lane detection and departure warning system. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):582–593.
- [84] Xing, Y., Lv, C., Wang, H., Cao, D., and Velenis, E. (2018). Dynamic integration and online evaluation of vision-based lane detection algorithms. *IET Intelligent Transport Systems*, 13(1):55–62.
- [85] Xing, Y., Lv, C., Wang, H., Cao, D., and Velenis, E. (2019). Dynamic integration and online evaluation of vision-based lane detection algorithms. *IET Intelligent Transport Systems*, 13(1):55–62.
- [86] Xiong, L., Deng, Z., Zhang, P., and Fu, Z. (2018). A 3d estimation of structural road surface based on lane-line information. *IFAC-PapersOnLine*, 51(31):778–783. 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018.
- [87] Yang, W., Liu, J., and Feng, J. (2019). Frame-consistent recurrent video deraining with dual-level flow.
- [88] Yang, W., Tan, R. T., Feng, J., Liu, J., Guo, Z., and Yan, S. (2017). Deep joint rain detection and removal from a single image.
- [89] Yingping Huang, Yangwei Li, X. H. W. C. (2018). Lane detection based on inverse perspective transformation and kalman filter. *KSII Transactions on Internet and Information Systems*, 12(2).
- [90] Yu, B. and Jain, A. (1997). Lane boundary detection using a multiresolution hough transform. In *Proceedings of International Conference on Image Processing*, volume 2, pages 748–751 vol.2.
- [91] Yusuf, M. M., Karim, T., and Saif, A. F. M. S. (2020a). A robust method for lane detection under adverse weather and illumination conditions using convolutional neural network.
- [92] Yusuf, M. M., Karim, T., and Saif, A. F. M. S. (2020b). A robust method for lane detection under adverse weather and illumination conditions using convolutional neural network.
- [93] Zakaria, N. J., Shapiai, M. I., Ghani, R. A., Yassin, M. N. M., Ibrahim, M. Z., and Wahid, N. (2023). Lane detection in autonomous vehicles: A systematic review. *IEEE Access*, 11:3729–3765.

- 
- [94] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. (2022). Restormer: Efficient transformer for high-resolution image restoration. pages 5728–5739.
- [95] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., and Shao, L. (2021). Multi-stage progressive image restoration. pages 14821–14831.
- [96] Zhang, H. and Patel, V. M. (2018). Density-aware single image de-raining using a multi-stream dense network.
- [97] Zhang, H. e. a. (2020). Parabolic lane detection using a fitting method. *IEEE Transactions on Intelligent Transportation Systems*, 21:1834–1843.
- [98] Zhang, R., Wu, Y., Gou, W., and Chen, J. (2021). RS-lane: A robust lane detection method based on ResNeSt and self-attention distillation for challenging traffic situations. *Journal of Advanced Transportation*, 2021:1–12.
- [99] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., and Wang, Q. (2020). Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Transactions on Vehicular Technology*, 69(1):41–54.

# Apéndice A

## Producción académica

Se presentó un poster de divulgación científica presentado en el 1er. Coloquio Internacional de Investigación, Tecnología y Sustentabilidad (Figura A.1).



Figura A.1 Constancia de la presentación del póster.

Se realizó la publicación del artículo "*Lane Detection Using Computer Vision and Convolutional Neural Networks for Autonomous Vehicles*" en la revista IJCOPI International Journal of Combinatorial Optimization Problems and Informatics (Figura A.2). La Figura A.2 muestra la primera página del artículo publicado.



www.editada.org

*International Journal of Combinatorial Optimization Problems and Informatics.*

## Lane Detection Using Computer Vision and Convolutional Neural Networks for Autonomous Vehicles

*Sergio Álvarez Silva, Dante Mújica Vargas, Andrés Antonio Arenas Muñiz*

Departamento de Ciencias de la Computación, Centro Nacional de Investigación y Desarrollo Tecnológico - CENIDET/TecNM, Cuernavaca Morelos 62594, México.  
m23ce001@cenidet.tecnm.mx, dante.mv@cenidet.tecnm.mx, m22ce043@cenidet.tecnm.mx

**Abstract.** This article presents an analysis of computer vision algorithms for Lane Maintenance Assistants (LMA), comparing traditional methods with Convolutional Neural Networks (CNNs). The objective is to evaluate their effectiveness under diverse driving conditions using recognized databases and testing in both real and simulated environments. A proprietary database containing scenarios from the state of Morelos was also used. Experiments covered adverse conditions, such as rain (light, moderate, heavy), solar glare, road shadows, curves, and night driving with/without artificial lighting. Fog simulations included uniform, heterogeneous, cloudy, and combined types. Results showed traditional methods, such as Sobel + Adaptive Thresholding (SA) and Canny + Hough (CH), perform well in normal conditions, achieving a high Intersection over Union (IoU) of 0.9729 and Precision of 0.9888 in Experiment 1. However, their performance declined in complex scenarios like sharp curves and degraded lane markings, with IoU dropping to 0.4838 and Precision to 0.4925 in Experiment 4. Conversely, CNN-based algorithms, like SCNN and VGG16, demonstrated greater adaptability and accuracy in challenging environments. In Experiment 5, VGG16 achieved a Recall of 0.9796 and a Dice Index of 0.6575, showing superior lane detection in dynamic conditions. Although traditional methods and CNNs showed comparable IoU and Precision, the higher Recall in CNNs indicates better robustness, crucial for real-world applications. This study highlights the advantages of deep learning in road safety, reinforcing CNNs particularly VGG16 as a preferred option for lane detection in autonomous vehicles.  
**Keywords:** ADAS, computer vision, lane detection, lane-keeping assistance.

### Article Info

*Received Dec 10, 2024*

*Accepted Feb 18, 2025*

## 1 Introduction

Humans make decisions constantly throughout the day, many of which are based on our perceptions of our surroundings. In driving, visual perception is essential to ensure safe and efficient driving. Autonomous vehicles require the ability to detect objects in their environment and react appropriately, which poses several challenges related to computer vision [1]. Among these challenges are the identification of traffic signs, traffic lights, pedestrians and, perhaps most importantly, lane lines. The latter play a crucial role in vehicle motion planning, as they help determine the path to follow and avoid unintended deviations [2]. Motion planning is a key technology in autonomous vehicles, but it faces significant challenges, especially when it comes to accurate lane detection. This process begins with lane identification and tracking along the vehicle's route, followed by detection of obstacles, signs, and other road features that can influence driving. This research focuses on evaluating the most effective methods for lane detection, dividing them into two main categories: traditional computer vision-based approaches and advanced deep learning techniques, such as convolutional neural networks (CNNs). In-vehicle perception systems are central to addressing these challenges. These systems are composed of cameras and sensors strategically placed in the vehicle, whose function is to capture, analyze and process information about the environment. This information is essential for real-time decision making and has a direct impact on driving safety and comfort. Advanced driver assistance systems (ADAS), such as Lane Keeping Assist (LKA) and automatic emergency braking, rely on this capability to interpret road conditions and help the driver avoid accidents. The integration of data from various sensors, such as LiDAR, cameras and radar, enables a more complete view of the

# Apéndice B

## Retribución social

Se presentó un poster de divulgación científica presentado en el 1er. Coloquio Internacional de Investigación, Tecnología y Sustentabilidad (Figura B.1).



Figura B.1 Constancia de la presentación del póster.