DESARROLLO DE UN APLICATIVO EMBEBIDO DE ALERTA TEMPRANA VISUAL Y AUDITIVA DE SALIDA DE CARRIL

JUAN MARTIN DÁVILA HERRERA

1842208

FACULTAD DE INGENIERÍA

UNIVERSIDAD DEL VALLE

2025

DESARROLLO DE UN APLICATIVO EMBEBIDO DE ALERTA TEMPRANA VISUAL Y AUDITIVA DE SALIDA DE CARRIL

JUAN MARTIN DÁVILA HERRERA

TRABAJO PRESENTADO PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO

EVAL BLADIMIR BACCA, PhD.

DIRECTOR

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

UNIVERSIDAD DEL VALLE

2025

Universidad del Valle

PSI
Percepción y Sistemas Inteligentes

Nota de aceptación:

_____
_____
_____
_____
_____
_____
_____

_____
Firma del director del trabajo

_____
Firma del evaluador

_____
Firma del evaluador

Santiago de Cali, 23 de mayo de 2025

DEDICATORIA

Dedico este trabajo a mi familia, quienes brindaron su apoyo y paciencia incondicional durante todo el proceso de este trabajo. Su amor, comprensión y entrega me han dado mucha motivación para no rendirme. También dedico este trabajo a mi compañera de mil aventuras, mi mascota Manuela, que lastimosamente no logró acompañarme hasta el final, pero su compañía inigualable jamás terminaré de agradecerla.

AGRADECIMIENTOS

Agradezco a todas las personas que hicieron parte de este recorrido de mi vida y me guiaron y/o acompañaron en el proceso de culminar este proyecto entre ellos familiares, profesores, personal administrativo, amigos y compañeros.

Agradezco especialmente a mi familia quienes nunca me dieron la espalda, me acompañaron en todos los altos y bajos, este trabajo también les pertenece a ustedes.

Expreso mi más sincero agradecimiento al profesor Eval Bladimir Bacca Cortés por su paciencia, comprensión, atención, dedicación y guía durante el desarrollo de este trabajo.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En este trabajo se propone el desarrollo de un aplicativo embebido de alerta temprana visual y auditiva ante una salida involuntaria de carril, enfocado en sistemas ADAS integrados. Dando respuesta a la pregunta: ¿Cómo desarrollar un aplicativo embebido que brinde alertas visuales y auditivas ante una salida de carril?, se realizó una revisión teórica de antecedentes académicos sobre técnicas de procesamiento digital de imágenes, métodos de estimación de carriles y algoritmos de segmentación, destacando el uso de la transformada de perspectiva, el filtro Sobel y el espacio de color HSL.

A partir de esta base teórica, se diseñó el sistema empleando la metodología RUP, definiendo requerimientos funcionales como la detección continua del carril, la generación automática de alertas, y la configuración personalizada por parte del usuario, junto a requerimientos no funcionales como el bajo consumo de recursos y el funcionamiento en tiempo real. El aplicativo fue estructurado en cuatro módulos: configuración, preprocesamiento, procesamiento y alertas, todos implementados sobre una plataforma NVIDIA Jetson Nano utilizando bibliotecas optimizadas para CUDA y una interfaz desarrollada en Qt5.

Se llevaron a cabo pruebas de integración, cuantitativas y de campo. En las pruebas cuantitativas realizadas en entornos reales del Valle del Cauca, ae obtuvieron precisiones consistentes en la detección de carriles y en la generación de alertas tempranas, alcanzando valores entre 65% y 75% en la etapa de detección y entre 70% y 80% en la emisión de alertas, dependiendo de las condiciones de la vía. Los mejores resultados se observaron en tramos con buena marcación vial, mientras que en escenarios más exigentes, como curvas o señalización deteriorada, la precisión disminuyó, aunque se mantuvo dentro de márgenes aceptables para una primera implementación embebida. Estos resultados validan la eficacia del sistema en contextos propios de la infraestructura vial colombiana.

Finalmente, se discuten limitaciones y se proponen mejoras futuras, resaltando el potencial del aplicativo como solución embebida accesible para aumentar la seguridad vial en vehículos que no cuentan con tecnologías ADAS de fábrica.

Palabras clave: ADAS, LDWS, Carriles, Seguridad Vial, Visión Artificial.

ABSTRACT

This work proposes the development of an embedded early warning application for unintentional lane departure, focused on integrated ADAS systems. Addressing the question: How to develop an embedded application that provides visual and auditory alerts in the event of lane departure? A theoretical review of academic literature was conducted on digital image processing techniques, lane estimation methods, and segmentation algorithms, highlighting the use of perspective transformation, Sobel filter, and HSL color space.

Based on this theoretical foundation, the system was designed using the RUP methodology, defining functional requirements such as continuous lane detection, automatic alert generation, and user-customizable configuration, along with non-functional requirements like low resource consumption and real-time operation. The application was structured into four modules: configuration, preprocessing, processing, and alerts, all implemented on an NVIDIA Jetson Nano platform using CUDA-optimized libraries and a Qt5-based interface.

Integration, quantitative, and field tests were conducted. In the quantitative evaluations carried out under real-world conditions in Valle del Cauca, the system achieved consistent accuracies in lane detection and early warning generation, with values ranging between 65% and 75% for the detection stage and 70% to 80% for the alert stage, depending on road conditions. The highest performance was observed on segments with well-defined lane markings, whereas in more challenging scenarios—such as curves or deteriorated signage—accuracy decreased but remained within acceptable margins for an initial embedded implementation. These results confirm the effectiveness of the system within the specific context of Colombian road infrastructure.

Finally, limitations are discussed, and future improvements are proposed, highlighting the potential of the application as an embedded solution to enhance road safety in vehicles lacking factory-installed ADAS technologies.

Keywords: ADAS, LDWS, Lanes, Road Safety, Computer Vision.

Capítulo 1 INTRODUCCIÓN

1.1. Introducción

En las últimas décadas, dado el crecimiento demográfico y los avances tecnológicos, se ha presentado un aumento exponencial en la compra de vehículos automotores y colateralmente también se ha dado un aumento en las cifras de los siniestros viales [20]. Para solventar esto, se han implementado múltiples soluciones para la seguridad vial con elementos pasivos los cuales se encargan de mitigar los daños causados al conductor en el momento del accidente tales como bolsas de aire, refuerzos estructurales, entre otros; y con elementos activos los cuales buscan anticiparse a los accidentes de tránsito con sistemas de prevención tales como sistema de alerta de punto ciego, sistema de alerta temprana de salida de carril, sistema de control crucero adaptativo, entre otros.

Los sistemas avanzados de asistencia a la conducción (ADAS) utilizan técnicas de visión artificial, procesamiento digital de imágenes [5], aprendizaje profundo [10],

aprendizaje de máquina [11], entre otros, para poner en funcionamiento los sistemas de seguridad activa. Bajo estas circunstancias, el objetivo de este trabajo de investigación consiste en la búsqueda de métodos de extracción y selección de características de imágenes para la detección del carril y así realizar una realimentación donde se compare la ubicación de los carriles respecto al vehículo que está circulando para generar una alerta visual y/o auditiva al conductor en caso de que esté saliendo del carril.

Figura 1.1.1. Imagen de referencia de métodos de extracción. Fuente: [5]

Para realizar una selección detallada y buscando la opción más adecuada, se hace una búsqueda exhaustiva de antecedentes relacionados con la detección de carriles (ver Figura 1.1 como ejemplo), sistemas de alerta temprana de salida de carril y métodos de reconocimiento de imágenes. En el marco referencial, se presenta un conglomerado resumido de los conceptos generales del procesamiento digital de imágenes lo cual permite el desarrollo de los sistemas avanzados de asistencia a la conducción (ADAS) para así tener una visión amplia para desarrollar un aplicativo embebido de alerta temprana de salida de carril.

Así mismo, se desarrollan cada uno de los objetivos y estrategias que dan paso a resolver el problema de este trabajo de investigación. Para esto, se define el planteamiento del problema junto con los objetivos generales y específicos para desarrollar un aplicativo embebido de alerta temprana de salida de carril haciendo uso de procesamiento digital de imágenes y métodos de detección de carriles con su respectiva justificación. Para completar satisfactoriamente los objetivos propuestos, se establece una metodología siguiendo los lineamientos de RUP para diseñar el funcionamiento del aplicativo para posteriormente implementar de manera satisfactoria.

1.2. Planteamiento del problema

Los accidentes de tránsito son una de las principales causas de muerte de personas en Colombia, y suponen una disminución notable de la esperanza de vida en un cuarto y dos años de vida en hombres y mujeres respectivamente [2] . El 89% de los accidentes de tránsito ocurren por fallas humanas (distracciones, exceso de confianza, entre otras) y tan solo un 11% ocurren por factores externos (fallas mecánicas, geográficas, climáticas, entre otras) [3]. En la mayor parte de los casos de accidentes de tránsito producidos por una falla humana, se presenta una salida involuntaria del carril por el cual se está circulando, terminando en una colisión contra un muro, vehículo, entre otros [4].

Para solucionar esta problemática, la compañía Iteris desarrolló “Lane Departure Warning System” que advierte al conductor de una salida involuntaria de carril mediante bocinas ubicadas a cada lado del vehículo, generando un sonido dependiendo del costado por el cual se dé esta salida siempre y cuando el conductor no haya hecho uso del intermitente antes de abandonar el carril. En el año 2000, comenzó la producción en masa de este sistema, inicialmente implementado en los camiones Mercedes-Benz Actros de manera opcional. Y en el 2002, la compañía Freightliner también comenzó a hacer uso de este sistema de forma opcional [5]. En la actualidad, se ofrece una amplia variedad de vehículos de uso particular y de transporte de carga que incluyen un sistema de alerta de salida de carril en su paquete de seguridad activa o de ADAS. Pero normalmente pertenecen a vehículos de gama media en adelante, lo cual es poco accesible a gran parte de la población, por tanto, se convierte en una asistencia de seguridad que posee cierto grado de exclusividad [6]. En el año 2019, se realizó un estudio del procesamiento de imágenes haciendo uso de múltiples datasets y diseñando un algoritmo que permita lograr robustez para el reconocimiento del carril y la posición del vehículo con respecto a este, así como reducir el tiempo de este proceso [1].

Por otra parte, los ADAS no suelen venderse como accesorio adicional en los vehículos actuales. Únicamente se ofrecen al mercado como equipamiento original de la marca (OEM) lo cual restringe el acceso a estas tecnologías enfocadas en favorecer la seguridad activa de los ocupantes de los vehículos que transitan por las carreteras.

Bajo estas circunstancias en las cuales los ADAS presentan un restringido acceso al público, así como la falta de normativas en Colombia y los altos precios de los vehículos comparado con los ingresos promedio de los colombianos, es casi inexistente la presencia de estas asistencias en el mercado del país lo cual conlleva en un atraso tecnológico importante en materia de seguridad vial.

En este proyecto se busca diseñar una solución embebida que pueda ser instalada la mayoría de los vehículos tradicionales que circulan sin este tipo de ayuda. Considerando lo anterior, se plantea la siguiente pregunta: **¿Cómo desarrollar un aplicativo embebido de alerta temprana de invasión de carril que proporcione alertas visuales y auditivas al conductor?**

El sistema propuesto haría uso de una cámara que realice un constante monitoreo de la ubicación del vehículo con respecto al carril en entornos extraurbanos, y mediante una pantalla se genere una alerta visual al piloto en caso de que tenga una tendencia de invasión de carril, o haciendo uso de alertas auditivas en caso de una invasión de carril que implique un alto riesgo para el piloto y los ocupantes del vehículo.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un aplicativo embebido de alerta temprana visual y auditiva de salida de carril usando una cámara y visión por computador.

1.3.2. Objetivos específicos

I. Sintetizar información bibliográfica con relación al preprocesamiento de imágenes, métodos de estimación de mantenimiento de carril y la realimentación para obtener los requerimientos funcionales del sistema.

II. Diseñar el aplicativo embebido de alerta temprana visual y auditiva de salida de carril.

III. Implementar el aplicativo embebido de alerta temprana visual y auditiva de salida de carril.

IV. Realizar pruebas de integración, cuantitativas y de campo para validar los requerimientos del aplicativo.

1.4. Solución planteada

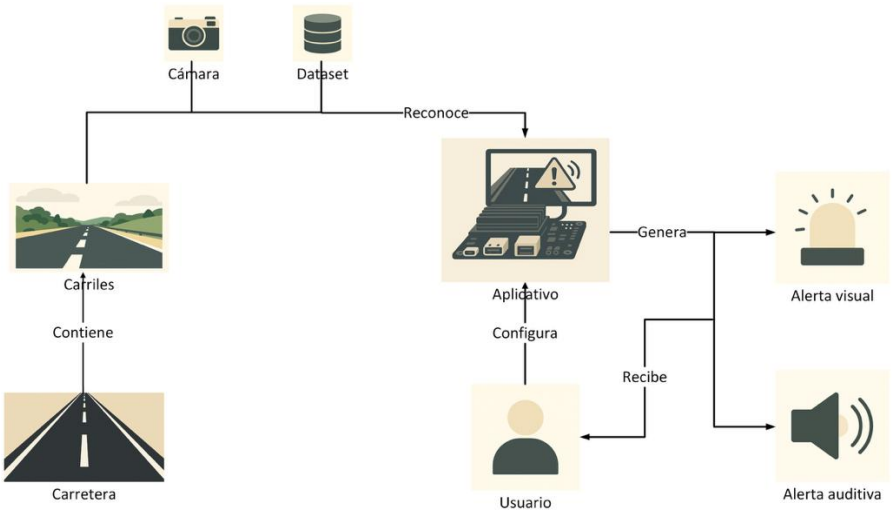


Figura 1.2. Diagrama conceptual del aplicativo.

En la Figura 1.2 se muestra el diagrama conceptual de la solución planteada del aplicativo donde se tienen los siguientes elementos principales:

- **Cámara:** Este dispositivo captura las imágenes en tiempo real del entorno donde se encuentra la carretera y los carriles que delimitan la misma por donde circula el vehículo.
- **Dataset:** Para realizar pruebas preliminares del correcto funcionamiento del aplicativo, se debe hacer uso de imágenes y/o videos precargados para garantizar el correcto funcionamiento del aplicativo.
- **Aplicativo:** Este software procesa la información recibida por la cámara y/o dataset. Haciendo uso de algoritmos de preprocesamiento de imágenes, se detectan los límites de las marcas de los carriles en las imágenes adquiridas para así obtener la ubicación espacial del vehículo respecto al carril.
- **Alertas generadas:** Son las notificaciones que se le proporcionan al conductor en caso de una posible salida de carril. La alerta visual corresponde a una luz intermitente en algún lugar visible del vehículo, mientras que la alerta auditiva es un sonido que informa al conductor.
- **Usuario:** Es el conductor del vehículo que obtiene las alertas generadas por el aplicativo para así tomar una decisión en la trayectoria del vehículo.

1.5. Estructura del Documento

En este documento, se presenta el desarrollo del aplicativo embebido de alerta temprana de salida de carril. Para ello se dispondrá de la siguiente estructura:

CAPÍTULO 1 – INTRODUCCIÓN: En este capítulo se abordan elementos generales introductorios al tema de investigación, así como una evaluación al panorama. Se presenta el planteamiento del problema el cual entrega una visión preliminar del porqué desarrollar el aplicativo, también se dan a conocer los objetivos del proyecto y la solución planteada representada en un diagrama conceptual.

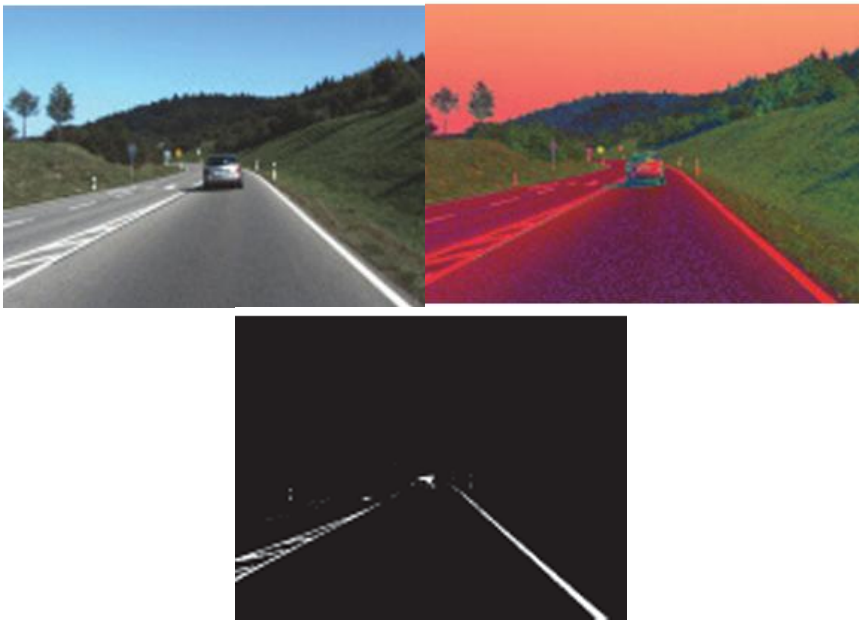
CAPÍTULO 2 – MARCO REFERENCIAL Y TEÓRICO: En este capítulo se contextualizan los trabajos previos relacionados nacional e internacionalmente de acuerdo con un conjunto de criterios presentados en una tabla comparativa para realizar un proceso de selección de las estrategias más adecuadas para la construcción del aplicativo. Del mismo modo, se presenta un contexto a los conceptos clave requeridos para la construcción y comprensión del proyecto.

CAPÍTULO 3 – DISEÑO E IMPLEMENTACIÓN: En este capítulo se exponen los elementos que se tuvieron en cuenta para el diseño del aplicativo haciendo uso de la metodología RUP. Del mismo modo, se presenta el proceso de implementación del aplicativo por módulos teniendo en cuenta los criterios de diseño mostrados anteriormente.

CAPÍTULO 4 – PRUEBAS Y RESULTADOS: En este capítulo se evidencian los resultados de las pruebas de campo e integración realizados con el aplicativo.

CAPÍTULO 5 – CONCLUSIONES Y TRABAJOS FUTUROS: En este capítulo se presentan las conclusiones del trabajo realizado, así como las aspiraciones de los trabajos futuros que puedan realizarse con base al presente.

Capítulo 2 MARCO REFERENCIAL Y TEÓRICO



2.1. Introducción

Este capítulo tiene como objetivo proporcionar una visión general de los antecedentes y el marco teórico para el desarrollo de un sistema de alerta temprana de salida de carril. Se realiza una revisión exhaustiva de los métodos de estimación de mantenimiento de carril utilizados en la actualidad, centrándose en la visión artificial. Además, se analizan en detalle los algoritmos de pre-procesamiento, realimentación y las tecnologías adicionales empleadas para reconocer la segmentación de los carriles y estimar el nivel de salida de este. Se destaca la transformada de perspectiva como la más adecuada para el reconocimiento de carriles. Todo esto sienta las bases teóricas para el desarrollo de la aplicación propuesta.

2.2. Antecedentes

Tabla 2.1. Antecedentes categorizados en los criterios principales.

Referencia	Métodos de pre-procesamiento de la imagen	Método de estimación para el mantenimiento de carril	Tipo de realimentación usada	Sensores usados	Precisión
[1]	Extracción ROI y pirámide Gaussiana	Algoritmo ED y EDLines	Comparación matemática	Datasets	99.25%
[7]	Escala de grises y segmentación por el método OTSU	Diseño de DROI e interpolación	Comparación matemática basado en coordenadas polares	Caltech datasets	99.21%
[8]	Escala de grises y algoritmo EDLines	Eliminación angular y estimación del punto de desvanecimiento	Comparación matemática mediante ángulos	Caltech datasets	98.80%
[9]	NA	Redes neuronales CNN, RNN y Encoder-Decoder	Comparación convolucional haciendo pooling y padding	tuSimple datasets	98.52%
[10]	Extracción ROI binario y filtrado RGB	Filtro de diferenciación horizontal y transformada de Hough	Comparación matemática mediante ángulos	Cámara de video en tiempo real	NA
[11]	Extracción de ROI bajo métodos de estimación gráfico	Redes neuronales CNN y búsqueda Breadth-first	Comparación matemática polinomial	Caltech y tuSimple datasets	99.87%
[12]	Extracción ROI	Algoritmos de visión artificial y machine learning	Comparación matemática	Cámara de video en tiempo real	91.70%
[13]	Escala de grises y ecualización del histograma	Transformada de Hough y ROI	Comparación angular	Cámara de video en tiempo real	NA
[14]	Escala de grises y difuminación de la imagen	Redes neuronales CNN y RANSAC	Comparación matemática	Datasets	94.70%
[15]	Extracción ROI y asignación de color blanco y negro mediante umbral estadístico	Aproximación a función cuadrática mediante la transformada de Hough	Comparación matemática bajo el método de gradientes	Video previamente capturado en una cámara	NA
[16]	Escala de grises	Método CRAL	Comparación matemática mediante los ángulos de la cámara	Cámara FL	N/A

[17]	Extracción ROI y conversión de color HLS	Transformada de perspectiva	Comparación matemática	KITTI dataset	84%
[18]	Extracción ROI y filtros de imagen	Clasificación en secuencias de imágenes	Comparación matemática entre las distintas imágenes	Imágenes capturadas desde un dispositivo móvil	N/A
[19]	Extracción de características de bajo perfil	Transformada de distancia	Comparación matemática	Caltech datasets	97.20%
[20]	Extracción de ROI tras aplicar filtro de color HSV y filtro Gaussiano	Transformada de Hough	Comparación matemática	Video previamente capturado en una cámara	94.70%
[21]	EDLines adaptativo	Filtro Gabor	Comparación matemática usando gradientes	Datasets e imágenes capturadas	95.24%
[22]	Escala de grises, pirámide Gaussiana y DROI	Transformada de Hough	Comparación matemática entre ángulos y posición en el plano	Cámara monocular	90.09%
[23]	N/A	Redes neuronales	Selección de características globales basado en filas	Caltech y tuSimple datasets	96.06%

En la Tabla 2.1. se exponen 15 antecedentes de detección de carriles clasificados por los siguientes criterios:

- **Método de pre-procesamiento de la imagen:** De acuerdo con el procedimiento de acondicionamiento de la imagen para optimizar el uso de recursos, incrementar la precisión y remover la información innecesaria.
- **Método de estimación del carril:** De acuerdo con el algoritmo utilizado para reconocer la ubicación de los carriles y estimar la posición de estos.
- **Tipo de realimentación usada:** De acuerdo con la estrategia utilizada para determinar la posición del vehículo con respecto al carril para generar las alertas.
- **Sensores:** De acuerdo con el tipo de información utilizado en las pruebas funcionales.
- **Precisión:** De acuerdo con el margen de error obtenido en las pruebas funcionales.

2.2.1. Antecedentes Nacionales

En la Universidad Pedagógica y Tecnológica de Colombia [12], se estudió un sistema de asistencia a la conducción mediante el uso de visión artificial y machine learning para identificar señales de tránsito, la posición del vehículo respecto al carril y obstáculos que se presenten en la carretera.

Haciendo uso de regiones de interés (ROI), se aplicaron procesos de segmentación y detección de bordes donde se selecciona la región de la imagen que contiene la mayor cantidad de información de las delimitaciones del carril y se aplica un filtrado de bordes donde se elimina todo el contenido irrelevante de la imagen. Posteriormente, se estima mediante cálculos matemáticos la posición del vehículo con respecto al carril para así estimar si es necesario generar una alerta.

Este sistema de asistencia fue implementado sobre un vehículo en tiempo real donde se realizó un viaje de 43 km a velocidades inferiores a 70 km/h donde se realizaba un conteo de éxitos cada vez que el sistema trabajaba y detectaba correctamente cada una de las variables puestas y medidas manualmente por los investigadores.

Mediante algoritmos de visión computacional, de bajo consumo de recursos, se logró satisfactoriamente la implementación del sistema de detección y alertas de salida de carril el cual posee ventajas mediante este método frente a las redes neuronales artificiales dado que es inmune a las alteraciones por ángulos y perturbaciones en la posición angular del vehículo aparte de lograr una tasa de refresco de 22 cuadros por segundo (FPS) lo cual permite una alta estabilidad en tiempo real y un aceptable tiempo de reacción.

2.2.2. Antecedentes internacionales

En [1] se propone un algoritmo para la detección de la salida involuntaria de carril haciendo énfasis en la robustez y que trabaje en tiempo real. El algoritmo propuesto se divide en cinco etapas. En primer lugar, se trabaja el pre-procesamiento de las imágenes donde se extraen las regiones de interés (ROI) y haciendo uso de la aproximación de pirámide Gaussiana, se logra un suavizado de la imagen para así disminuir los detalles innecesarios de la misma y reducir el tiempo de procesamiento; en segundo lugar, se hace uso del algoritmo de detección de bordes (ED) el cual trabaja con la imagen en escala de grises y se degradan para así detectar los pixeles de gradiente máximo (anchors) y determinan los bordes. Con estos bordes, mediante el algoritmo EDLines con el método de mínimos cuadrados (LS) se ajustan puntos a un determinado sistema de coordenadas que determina la ubicación de los segmentos de la línea. En tercer lugar, se trabaja la etapa de filtrado y agrupamiento de las líneas halladas con el anterior algoritmo EDLines donde dichos segmentos se filtren y se seleccionan solo los que están relacionados con los límites de los carriles y posteriormente se agrupan de forma matemática para detectarlos como dos líneas paralelas con sus respectivas divisiones e intersecciones; en cuarto lugar para el seguimiento de la línea, se mantiene en constante actualización la lista de seguimiento y se correlacionan con las líneas agrupadas en tiempo real para así detectar si concuerdan. Por último, se realiza una comparación matemática entre la ubicación de las líneas y la posición estimada de la cámara para así detectar el nivel de salida del carril.

Haciendo uso de este algoritmo, se consiguió implementar un sistema de alerta temprana de salida de carril con un tiempo promedio de procesamiento de cuadro de 12.5 milisegundos y una precisión de un 99.25%.

Por otro lado, en [7] se establece un algoritmo basado en similaridad de regiones dinámicas de interés (DROI) la cual busca mejorar la precisión de detección del carril. Para el preprocesamiento de la imagen se convierten las imágenes a escala de grises para reducir el tiempo de procesamiento, posteriormente se resalta la información principal mediante la ecualización por histograma y finalmente se realiza el segmentado de los bordes mediante el método OSTU. Para obtener unas mejores regiones de interés (ROI) se debe de tener un campo de visión seguro adaptativo y mediante este se seleccionan las DROI. La ubicación espacial en coordenadas polares de cada punto del borde de la calzada y la distancia máxima de seguridad previamente definida, se resuelven mediante cálculos matemáticos de la línea del carril y los datos de velocidad del vehículo extraídos del cuadro inmediatamente anterior para así definir la precisión del DROI en el momento actual. Después de esto, los puntos discontinuos hallados son procesados mediante la interpolación lineal para las líneas con curvatura ligera e interpolación cuadrática para complementar los datos en caso de una curvatura amplia. Para el seguimiento de la línea se aplica el método de predicción Grey que permite tener un buen rendimiento y no requiere un alto volumen de datos para realizar la predicción de líneas. Por último, se estiman las coordenadas de las líneas mediante el sistema polar para así tener una mayor precisión a la hora de reconocer las líneas rectas y curvas. Mediante el algoritmo mencionado, se verificó la detección de las líneas haciendo uso de datasets Caltech donde se tuvo una tasa de precisión del 99.21% y se probó con 5683 fotogramas para comprobar la robustez.

También en [8], se presenta un algoritmo de detección de líneas basado en video el cual hace uso de un método rápido de estimación de punto de desvanecimiento buscando la eficiencia y la precisión en tiempo real. En principio, se hace uso del algoritmo EDLines para la extracción de los segmentos de la imagen dividiendo por colores ambas líneas del carril. Posterior a esto, se realiza un proceso de eliminación angular la cual consiste en seleccionar dos puntos reflejados para definir los posibles ángulos sobre los cuales se encuentra la línea. Con esta información, se busca la convergencia de ambas líneas hacia el punto de desvanecimiento realizando extrapolación para lograr que sea completamente visible para así estimar la posición completa de las líneas y realizar una predicción. En último lugar, se realiza una comparación matemática mediante ángulos para verificar la posición del vehículo respecto al carril. Bajo este método, se obtuvo un tiempo de respuesta por fotograma aproximado de 12 milisegundos y se verificaron 1225 cuadros obteniendo una precisión del 98.80%.

Así mismo, en [9] se busca la flexibilidad ante condiciones adversas tales como sombras, degradación de la marcación, entre otras haciendo uso de una arquitectura profunda que combina redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN). En principio, se busca hacer uso de varias imágenes solapando varios fotogramas de tal forma que se pueda realizar una predicción precisa de la línea. RNN es un mecanismo adaptativo para las labores de detección y predicción de los carriles dado su potencial en el procesamiento continuo de señales y la extracción de las características secuenciales. Por otro lado, CNN se destaca por el procesamiento de imágenes pesadas por lo cual puede reducirse el tamaño de la imagen para mejorar el rendimiento haciendo uso de operaciones recursivas de convolución y pooling. También se hace uso de una red convolucional Encoder-Decoder la cual se encarga de detectar las líneas de forma semántica. Por último, se entrenan las redes mediante métodos adaptativos de estimación para así buscar la mayor precisión y robustez. Con este método, se logró una mayor flexibilidad a la hora de detectar los carriles, así como un tiempo de respuesta de 6.7 milisegundos con requerimientos de hardware superiores a los anteriores obteniendo una precisión del 98.52%.

Igualmente, en [10] se propone un algoritmo basado en discontinuidad donde se busca garantizar la estabilidad en tiempo real. Para el preprocesamiento de la imagen, se hace uso de ROI para mejorar el desempeño, después se usa de un filtrado RGB para intensificar los colores de la imagen y reconocer más fácilmente los colores amarillo y blanco para así obtener una imagen binaria con línea gruesas. Para la detección de los bordes, se utiliza un filtro de diferenciación horizontal [-1 0 1] el cual se convoluciona con la imagen binaria obtenida anteriormente. Finalmente, hace uso de la transformada de Hough para así detectar líneas continuas y segmentadas. Bajo este algoritmo, se consigue una mejora en el consumo de recursos obteniendo un tiempo de computación de 30ms.

En el documento [11] se presenta un método que permite detectar múltiples líneas sin sacrificar en gran parte el rendimiento. Inicialmente, se hace un recorte de la imagen mediante la extracción de ROI y se detectan los casos de los bordes de carril segmentadas mediante métodos basados en gráficos. En segundo lugar, se hace uso de una aproximación basada en CNN para detectar y seleccionar los segmentos de línea para después realizar una búsqueda Breadth-first para identificar los casos mencionados

y agregarlos a la red de segmentación de carril. Por último, mediante la vista de pájaro se realiza una comparación matemática polinomial de la posición del carril. Este método permitió aumentar la cantidad de carriles procesados obteniendo una velocidad de procesamiento de cuadro aproximada de 18,33 milisegundos, una tasa de cuadros por segundo de 54,535 y precisión del 99.87%.

En [13] se expone una técnica de detección de carril haciendo uso de la transformada de Hough en tramos rectos mientras que en los tramos curvos se utiliza el algoritmo de seguimiento. Se comienza con la etapa de preprocesamiento de la imagen convirtiendo la imagen a escala de grises y ecualizando el histograma para suprimir los valores innecesarios de la imagen. Posteriormente en la etapa de detección de bordes haciendo uso del operador de Sobel obteniendo así una imagen únicamente con los bordes más relevantes de la imagen. Después se hace uso de la transformada de Hough la cual detecta objetos con una figura específica, en este caso línea recta detectando así el carril. Para mejorar la respuesta en tiempo real, se combina la técnica de extracción de regiones de interés (ROI) para solo detectar los segmentos blancos y amarillos, así como la técnica del método rápido de punto de desvanecimiento para tener mayor estabilidad en el seguimiento. Este método permitió una buena robustez respecto a los cambios de iluminación, pero una baja estabilidad a la hora de tener obstáculos en frente.

Así mismo, en [14] se muestra un método robusto de detección de carriles haciendo uso de redes neuronales convolucionales (CNN) combinada con el algoritmo de consenso de muestras aleatorias (RANSAC) donde se busca mejorar el rendimiento y minimizar lo máximo posible las líneas de ruido. Inicialmente, se convierte la imagen a escala de grises y se difumina la imagen para eliminar el ruido ambiental y tener información con mayor fidelidad y así realizar la detección de bordes mediante los puntos blancos de la imagen. Para mejorar la precisión del preprocesamiento, se realiza una convolución entre los bordes detectados y un núcleo forma de sombrero. Después, se realiza una extracción de regiones de interés (ROI) para aligerar la carga computacional para así entregar un conjunto de puntos de datos para que el algoritmo RANSAC detecte el carril. De igual forma, se hace uso de CNN para hacer más robusta la detección del carril en condiciones no tan favorables tales como la detección de más de dos líneas, el desplazamiento del carril entre fotogramas es muy grande o el punto de desvanecimiento entre ambos carriles es impreciso. Bajo este conjunto de algoritmos basados en redes neuronales, se obtuvieron resultados favorables y una buena robustez ante condiciones no favorables a la hora de detectar carriles con una precisión del 94.70%.

En el texto [15] se define un método basado en la información de color donde se busca la aplicación en entornos complejos y que requiera de poca potencia computacional. En primer lugar, se realiza una extracción de regiones de interés (ROI) para encontrar un umbral sobre el cual se puede clasificar los puntos blancos de la imagen. Posteriormente, se hace uso de la transformada de Hough para obtener una aproximación mediante una función cuadrática a la geometría del carril y el uso de mínimos cuadrados para la optimización de los recursos utilizados. Haciendo uso de este algoritmo, se obtuvo un alto grado de robustez frente a sombras, tipo de pavimento, reflejo del sol, entre otros y un bajo consumo a nivel de recursos.

Por otro lado, en [16] se propone un método de estimación de carril haciendo uso de una cámara de visión frontal a bordo se calculan las distancias mediante los ángulos obtenidos por la visión periférica de la cámara teniendo en cuenta el alcance de esta para así reconocer la desviación del vehículo con respecto a las marcaciones de los carriles en la carretera. Al ser un algoritmo experimental, solo se analizan algunos alcances que podría ofrecer, pero careciendo de pruebas que comprueben el funcionamiento de este.

En [17] se propone un método de detección de líneas fundamentado en técnicas tradicionales de visión artificial para reconocer los carriles enfocado en la resistencia a cambios bruscos de iluminación. En la etapa de preprocesamiento de la imagen, se inicia con la extracción de la región de interés (ROI) para eliminar información innecesaria, posteriormente se aplica un filtro de color HSL para extraer los elementos más destacables en cuestión de iluminación y saturación de la imagen, a la misma vez que un filtro de Sobel para lograr una umbralización combinada. Posterior a esto, se aplicó la transformada de perspectiva para convertir la imagen en un espacio tridimensional a uno bidimensional para tener mucha más facilidad a la hora de reconocer las líneas. Por último, se hace uso de búsqueda por ventanas deslizantes que utiliza la información de los centros de las líneas para rastrear y detectar las líneas del carril desde la parte inferior a la superior de la imagen.

Este algoritmo mostró buen comportamiento y fue consistente en las condiciones en las que se realizaron las pruebas obteniendo así una precisión del 84%.

En [18] se plantea un método de detección de líneas basado en las técnicas de clasificación de secuencias de imágenes para así detectar los carriles priorizando la robustez y la precisión del sistema. En la etapa de preprocesamiento de la imagen, se aplican técnicas para mejorar la legibilidad incluyendo la corrección de la distorsión, el equilibrio de contraste y la reducción de ruido para posteriormente aplicar la técnica de regiones de interés (ROI).

El sistema propuesto demostró alta precisión y adaptabilidad frente a condiciones adversas tales como variaciones en la iluminación y cambios repentinos en el patrón de las marcaciones de la carretera con un tiempo de respuesta de 1.930ms el cual es lo suficientemente rápido como para ofrecer una buena respuesta en tiempo real.

Así mismo, en el trabajo [19] se presenta una técnica que se fundamenta en la aplicación de la transformada de la distancia buscando mejorar la fiabilidad de la detección de carriles ante condiciones adversas donde se presenten ambientes variados al hacer uso de una metodología de regresión adaptativa.

En esta técnica se implementa un preprocesamiento de imagen de bajo nivel el cual está dividido en 3 pasos. El primero consiste en modificar la imagen dada por la cámara teniendo en cuenta la posición de esta para simular una vista superior del camino para así obtener un polígono de 4 puntos. Posteriormente, se implementa la detección de bordes separando blanco de negro y eliminando el ruido del mismo polígono para detectar y diferenciar las líneas que demarcan el carril. Por último, se aplica la transformada de la distancia donde se encuentran ambos bordes del carril.

En las pruebas realizadas basadas en datasets Caltech, se obtuvo una precisión del 97.2% y un tiempo de respuesta promedio de 63 ms el cual permite aplicación en tiempo real.

En [20] se presenta un método de detección de líneas que integra varios de los métodos previamente mencionados tales como la extracción de ROI, transformación de color a HSV y la transformada de Hough. Inicialmente, se aplica el preprocesamiento de la imagen donde se convierte la codificación de color de RGB a HSV buscando disminuir el ruido y la sensibilidad a los cambios de iluminación de la imagen para así ofrecer mayor robustez y posteriormente aplicar las técnicas de preprocesamiento para llegar a la imagen binaria. Para la detección de bordes, se aplica un filtro Gaussiano a la imagen para suavizar la imagen y realizar la extracción de bordes bajo el método Canny y después extraer las regiones de interés (ROI). Por último, se realiza la detección de carril haciendo uso de la transformada de Hough.

Bajo esta metodología, se obtuvo mayor robustez al combinar apropiadamente varias técnicas permitiendo una mayor fiabilidad frente a cambios de iluminación o cambios inesperados en la geometría del carril ofreciendo una precisión del 94.70% haciendo uso de un video pregrabado.

De igual forma, en [21] se presenta un modelo para mejorar algunas de las metodologías presentadas anteriormente. Inicialmente se presenta un modelo de preprocesamiento de imagen donde se inicia con un proceso de detección de texturas el cual consiste en la aplicación de un filtro Gabor para distinguir las líneas de otros agentes que estén en la carretera. En segundo lugar, se tiene un método de detección de bordes adaptativo el cual funciona de manera similar al presentado anteriormente añadiendo la capacidad de ajustar los valores umbrales de los bordes de acuerdo con las condiciones lumínicas detectadas con el filtro Gabor usado anteriormente. Para finalizar, se implementa un algoritmo de selección de candidatos que busca la mejor opción de línea derecha e izquierda usando los bordes previamente procesados y los selecciona aplicando un análisis de gradientes.

Posteriormente, se utilizan los candidatos de línea hallados anteriormente para así calcular la desviación del vehículo respecto al carril haciendo uso de la distancia Euclidiana y el ángulo respecto a las líneas. Se realizaron pruebas con datasets e imágenes y se obtuvieron resultados del 95.24% de efectividad y logrando una tasa de cuadros de 20fps lo cual es válido para el sistema en tiempo real.

Además, en [22] se muestra un algoritmo simple que se enfoca en el bajo uso de recursos y presentar robustez a cambios en la iluminación. Inicialmente, se realiza un preprocesamiento de la imagen convirtiendo la imagen a escala de grises para aplicar un filtro Gaussiano que elimina el ruido e información innecesaria y se binariza la imagen mediante el método de umbralización Otsu de tal forma que solo se obtienen las líneas de interés para así realizar una extracción DROI. Posteriormente, se aplica una transformada de Hough para hallar los carriles y la distancia de ellos para obtener el nivel de salida de carril.

Por último, en [24] se muestra un algoritmo basado en redes neuronales haciendo uso de selección de filas con características globales de la imagen, en lugar de la segmentación píxel por píxel. Además, incluyen una pérdida estructural que permite modelar explícitamente la estructura de los carriles aumentando la efectividad. Esto permite un mejor desempeño y precisión en condiciones desafiantes.

Analizando la información presente en las referencias mostradas anteriormente, se puede concluir que un sistema de alerta temprana de salida de carril tiene la posibilidad de ser diseñado e implementado bajo métodos muy diferentes entre sí, pero sosteniendo varios puntos en común los cuales se pueden dividir en tres categorías principales: preprocesamiento de la imagen, métodos de estimación del carril y la detección de la salida respecto al carril. En el preprocesamiento de imagen, hay muchas similitudes entre los métodos y se puede observar que ROI permite flexibilidad y baja potencia de cálculo lo cual es un factor clave para sistemas embebidos. Para los métodos de estimación de carril, se presentan distintas metodologías y algoritmos basados en varias expresiones matemáticas que permiten tener una ubicación espacial de forma teórica. Para detección de salida de carril se usa generalmente la distancia Euclidiana y el ángulo de desviación, pero también existen otros medios útiles.

2.2.3. Selección del método de preprocesamiento de la imagen

De acuerdo con la tabla 2.1, se encontró que en [1], [10], [11], [12], [15], [18], [20] y [22] hacen uso de la extracción ROI para eliminar las regiones que no son de interés en el aplicativo, en este caso, el paisaje que está por encima de la carretera. El punto de desvanecimiento para la selección de la ROI es el punto de conexión entre las líneas y/o el punto de finalización de ambas líneas, posteriormente, se elimina todo el contenido que se encuentre arriba de este punto de desvanecimiento.

En [22] se realiza un cambio de espacio de color desde RGB al HLS que permite distorsionar la imagen para destacar únicamente las líneas más gruesas como son los carriles omitiendo otras que no son de interés como pueden ser las líneas de los árboles, vehículos, montañas, etc. lo cual reduce la capacidad de cómputo y omite posibles fuentes de error.

Tal como se observa en [7], y en [22], el método de umbralización OTSU realiza una binarización de la imagen de tal forma que solo se utiliza una matriz de color negro y blanco de tal forma que las líneas del carril son claras y se pueden trabajar de una forma más sencilla y precisa pero es muy dependiente de la calidad de la imagen y presenta limitaciones en condiciones de variabilidad lumínica. Así mismo, en [20], y en [21] se hace uso del detector de bordes Canny el cual se utilizó para identificar los bordes de manera robusta y reduciendo el ruido, sin embargo presenta alta complejidad computacional restringiendo el uso en aplicaciones en tiempo real y sistemas embebidos. En [13], y en [17] se presenta el filtro Sobel el cual permite resaltar áreas con cambios significativos de intensidad arrojando indicativos de existencia de bordes. Este último método presenta un equilibrio entre robustez y simplicidad a nivel computacional así como un comportamiento interactivo en sus valores umbrales para entornos dinámicos donde las condiciones de iluminación y del pavimento varían constantemente.

Estos fueron los métodos que se escogieron para el preprocesamiento de imágenes dado que presentan características similares y al ser utilizados en conjunto se obtiene una excelente priorización de los elementos más importantes de la imagen de tal forma que se obtienen únicamente los valores esenciales de la imagen que corresponden a la ubicación de las líneas del carril.

2.2.4. Selección del método de estimación para el mantenimiento de carril

De acuerdo con la tabla mostrada previamente, se encontró que en [19] se hace uso de la transformada de perspectiva la cual consiste en modificar la alineación de la imagen de tal forma que se cambia la perspectiva de una vista frontal a una vista aérea de la carretera con los parámetros obtenidos para así hallar las líneas de los carriles.

En [8], [10], [11] y [13] se utilizan métodos basados en redes neuronales CNN, RNN y RANSAC de tal forma que se entrena el modelo para que aprenda las características más relevantes de la imagen y descarte elementos que no son de interés. De esta forma, el modelo predice la presencia y la ubicación de las líneas del carril en las imágenes posteriores y mejorar la precisión de éste.

Se escogió el método basado en la transformada de perspectiva mostrado [10], en [11], y en [17], que permite obtener las líneas de los carriles y la comparación entre ellas de tal forma que se puede obtener la información esencial para la estimación de la salida del carril.

Dado que la transformada de perspectiva presenta vulnerabilidad frente a los cambios de iluminación y/o cambios de carril, en [9] se muestra una modificación a la transformada de perspectiva donde se realiza un seguimiento de las líneas restringiendo en rango de (src, dst) basado en el fotograma inicial de tal forma que se ofrece una mayor robustez frente a los cambios de iluminación y evitar falsos positivos.

2.2.5. Selección del tipo de realimentación

En [17] se puede observar que se emplean los valores de distancia y ángulo hallados con la transformada de perspectiva para estimar la salida del carril del vehículo tanto en la trayectoria como en la posición. Al comparar los valores de los carriles detectados, se define el ancho total del carril en la base de la imagen así como el punto intermedio del vehículo siendo este el punto central del ancho de la imagen.

En [5] se emplean los valores utilizados anteriormente para hallar un valor porcentual de la salida del carril haciendo uso de la ecuación 2.1.:

$$Desviación\ (d) = \frac{L - C}{W/2}$$

Ecuación 2.1. Valor porcentual de la desviación respecto al carril.

Donde L corresponde al punto intermedio de la línea, C corresponde al punto intermedio del vehículo y W corresponde al ancho del carril.

El valor de *(d)* inicia desde el -100% (salida completa por el límite derecho del carril) hasta el 100% (salida completa por el límite derecho del carril). Para obtener una respuesta

válida de la salida del carril se definieron los valores clasificados en regiones tal como se muestra en la Tabla 2.2.:

Región	Valores de desviación
Región segura	-40% a 40%
Región de precaución	-60% a -40% 40% a 60%
Región de peligro	<-60% y >60%

Tabla 2.1. Clasificación de los valores de desviación.

2.3. Marco teórico

En este apartado se definen conceptos fundamentales presentes en este trabajo de grado, que permiten abordar el contexto de la investigación de una mejor manera. Las definiciones abarcan tanto aspectos de la estructura vial, como del área de la ingeniería.

2.3.1. Sistemas Avanzados de Ayuda a la Conducción

Los sistemas avanzados de ayuda a la conducción son sistemas inteligentes que están presentes en el interior del vehículo y asisten al conductor en múltiples formas. Estos sistemas también pueden ser empleados para analizar la fatiga y la falta de concentración del conductor humano, lo que les permite emitir advertencias preventivas o evaluar el desempeño al volante y brindar recomendaciones en consecuencia. Estos sistemas pueden asumir el control de ciertas funciones del conductor, como la evaluación de posibles peligros, la ejecución de tareas simples como mantener una velocidad constante (usando el control de crucero) o realizar maniobras más difíciles como adelantar y estacionar. Esto facilita el intercambio de información para mejorar la percepción, ubicación, planificación y toma de decisiones por parte de los vehículos [15].

2.3.2. Niveles de Automatización.

De acuerdo con la Sociedad de Ingenieros Automotrices (SAE), los vehículos pueden clasificarse según su nivel de automatización bajo los siguientes parámetros: (1) Dirección, aceleración y deceleración, (2) Supervisión del entorno, (3) Rendimiento de retroceso de la tarea de conducción dinámica [18].

- **Nivel 0:** Corresponde a los vehículos tradicionales donde no existe ningún tipo de ayuda a la conducción y ésta depende completamente del conductor humano.
- **Nivel 1:** Permite brindar una asistencia específica que implica la dirección o la aceleración/deceleración mientras que el resto de las tareas deben de ser realizadas por el conductor humano.
- **Nivel 2:** En escenarios específicos, se presenta una automatización parcial donde es posible una asistencia al conductor para la dirección y aceleración/deceleración de forma simultánea.

- **Nivel 3:** Permite realizar toda la conducción automatizada bajo todas las tareas dinámicas, pero se espera que el conductor recupere el control cuando sea necesario tras la intervención.
- **Nivel 4:** Permite tomar todo el control de la conducción incluso sin la intervención humana siempre y cuando éste no intervenga tras el envío de la solicitud.
- **Nivel 5:** Es capaz de gestionar todas las situaciones haciendo innecesaria la intervención del conductor humano

2.3.3. Procesamiento Digital de Imágenes.

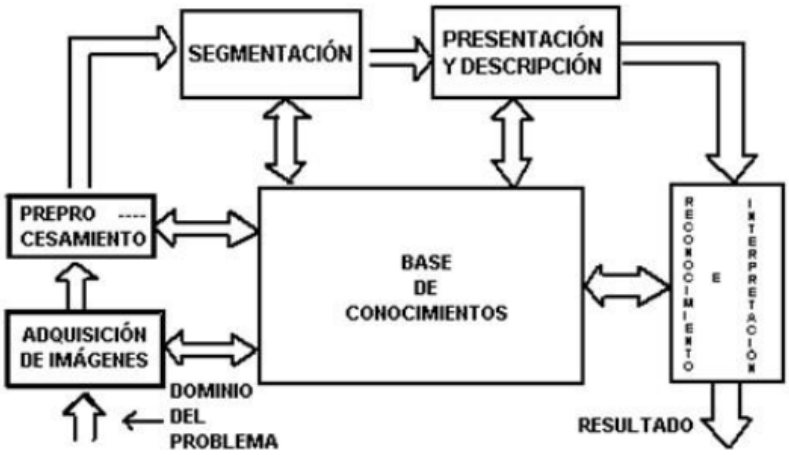


Figura 2.1. Etapas del procesamiento de imágenes.

De acuerdo con [23], una imagen digital es una función bidimensional discretizada la cual puede ser vista como una matriz donde el valor de cada celda corresponde a un nivel específico de gris y la ubicación espacial de ésta pertenece a un punto específico de la imagen.

En la figura anterior, se puede observar un diagrama de bloques donde se exponen las etapas del procesamiento digital de imágenes los cuales son:

- **Adquisición de la imagen:** Consiste en la obtención de la imagen digital haciendo uso de dispositivos como cámaras digitales.
- **Preprocesamiento:** Es un conjunto de técnicas que permiten mejorar la apariencia visual de la imagen para enriquecer su estética o adecuarla para un procesamiento posterior de forma que facilite su análisis.
- **Segmentación:** Es la división de la imagen en varias secciones para procesarlas por separado.
- **Representación y descripción:** Los segmentos obtenidos anteriormente se adecúan para su procesamiento y se definen las secciones de interés.
- **Reconocimiento e interpretación**: Se clasifican de acuerdo con la necesidad.

2.3.4. Métodos de Preprocesamiento de la Imagen.

La consulta bibliográfica sobre los métodos de preprocesamiento de la imagen ha presentado una cantidad significativa de variaciones y métodos de acuerdo con las condiciones y necesidades de cada uno de ellos.

Extracción de Regiones de Interés (ROI): La extracción de Regiones de Interés (Regions Of Interest, "ROI" por sus siglas en inglés) es una técnica en el preprocesamiento digital de imágenes donde se busca enfocar el análisis de la imagen en una porción en específico de la imagen para extraer los elementos más relevantes y descartar la información poco significativa. Para ello, se define un polígono trapezoidal tal como se muestra en la Figura 2.2. que abarca la zona donde se encuentran los carriles, generalmente en la mitad inferior de la imagen. En esta técnica se asume que las líneas de los carriles se unen en un punto específico llamado "punto de fuga" el cual establece el límite superior de la región de interés permitiendo así reducir la carga computacional y evitando hacer uso de información no relevante que pueda generar ruido o respuestas no esperadas.

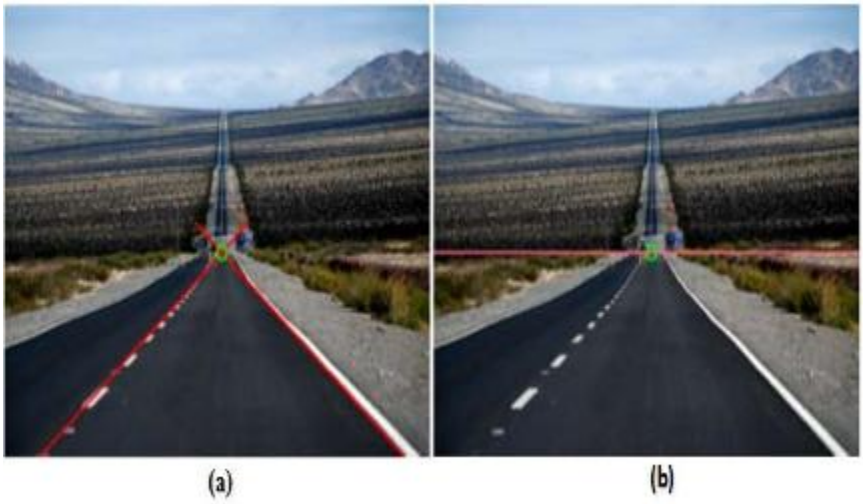


Figura 2.2. (a) Punto de fuga (b) Definición de la ROI. Tomado de [25].

Umbralización: La umbralización es un proceso relevante en el preprocesamiento digital de imágenes donde se busca generar una imagen binaria donde únicamente se destacan los elementos relevantes en un fondo uniforme; en este caso, la demarcación del carril sobre el pavimento permitiendo tener una segmentación precisa y un procesamiento eficiente.

Método de OTSU: El método de OTSU es un algoritmo de umbralización global que determina un valor óptimo basado en la maximización de la varianza entre clases. Para ello, se establece una división entre los píxeles que están por debajo del umbral y aquellos que están por encima. (ver Figura 2.3.)

Figura 2.3. (a) Imagen original (b) Imagen con umbralización OTSU.

Detección de bordes Canny: El detector de bordes Canny es una técnica de preprocesamiento que combina técnicas de suavizado, derivadas y umbralización para la detección de los contornos de la imagen tal como se muestra en la Figura 2.4. Inicialmente, se aplica un filtro Gaussiano para reducir el detalle de la imagen evitando el ruido para luego realizarse el cálculo del gradiente en la imagen. Por último, se establecen dos valores umbrales (superior e inferior) para determinar los pixeles candidatos y eliminar el resto.

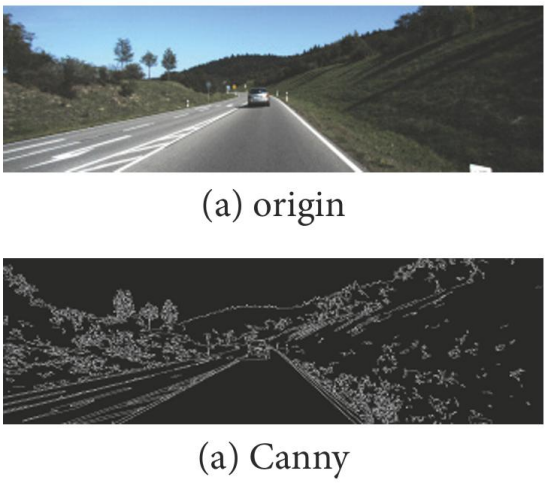


Figura 2.4. Detección de bordes Canny. Tomado de [20].

Filtro Sobel: El filtro Sobel es un método de detección de bordes fundamentado en el cálculo del gradiente de intensidad de la imagen. Para ello, se debe realizar una convolución de la imagen con una máscara que reconoce estos cambios de intensidad en un solo eje, entre el eje x (horizontal) o en el eje y (vertical). (ver Figura 2.5.)

Figura 2.5. (a) Imagen original (b) Filtro Sobel aplicado. Tomado de [13].

Espacio de Color HSL: El espacio de Color HSL (Hue, Saturation, Lightness en inglés) es un modelo de color que asemeja los colores a la forma en que los ve el ojo humano en las siguientes tres componentes:

- **Tono (Hue):** Aquí se define el tono del color en sí visto en forma circular donde en 0° está el rojo, 120° verde, y a 240° está el color azul.
- **Saturación (Saturation):** Aquí se define qué tan vibrante es el color. Se representa en una escala de 0% a 100%, donde 0% corresponde a un tono muerto asemejándose a un gris y 100% al color puro.
- **Luminosidad (Lightness):** Aquí se representa el brillo del color medido en un rango de 0% a 100%, donde 0% corresponde a un tono completamente oscuro asemejándose a negro y 100% corresponde a un tono completamente claro asemejándose al blanco.

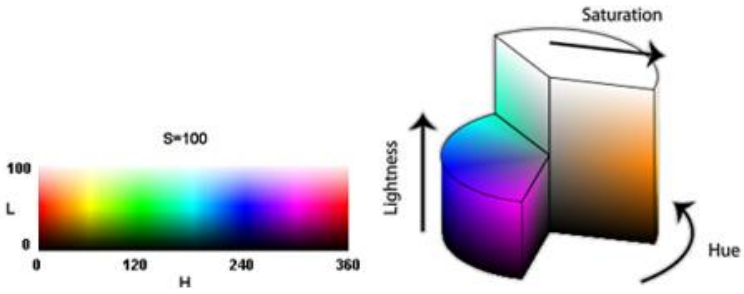


Figura 2.6. Espacio de color HSL. Tomado de [26].

2.3.5. Métodos de Segmentación de Carriles.

La consulta bibliográfica sobre los métodos de segmentación de carriles ha presentado avances importantes, motivado por el incremento en los ADAS y la necesidad de mejorar la seguridad vial.

Segmentación de Carriles Basada en Técnicas Clásicas de Procesamiento de Imágenes

Las técnicas clásicas de procesamiento de imágenes están basadas principalmente en métodos geométricos para transformar la imagen de acuerdo con los requerimientos.

Transformada de Hough: La transformada de Hough presentada en [10], [13], [15], [20] y [22] es una técnica matemática utilizada para identificar líneas en una imagen al transformar puntos en el espacio de la imagen a una representación en el espacio de parámetros. El principio de esta transformación es la conversión de punto a curva, donde se transforma el sistema de coordenadas cartesiano *P(x,y)* a un espacio de coordenadas polar de Hough *P(ρ,Θ)* como se muestra en la Figura 2.7.

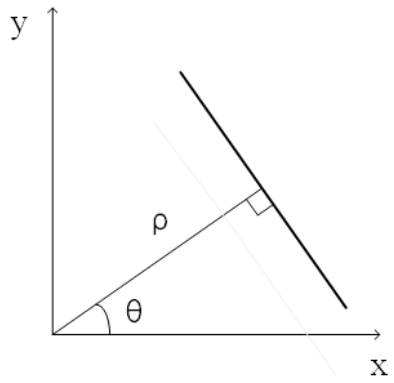


Figura 2.7.Parámetros de la transformada de Hough.

Transformada de Perspectiva: La transformada de perspectiva presentada en [19] permite proyectar una imagen desde una vista tridimensional a una bidimensional facilitando la identificación de los límites del carril mediante la rectificación de la imagen y la reducción de la distorsión de la misma (ver Figura 2.8). Este proceso permite convertir la vista de perspectiva de la carretera en una vista aérea simulada lo que permite tener más robustez frente a escenarios donde la geometría de la carretera cambia constantemente, así como una mejor adaptabilidad a entornos complejos como la infraestructura vial nacional de Colombia.

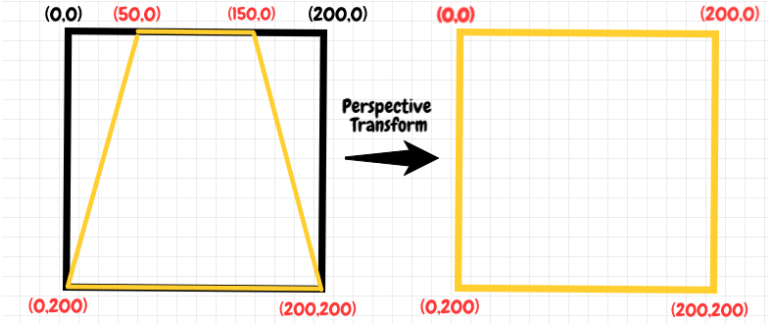


Figura 2.8. Transformada de Perspectiva.

Aprendizaje profundo: El aprendizaje profundo, y en particular las redes neuronales son una categoría moderna en la segmentación de carriles debido a su capacidad para aprender características complejas y abstractas de las imágenes de manera automática.

En [9] se hace uso de las redes neuronales convolucionales (CNN) donde se tiene una especial efectividad para enfrentar escenarios desafiantes en tiempo real. Este enfoque se fundamenta en la capacidad de las redes neuronales para aprender de manera supervisada las características complejas de las imágenes y adaptarse a las variaciones en las condiciones de iluminación y del entorno. (Ver Figura 2.9.)

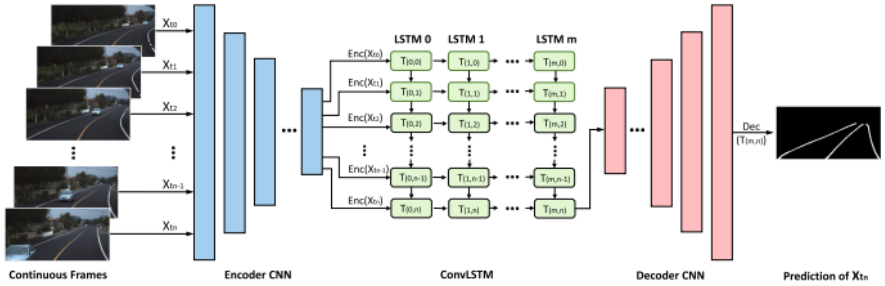


Figura 2.9. Arquitectura de la red neuronal. Tomado de [9]

En [14] también se aplicaron CNN para la mejora de la detección de carriles combinándolas con el método de consenso de muestras aleatorias (RANSAC) para lograr una detección robusta en escenarios desafiantes. La unión de CNN con RANSAC permite filtrar las detecciones erróneas y mejorar la estabilidad del reconocimiento, especialmente en situaciones donde las marcas de los carriles están deterioradas.

2.3.6. NVIDIA Jetson Nano

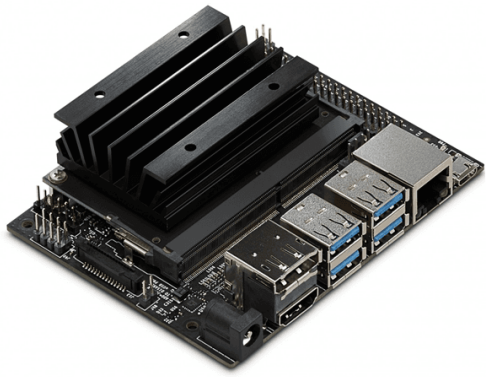


Figura 2.10. NVIDIA Jetson Nano Developer Kit.

La NVIDIA Jetson Nano es una plataforma de hardware embebido enfocada al procesamiento gráfico ideal para aplicaciones de visión artificial y aprendizaje profundo

(ver Figura 2.10). Esta plataforma cuenta con una GPU NVIDIA de arquitectura Maxwell con 128 núcleos CUDA, un CPU de arquitectura ARM Cortex-A57 de cuatro núcleos y 4GB de memoria RAM LPDDR4. Hace uso del sistema operativo Ubuntu 16.04 LTS y soporta bibliotecas como TensorFlow, PyTorch y OpenCV.

2.3.7. Metodología RUP.

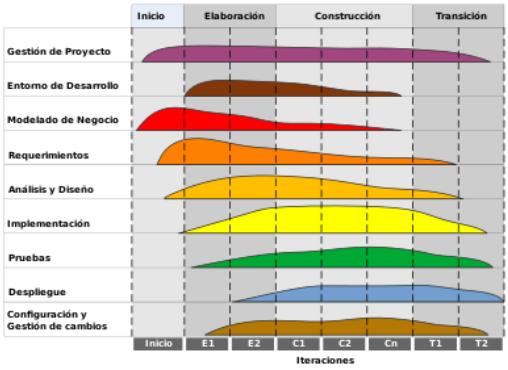


Figura 2.11. Metodología RUP

La metodología Rational Unified Process (RUP) es un proceso de ingeniería de software el cual ofrece un método de asignación de tareas enfocadas al desarrollo de un producto enfocándose en garantizar la mayor calidad posible dentro de un cronograma y presupuesto especificado [27].

En la Figura 2.11, se puede describir la metodología de acuerdo con sus dimensiones:

Dimensión vertical: Corresponde al apartado estático del proyecto en términos de actividades, énfasis o flujos de trabajo.

Dimensión horizontal: Corresponde al apartado dinámico del proyecto en término de tiempo, fases e iteraciones.

2.4. Observaciones finales

Teniendo en cuenta la información mostrada en la recolección de antecedentes condensada en la Tabla 2.1 es relevante afirmar que el método de estimación de carril es un elemento clave para definir qué tecnologías y herramientas utilizar para diseñar un sistema de alerta temprana de salida de carril. La precisión medida en porcentaje permite tener una visión de la efectividad del algoritmo utilizado para validar la relevancia que ofrece en la labor de selección. De acuerdo con estos criterios, se puede realizar un consolidado de los algoritmos más relevantes y así seleccionar una serie de características entre estos para el desarrollo de este proyecto. Dada la acelerada evolución en cuestión de hardware gráfico, varios algoritmos fueron desarrollados en entornos de desarrollo limitados lo cual puede significar un rendimiento superior al

presentado por los autores. Los métodos tradicionales basados en extracción de características y recopilación matemática de los datos deben hacer uso de métodos de preprocesamiento para simplificar y optimizar la carga de recursos mientras que, para los algoritmos basados en redes neuronales, no es necesario métodos de preprocesamiento dado que en el entrenamiento se usan patrones de imágenes similares por lo tanto las regiones de interés se intuyen automáticamente. Es importante mencionar que para el desarrollo de algunos aplicativos implementados se utilizan lenguajes orientados a la programación de objetos tales como C++ o multiparadigma como lo es Python.

En la selección del método de preprocesamiento se destaca principalmente el cambio de espacio de color RGB-HSL para facilitar la segmentación de carriles en condiciones de mucha variabilidad en la iluminación del entorno. Al separar el canal de saturación (S) e iluminación (L), se obtiene una mejor diferenciación de las marcas del carril para mejorar la robustez en situaciones donde las sombras o la luz solar directa afecte la calidad y estado de la imagen en el espacio de color RGB tradicional.

Así mismo, el filtro Sobel fue seleccionado dado que facilita una segmentación precisa de las líneas del carril. A diferencia de OTSU, el filtro Sobel permite un control local sobre la gradiente permitiendo un ajuste más interactivo y preciso para adaptarse a diferentes condiciones. Por otro lado, el filtro Canny aumenta la complejidad a nivel computacional al requerir varios procesos de preprocesamiento en la definición de los valores mínimos y máximos de los umbrales; esto también aumenta la probabilidad de la detección de ruido y elementos no relevantes de la imagen.

Para la selección del método de estimación del carril, se seleccionó la transformada de perspectiva por su capacidad de convertir la vista frontal tridimensional de la carretera en una imagen bidimensional simulando una vista aérea o vista de pájaro facilitando el reconocimiento de los límites del carril y eliminando la distorsión generada por la perspectiva natural de la posición de la cámara obteniendo una vista paralela de las líneas del carril. Respecto con la transformada de Hough, la transformada de perspectiva presenta mejor comportamiento cuando la carretera presenta características especiales como curvatura y líneas discontinuas como es común en la topología e infraestructura vial colombiana. Por su parte, los métodos basados en CNN entregan una alta precisión y robustez al recopilar características complejas en el entrenamiento, pero requieren de un alto poder de cómputo y un tiempo de entrenamiento significativo lo cual restringe el uso en sistemas embebidos como éste, donde se requiere su operación en tiempo real.

De la misma forma, el uso de ventanas deslizantes habilita la identificación de manera efectiva de las posiciones de los carriles en la imagen procesada realizando una subsegmentación de la imagen en diferentes partes para así hallar los pixeles correspondientes a los carriles para garantizar el resultado y reduciendo el margen de error evitando la influencia de elementos no deseados en la imagen.

Es importante resaltar la importancia del uso de la metodología RUP en el diseño del aplicativo para establecer los requerimientos funcionales y no funcionales, así como los casos de uso real y tener una idea más amplia del desarrollo del aplicativo. Si bien en el

marco teórico se describen múltiples técnicas de segmentación y estimación de carriles (Canny, Hough, RANSAC, entre otros), en el desarrollo del sistema se implementaron aquellas que mostraron mayor compatibilidad con la plataforma Jetson Nano y el enfoque en tiempo real: el filtro Sobel, la conversión al espacio de color HSL, la transformada de perspectiva y el método de ventanas deslizantes para ajuste polinomial. Las demás se incluyen únicamente como referencia conceptual.

Capítulo 3 DISEÑO E IMPLEMENTACIÓN DEL APLICATIVO

3.1. Introducción

En este capítulo se expone el procedimiento del desarrollo del aplicativo embebido con el fin de dar solución a la problemática planteada en este trabajo de grado en búsqueda de resultados satisfactorios al reconocer una salida de carril y generar una alerta.

Inicialmente, se realizará el diseño del aplicativo haciendo uso de la metodología RUP evaluando los aspectos clave que componen el desarrollo de esta solución tecnológica. Posteriormente, se realizará la implementación dividida en cuatro módulos: configuración, preprocesamiento, procesamiento y alerta.

3.2. Diseño del aplicativo

En el diseño del aplicativo se aplicó la metodología RUP (Rational Unified Process) [27] para desarrollar el software de manera iterativa priorizando la calidad del aplicativo y simplificando las tareas de implementación para facilitar la comprensión de este.

Mediante el uso de la metodología RUP, se definieron los requerimientos funcionales y no funcionales y de acuerdo con ellos, se define el diagrama de concepto, los casos de uso real, los diagramas de secuencias y el diagrama de clases. Por razones de espacio, los casos de uso real y los diagramas de secuencias hacen parte de los anexos de este documento.

De esta manera, se plantea de manera más eficiente el proyecto y se puede realizar una planeación y entrega del aplicativo de manera más organizada y efectiva.

3.2.1. Requerimientos funcionales y no funcionales

Los requerimientos funcionales establecen las características que el software debe tener para cumplir con los objetivos del usuario y encaminados a resolver cierta problemática. Con esto presente, se proponen los siguientes requerimientos funcionales, y se categorizan en dos vertientes: Esenciales (E) y Opcionales (O). Esto se observa en Tabla 3.1.

Los requerimientos no funcionales son aquellos que definen los atributos del sistema que no están directamente relacionados con las funciones y capacidades del aplicativo, sino con su rendimiento, calidad, seguridad, usabilidad y otros aspectos que no se refieren a las funcionalidades específicas que debe tener el software tal como se observa en la Tabla 3.2.

Ref #	Funciones	Categoría
1.0	La fuente de captura de la imagen debe de ser configurable por el usuario	**E**
2.0	Se deben aplicar técnicas de preprocesamiento de imagen para mejorar la calidad de las imágenes de entrada al sistema	**E**
3.0	El sistema debe reconocer los límites del carril donde el vehículo se desplaza.	**E**
4.0	Definir la ubicación del vehículo respecto al carril por el cual se está circulando	**E**
5.0	Detectar la proximidad del vehículo a los límites preestablecidos	**E**
6.0	El sobrepaso de los límites debe ser configurable por el usuario	**E**
7.0	El aplicativo puede generar diversas alertas dependiendo de los límites preestablecidos y la sensibilidad de éstas	**O**
8.0	El aplicativo debe tener una interfaz gráfica de usuario (GUI) para configurar los parámetros necesarios para el correcto funcionamiento del sistema	**E**

Tabla 3.1. Requerimientos funcionales

Ref #	Descripción	Categoría
1.0	Utilizar el sistema NVIDIA Jetson Nano para el desarrollo e implementación	
2.0	Soportar la version 4.6.5 JetPack (Jetson SDK) que incluye varios paquetes para el correcto funcionamiento del dispositivo	**E**
3.0	Hacer uso de NVIDIA Jetson Linux Driver Package (L4T) versión 32.7.6	**E**
4.0	Soportar Ubuntu 22.04.3 LTS en la máquina host que será donde se desarrolle el aplicativo	**O**
5.0	Hacer uso del paquete OpenCV 4.1.1 que contiene herramientas de visión artificial previamente desarrolladas	**E**
6.0	Soportar Ubuntu 18.04 en el kit de desarrollo NVIDIA Jetson Nano que será donde se implemente el aplicativo	**E**
7.0	Implementar las librerías CUDA versión 10.2 en el desarrollo del aplicativo para incrementar la compatibilidad y optimización del sistema NVIDIA	**E**
8.0	Diseñar la interfaz gráfica de usuario (GUI) con el framework QT 5	**E**

Tabla 3.2. Requerimientos no funcionales

3.2. Diagrama de clases

Con el fin de completar los requerimientos establecidos en la sección anterior, se desarrollaron las clases mostradas en el diagrama de la Figura 3.1.

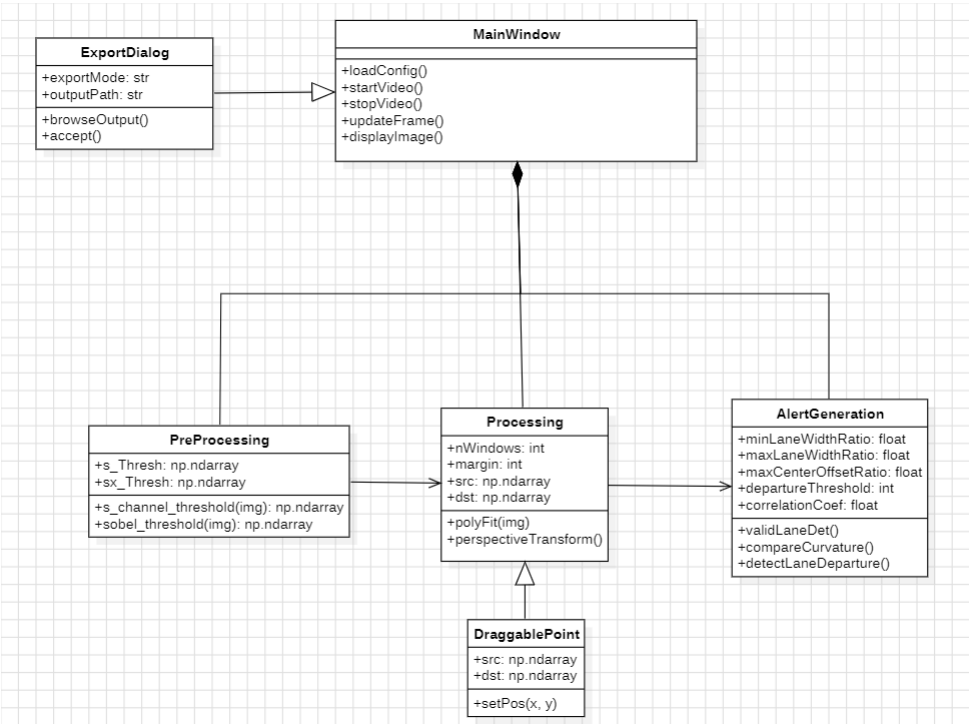


Figura 3.1. Diagrama de clases

- Clase **MainWindow:** Es la clase principal del aplicativo donde se disponen la mayoría de los métodos correspondientes a la interacción del aplicativo con el usuario. Esta se encarga de gestionar la GUI, seleccionar la fuente de adquisición, ejecutar el aplicativo y realizar la asignación de los valores de la configuración.
- Clase **PreProcessing**: En esta clase se condensan los métodos encargados de la etapa de preprocesamiento de imágenes mediante la extracción y filtrado del canal S, así como la detección de bordes horizontales tras la binarización con el filtro de Sobel.
- Clase **Processing**: En esta clase se implementa el pipeline principal de la detección de carriles incluyendo la transformada de perspectiva, el ajuste polinomial y la búsqueda por ventanas deslizantes.
- Clase **AlertGeneration**: Esta clase contiene la lógica de la implementación de la supervisión de validez de carriles, así como la generación de alertas calculando cuando se ha salido de su trayectoria en una detección válida de carril.

- Clase **DraggablePoint**: En esta clase se determinan los puntos editables dentro de la GUI para ajustar la región de interés que toma la transformada de perspectiva.
- Clase **ExportDialog**: En esta clase se implementa la gestión de exportación donde el usuario define el formato de salida entre video o imágenes, definir la ruta de almacenamiento y confirmar la selección.

3.3. Implementación del aplicativo

En esta sección se establecen los módulos que estructuran al aplicativo para mostrar el desarrollo de cada uno de ellos y sintetizar la implementación.

El aplicativo se compone de 4 principales módulos: Configuración, Preprocesamiento, Procesamiento y Alerta.

3.3.1. Módulo de Configuración

El módulo de configuración consiste en la interfaz gráfica de usuario (GUI) diseñada en QT Designer que contiene las configuraciones previas a la ejecución, así como la visualización de la ejecución.

Inicialmente, se tiene la ventana principal dividida en tres pestañas: (1) Ejecución, (2) Configuración y (3) Debug. Tal como se muestra en la Figura 3.2. se inicia con la pestaña de Ejecución por defecto, esta cuenta con la pestaña de visualización (4) donde se muestra el funcionamiento en tiempo real del aplicativo, así como los botones de iniciar (5) y detener (6) el algoritmo. También se hace uso de un botón de chequeo para habilitar o deshabilitar el modo de edición (7).

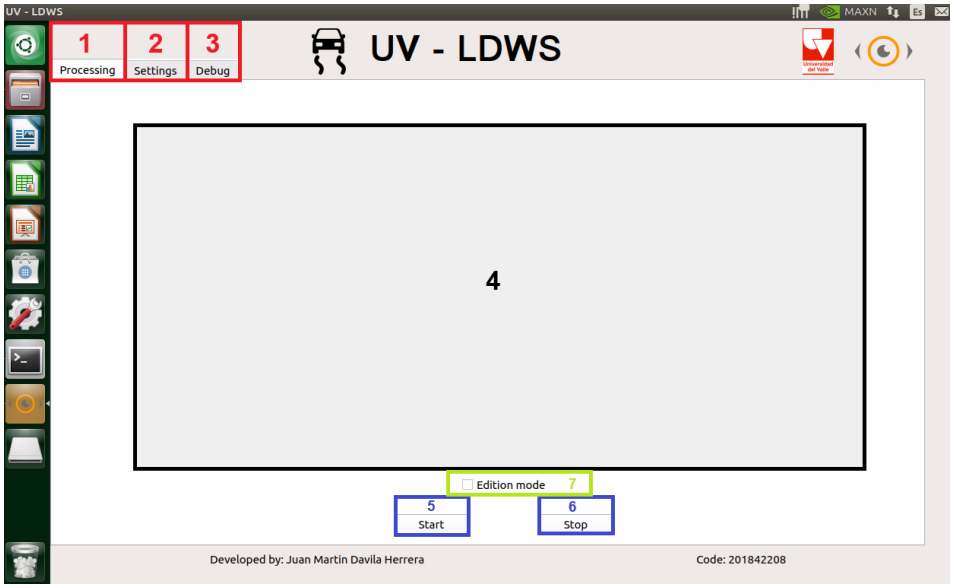


Figura 3.2. Ventana inicial

El modo edición está controlado por un botón de chequeo el cual funciona de la siguiente manera:

- Al habilitarse, se activa un conjunto de cuatro puntos de color rojo posicionados en unos valores por defecto, los cuales son arrastrables por el usuario haciendo uso del puntero para registrar una nueva matriz de origen en la transformada de perspectiva tal como se muestra en la Figura 3.3.
- Al deshabilitarse, se guarda y se asigna la nueva matriz de origen de la transformada de perspectiva.

Este elemento es fundamental para adaptarse a diferentes condiciones, topografías y posicionamiento de la cámara aumentando la robustez y la facilidad de uso para el usuario.

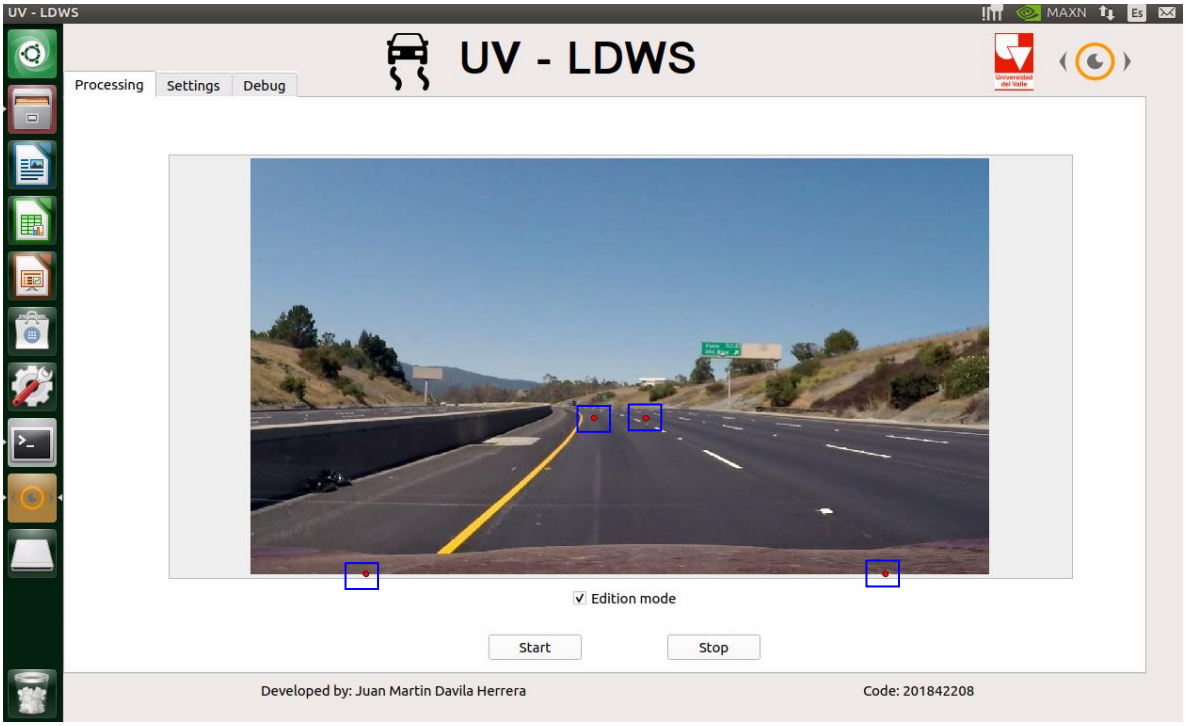


Figura 3.3. Modo edición

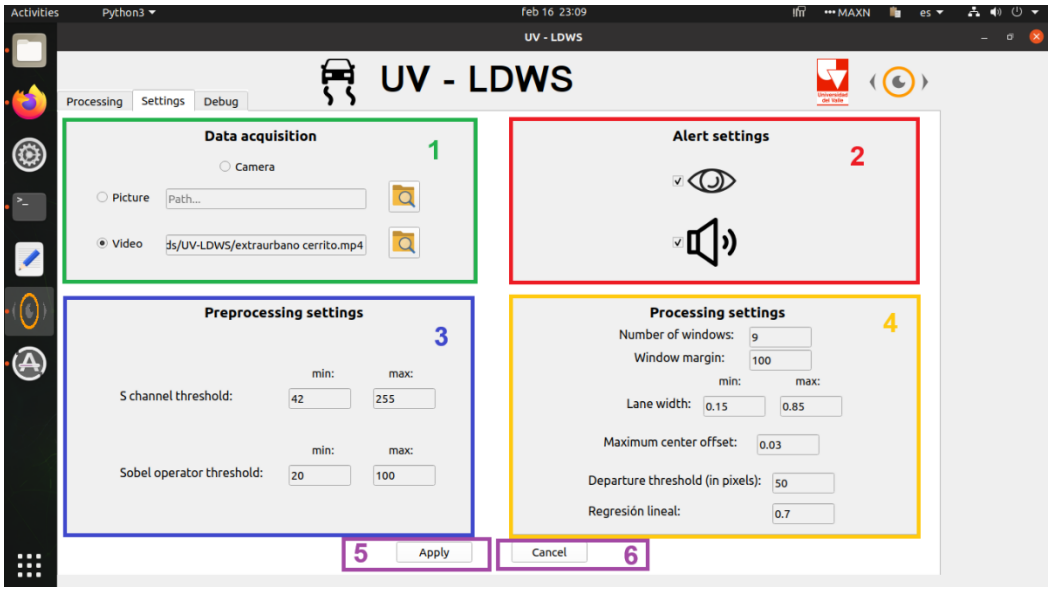


Figura 3.4. Ventana de configuración

En la Figura 3.4. se muestra la pestaña de Configuración dividida en 4 principales secciones: (1) Adquisición de datos, (2) Configuración de alerta, (3) Configuración de preprocesamiento y (4) Configuración de procesamiento. Así como (5) el botón de Aplicar para guardar la configuración y (6) el botón de Cancelar para volver a la ventana inicial.

- **Adquisición de datos:** En la figura 3.5. se muestra esta sección donde se establece la fuente de adquisición de los datos divididos en tres RadioButton para (1) cámara, (2) imagen o (3) video.

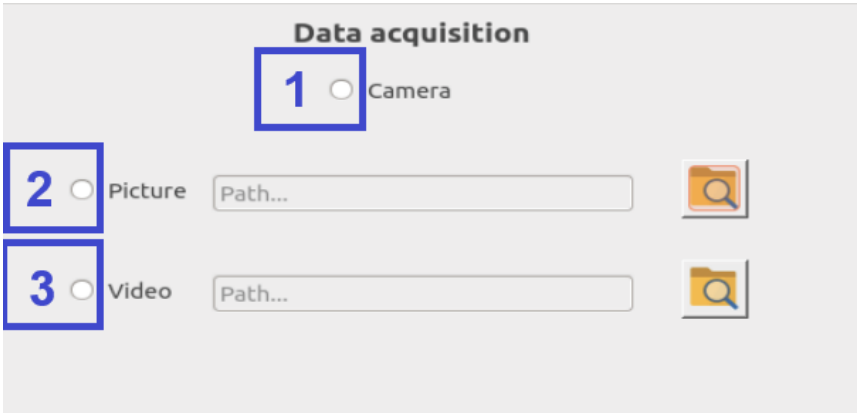


Figura 3.5. Sección de adquisición de datos

Adicionalmente, se tiene un botón para la carga de imagen y video que permite cargar un archivo con una ventana de selección. Posteriormente se carga junto con la ruta del archiva mostrada en el cuadro de texto “Path…” asignado a cada uno tal como se muestra en la Figura 3.6.

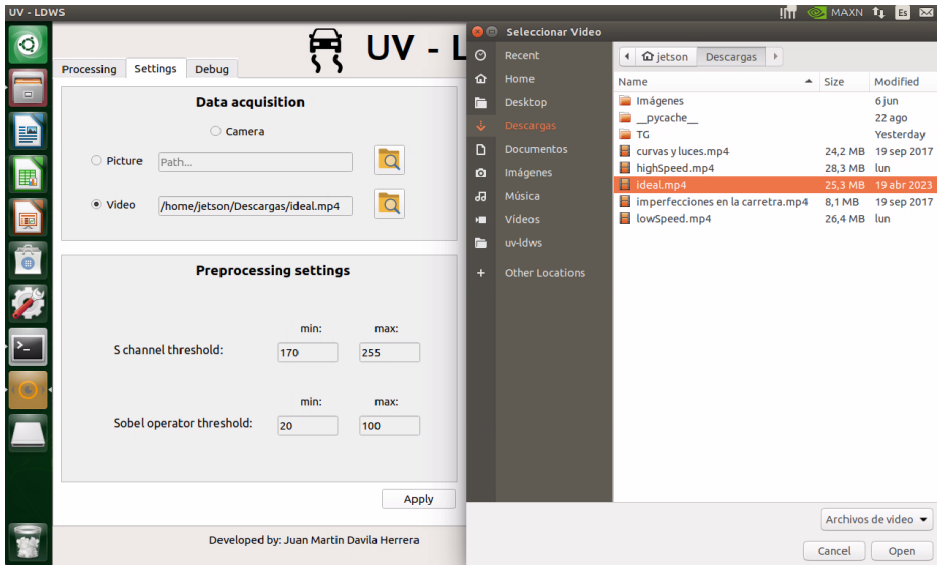


Figura 3.6. Ventana de selección de archivo.

- **Configuración de alerta:** En la Figura 3.7. y Figura 3.8. se muestra esta sección donde hay dos botones de chequeo indicados de forma visual con imágenes dinámicas para activar o desactivar cada modo de alerta (visual y auditivo respectivamente).

Figura 3.7. Sección de configuración de alerta deshabilitada

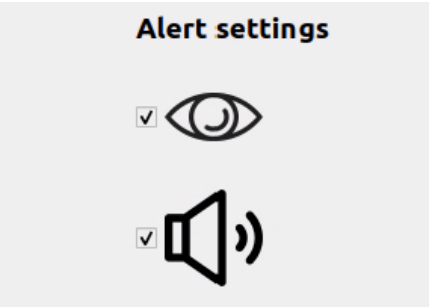


Figura 3.8. Sección de configuración de alerta habilitada

- **Configuración de preprocesamiento:** En la figura 3.9. se muestra esta sección compuesta por cuatro cuadros de texto correspondientes a (1) los valores mínimos y máximo del umbral en el canal S del espacio de color HSL y (2) los valores mínimos y máximos del umbral del filtro de Sobel sobre el eje X. Los cuales son necesarios para el procesamiento de las imágenes usando el método Sobel [20] y HSL [22].

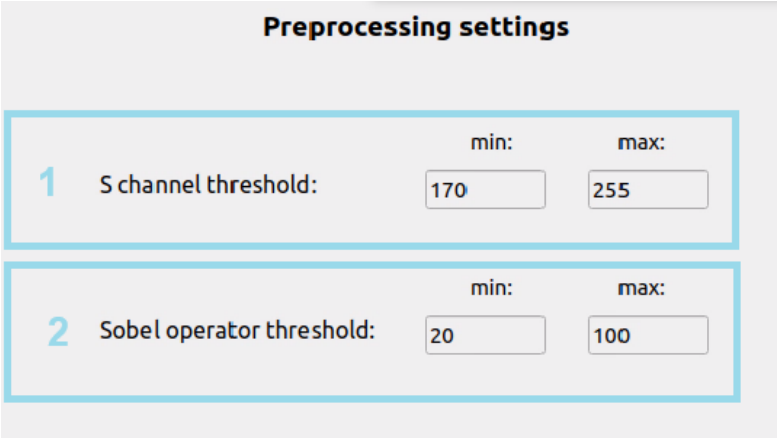


Figura 3.9. Sección de configuración del preprocesamiento

- **Configuración de procesamiento:** En la figura 3.10. se muestra esta sección donde se realizan diversos ajustes a la configuración del procesamiento correspondiente al algoritmo de reconocimiento del carril y la generación de alertas de salida de éste.

Inicialmente se tiene en (1) el ajuste de la cantidad de ventanas deslizantes a utilizar y en (2) el margen de separación entre estas representado en cantidad de pixeles. Para la supervisión de los carriles, se tiene en (3) el parámetro de validación del ancho del carril con respecto al ancho total de la imagen representado en porcentajes. El valor mínimo corresponde a que debe de ser al menos este valor para considerar la detección del carril como válida y el valor máximo significa que el ancho del carril detectado no puede superar este valor.

Para la detección de un solo carril y de las alertas, se tiene en (4) un parámetro para ajustar un offset del centro del carril con respecto a la imagen representado en porcentaje para así evitar falsos positivos en la generación de alertas en caso de movimientos bruscos como maniobras de esquivar. También, se tiene en (5) el ajuste en pixeles de la generación de alerta visual. Por último, en (6) se asignó un cuadro de texto para ajustar el coeficiente de regresión lineal para evitar falsos positivos y polígonos incoherentes en la imagen.

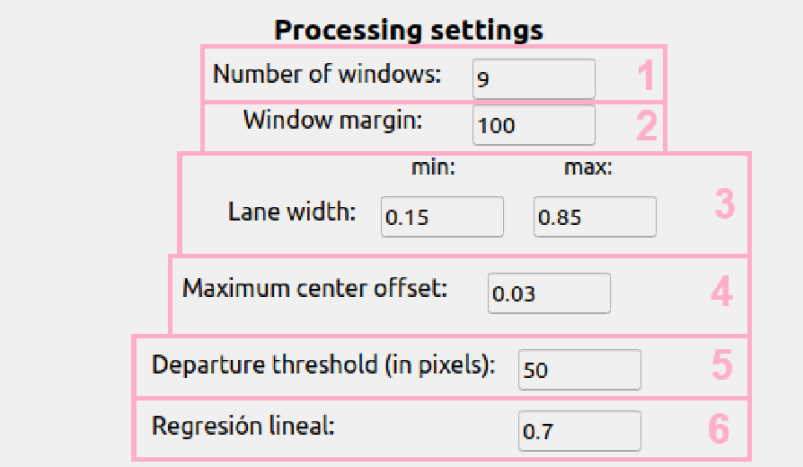


Figura 3.10. Configuración de procesamiento.

Para validar que la configuración se aplique correctamente, se asignó el botón de Aplicar como guardado en el archivo *settings.YAML* el cual permite visualizar la asignación y jerarquización de los apartados mostrados anteriormente en la configuración así como su posterior lectura e importación en las variables correspondientes. A continuación, se muestra una prueba realizada en la Figura 3.11, la confirmación en la Figura 3.12 y su resultado en la Figura 3.13.

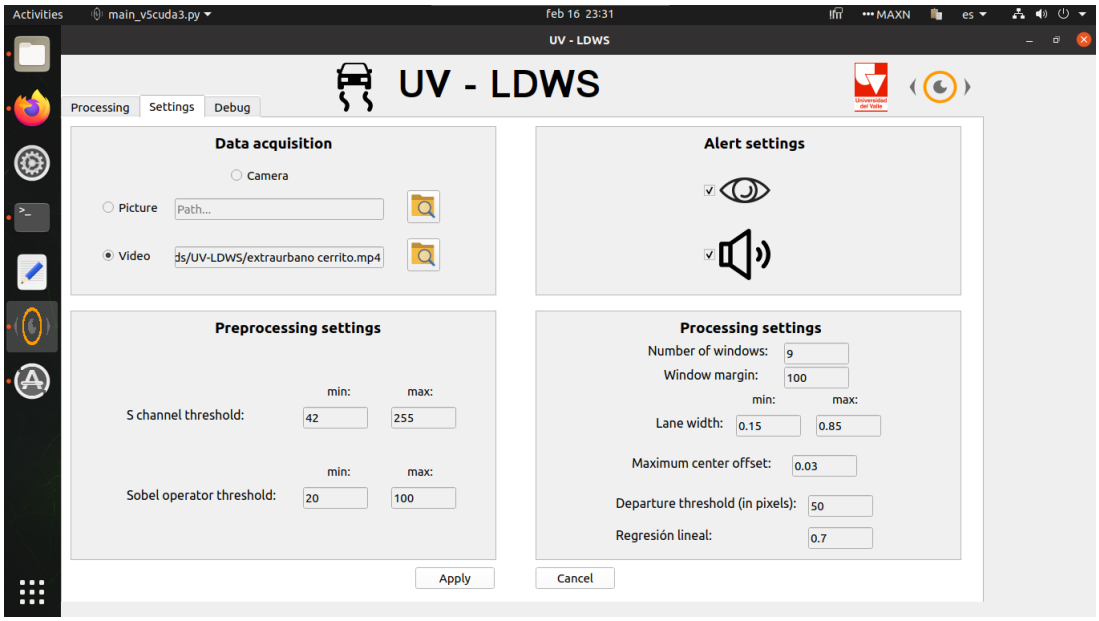


Figura 3.11. Parámetros de la prueba

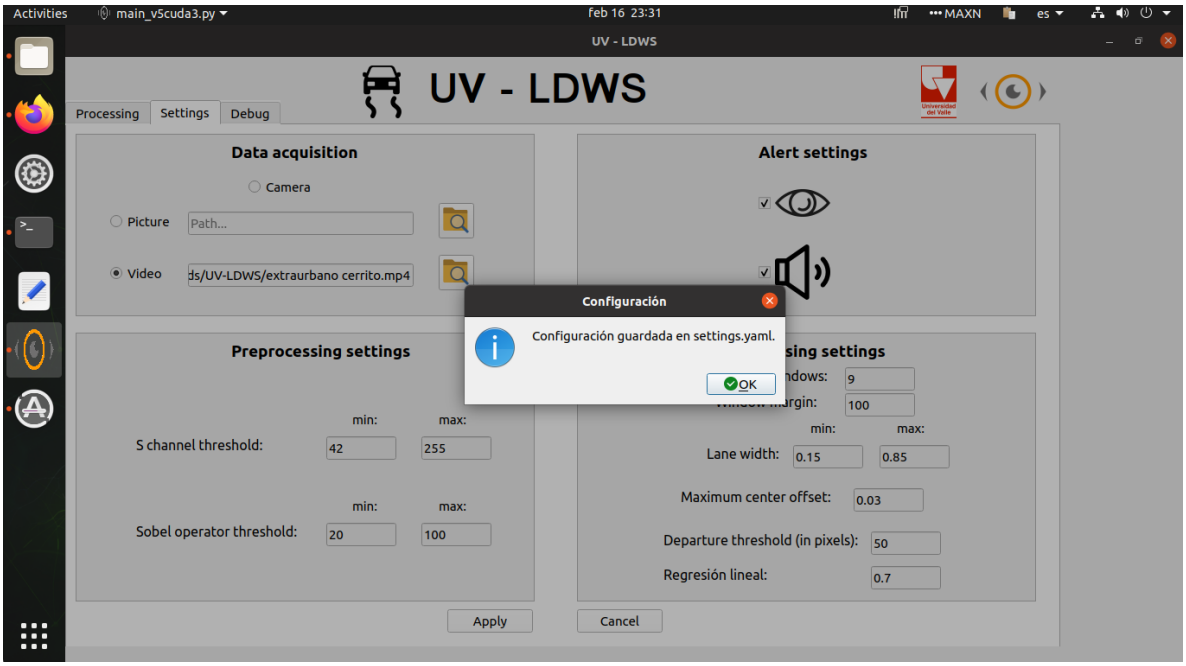


Figura 3.12. Validación de la prueba

Open

*settings.yaml
~/Downloads/UV-LDWS

```yaml
alertConf:
  soundAlert: 'ON'
  visualAlert: 'ON'
dataAcq:
  dataSource: Video
  picSource: ''
  vidSource: /home/jetson/Downloads/UV-LDWS/extraurbano cerrito.mp4
debugConf:
  HSLButton: false
  SobelButton: false
  fullButton: true
  fullPreButton: false
  resultantButton: false
  slidingWindowButton: false
  warpButton: false
preproConf:
  SChannelThreshold:
    maximum: 255
    minimum: 42
  SobelThreshold:
    maximum: 100
    minimum: 20
proConf:
  curveSimilarity: 0.7
  departureThreshold: 50
  laneWidth:
    maximum: 0.85
    minimum: 0.15
  margin: 100
  maxCenterOffset: 0.03
  nWindows: 9
```

Figura 3.13. Resultados de la prueba

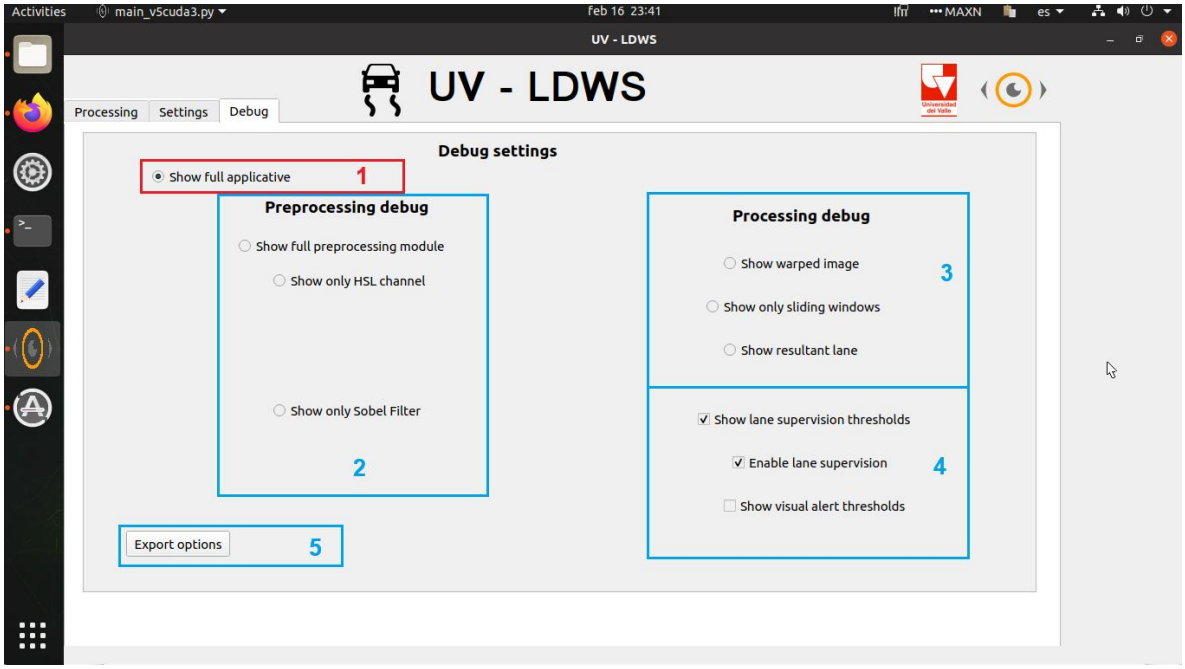


Figura 3.14. Pestaña de Depuración.

En la Figura 3.14. se muestra la pestaña “Depuración” la cual contiene en (1) el botón para visualizar el resultado final y está dividida en 3 secciones: en (2) Depuración del preprocesamiento, en (3) Depuración del procesamiento, en (4) Opciones varias y en (5) Opciones de exportación.

- **Depuración del preprocesamiento:** En esta sección, se tienen opciones para mostrar elementos específicos del módulo de preprocesamiento para así revisar los parámetros y funcionamiento de este para tener un mayor control y precisión sobre esta etapa y poder corregir errores o mejorar la respuesta del aplicativo.

Para ello en la Figura 3.15, se tiene en (1) un botón para observar el funcionamiento de la conversión al espacio de color HSL, en (2) un botón para observar el funcionamiento del filtro Sobel y en (3) un botón para observar el funcionamiento completo del módulo de preprocesamiento.

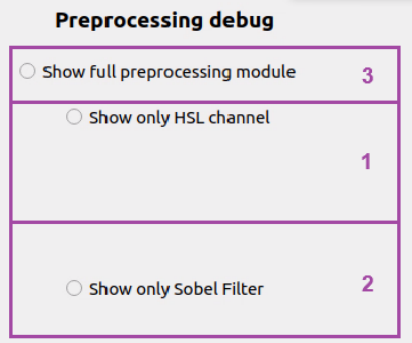


Figura 3.15. Depuración del preprocesamiento.

- **Depuración del procesamiento:** En esta sección, se tienen opciones para mostrar elementos específicos del módulo de procesamiento para así revisar los parámetros y funcionamiento de este para tener un mayor control y precisión sobre esta etapa y poder corregir errores o mejorar la respuesta del aplicativo.

Para ello en la Figura 3.16, se tiene en (1) un botón para observar el funcionamiento de la transformada de perspectiva respecto a los puntos establecidos previamente en el modo de edición, en (2) un botón para observar el comportamiento de las ventanas deslizantes y en (3) un botón para observar el carril resultante del reconocimiento.

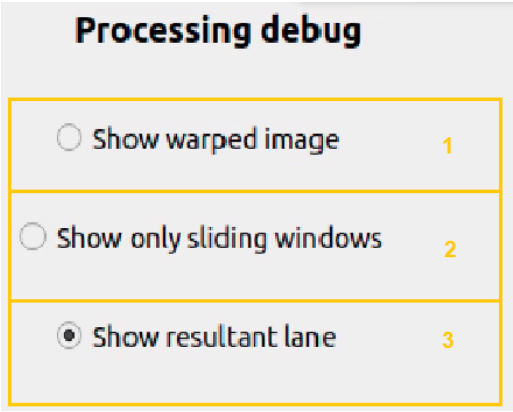


Figura 3.16. Depuración del procesamiento

- **Opciones varias:** En esta sección, se tienen opciones para visualizar opciones varias y depurar los parámetros previamente establecidos. Se tiene en (1) un botón para ver los umbrales de la supervisión del reconocimiento de carriles y así definir su validez, (2) un botón para habilitar o deshabilitar la supervisión de carriles y (3) un botón para mostrar los umbrales de la generación de las alertas tal como se muestra en la Figura 3.17.

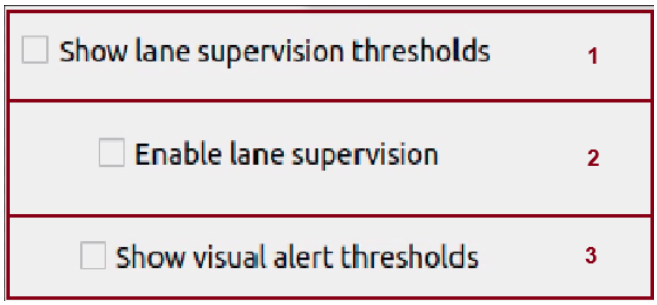


Figura 3.17. Opciones varias

- **Opciones de exportación:** En esta sección, se tienen opciones para exportar el resultado obtenido en un video en formato *.mp4 o una sucesión de imágenes extraídas de un frame por cada segundo de procesamiento.

3.3.2. Módulo de Preprocesamiento

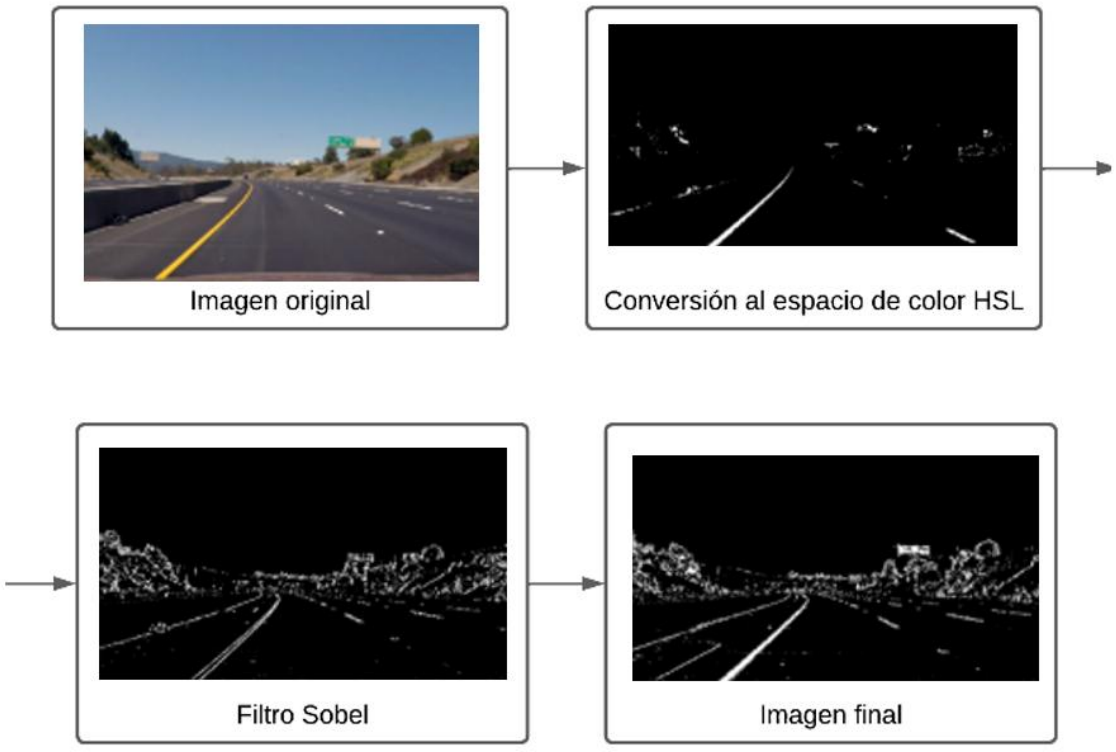


Figura 3.18. Diagrama de bloques del preprocesamiento de la imagen.

En este módulo se realiza el preprocesamiento de la imagen acondicionándola para mejorar la precisión y disminuir la carga computacional del algoritmo mostrado en la Figura 3.18. Para ello, inicialmente se aplica una conversión del espacio de color RGB al

HSL donde se desacoplan los canales de tono (H) y de luminosidad (L) para así reducir el impacto en las variaciones de iluminación como sombras o reflejos.

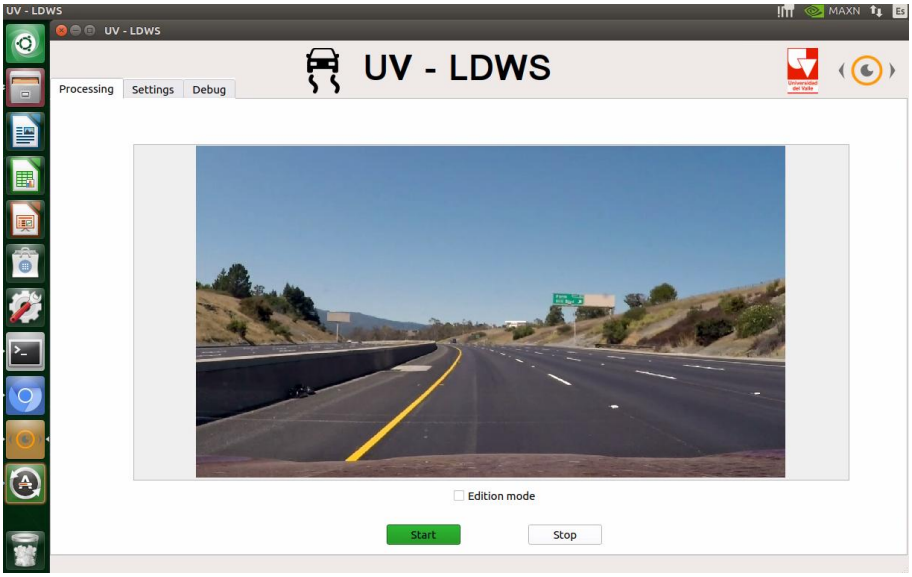


Figura 3.19. Imagen de prueba original.

Para lograr una mayor robustez frente a variaciones de iluminación, se utiliza únicamente el canal de saturación (S) para resaltar las líneas del carril que tienen un valor de saturación considerablemente más alto comparado con el pavimento. Al establecer un umbral específico, el sistema puede aislar las líneas del carril, que generalmente son amarillas o blancas, obtienen un valor alto de saturación lo que permite obtener un bosquejo preciso de las mismas tal como se muestra en la Figura 3.19 y Figura 3.20.

Figura 3.20. Imagen en espacio de color HSL

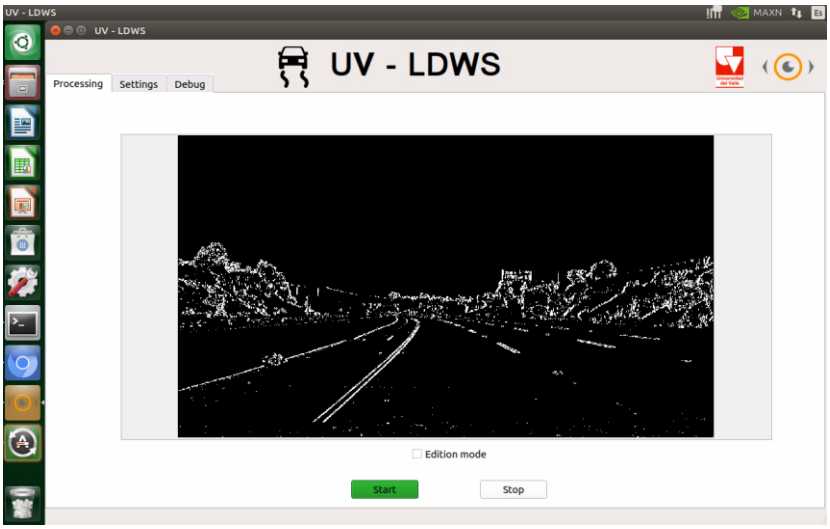


Figura 3.21. Imagen con filtro Sobel

Posteriormente, se aplica un filtro Sobel a la imagen para eliminar el ruido e información innecesaria tal como se muestra en la Figura 3.21. En el aplicativo, se aplica el filtro Sobel en el eje X para así priorizar la detección de bordes verticales tales como las líneas de los carriles mejorando la precisión de la detección. Al ser un elemento parametrizable, se puede reducir el ruido mediante el control de la sensibilidad de la binarización mediante este método y evitar falsas detecciones debido a imperfecciones en la carretera o cambios en la textura del pavimento.

Finalmente, se integran ambos métodos de manera secuencial y complementaria para resaltar las características más relevantes de la imagen para así mejorar la precisión de la detección evitando ruido y elementos descartables de la imagen.

Figura 3.22. Imagen completamente preprocesada.

Para ello, se combinan ambas imágenes binarizadas haciendo uso de la operación lógica OR para así obtener una respuesta uniforme y consistente con base en la información clave de saturación y los bordes de la imagen tal como se muestra en la Figura 3.22.

3.3.3. Módulo de Procesamiento

En este módulo se implementa la segmentación, reconocimiento y estimación de la trayectoria del carril.

Inicialmente, se aplica la transformada de perspectiva para convertir la imagen original de la carretera en una vista aérea simulada para identificar de forma más precisa los carriles, la ubicación de estos y su respectiva trayectoria. Para llevar a cabo este proceso se definen los puntos de origen y destino para así definir una matriz de transformación. Los puntos se eligen cuidadosamente en el modo edición visto anteriormente en el módulo de configuración para convertir la vista de la cámara (normalmente oblicua) en una vista superior que alinea los carriles paralelamente tal como se muestra en la Figura 3.23.

Figura 3.23. Transformada de perspectiva aplicada

Posteriormente, se realiza la detección de los carriles mediante ventanas deslizantes (ver Figura 3.24) lo cual permite localizar los pixeles correspondientes a las líneas del carril y establecer un polinomio para modelar la forma y trayectoria del carril de la siguiente manera:

1. **Cálculo del histograma:** Se calcula el histograma de la imagen binaria transformada para identificar las posiciones iniciales de los carriles siendo los picos más altos de este los lugares donde hay mayor probabilidad de posición del carril.
2. **Configuración de las ventanas:** La imagen se divide verticalmente en ventanas deslizantes que se utilizan para seguir las líneas del carril desde la parte inferior hasta la parte superior donde cada una de ellas tiene un alto definido para cubrir una porción de la altura de la imagen. Adicionalmente, se parametriza el margen horizontal de las mismas

para delimitar la búsqueda de los carriles para evitar falsos positivos con elementos no correspondientes al carril.

3. **Búsqueda de los pixeles de los carriles:** De acuerdo con las posiciones iniciales encontradas en el histograma, se utilizan las ventanas deslizantes para localizar los pixeles que pertenecen a los carriles. Dentro de cada ventana, se buscan los pixeles que se detectaron y se agregan a una lista de pixeles por carril. En caso de encontrar suficientes pixeles (este valor es configurable, por defecto se tienen 50 pixeles), se reajustan las ventanas en la siguiente iteración para tener un mejor seguimiento de la línea.

Figura 3.24. Búsqueda por ventanas deslizantes

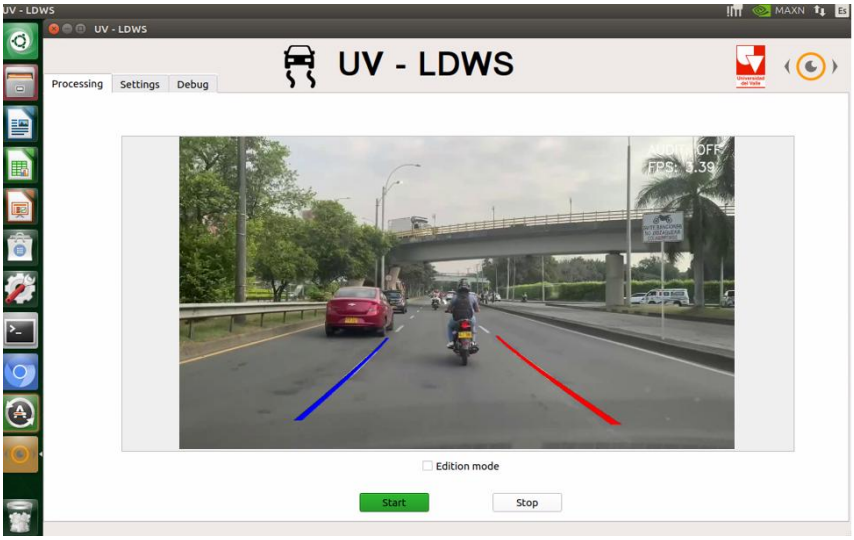


Figura 3.25. Carriles detectados

Finalmente, se realiza un ajuste polinomial para graficar los carriles detectados. Se hace uso de polinomios de segundo grado para mostrar la detección previa de los carriles tal como se muestra en la Figura 3.25.

3.3.4. Módulo de Alertas

En este módulo se implementa la generación de la alarma con los valores obtenidos de la longitud y posición de cada una de las líneas del carril para compararlas con la ubicación del vehículo y determinar un nivel porcentual de desviación para así generar una alerta visual y/o auditiva de acuerdo con los valores de sensibilidad configurados previamente.

Inicialmente, se realiza una supervisión a los carriles para determinar si la detección fue válida. El primer paso para determinar si el carril detectado es válido es verificar que el ancho del carril esté dentro de un rango razonable. Esto asegura que el sistema está detectando correctamente las líneas del carril y que no se están considerando elementos erróneos como carriles adyacentes o bordes de la carretera. Lo anterior, se parametriza con umbrales de separación respecto a un porcentaje del ancho de la imagen tal como se muestra en la Figura 3.26.

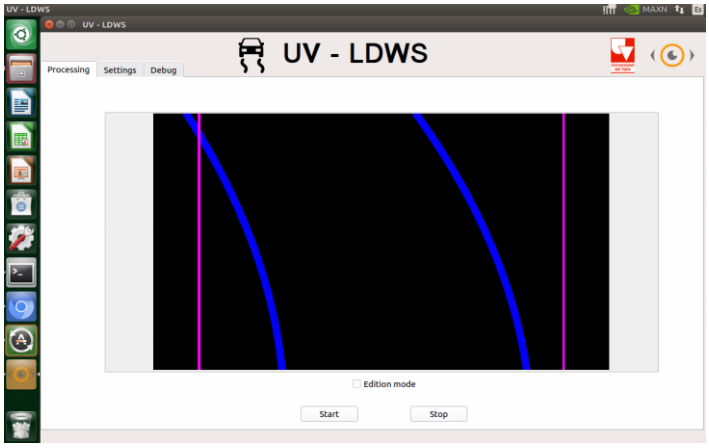


Figura 3.26. Restricción por ancho de la imagen.

Al definir la validez de cada carril, se aplica de manera individual el desplazamiento del carril respecto al centro del vehículo (imagen) permitiendo un offset por carril en caso de movimientos bruscos tal como se muestra en la Figura 3.27.

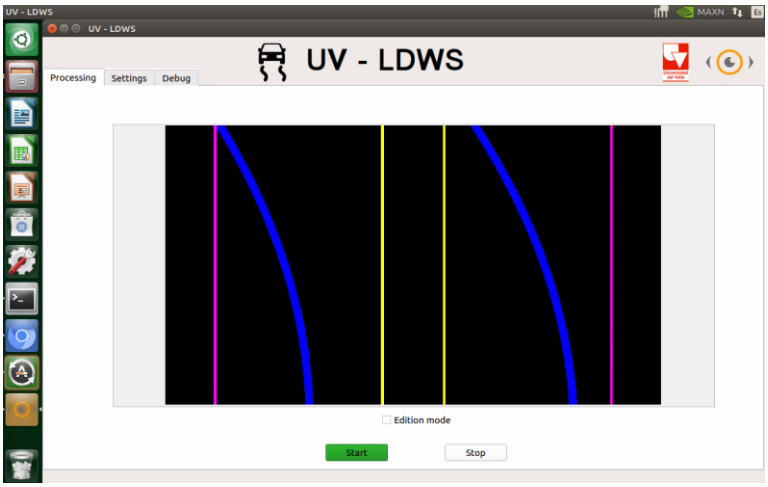


Figura 3.27. Restricción individual de carril.

Adicionalmente, se implementó un coeficiente basado en la comparación del coeficiente de regresión lineal entre curvas que representan las líneas detectadas, así como en la relación de sus radios de curvatura para evaluar la coherencia geométrica entre ellas. Teniendo en cuenta que cada línea se ajusta a un polinomio cuadrático como se muestra en la Ecuación 3.1.

$$x = Ay^2 + By + C$$

Ecuación 3.1

Donde *A, B, C* son los coeficientes del polinomio para cada línea, se calcula el radio de curvatura en un punto de evaluación mediante la fórmula propuesta en la Ecuación 3.2.

$$R_c = \frac{(1 + (2Ay_{eval} + B)^2)^{3/2}}{|2A|}$$

Ecuación 3.2

Donde el radio describe la agudeza de la curva, es decir, un valor grande que tiende a infinito indica una línea casi recta mientras que valores pequeños corresponden a curvas más pronunciadas. Para evaluar la similitud de curvatura entre las dos líneas (denotadas con radios mínimo y máximo, siendo radio mínimo menor o igual a radio máximo), se define el parámetro adimensional en la Ecuación 3.3.

$$S = \frac{R_{min}}{R_{max}} \in [0,1]$$

Ecuación 3.3

En caso de que *S≈1* indica curvas con radios muy similares y homogéneos, valores cercanos a 0 indican alta disparidad.

Finalmente se compara este valor con el umbral configurable *T* (por defecto *T=0.6*) para determinar la validez del carril bajo este parámetro restrictivo:

$$S \geq T$$

Ecuación 3.4

Tras determinar la validez de cada carril, se realiza una comparación de desplazamiento entre la línea del carril hallada y el centro de forma independiente y de acuerdo con esta diferencia, se tiene un umbral parametrizable de desviación en pixeles para generar la alerta visual. Para la alerta auditiva, se tiene un valor 20% mayor a la alerta visual. En caso de existir una desviación leve, se ilumina ese carril de color rojo en la visualización (ver Figura 3.28). En caso de presentarse una desviación moderada, se genera un sonido "beep".

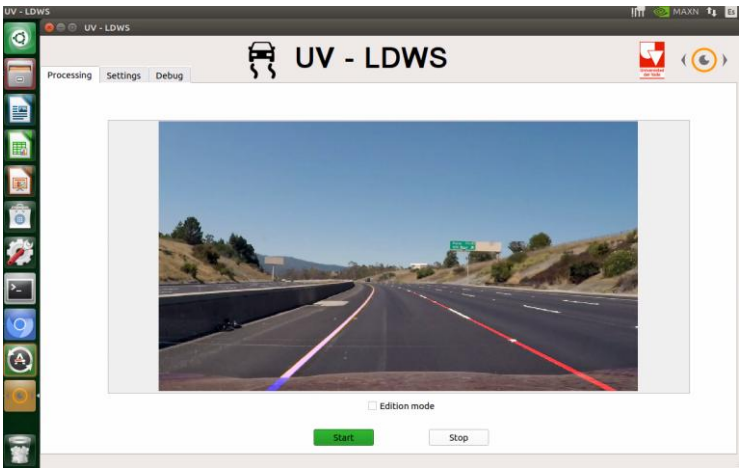


Figura 3.28. Alerta visual de salida de carril

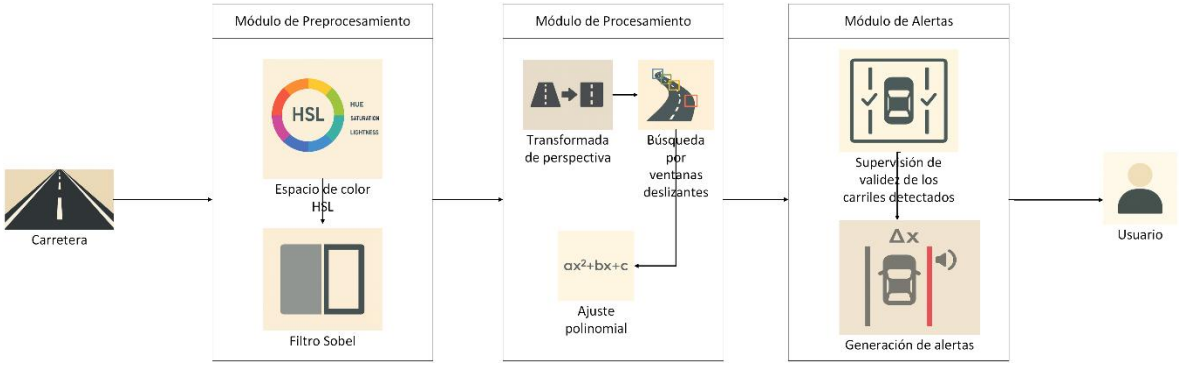


Figura 3.29. Diagrama de bloques de la implementación del aplicativo

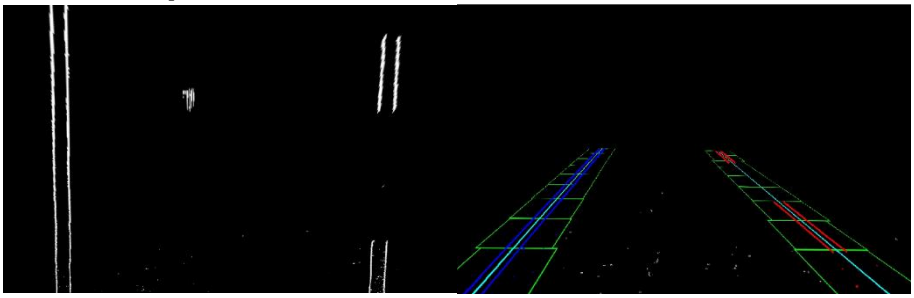
En la Figura 3.29 se resume la implementación mostrada en el presente capítulo mediante un diagrama de bloques que condensa la secuencia de los elementos utilizados en el aplicativo.

3.4. Conclusiones

En este capítulo, se realizó el desarrollo del aplicativo haciendo uso de la metodología RUP dando cumplimiento a los requerimientos funcionales vistos en la Tabla 3.1 y no funcionales establecidos en la Tabla 3.2. Esto impulsa el diseño ordenado y efectivo de la solución tecnológica propuesta permitiendo que el proceso de implementación tenga un esqueleto y una guía para el proceso de estructuración y montaje del aplicativo. El correcto diseño e implementación de una GUI intuitiva y fácil de usar para el usuario permite cumplir con el requerimiento 8. Dando cumplimiento al requerimiento 1, 6 y 7 donde la fuente de captura de la imagen y los límites establecidos para las alertas son elementos configurables desde el módulo de configuración en la pestaña *Settings*. También se dio cumplimiento al requerimiento 2 realizando un exhaustivo proceso de preprocesamiento de la imagen capturada bajo dos métodos de binarización. Con el método de detección de carril haciendo uso de la transformada de perspectiva junto con la búsqueda por ventanas deslizantes, se cumple con los requerimientos 3 y 4; haciendo uso de la información anterior y estableciendo reglas para definir la salida del carril, se cumple con el requerimiento 5.

El desarrollo del aplicativo se expande en cuatro principales módulos: módulo de configuración, módulo de preprocesamiento, módulo de procesamiento y módulo de generación de alerta. Cada uno de estos módulos representa un paso indispensable y secuencial en el correcto funcionamiento del aplicativo facilitando la interacción del usuario con los diferentes parámetros a utilizar y la respuesta del sistema respecto con estos. En la implementación del aplicativo, se ve plasmado el cumplimiento de los requerimientos funcionales y no funcionales, así como el uso del diagrama conceptual como una guía para las interacciones del usuario, los módulos y el aplicativo en general. También se ve descrito el diseño del aplicativo en el código funcional que compone el aplicativo haciendo uso de scripts, la integración de bibliotecas y tras pruebas exhaustivas, la verificación del funcionamiento de cada apartado del aplicativo.

Capítulo 4 PRUEBAS Y RESULTADOS



4.1. Introducción

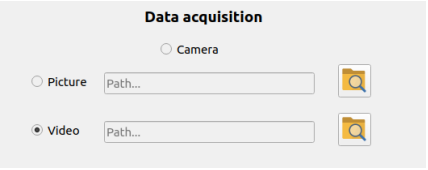
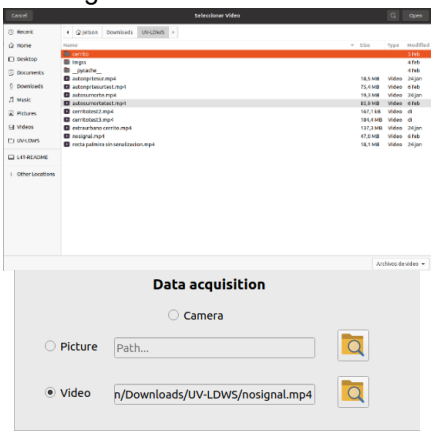
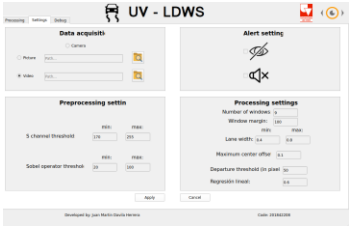
En este capítulo se presenta el proceso de validación del aplicativo embebido de alerta temprana de salida de carril mediante un conjunto riguroso de pruebas estructuradas en tres categorías fundamentales: pruebas de integración, pruebas cuantitativas y pruebas de campo. Estas pruebas permiten evaluar la calidad del sistema, su robustez operativa y su fidelidad al cumplimiento de los requerimientos funcionales y no funcionales previamente establecidos.

Las pruebas de integración tienen como propósito verificar la correcta interacción entre los distintos módulos que conforman el sistema, garantizando la interoperabilidad y el flujo coherente de datos a través de sus interfaces. Posteriormente, se exponen las pruebas cuantitativas, diseñadas para medir objetivamente el rendimiento del sistema mediante métricas como la precisión, el recall y la tasa de falsos positivos y negativos en distintos entornos controlados. Finalmente, se detallan las pruebas de campo, ejecutadas en condiciones reales de operación, las cuales evalúan la capacidad del sistema para adaptarse a variables impredecibles como condiciones climáticas adversas, variabilidad en la señalización vial y vibraciones durante la conducción.

4.2. Pruebas de integración

Tras finalizar el proceso de diseño e implementación del aplicativo embebido de alerta temprana de salida de carril, se procede a realizar las pruebas de integración con el objetivo de validar el funcionamiento conjunto de los módulos desarrollados. Estas pruebas permiten validar la correcta comunicación entre los componentes del sistema y su respuesta ante distintas condiciones operativas. De la misma forma, se busca garantizar que el flujo de datos entre la adquisición, el procesamiento y la generación de alertas se realice de manera continua, eficiente y coherente.

PRUEBA DE INTEGRACIÓN – MÓDULO DE CONFIGURACIÓN	
Tipo de Prueba	Selección de fuente de adquisición de datos, ajuste de parámetros del preprocesamiento, procesamiento, supervisión y alertas, y visualización de datos.
Hardware Requerido	NVIDIA Jetson Nano 4GB Developer Kit, microSD ADATA 64GB Class U10
Software Requerido	Aplicativo del Sistema de alerta temprana de salida de carril (Python 3.8.10), Jetpack 4.6, L4T 32.6.1, Ubuntu 18.04 LTS, PyQt 5.14.1
Objetivo	Validar el correcto funcionamiento del módulo de configuración
Carga de imágenes y videos de vehículo circulando en carretera, ajuste de parámetros y visualización de datos	
Objetivo	Interactuar con los archivos de imágenes o videos almacenados en el dispositivo para poder seleccionarlos, cargarlos y mostrarlos en una ventana. Además, definir valores a los parámetros usando los cuadros de texto para cambiar valores que pueden modificar el resultado del funcionamiento del aplicativo.
Datos de la Prueba	Imágenes o videos de carreteras. Parámetros
Procedimiento	1. Seleccionar el comboBox de “Video” 2. Seleccionar un video y presionar “Open” 3. Presionar el botón “Apply” 4. Visualizar el contenido en la pestaña “Processing” tras presionar el botón “Start” 5. Escribir los valores de preprocesamiento: (canal S (min: 170, max:255), Sobel (min: 20, max: 100), procesamiento: (ventanas: 9, margen de ventana: 100, ancho de línea: (min: 0.4, max: 0.8), offset máximo del centroide: 0.1, coeficiente de regresión lineal mínimo: 0.6) 6. Presionar el botón “Apply”
Resultado esperado	1. Se abre una ventana de diálogo en el sistema donde se puede navegar y escoger un archivo de vídeo en formato *.mp4 2. Se carga el archivo y el cuadro de texto se completa automáticamente indicando que la carga del archivo fue exitosa 3. Se visualiza el video presionando el botón de Start. 4. Se actualiza el archivo .YAML

Resultado obtenido	1. Se abre una ventana de diálogo en el sistema donde se puede navegar y escoger un archivo de vídeo en formato *.mp4  2. Se abre la ventana de exploración para cargar el archivo y el cuadro de texto se completa automáticamente indicando que la carga del archivo fue exitosa  3. Se visualiza el video presionando el botón de Start.  4. Se actualiza el archivo .YAML 

	
Comentarios	El modo seleccionado y la ruta del archivo seleccionado se almacena en el archivo settings.YAML.

Tabla 4.1. Pruebas de integración del módulo de configuración

PRUEBA DE INTEGRACIÓN – MÓDULO DE PROCESAMIENTO	
Tipo de Prueba	Ejecución del algoritmo de generación de alertas tempranas de salida de carril
Hardware Requerido	NVIDIA Jetson Nano 4GB Developer Kit, microSD ADATA 64GB Class U10
Software Requerido	Aplicativo del Sistema de alerta temprana de salida de carril (Python 3.8.10), Jetpack 4.6, L4T 32.6.1, Ubuntu 18.04 LTS, PyQt 5.14.1
Objetivo	Comprobar el funcionamiento del procesamiento etapa por etapa.
Procesamiento y visualización de las imágenes de los carriles	
Objetivo	Revisar cada etapa del preprocesamiento y procesamiento del aplicativo para tener una vista previa de cada etapa.
Datos de la Prueba	Imágenes o videos de carreteras. Parámetros
Procedimiento	1. Seleccionar el comboBox de HSL channel 2. Visualizar el resultado en la ventana principal 3. Seleccionar el comboBox de Sobel 4. Visualizar el resultado en la ventana principal 5. Seleccionar el comboBox de Warped image 6. Visualizar el resultado en la ventana principal 7. Seleccionar el comboBox de Sliding Windows 8. Visualizar el resultado en la ventana principal 9. Seleccionar el comboBox de Resultant Lane 10. Visualizar el resultado en la ventana principal
Resultado esperado	1. Visualizar el funcionamiento del canal HSL 2. Visualizar el funcionamiento del filtro Sobel 3. Visualizar el funcionamiento de la transformada de perspectiva 4. Visualizar el funcionamiento de las ventanas deslizantes 5. Visualizar el funcionamiento de la línea resultante

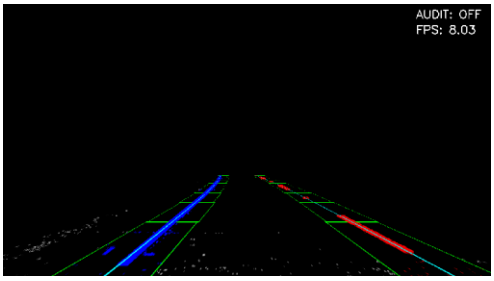
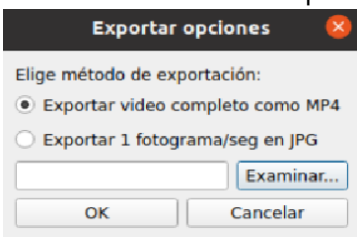
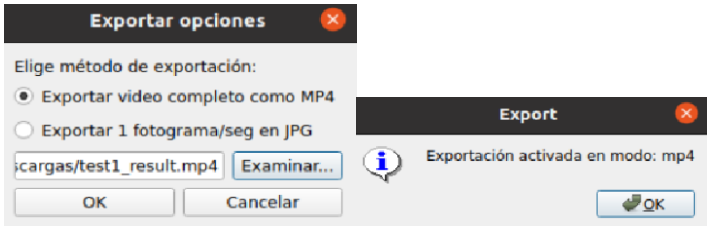
Resultado obtenido	1. Visualizar el funcionamiento del canal HSL  2. Visualizar el funcionamiento del filtro Sobel  3. Visualizar el funcionamiento de la transformada de perspectiva  4. Visualizar el funcionamiento de las ventanas deslizantes  5. Visualizar el funcionamiento de la línea resultante 
Comentarios	

Tabla 4.2. Pruebas de integración del módulo de procesamiento

PRUEBA DE INTEGRACIÓN – REPORTES	
Tipo de Prueba	Extracción de reportes en formato de vídeo o fotos
Hardware Requerido	NVIDIA Jetson Nano 4GB Developer Kit, microSD ADATA 64GB Class U10
Software Requerido	Aplicativo del Sistema de alerta temprana de salida de carril (Python 3.8.10), Jetpack 4.6, L4T 32.6.1, Ubuntu 18.04 LTS, PyQt 5.14.1
Objetivo	Comprobar la generación de reportes en formato de video o múltiples fotos en representación de fotogramas
Descarga de imágenes y videos con el resultado del procesamiento de los mismos	
Objetivo	Interactuar con los archivos de imágenes o videos almacenados en el dispositivo para poder procesarlos y posteriormente extraer sus resultados en formato de fotogramas o video.
Datos de la Prueba	Imágenes o videos de carreteras.
Procedimiento	1. Presionar el botón “Export Options” en la pestaña “Debug” 2. Seleccionar los parámetros de ruta y tipo de formato. 3. Presionar el botón “Ok” 4. Ejecutar el aplicativo hasta que termine el procesamiento 5. Visualizar el video o fotogramas resultantes
Resultado esperado	1. Abrir la ventana de selección de parámetros 2. Seleccionar los parámetros de ruta y tipo de archivo 3. Ejecución del aplicativo 4. Confirmación de la generación del archivo de reportes 5. Visualización del reporte
Resultado obtenido	1. Abrir la ventana de selección de parámetros  2. Seleccionar los parámetros de ruta y tipo de archivo  3. Ejecución del aplicativo

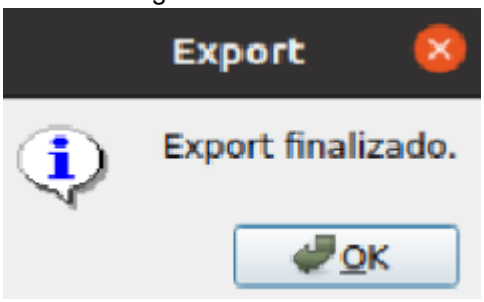
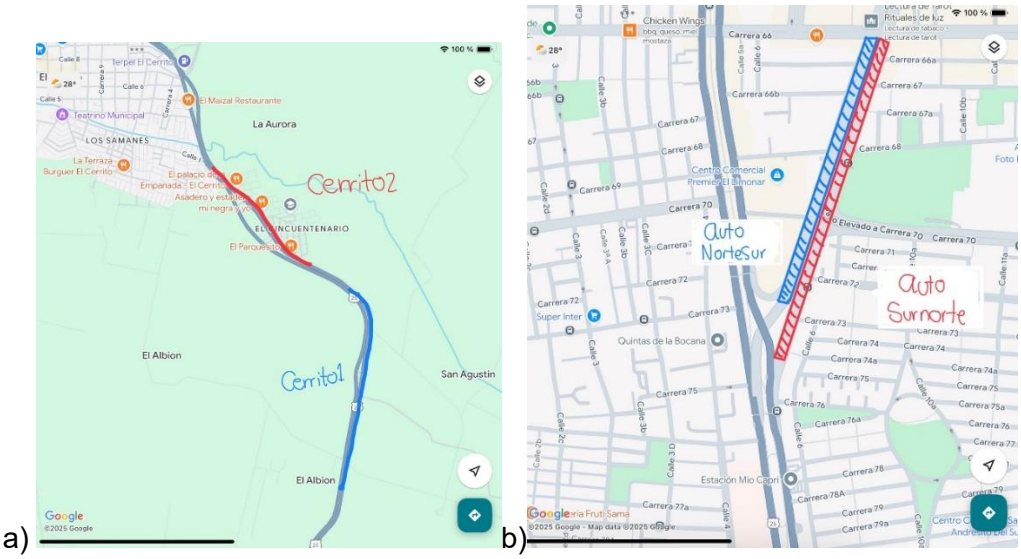
	 4. Confirmación de la generación del archivo de reportes  5. Visualización del reporte 
Comentarios	

Tabla 4.3. Prueba de integración del módulo de reportes

4.3. Pruebas cuantitativas

Para el proceso de validación del sistema de alerta temprana de salida de carril, se realizaron cuatro pruebas de campo en diferentes escenarios representativos. Las primeras dos pruebas, llamadas respectivamente **Cerrito 1** y **Cerrito 2**, se hicieron en vías extraurbanas (carretera), más específicamente en la **Vía Panamericana**, también conocida como **Ruta Nacional 25** en el recorrido **Palmira-Cerrito** reconocida por ser un trayecto de alta velocidad, líneas de carril relativamente claras y consistentes, así como condiciones ambientales relativamente estables propias de carreteras abiertas en el departamento del Valle del Cauca y un estado vial óptimo (ver Figura 4.1).

Por otro lado, las pruebas llamadas **AutoSurNorte** y **AutoNorteSur** se desarrollaron en entornos urbanos (ciudad), más específicamente en la **Calle 10**, también conocida como **Autopista Suroriental** en el tramo entre la **Carrera 66** y la **Carrera 73** enfrentando condiciones más problemáticas como tráfico de automóviles y motocicletas, delineado de carriles menos preciso e inconsistente, señales de tránsito impresas en el suelo y un estado vial en malas condiciones (ver Figura 4.2). Cada una de estas pruebas permitió adquirir información valiosa respecto al comportamiento del sistema tanto en la detección de carriles como en la generación de alertas de salida de carril.



a) b)

Figura 4.1. a) Resumen de recorrido extraurbano b) Resumen de recorrido en ciudad

Buscando mantener la consistencia metodológica y minimizar la variabilidad de la adquisición de los datos, así como reducir el margen de error entre pruebas, todas las pruebas experimentales fueron registradas bajo condiciones homogéneas utilizando un teléfono iPhone 14 Pro de almacenamiento 128GB mostrado en la Figura 4.2. Los videos se capturaron a una resolución Full HD 1080p (1920x1080) a una tasa de 30 fotogramas por segundo (fps), como se muestra en la Figura 4.4. El sensor principal del iPhone 14 Pro es un lente de 24mm de distancia focal con una apertura de diafragma f/1.78, asegurando una óptima entrada de luz. El procesamiento de los videos se realizó en tiempo real, con un tiempo de respuesta promedio por imagen entre 61 ± 4 ms y 102 ± 7 ms, equivalente a tasas de 9.8 ± 0.6 a 16.5 ± 0.9 fps, dependiendo del escenario de prueba (). En particular, el escenario PalmiraNoSignal presentó un mejor rendimiento, alcanzando valores superiores al promedio general debido a la menor complejidad en la marcación vial.

Escenario	FPS promedio (±σ)	Tiempo por fotograma (ms ±σ)
Cerrito 1	9.8 ± 0.6	102 ± 7
Cerrito 2	10.3 ± 0.5	97 ± 5
AutoNorteSur	9.9 ± 0.8	101 ± 8
AutoSurNorte	10.0 ± 0.7	100 ± 7
PalmiraNoSignal	16.5 ± 0.9	61 ± 4
Promedio global	11.3 ± 2.7	92 ± 17

Tabla 4.4 Tiempo de procesamiento de imagen por prueba

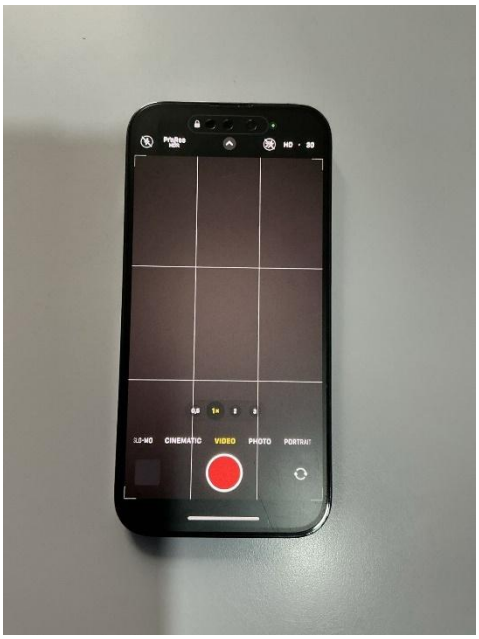
a)  b) 

Figura 4.2. a) Dispositivo de captura de imagen, b) Software nativo de cámara con las configuraciones aplicadas

Durante la recolección de los datos, el “modo acción” que ofrece el dispositivo fue deshabilitado manualmente para mantener la fidelidad del sistema de estabilización nativo y evitar posibles respuestas sobre procesadas. El sistema de estabilización del iPhone 14 Pro está compuesto por la combinación de estabilización óptica de imagen (OIS) y estabilización digital lo que entregó un registro visual estable a pesar de las vibraciones que genera el propio movimiento vehicular y más estando anclado al espejo retrovisor que transmite directamente las vibraciones del movimiento vehicular y del motor.

El teléfono se fijó mediante adhesión mecánica al espejo retrovisor central del vehículo utilizando cinta de enmascarar genérica. Para evitar la desviación angular en el plano horizontal, se niveló el eje óptico del sensor mediante un nivel de burbuja manual y se utilizó la rejilla de composición del software nativo de la cámara como referencia para mantener la alineación panorámica durante el montaje mostrado a continuación en la Figura 4.3.

Figura 4.3. Montaje de cámara nivelado

Para mantener la eficiencia computacional y reducir la carga de procesamiento sobre la NVIDIA Jetson Nano, cada archivo de video *.mp4 fue sometido a una única etapa de posprocesamiento donde se redujo su resolución de Full HD 1080p (1920x1080) a HD 720p (1280x720) y no se tuvo ninguna alteración a la estructura del contenido visual.

A partir de la información recopilada, se diseñaron matrices de confusión para la etapa de detección de carril y para la alerta temprana de salida de carril abriendo la oportunidad de identificar de manera cuantitativa la cantidad de aciertos y fallos, así como la razón cualitativa de cada tipo de fallo.

4.3.1. Cerrito 1

La primera prueba cuantitativa fue diseñada para evaluar el desempeño del sistema de alerta temprana de salida de carril en condiciones extraurbanas (carretera) donde se tiene la ventaja de contar con tramos rectos y curvos, velocidades moderadas a altas, condiciones de visibilidad favorables, un buen estado de la carretera, líneas correctamente demarcadas y señalización debidamente indicada.

La duración de la prueba fue de 158 segundos (2 minutos, 38 segundos) donde se extrajeron 2 fotogramas por cada segundo obteniendo así 317 muestras haciendo uso de un código de Python que teniendo en cuenta la tasa de actualización (30fps), extrae la imagen *.png cada 15 frames. Esta prueba fue realizada en el tramo Palmira-Cerrito tal como se muestra en la Figura 4.4 a una velocidad aproximada de 50km/h sobre el carril derecho manteniendo la prudencia y respetando las normativas nacionales [28] que indican para las vías de doble sentido “De dos (2) carriles: Por el carril de su derecha y utilizar con precaución el carril de su izquierda para maniobras de adelantamiento y respetar siempre la señalización respectiva”.

Figura 4.4. Trayectoria de la prueba Cerrito 1

El propósito de esta prueba fue observar el comportamiento del sistema en condiciones reales para la detección de carriles y la generación de alertas tempranas de salida de carril. Para ello, se hicieron dos matrices de confusión independientes donde la primera

corresponde a la detección de carril y con los valores que arrojaron verdadero positivo, se realiza la matriz de generación de alertas de salida temprana.

Detección de carril	
Verdadero positivo: 211 (66,56%)	Falso positivo: 15 (04,73%)
Falso negativo: 91 (28,71%)	Verdadero negativo: 0 (00,00%)

Tabla 4.5. Matriz de confusión de detección de carril en Cerrito1.

Como se observa en la Tabla 4.4. se tomaron 317 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 211 de las 317 muestras, el sistema detectó correctamente la presencia de los carriles mostrados y trazó de manera correcta su forma y estructura. Esto corresponde al 66.56% del total de muestras lo cual sugiere aproximadamente que 2 de cada 3 muestras el sistema cumplió adecuadamente con su primera etapa de detección de carril.
- **Falsos negativos:** Este valor crítico indica que, en 91 de las 317 muestras, el sistema no detectó los carriles así estuvieran presentes. Esto corresponde al 28.71% indicando que se puede mejorar la robustez del sistema en esta prueba.
- **Falsos positivos:** Este valor crítico indica que, en 15 de las 317 muestras, el sistema detectó carriles que en realidad no estaban. Esto corresponde al 4.73% y puede derivarse a señales de tránsito en el piso, confusión con vehículos circundantes y/o cambios bruscos de iluminación lo cual puede producir una alerta errónea.
- **Verdaderos negativos:** Este valor no tiene peso en esta prueba dado que en el proceso de toda la prueba hubo presencia constante e ininterrumpida de los carriles sobre los cuales se estaba circulando por lo cual se mantiene un valor de 0.

En la Tabla 4.5. se mostrará la matriz de confusión de la generación de alertas tempranas de salida de carril donde se analizaron los valores respecto a la matriz anterior en los fotogramas donde la detección de carril fue verdadero positivo.

Alerta temprana de salida	
Verdadero positivo: 32 (15,17%)	Falso positivo: 4 (01,90%)
Falso negativo: 21 (09,95%)	Verdadero negativo: 154 (72,99%)

Tabla 4.6. Matriz de confusión de generación de alertas en Cerrito1.

Como se observa, se tomaron 211 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 32 de las 211 muestras, el sistema detectó correctamente una salida de carril y emitió una alerta temprana correspondiente al 15.1%.
- **Falsos negativos:** Este valor indica que, en 21 de las 211 muestras, el vehículo salió del carril, pero el sistema no detectó la respectiva salida correspondiente al 9.9%.

- **Falsos positivos:** Este valor indica que, en 4 de las 211 muestras, el sistema reconoció una falsa salida de carril correspondiente al 1.90%.
- **Verdaderos negativos:** Este valor indica que, en 154 de las 211 muestras, el vehículo circuló correctamente y el sistema no emitió ninguna alerta. Esta es la situación más común lo que representa un comportamiento silencioso y tranquilo presentado en las condiciones normales de manejo correspondiente al 72.99%.

De acuerdo con lo consignado en la matriz de confusión en el apartado de la detección de carril, se puede identificar el comportamiento del sistema en los escenarios presentados en la prueba. En los verdaderos positivos que representan un 66,56% de los fotogramas analizados lo cual significa que se reconoció correctamente la presencia del carril donde los tramos de carretera presentan líneas de delimitación continuas, bien contrastadas y sin interferencias visuales. Por el lado contrario, los falsos negativos que representan el 28,71%, corresponden a situaciones donde los carriles estaban presentes y visibles pero el sistema no fue capaz de reconocerlos. Esto se presentó con mayor frecuencia en situaciones donde las líneas son discontinuas, están cubiertas por sombras o coexistían con otros elementos ajenos a la delimitación del carril generando ambigüedad visual.

También se presentaron falsos positivos representando el 04.73%, cuya proporción es notablemente baja, fueron generados por interpretaciones erróneas dadas por grietas, fisuras, señalización de velocidad en pavimento y señalización de dirección en forma de flecha en pavimento que siguen estando dentro del rango de análisis y cumpliendo con el coeficiente de correlación. En este caso, no se presentaron verdaderos negativos dado que en todo el tramo hay existencia visible de líneas debidamente demarcadas y las condiciones ambientales fueron favorables para la toma de muestras.

Teniendo en cuenta lo registrado en la matriz de confusión en la sección de la generación de alertas tempranas de salida de carril, se tomaron **únicamente** los valores donde la detección de carril fue correcta, es decir, los verdaderos positivos de la etapa anterior. Esto se realiza para que el análisis del sistema de alertas tempranas de salida de carril no se vea contaminado por errores de segmentación del carril. En los verdaderos positivos correspondientes al 15,17%, se observan los eventos donde se identificó una desviación significativa de la trayectoria hacia afuera del carril y se generó una correcta activación de la alarma. En los falsos positivos correspondientes al 01,90%, se tienen los escenarios donde se generaron falsas alarmas. Es decir, casos en los cuales el vehículo mantuvo su centroide dentro del carril y aún así se generó una alerta, provocada por maniobras evasivas o curvaturas que afecten en exceso la extensión de la línea de carril generada.

En los verdaderos negativos correspondientes al 72,99% se reflejan los casos donde no se generaron alertas cuando no era necesario. Esto es un resultado positivo dado que se evita la generación de falsas alarmas perdiendo credibilidad por parte del usuario y generando fatiga en el mismo. Por último, se tienen los falsos negativos los cuales son escenarios donde a pesar de haberse presentado una salida inminente del carril, el sistema no generó la debida alerta. Esto se generó principalmente por oscilaciones suaves y progresivas en la trayectoria del vehículo que no superaron el umbral predefinido y a desajustes en la interpretación del offset lateral en algunas curvas leves.

4.3.2. Cerrito 2

La segunda prueba cuantitativa es una extensión de la primera prueba para evaluar el desempeño del sistema de alerta temprana de salida de carril en condiciones extraurbanas (carretera) similares a la prueba anterior donde se mantienen las condiciones favorables en un tramo diferente de la misma vía.

La duración de la prueba fue de 101 segundos (1 minuto, 41 segundos) donde se extrajeron 2 fotogramas por cada segundo obteniendo así 202 muestras haciendo uso de un código de Python que teniendo en cuenta la tasa de actualización (30fps), extrae la imagen *.png cada 15 frames. Esta prueba fue realizada en el tramo Palmira-Cerrito tal como se muestra en la Figura 4.5 a una velocidad aproximada de 60km/h sobre el carril derecho manteniendo la prudencia y respetando las normativas nacionales [cntt] que indican para las vías de doble sentido “De dos (2) carriles: Por el carril de su derecha y utilizar con precaución el carril de su izquierda para maniobras de adelantamiento y respetar siempre la señalización respectiva”.

Figura 4.5. Trayectoria de la prueba Cerrito 2

El propósito de esta prueba fue observar el comportamiento del sistema en condiciones reales para la detección de carriles y la generación de alertas tempranas de salida de carril. Para ello, se hicieron dos matrices de confusión independientes donde la primera corresponde a la detección de carril y con los valores que arrojaron verdadero positivo, se realiza la matriz de generación de alertas de salida temprana.

Detección de carril	
Verdadero positivo: 140 (69,31%)	Falso positivo: 12 (05,94%)
Falso negativo: 50 (24,75%)	Verdadero negativo: 0 (00,00%)

Tabla 4.7. Matriz de confusión de detección de carril en Cerrito2.

Como se observa en la Tabla 4.6. se tomaron 202 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 140 de las 202 muestras, el sistema detectó correctamente la presencia de los carriles mostrados y trazó de manera correcta su forma y estructura. Esto corresponde al 69.31% del total de muestras lo cual sugiere aproximadamente que 2 de cada 3 muestras el sistema cumplió adecuadamente con su primera etapa de detección de carril.
- **Falsos negativos:** Este valor crítico indica que, en 50 de las 202 muestras, el sistema no detectó los carriles así estuvieran presentes. Esto corresponde al 24,75% indicando que se puede mejorar la robustez del sistema en esta prueba. Varios de estos falsos negativos fueron causados por el tramo de cambio de continuidad en las líneas sobre el cual el sistema no tuvo la adaptación esperada al cambio de geometría de las líneas.
- **Falsos positivos:** Este valor crítico indica que, en 12 de las 202 muestras, el sistema detectó carriles que en realidad no estaban. Esto corresponde al 05,94% y puede derivarse a señales de tránsito en el piso, confusión con vehículos circundantes y/o cambios bruscos de iluminación lo cual puede producir una alerta errónea.
- **Verdaderos negativos:** Este valor no tiene peso en esta prueba dado que en el proceso de toda la prueba hubo presencia constante e ininterrumpida de los carriles sobre los cuales se estaba circulando por lo cual se mantiene un valor de 0.

En la Tabla 4.7. se mostrará la matriz de confusión de la generación de alertas tempranas de salida de carril donde se analizaron los valores respecto a la matriz anterior en los fotogramas donde la detección de carril fue verdadero positivo.

Alerta temprana de salida	
Verdadero positivo: 44 (31,43%)	Falso positivo: 2 (05,94%)
Falso negativo: 21 (15,00%)	Verdadero negativo: 73 (52,14%)

Tabla 4.8. Matriz de confusión de generación de alertas en Cerrito 2.

Como se observa, se tomaron 140 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 44 de las 140 muestras, el sistema detectó correctamente una salida de carril y emitió una alerta temprana correspondiente al 31,43%.
- **Falsos negativos:** Este valor indica que, en 21 de las 140 muestras, el vehículo salió del carril, pero el sistema no detectó la respectiva salida correspondiente al 15%. En la primera curva se presentó un comportamiento donde se tomó más cerrada en pro de generar la alerta, pero el sistema no reconoció la salida pues no alcanzó a cumplir con el rango en pixeles a pesar de la salida.
- **Falsos positivos:** Este valor indica que, en 2 de las 140 muestras, el sistema reconoció una falsa salida de carril correspondiente al 05,94%.
- **Verdaderos negativos:** Este valor indica que, en 73 de las 140 muestras, el vehículo circuló correctamente y el sistema no emitió ninguna alerta. Esta es la

situación más común lo que representa un comportamiento silencioso y tranquilo presentado en las condiciones normales de manejo correspondiente al 52,14%.

De acuerdo con lo consignado en la matriz de confusión en el apartado de la detección de carril, se puede identificar el comportamiento del sistema en los escenarios presentados en la prueba. En los verdaderos positivos que representan un 66,56% de los fotogramas analizados lo cual significa que se reconoció correctamente la presencia del carril donde los tramos de carretera presentan líneas de delimitación continuas, bien contrastadas y sin interferencias visuales. En contraste, los falsos negativos que representan el 28,71%, se dieron en situaciones donde los carriles estaban presentes y visibles pero el sistema no fue capaz de reconocerlos. Esto se presentó con mayor frecuencia en situaciones donde las líneas son discontinuas, están cubiertas por sombras o coexistían con otros elementos ajenos a la delimitación del carril generando ambigüedad visual. Entre los segundos 28 y 35 del video existe un tramo de la vía en el cual está presente un paradero de transporte público extraurbano donde cambia la geometría del carril generando una disparidad en la trayectoria del carril y generando falso negativo como se observa en la Figura 4.6. Haciendo uso de algoritmos predictivos tales como el filtro de Kalman extendido, se puede mejorar esta respuesta y presentar una mejor adaptabilidad ante este tipo de escenarios inconsistentes donde hay cambios inesperados en la geometría de la línea del carril.

Figura 4.6. Cambio en la trayectoria del carril

También se presentaron falsos positivos representando el 04.73%, cuya proporción es notablemente baja, fueron generados por interpretaciones erróneas dadas por grietas, fisuras, señalización de velocidad en pavimento y señalización de dirección en forma de flecha en pavimento que siguen estando dentro del rango de análisis y cumpliendo con el coeficiente de correlación. En este caso, no se presentaron verdaderos negativos dado que en todo el tramo hay existencia visible de líneas debidamente demarcadas y las condiciones ambientales fueron favorables para la toma de muestras.

Teniendo en cuenta lo registrado en la matriz de confusión en la sección de la generación de alertas tempranas de salida de carril, se tomaron **únicamente** los valores donde la detección de carril fue correcta, es decir, los verdaderos positivos de la etapa anterior.

Esto se realiza para que el análisis del sistema de alertas tempranas de salida de carril no se vea contaminado por errores de segmentación del carril. En los verdaderos positivos correspondientes al 15,17%, se observan los eventos donde se identificó una desviación significativa de la trayectoria hacia afuera del carril y se generó una correcta activación de la alarma. En los falsos positivos correspondientes al 01,90%, se tienen los escenarios donde se generaron falsas alarmas. Es decir, casos en los cuales el vehículo mantuvo su centroide dentro del carril y aun así se generó una alerta, provocada por maniobras evasivas o curvaturas que afecten en exceso la extensión de la línea de carril generada.

En los verdaderos negativos correspondientes al 72,99% se reflejan los casos donde no se generaron alertas cuando no era necesario. Esto es un resultado positivo dado que se evita la generación de falsas alarmas perdiendo credibilidad por parte del usuario y generando fatiga en el mismo. Por último, se tienen los falsos negativos los cuales son escenarios donde a pesar de haberse presentado una salida inminente del carril, el sistema no generó la debida alerta. Esto se generó principalmente por oscilaciones suaves y progresivas en la trayectoria del vehículo que no superaron el umbral predefinido y a desajustes en la interpretación del offset lateral en algunas curvas leves.

4.3.3. AutoNorteSur

La tercera prueba cuantitativa se realizó con el objetivo de evaluar el desempeño del sistema de alerta temprana de salida de carril en un entorno urbano. Esta prueba se llevó a cabo en un tramo de la Autopista Suroriental (también conocida como Calle 10) comprendido entre las carreras 66 y 73 en el sentido Norte-Sur (ver Figura 4.7); una vía de alta circulación vehicular y condiciones desafiantes tales como señalización en el pavimento más constante y un alto flujo vehicular mixto de automóviles y motocicletas. Esta prueba se realizó a una velocidad promedio de 30km/h respetando límites de velocidad de zona urbana y se circuló por el carril central para mantener un escenario más realista.

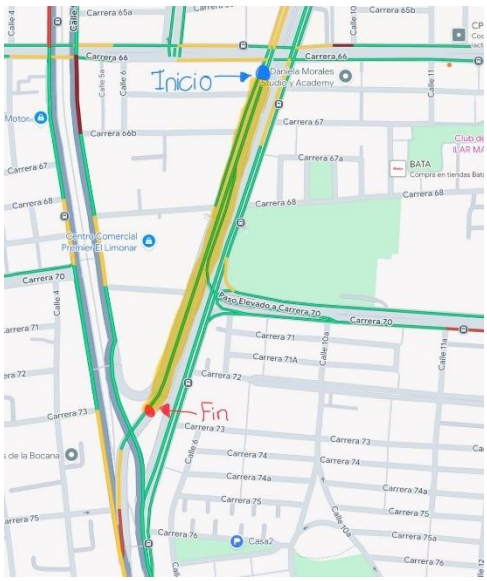


Figura 4.7. Recorrido realizado en AutoNorteSur

La duración total del recorrido fue de 26 segundos, durante los cuales se extrajeron 2 imágenes por cada segundo obteniendo así un total de 52 muestras. Para esto, se utilizó un script en Python que extrajo una imagen en formato *.png cada 15 fotogramas teniendo en cuenta que la tasa de actualización del video es de 30 fotogramas por segundo (fps).

El objetivo de esta prueba fue analizar el comportamiento del sistema en entorno urbano real que suelen ser más desafiantes para la detección de carriles y la generación de alertas tempranas de salida de carril. Para ello, se hicieron dos matrices de confusión independientes donde la primera corresponde a la detección de carril y con los valores que arrojaron verdadero positivo, se realiza la matriz de generación de alertas de salida temprana.

Detección de carril	
Verdadero positivo: 24 (46,15%)	Falso positivo: 7 (13,46%)
Falso negativo: 21 (40,38%)	Verdadero negativo: 0 (00,00%)

Tabla 4.9. Matriz de confusión de detección de carril en AutoNorteSur.

Como se observa en la Tabla 4.8. se tomaron 52 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 24 de las 52 muestras, el sistema detectó correctamente la presencia de los carriles mostrados y trazó de manera correcta su forma y estructura. Esto corresponde al 46,15% del total de muestras lo cual sugiere aproximadamente que 1 de cada 2 muestras el sistema cumplió adecuadamente con su primera etapa de detección de carril.
- **Falsos negativos:** Este valor crítico indica que, en 21 de las 52 muestras, el sistema no detectó los carriles así estuvieran presentes. Esto corresponde al 40,38% indicando que se puede mejorar la robustez del sistema en esta prueba. Esto producido por vehículos circundantes en zonas cercanas a las líneas del carril que desencadenaban cambios de iluminación produciendo alteraciones en el reconocimiento del carril y generando falsas detecciones con parámetros inválidos que el supervisor descarta y omite.
- **Falsos positivos:** Este valor crítico indica que, en 7 de las 52 muestras, el sistema detectó carriles que en realidad no estaban. Esto corresponde al 13,46% y puede derivarse a señales de tránsito en el piso, confusión con vehículos circundantes y/o cambios bruscos de iluminación lo cual puede producir una alerta errónea. En algunos casos donde las mencionadas falsas detecciones cumplieron con los parámetros restrictivos establecidos en la etapa del supervisor, se mostró una imagen errónea.
- **Verdaderos negativos:** Este valor no tiene peso en esta prueba dado que en el proceso de toda la prueba hubo presencia constante e ininterrumpida de los carriles sobre los cuales se estaba circulando por lo cual se mantiene un valor de 0.

En la Tabla 4.9. se mostrará la matriz de confusión de la generación de alertas tempranas de salida de carril donde se analizaron los valores respecto a la matriz anterior en los fotogramas donde la detección de carril fue verdadero positivo.

Alerta temprana de salida	
Verdadero positivo: 2 (08,33%)	Falso positivo: 8 (33,33%)
Falso negativo: 0 (00,00%)	Verdadero negativo: 14 (58,33%)

Tabla 4.10. Matriz de confusión de generación de alertas en AutoNorteSur.

Como se observa, se tomaron 24 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 2 de las 24 muestras, el sistema detectó correctamente una salida de carril y emitió una alerta temprana correspondiente al 08,33%.
- **Falsos negativos:** Este valor indica que no hubo detecciones fallidas de la salida del carril.
- **Falsos positivos:** Este valor indica que, en 8 de las 24 muestras, el vehículo salió del carril, pero el sistema no detectó la respectiva salida correspondiente al 33,33%. Debido a la naturaleza de un entorno urbano donde se presentan maniobras evasivas y variaciones en la posición del vehículo, se generan falsas alertas.
- **Verdaderos negativos:** Este valor indica que, en 14 de las 24 muestras, el vehículo circuló correctamente y el sistema no emitió ninguna alerta. Esta es la situación más común lo que representa un comportamiento silencioso y tranquilo presentado en las condiciones normales de manejo correspondiente al 58,33%.

Teniendo en cuenta lo registrado en la matriz de confusión en el apartado de detección de carril, se puede observar el comportamiento del sistema en el escenario urbano evaluado. Inicialmente, los verdaderos positivos representan un 46,15% de los fotogramas analizados. Esto indica que este porcentaje a pesar de ser menor respecto a las pruebas en entorno extraurbano refleja la mayor complejidad que posee un entorno urbano donde factores como la señalización sobre el pavimento (límite de velocidad, flechas paralelas) y el paso constante de vehículos adyacentes como automóviles y motocicletas debido al denso flujo vehicular agregan ruido que afecta la correcta segmentación de los carriles.

A diferencia de lo anterior, los falsos negativos representan un 40,38% lo cual indica que en 2 de cada 5 pruebas donde había un carril presente, el sistema no lo detectó. Estas fallas se deben principalmente a obstrucciones visuales causadas por vehículos invadiendo la línea del carril, la sombra de estos mismos y la presencia de la señalización horizontal que entra en conflicto con la detección y genera valores considerados como inválidos en la etapa de supervisión descartando así la detección. También, el reducido tiempo de la prueba dado por la distancia del trayecto implicó una menor resiliencia del sistema ante cambios bruscos en el entorno. Debido a que se pierde la información del carril con los vehículos circundantes, la implementación de una región de interés dinámica permite cambiar automáticamente la posición del plano que se toma eliminando estos vehículos que generan errores.

En los falsos positivos representando un 13,46% (superior al valor obtenido en las pruebas extraurbanas) evidencian la incorrecta interpretación de las líneas del carril. Esto se dio principalmente por las señales horizontales y falsas detecciones en las siluetas de los vehículos adyacentes generando una incorrecta detección que cumplió con los parámetros restrictivos del aplicativo. Al realizar la implementación de un filtro predictivo, se puede estimar dónde estaría realmente la línea evitando un falso positivo.

A partir de la matriz de confusión en la sección de la generación de alertas tempranas de salida de carril, se tomaron únicamente los fotogramas donde la detección de carril fue satisfactoria (verdaderos positivos en la etapa anterior). Con esta información, los verdaderos positivos correspondieron al 08,33% reflejando solo en dos ocasiones se identificó una correcta salida del carril emitiendo una alerta. La baja sensibilidad obtenida se puede explicar por la naturaleza del entorno urbano donde las salidas son poco frecuentes, así como el alto flujo vehicular que no permitieron realizar salidas voluntarias para evitar colisiones.

Por otro lado, los falsos positivos representaron un 33,33% indicando los casos donde se generaron alertas sin presentarse una desviación real del carril. Esto causado por maniobras evasivas comunes en la ciudad donde se realizan ligeros cambios de trayectoria para esquivar vehículos que invaden el carril, motocicletas que adelantan en medio de los carros además de las tapas de las alcantarillas que pueden ocasionar daños al vehículo. Si no se gestiona adecuadamente este comportamiento, se puede reducir la confianza del usuario en el sistema. En esta prueba tampoco se registraron verdaderos negativos dado que siempre estuvo presente la continua delimitación vial. De la misma forma, implementando una región de interés dinámica se pueden balancear los movimientos involuntarios o evasivos.

Los verdaderos negativos representaron el 58,33% siendo la mayoría de los casos en que el vehículo mantuvo la trayectoria dentro del carril y el sistema no generó ningún tipo de alerta. Por último, no se presentaron falsos negativos en esta prueba lo cual es favorable e indica una baja sensibilidad general del sistema en entornos urbanos donde las salidas son graduales y mucho menos evidentes.

4.3.4. AutoSurNorte

La cuarta y última prueba cuantitativa se realizó con el objetivo de reforzar el análisis del desempeño del sistema de alerta temprana de salida de carril en un entorno urbano. Esta prueba se desarrolló en un tramo de la Autopista Suroriental (también conocida como Calle 10) comprendido entre las carreras 74a y 66a en sentido Sur-Norte (ver Figura 4.8); una vía de alta circulación vehicular y condiciones desafiantes tales como señalización en el pavimento más constante y un alto flujo vehicular mixto de automóviles y motocicletas. Esta prueba se realizó a una velocidad promedio de 30km/h respetando límites de velocidad de zona urbana y se circuló por el carril central para mantener un escenario más realista.

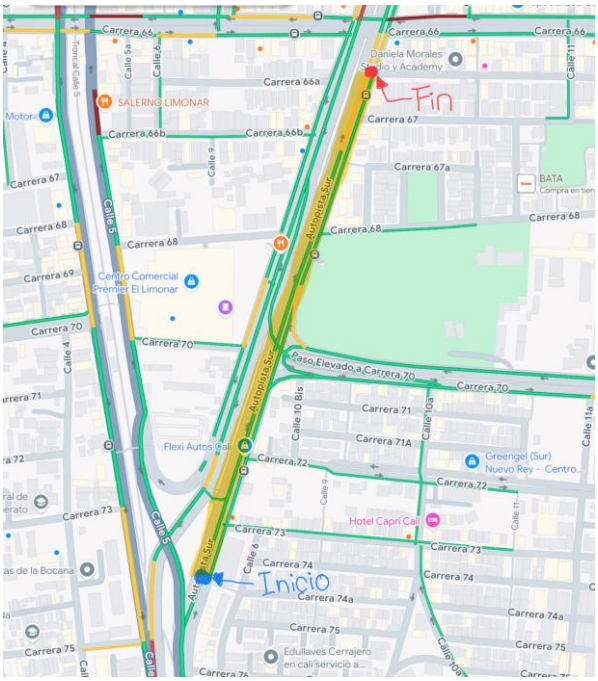


Figura 4.8. Recorrido realizado en AutoSurNorte

La duración total del recorrido fue de 27 segundos, durante los cuales se extrajeron 2 imágenes por cada segundo obteniendo así un total de 54 muestras. Para esto, se utilizó un script en Python que extrajo una imagen en formato *.png cada 15 fotogramas teniendo en cuenta que la tasa de actualización del video es de 30 fotogramas por segundo (fps).

El objetivo de esta prueba fue extender la observación del comportamiento del sistema en un entorno urbano real similar al anterior que igualmente suele ser desafiantes para la detección de carriles y la generación de alertas tempranas de salida de carril. Para ello, se hicieron dos matrices de confusión independientes donde la primera corresponde a la detección de carril y con los valores que arrojaron verdadero positivo, se realiza la matriz de generación de alertas de salida temprana.

Detección de carril	
Verdadero positivo: 24 (44,44%)	Falso positivo: 13 (24,07%)
Falso negativo: 17 (31,48%)	Verdadero negativo: 0 (00,00%)

Tabla 4.11. Matriz de confusión de detección de carril en AutoSurNorte.

Como se observa en la Tabla 4.10. se tomaron 54 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 24 de las 54 muestras, el sistema detectó correctamente la presencia de los carriles mostrados y trazó de manera correcta su forma y estructura. Esto corresponde al 44,44% del total de muestras lo cual sugiere aproximadamente que 1 de cada 2 muestras el sistema cumplió adecuadamente con su primera etapa de detección de carril.

- **Falsos negativos:** Este valor crítico indica que, en 17 de las 54 muestras, el sistema no detectó los carriles así estuvieran presentes. Esto corresponde al 31,48% indicando que se puede mejorar la robustez del sistema en esta prueba.
- **Falsos positivos:** Este valor crítico indica que, en 13 de las 54 muestras, el sistema detectó carriles que en realidad no estaban. Esto corresponde al 24,07% y puede derivarse a señales de tránsito en el piso, confusión con vehículos circundantes y/o cambios bruscos de iluminación lo cual puede producir una alerta errónea.
- **Verdaderos negativos:** Este valor no tiene peso en esta prueba dado que en el proceso de toda la prueba hubo presencia constante e ininterrumpida de los carriles sobre los cuales se estaba circulando por lo cual se mantiene un valor de 0.

En la Tabla 4.11. se mostrará la matriz de confusión de la generación de alertas tempranas de salida de carril donde se analizaron los valores respecto a la matriz anterior en los fotogramas donde la detección de carril fue verdadero positivo.

Alerta temprana de salida	
Verdadero positivo: 2 (08,33%)	Falso positivo: 5 (20,83%)
Falso negativo: 1 (04,17%)	Verdadero negativo: 16 (66,67%)

Tabla 4.12. Matriz de confusión de generación de alertas en AutoNorteSur.

Como se observa, se tomaron 24 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 2 de las 24 muestras, el sistema detectó correctamente una salida de carril y emitió una alerta temprana correspondiente al 08,33%.
- **Falsos negativos:** Este valor indica que, en 1 de las 24 muestras, salió del carril, pero el sistema no detectó la respectiva salida.
- **Falsos positivos:** Este valor indica que, en 8 de las 24 muestras, el vehículo no salió del carril, pero el sistema detectó una falsa salida correspondiente al 33,33%.
- **Verdaderos negativos:** Este valor indica que, en 16 de las 24 muestras, el vehículo circuló correctamente y el sistema no emitió ninguna alerta. Esta es la situación más común lo que representa un comportamiento silencioso y tranquilo presentado en las condiciones normales de manejo correspondiente al 66,67%.

Esta prueba mantuvo condiciones muy similares a la prueba anterior realizada en el trayecto inverso con relación a la complejidad visual dada por la señalización horizontal y el alto flujo vehicular. Para la detección de carril, se logró un 44,44% de verdaderos positivos lo cual indica que en casi la mitad de las veces el sistema logró identificar el carril satisfactoriamente.

En contraste, el sistema presentó una tasa considerable de falsos negativos del 31,48% justificado parcialmente por un trayecto de aproximadamente 3-4 segundos donde una motocicleta circulaba justo en frente del vehículo sobre la línea izquierda del carril, así

como la señalización horizontal en un tramo diferente que también generó conflicto con la detección del carril. Asimismo, se tiene un 24,07% de falsos positivos causados por la detección errónea de los símbolos pintados en la vía mencionados anteriormente que compartían patrones visuales similares a la figura y distancia real de la delimitación del carril. Igual que en todas las pruebas anteriores, no se tienen verdaderos negativos dado que existe una presencia constante y continua de las líneas que denotan los carriles a lo largo del recorrido. Esto podría ser solventado con el uso de un filtro de Kalman extendido para estimar la continuidad del carril o el uso de redes neuronales donde se incluyan resultados de este tipo en la etapa de entrenamiento.

En lo que respecta al sistema de alerta temprana de salida de carril, de la misma forma que en las pruebas anteriores donde únicamente se evaluaron los casos donde la detección de carril fue acertada, se observó una activación precisa de la alerta temprana en el 08,33% de los casos. A pesar de que este valor es bajo, corresponde a la naturaleza del trayecto urbano donde los movimientos laterales son bajos y se realizan con precaución. También se presentaron falsos positivos en un 20,83% provocados por maniobras evasivas comunes en este tipo de escenarios tales como evitar baches, tapas de alcantarilla, motociclistas o algunos vehículos que invaden el carril, así como cambios en la geometría de la carretera. Esto puede solventarse mediante el uso de región dinámica de interés donde se tenga una ligera tolerancia a los movimientos involuntarios o irrelevantes para la detección. Se presentó un falso negativo correspondiente al 04,17% restante donde se atribuyó al momento preciso donde hubo salida del carril, pero el sistema tuvo un retraso (lag) para la respuesta ante esa detección. Por último, el sistema evitó alertas innecesarias en un 66,67% de las muestras clasificados como verdaderos negativos lo cual representa un buen comportamiento.

4.4. Pruebas de campo

En las pruebas de campo, se analizaron los siguientes escenarios:

- **PalmiraNoSignal:** Esta prueba se realizó en el trayecto Palmira-Cali en la vía Panamericana (también conocida como Ruta 25) en una zona donde se hizo una reciente reparación completa del asfalto de la carretera por lo cual la vía se encontraba sin los carriles debidamente marcados, únicamente indicados con tiza por lo cual es un escenario completamente desafiante donde se busca que el sistema no haga ningún reconocimiento para así evaluar la capacidad de afrontar escenarios donde no debe de haber ningún tipo de respuesta por parte del sistema.
- **NoiseCerrito:** Esta prueba se realizó en el mismo trayecto Cerrito-Palmira indicado en Cerrito1 en sentido contrario con fallas en la adquisición de datos. Como primer inconveniente, se tiene que las condiciones climáticas eran muy poco favorables en términos de visibilidad debido a que había neblina en la carretera. En segundo lugar, se estaba conduciendo a una velocidad de 120km/h lo cual indica un exceso de velocidad provocando más vibraciones, movimientos involuntarios de la cámara y distorsión en la imagen. Por último, también se tiene que el soporte del dispositivo de adquisición de video fue biomecánico, es decir, lo sostuvo un acompañante sobre el parabrisas con el brazo lo cual limitó la

estabilidad, nivelación, ajuste y se agregó ruido dado a las vibraciones del vehículo junto con los movimientos involuntarios naturales de la mano del sujeto que lo sostuvo.

4.4.1. PalmiraNoSignal

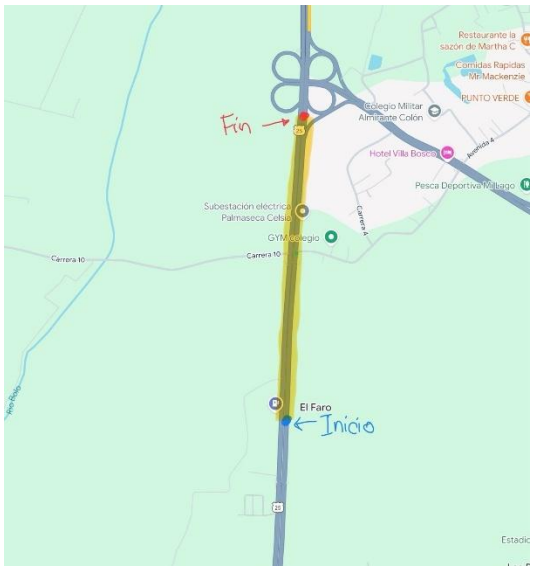


Figura 4.9. Recorrido de PalmiraNoSignal

El objetivo principal de la primera prueba de campo es evaluar la robustez del sistema de alerta temprana de salida de carril en condiciones completamente desfavorables donde no existe demarcación visible de los carriles sobre la carretera. Esto permite validar que el sistema no genera alertas innecesarias cuando hay ausencia de datos visuales válidos minimizando la incidencia de falsos positivos que ponen en riesgo la confianza del usuario.

La prueba se desarrolló sobre un segmento de la Ruta 25 en el trayecto Palmira-Cali, en sentido norte-sur, desde la estación de Terpel San Agustín hasta la intersección vial hacia el municipio de Rozo-Yumbo tal como se muestra en la Figura 4.9. Este recorrido fue seleccionado estratégicamente dado que en el momento de la grabación se presenta la ausencia total de las demarcaciones viales centrales y laterales lo cual constituye una prueba de resiliencia del sistema para evidenciar que el algoritmo de detección de carriles no genera resultados innecesarios o inválidos.

La duración total del recorrido fue de 40 segundos, durante los cuales se extrajeron 2 imágenes por cada segundo obteniendo así un total de 80 muestras. Para esto, se utilizó un script en Python que extrajo una imagen en formato *.png cada 15 fotogramas teniendo en cuenta que la tasa de actualización del video es de 30 fotogramas por segundo (fps).

Figura 4.10. Fotograma aleatorio extraído.

Durante la ejecución de la prueba, el sistema de alerta temprana de salida de carril presentó un desempeño óptimo y acorde a los objetivos planteados. No se registraron falsos positivos en ninguna de las imágenes analizadas lo cual representa la capacidad del sistema de discriminar la ausencia de líneas de carril reales y obviar interpretaciones erróneas causadas por elementos ambientales como bordes de vía, sombras o irregularidades en el pavimento.

4.4.2. NoiseCerrito

La segunda prueba de campo se desarrolló en el trayecto Cerrito-Palmira en sentido contrario al registrado en la prueba inicial Cerrito1 tal como se muestra en la Figura 4.11 y tuvo factores que constituyeron dificultades que afectaron la confiabilidad de los datos adquiridos.

La duración total del recorrido fue de 2 minutos y 30 segundos (150 segundos), durante los cuales se extrajeron 2 imágenes por cada segundo obteniendo así un total de 80 muestras. Para esto, se utilizó un script en Python que extrajo una imagen en formato *.png cada 15 fotogramas teniendo en cuenta que la tasa de actualización del video es de 30 fotogramas por segundo (fps).

Figura 4.11. Recorrido de NoiseCerrito

Las condiciones climáticas eran adversas presentándose neblina densa en la carretera lo cual disminuyó significativamente la visibilidad desencadenando en un reto importante para el aplicativo dado que este factor reduce la claridad de la demarcación de los carriles y se vuelven más propensos a ser confundidos con otros elementos viales. La velocidad de conducción registrada fue de aproximadamente 120km/h lo cual excede el límite recomendado para este tipo de vías aumentando las vibraciones y movimientos involuntarios del dispositivo de adquisición del video generando aún más distorsión en el video. Por último, el método de sujeción del dispositivo complementó la inestabilidad el cual fue sostenido por un acompañante diestro con el brazo derecho sobre el parabrisas. La sujeción biomecánica mencionada limitó la estabilidad, nivelación y ajuste fino del dispositivo agregando ruido visual debido a las vibraciones del vehículo combinadas con los movimientos naturales e involuntarios de la mano del operador. Para ello, se hicieron dos matrices de confusión independientes donde la primera corresponde a la detección de carril y con los valores que arrojaron verdadero positivo, se realiza la matriz de generación de alertas de salida temprana.

Detección de carril	
Verdadero positivo: 37 (12,33%)	Falso positivo: 2 (0,67%)
Falso negativo: 261 (87,00%)	Verdadero negativo: 0 (00,00%)

Tabla 4.13. Matriz de confusión de detección de carril en NoiseCerrito.

Como se observa en la Tabla 4.12. se tomaron 300 muestras de las cuales:

- **Verdaderos positivos:** Este valor indica que, en 37 de las 300 muestras, el sistema detectó correctamente la presencia de los carriles mostrados y trazó de manera correcta su forma y estructura. Esto corresponde al 12,33% del total de muestras lo cual sugiere un nivel muy bajo de reconocimiento.
- **Falsos negativos:** Este valor crítico indica que, en 261 de las 300 muestras, el sistema no detectó los carriles así estuvieran presentes. Esto corresponde al 87,00% indicando un significativo margen de error. Esto se puede mejorar teniendo un sistema de soporte más firme y estable como el usado en las pruebas cuantitativas así como un algoritmo más robusto como redes neuronales para situaciones más desafiantes como la presentada en esta prueba,
- **Falsos positivos:** Este valor crítico indica que, en 13 de las 54 muestras, el sistema detectó carriles que en realidad no estaban. Esto corresponde al 24,07% y puede derivarse a señales de tránsito en el piso, confusión con vehículos circundantes y/o cambios bruscos de iluminación lo cual puede producir una alerta errónea.
- **Verdaderos negativos:** Este valor no tiene peso en esta prueba dado que en el proceso de toda la prueba hubo presencia constante e ininterrumpida de los carriles sobre los cuales se estaba circulando por lo cual se mantiene un valor de 0.

En la Tabla 4.13. se mostrará la matriz de confusión de la generación de alertas tempranas de salida de carril donde se analizaron los valores respecto a la matriz anterior en los fotogramas donde la detección de carril fue verdadero positivo.

Alerta temprana de salida	
Verdadero positivo: 0 (00,00%)	Falso positivo: 0 (00,00%)
Falso negativo: 0 (00,00%)	Verdadero negativo: 37 (100,00%)

Tabla 4.14. Matriz de confusión de generación de alertas en NoiseCerrito.

Como se observa, se tomaron 37 muestras de las cuales solo se registraron verdaderos negativos sin presencia de falsos positivos, falsos negativos ni verdaderos positivos. Este resultado indica que bajo de las escasas condiciones donde hubo un acertado reconocimiento de carril, el sistema no generó alertas debido a que el vehículo se encontraba debidamente centrado en su carril en estos momentos específicos.

4.5. Conclusiones

Las pruebas realizadas para comprobar el correcto funcionamiento del sistema de alerta temprana de salida de carril permitieron obtener resultados relevantes sobre el funcionamiento, desempeño, robustez y confiabilidad del aplicativo desarrollado.

Inicialmente, se tienen las pruebas de integración donde se observó el correcto funcionamiento del aplicativo bajo los 3 principales módulos: configuración, procesamiento y reportes. El módulo de configuración se encarga de almacenar los parámetros dados por el usuario en cada una de las etapas del procesamiento en el archivo settings.yaml para posteriormente ser tomadas por el aplicativo. Así mismo, se encarga del funcionamiento de la GUI con las interacciones con el usuario para ejecutar correctamente el aplicativo. En el módulo de procesamiento, se tiene la validación del funcionamiento de cada etapa del preprocesamiento y el procesamiento para probar valores en alguna etapa que presente inconvenientes o revisar que todo funciona cómo es debido. Por último, en el módulo de reportes, se tiene la generación de los archivos de video o una carpeta de imágenes correspondientes a un fotograma por segundo para almacenar el resultado de cada prueba para su posterior análisis.

Adicionalmente, se tienen las pruebas cuantitativas realizadas tanto en entornos urbanos como extraurbanos mostraron que el sistema es capaz de reconocer el trazado de los carriles en condiciones favorables logrando hasta un 69,31% de verdaderos positivos en vías extraurbanas (Prueba Cerrito2) y un 46,15% en entorno urbano (AutoNorteSur). A pesar de esto, también hay que resaltar algunos desafíos importantes relacionados con los falsos negativos alcanzando un 40,38% en escenarios urbanos causado principalmente por obstrucciones visuales producidas por vehículos, imperfecciones en el pavimento y señalización horizontal que generan ambigüedades en la detección y activan el supervisor de validez de línea. Los falsos positivos, aunque menos en vías extraurbanas (05,94%), fue más evidente en vías urbanas (13,46%) relacionadas con interpretaciones erróneas de las señales horizontales o los vehículos adyacentes con geometría similar y posición cercana al carril real.

Respecto con la generación de alertas tempranas, se presentó una baja sensibilidad en las pruebas urbanas con verdaderos positivos alrededor del 08,33% dado que la naturaleza del entorno y la duración de las pruebas hacen que las salidas de carril sean menos frecuentes y los movimientos laterales sean más controlados. Se destaca la alta tasa de falsos positivos en general resaltando la necesidad de mejora en la discriminación entre maniobras evasivas y salidas de carril genuinas.

En las pruebas de campo se contrastan dos pruebas con naturalezas y resultados muy diferentes. Por un lado, en la prueba PalmiraNoSignal el sistema demostró un desempeño ideal al no generar falsas alertas debido a la ausencia de líneas visibles validando la capacidad de discriminar condiciones no óptimas para la detección y evitando la generación de alertas falsas que afectan la confiabilidad del sistema.

Por otro lado, la prueba NoiseCerrito presentó una pérdida considerable con un valor importante de falsos negativos (87,00%) y contados valores de verdadero positivo (12,33%) atribuible a factores externos como la neblina, el exceso de velocidad y la inestabilidad del dispositivo de captura debido a la sujeción manual biomecánica.

Capítulo 5 CONCLUSIONES Y TRABAJOS FUTUROS

5.1. Conclusiones

A lo largo del desarrollo de este proyecto se cumplió de manera ordenada y concreta con los objetivos específicos planteados. El primero de ellos, relacionado con la revisión bibliográfica, permitió reunir información clave sobre técnicas de procesamiento de imágenes y estimación de carriles, lo cual fue fundamental para identificar los requerimientos funcionales del sistema. Entre ellos, destacan funciones como la posibilidad de configurar parámetros del sistema desde una interfaz gráfica, detectar de forma continua la ubicación del vehículo respecto a las líneas del carril, y generar alertas visuales o auditivas en caso de desviación. También se establecieron requerimientos no funcionales enfocados en garantizar una respuesta rápida, mantener el sistema estable en tiempo real, y permitir su uso en condiciones lumínicas cambiantes, como suele ocurrir en las vías colombianas.

Para cumplir el segundo objetivo, se empleó la metodología RUP, que facilitó la organización del trabajo mediante la elaboración de diagramas conceptuales, de clases y casos de uso. Esto ayudó a definir claramente la funcionalidad esperada de cada módulo y permitió prever cómo se integrarían entre sí.

En cuanto al tercer objetivo, se logró implementar el aplicativo en cuatro módulos bien definidos: el de configuración permite ajustar valores desde la interfaz; el de preprocesamiento transforma la imagen captada por la cámara usando filtros como Sobel y HSL; el de procesamiento identifica los carriles mediante transformaciones de perspectiva; y el módulo de alertas notifica al conductor con señales visuales y/o auditivas. Todos estos módulos funcionan de manera coordinada en una plataforma Jetson Nano, optimizada con CUDA para mejorar el rendimiento.

Finalmente, se cumplió el cuarto objetivo al realizar pruebas de integración funcional en la interfaz, así como evaluaciones cuantitativas y en campo. Estas se llevaron a cabo en distintas rutas del Valle del Cauca, y los resultados mostraron que el sistema cumple con lo requerido: las alertas se activan correctamente cuando se detecta una salida del carril, y la detección se mantiene precisa incluso en escenarios reales con variaciones en el entorno. Esto demuestra que el aplicativo no solo cumple con lo propuesto, sino que es apto para ser utilizado como una herramienta práctica de seguridad vial en contextos como el colombiano, donde este tipo de tecnologías aún no están ampliamente disponibles.También se destaca que para el uso de técnicas de visión artificial y de

estructuración de un aplicativo completo, confiable y robusto se debe hacer uso de un lenguaje de programación orientado a objetos. Para ello, se escoge el lenguaje interpretado Python, debido a su gran comunidad, documentación en visión artificial y por su facilidad para desplegar el aplicativo, así como la compatibilidad con CUDA.

Además, se puede evidenciar que a través de las pruebas realizadas en el capítulo 4 se puede demostrar que el sistema presenta un buen comportamiento en condiciones reales de los entornos urbanos y extraurbanos en el territorio colombiano en condiciones climáticas favorables.

En este trabajo se puede demostrar que bajo la metodología establecida como RUP, se puede desarrollar un aplicativo que reconozca las demarcaciones de los carriles y determinar la desviación del vehículo para así generar una alerta temprana al usuario, el cual, además de presentar resultados favorables, permite al usuario afinar los parámetros mejorando la flexibilidad y adaptabilidad del sistema. Además, como se demuestra en el capítulo 4, se presenta versatilidad entre ambos tipos de entorno en el territorio colombiano el cual se destaca por su bajo control de calidad en el pavimento y marcaciones de este.

Se detectó una alta vulnerabilidad del sistema ante cualquier alteración a nivel climatológico que afecte la visibilidad, así como la inestabilidad del dispositivo de captura evidenciado en la prueba de campo NoiseCerrito donde los resultados se alejaron de un comportamiento correcto por lo cual la robustez del aplicativo no fue la mejor.

Es importante recalcar que mediante Python es posible manipular grandes volúmenes de datos de manera fácil y precisa, como fue este caso donde se almacenaron y manejaron los arreglos de visión, las adaptaciones a la imagen en su procesamiento y la extracción de resultados en diferentes procesos como la construcción de la matriz de confusión.

5.2. Trabajos futuros

Con el fin de mejorar el trabajo realizado, se proponen diferentes alternativas que pueden mejorar las limitaciones encontradas y mejorar el rendimiento general del aplicativo. Para esto se plantean las siguientes alternativas para implementar en trabajos futuros:

- **Filtro predictivo:** Esta característica puede ser implementada bajo un filtro de Kalman extendido o similares para mejorar la respuesta ante obstáculos intermitentes en la vía como señalización horizontal, inconsistencia en la trayectoria de la línea marcada o pérdidas rápidas de la visibilidad por cambios de iluminación
- **Redes neuronales:** El uso de un modelo basado en redes neuronales puede mejorar la precisión del sistema debido a que en el entrenamiento de estas se puede discriminar los valores erróneos que pudieron presentarse en este trabajo sacrificando rendimiento.
- **Región dinámica de interés:** Esta característica facilita el uso y reduce el tiempo de preparación entre pruebas dado que los puntos de origen de la transformada de perspectiva se asignan manualmente y se debe comprobar mediante el Debug que

hayan quedado bien calibrados. Al realizar esto automáticamente, se mejora la precisión de los puntos y facilitan el uso al usuario.

- **Cámara en tiempo real:** Es importante realizar el montaje de la cámara con soportes estables y en tiempo real en un vehículo para así realizar una prueba de ruta y continuar con el proceso de implementación.

Bibliografía

[1] I. Gamal and A. Badawy, *A Robust, Real-Time and Calibration-Free Departure Warning System*.

[2] K. Acosta and J. Romero, "Cambio recientes en las principales causas de mortalidad en Colombia," 2014.

[3] W. Arias-Rojas Politecnico di Milano, "Road safety Audit Experience in Puerto Rico." [Online]. Available: http://www.roadwaysafetyaudits.org

[4] J. B. Cicchino and D. S. Zuby, "Prevalence of driver physical factors leading to unintentional lane departure crashes," *Traffic Inj. Prev.*, vol. 18, no. 5, pp. 481–487, 2017, doi: 10.1080/15389588.2016.1247446.

[5] TJ, "Lane Departure Warning System," http://usedsemitrailers.com/lane-departure-warning-system/.

[6] C. Marin and J. Sanchez, "ESTUDIO DE LAS TECNOLOGÍAS DE SEGURIDAD PASIVA Y ACTIVAPRESENTES EN LOS VEHÍCULOS VENDIDOS EN COLOMBIA".

[7] J. Hu, S. Xiong, Y. Sun, J. Zha, and C. Fu, "Research on lane detection based on global search of dynamic region of interest (DROI)," *Appl. Sci.*, vol. 10, no. 7, Apr. 2020, doi: 10.3390/app10072543.

[8] B. Benligiray, C. Topal, and C. Akinlar, "Video-based lane detection using a fast vanishing point estimation method," in *Proceedings - 2012 IEEE International Symposium on Multimedia, ISM 2012*, 2012, pp. 348–351. doi: 10.1109/ISM.2012.70.

[9] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020, doi: 10.1109/TVT.2019.2949603.

[10] J. Baili, M. Marzougui, A. Sboui, S. Lahouar, and M. Hergli, *Lane Departure Detection Using Image Processing Techniques*. IEEE, 2017.

[11] D. Chang *et al.*, *Multi-lane Detection Using Instance Segmentation and Attentive Voting*. 2019.

[12] C. Valencia-Payan, J. Muñoz-Ordóñez, and L. Pencue-Fierro, "Driver-Assistant System Using Computer Vision and Machine Learning," *Rev. Fac. Ing.*, vol. 29, no. 54, 2020, doi: 10.19053/01211129.v29.n54.2020.11760.

[13] X. Wei, Z. Zhang, C. Zongjun, and W. Feng, *Research on Lane Detection and Tracking Algorithm Based on Improved HoughTransform*.

[14] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8834, pp. 454–461, 2014, doi: 10.1007/978-3-319-12637-1_57.

[15] R. Kala, “Advanced Driver Assistance Systems,” *On-Road Intell. Veh.*, pp. 59–82, Jan. 2016, doi: 10.1016/B978-0-12-803729-4.00004-0.

[16] X. An, M. Wu, and H. He, “A novel approach to provide lane departure warning using only one forward-looking camera,” *Proc. 2006 Int. Symp. Collab. Technol. Syst. CTS 2006*, vol. 2006, pp. 356–362, 2006, doi: 10.1109/CTS.2006.9.

[17] M. R. Haque, M. M. Islam, K. S. Alam, H. Iqbal, and M. E. Shaik, “A Computer Vision based Lane Detection Approach,” *Int. J. Image, Graph. Signal Process.*, vol. 11, no. 3, pp. 27–34, 2019, doi: 10.5815/ijigsp.2019.03.04.

[18] S. Lim, D. Lee, and Y. Park, “Lane Detection and Tracking Using Classification in Image Sequences,” *KSII Trans. Internet Inf. Syst.*, vol. 8, pp. 4489–4501, Dec. 2014, doi: 10.3837/tiis.2014.12.014.

[19] R. Jiang, R. Klette, S. Wang, and T. Vaudrey, “Lane Detection and Tracking Using a New Lane Model and a Distance Transform,” *http://www.mi.auckland.ac.nz/tech-reports/MItech-TR-39.pdf*, vol. 22, Jul. 2011, doi: 10.1007/s00138-010-0307-7.

[20] M. Li, Y. Li, and M. Jiang, “Lane Detection Based on Connection of Various Feature Extraction Methods,” *Adv. Multimed.*, vol. 2018, pp. 1–13, Aug. 2018, doi: 10.1155/2018/8320207.

[21] T. Y. Teo, R. Sutopo, J. M.-Y. Lim, and K. Wong, “Innovative lane detection method to increase the accuracy of lane departure warning system,” *Multimed. Tools Appl.*, vol. 80, no. 2, pp. 2063–2080, 2021, doi: 10.1007/s11042-020-09819-0.

[22] Z. W. Kim, “Robust lane detection and tracking in challenging scenarios,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008, doi: 10.1109/TITS.2007.908582.

[23] C. M. Martínez and D. Cao, “iHorizon-enabled energy management for electrified vehicles,” *iHorizon-Enabled Energy Manag. Electrified Veh.*, pp. 1–425, Jan. 2018, doi: 10.1016/C2017-0-02869-0.

[24] C. M. Martínez and D. Cao, “iHorizon-enabled energy management for electrified vehicles,” pp. 1–425, 2018, doi: 10.1016/C2017-0-02869-0.

[25] Y. Kortli, M. Marzougui, B. Bouallegue, J. S. C. Bose, P. Rodrigues, and M. Atri, “A novel illumination-invariant lane detection system,” *2017 2nd Int. Conf. Anti-Cyber Crimes, ICACC 2017*, vol. 1, pp. 166–171, 2017, doi: 10.1109/Anti-Cybercrime.2017.7905284.

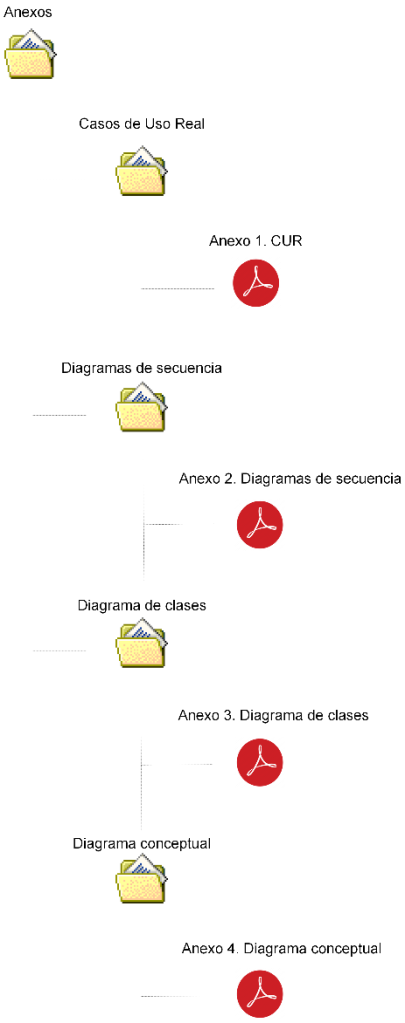
[26] S. H. Tsai and Y. H. Tseng, “A novel color detection method based on HSL color space for robotic soccer competition,” *Comput. Math. with Appl.*, vol. 64, no. 5, pp. 1291–1300, 2012, doi: 10.1016/j.camwa.2012.03.073.

[27] C. Ashbacher, “IBM Rational Unified Process Reference and Certification Guide Solution Designer.,” *J. Object Technol.*, vol. 7, no. 6, p. 53, 2008, doi: 10.5381/jot.2008.7.6.r1.

[28] A. Pastrana, “LEY 769 DE 2002,” *D. Of. 44.893 Agosto 07 2002*, vol. 2002, no. Agosto 6, p. 649, 2002.

Anexos

En esta sección, se muestran los Anexos relacionados con la metodología RUP alojados en la siguiente ruta de GoogleDrive tal como se muestra en la Figura Anexos.



En la carpeta raíz, se encuentran divididos los anexos por subcarpetas distribuidas de la siguiente forma:

- **Casos de Uso Real:** En esta subcarpeta se encuentra el archivo Anexo 1. CUR.pdf correspondiente a las tablas que refieren a los Casos de Uso Real planteados en la metodología RUP.
- **Diagramas de secuencia:** En esta subcarpeta se encuentra el archivo Anexo 2. Diagramas de secuencia.pdf correspondiente a los Diagramas de Secuencia planteados en la metodología RUP.
- **Diagrama de clases:** En esta subcarpeta se encuentra el archivo Anexo 3. Diagrama de clases.pdf correspondiente al Diagrama de clases planteados en la metodología RUP.
- **Diagrama conceptual:** En esta subcarpeta se encuentra el archivo Anexo 4. Diagrama conceptual.pdf correspondiente al Diagrama conceptual planteados en la metodología RUP.