

Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería



Sistema de Realidad Aumentada Basado en
Visión para Sistemas de Seguridad Activa en
Motocicletas

TESIS

que para obtener el grado de
Maestro en Ingeniería

Presenta:

Jose Isabel Garcia Rocha

Bajo la dirección de:

Dr. Leocundo Aguilar Noriega

Dr. Mauricio Alonso Sanchez Herrera

Tijuana B.C., México

Diciembre de 2022

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

Folio No.338

Tijuana, B.C., a 24 de noviembre del 2022

C. Jose Isabel Garcia Rocha
Pasante de: Maestría en Ingeniería
Presente

El tema de trabajo y/o tesis para su examen profesional, en la

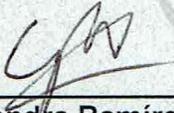
Opción TESIS

Es propuesto, por los C. Dr. Leocundo Aguilar Noriega y
Dr. Mauricio Alonso Sánchez Herrera

Quienes serán los responsables de la calidad del trabajo que usted presente, referido al tema “Sistema de realidad aumentada basado en visión para sistemas de seguridad activa en motocicletas”.

El cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I. INTRODUCCIÓN
- II. MARCO TEÓRICO
- III. DESARROLLO
- IV. RESULTADOS
- V. CONCLUSIONES Y TRABAJO FUTURO


Dra. Ana Alejandra Ramírez Rodríguez
Sub-Directora


Dr. Mauricio Alonso Sánchez Herrera
Co - Director de Tesis


M.C. Roberto Alejandro Reyes Martínez
Director


Dr. Leocundo Aguilar Noriega
Director de Tesis



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA
COORDINACIÓN DE
POSGRADO E INVESTIGACIÓN

Agradecimientos

Si dos personas merecen ser mencionadas en los agradecimientos, así como dedicarles este trabajo sin lugar a dudas, son mis directores, maestros y tutores: el Dr. Leocundo Aguilar y el Dr. Mauricio A. Sánchez a quienes les agradezco su apoyo, enseñanzas, consejos y correcciones. Si este trabajo se logró satisfactoriamente, tiene en gran mérito el estar bajo su tutela.

Así mismo le dedico este trabajo a la persona que me apoyo de manera personal con sus enseñanzas y el ánimo antes y durante este trabajo, con cariño te agradezco Andrea Sánchez por tu compañía y paciencia.

A todos ellos, les agradezco.

Resumen

En este trabajo se desarrolla un sistema de seguridad activa vehicular basado en la implementación de mono visión utilizando realidad aumentada el cual estará instalado en la parte trasera de una motocicleta apuntando hacia los vehículos que se acercan desde atrás a esta, sirviendo como una alternativa a los métodos que hacen uso de sensores ultrasónicos, LIDAR, Radar, óptico estéreo o sensores de proximidad que suelen tener un costo mayor y/o requieren de mayor computo. Se aprovechará de la visión por computadora para detectar los vehículos, del *tracking* en tiempo real para realizar un seguimiento identificando y asignando atributos únicos a cada uno de ellos, con los que se marcarán los que representen un riesgo para el motociclista.

Con ello se permitirá aumentar la seguridad vial del motociclista al otorgarle mayor información sobre la vía en la que transita. Estará dividido en dos módulos, uno colocado en el casco (siendo este de reducido tamaño y peso) y otro en la motocicleta, por lo que se podrá utilizar de forma ambulatoria así como también con adaptaciones pasar a otro tipo de vehículo. Las capturas obtenidas de la vía permitirán ser representadas como obstáculos con atributos como distancia, velocidad relativa y ángulo para así poder analizar la situación vial en tiempo real y darle esa información al usuario de manera entendible logrando reducir el riesgo de sufrir un incidente en la vía pública.

Abstract

In this work we develop an active vehicle safety system based on the implementation of mono vision using augmented reality which will be installed on the rear of a motorcycle pointing towards vehicles approaching from behind it, serving as an alternative to methods that make use of ultrasonic sensors, LIDAR, Radar, optical stereo or proximity sensors that usually have a higher cost and / or require more computation. We will take advantage of computer vision to detect vehicles, real-time tracking to track them, identifying and assigning unique attributes to each one of them, which will be used to mark those that represent a danger to the motorcyclist.

This will increase motorcyclists' road safety by providing them with more information about the road on which they are traveling. It will be divided into two modules, one placed on the helmet (being of reduced size and weight) and the other on the motorcycle, so it can be used in an ambulatory way as well as with adaptations to pass to another type of vehicle. The captures obtained from the road will be represented as obstacles with attributes such as distance, relative speed and angle in order to analyze the road situation in real time and give this information to the user in an understandable way, thus reducing the risk of suffering an incident on public roads.

Tabla de Contenido

	Pag.
Agradecimientos	3
Resumen	4
Abstract	5
1 Introducción	14
1.1 Planteamiento del problema	14
1.2 Justificación	17
1.3 Objetivos	18
1.3.1 Objetivo General	18
1.3.2 Objetivos particulares	18
1.4 Hipótesis	19
1.5 Metas	19
1.6 Metodología	19
2 Marco Teórico	21
2.1 Antecedentes	21
2.1.1 Sistemas de seguridad activa	21
2.1.2 Inteligencia artificial	24
2.1.3 Machine Learning	25
2.1.4 Deep Learning	25
2.1.5 Redes neuronales convolucionales	26
2.1.6 Arquitecturas de detección de objetos	29
2.1.7 Visión por computadora	31

2.1.8	Filtro de Kalman	36
2.1.9	Método de evaluación IOU	37
2.1.10	Algoritmo Húngaro	38
2.1.11	Seguidor SORT	38
2.1.12	Realidad aumentada	39
2.1.13	Detección de vehículos con visión	43
2.1.14	Microcontroladores	44
2.1.15	Estado del arte	45
3	Desarrollo	51
3.1	Módulo principal	51
3.1.1	Componentes	51
3.1.2	Implementación	54
3.2	Módulo escáner	74
3.2.1	Componentes principales	74
3.2.2	Microcontrolador ESP32-CAM	74
3.2.3	OBDII Serial	75
3.2.4	Diseño y conexión	77
3.2.5	Funcionalidad	78
3.3	Módulo realidad aumentada	80
3.3.1	Componentes principales	80
3.3.2	Microcontrolador ESP8266	80
3.3.3	Pantalla TFT	81
3.3.4	Diseño y conexión	83
3.3.5	Funcionalidad	84
4	Resultados	87
4.1	Pruebas	87
4.1.1	Detección	87
4.1.2	Seguimiento	90

4.1.3	Estimación de distancia	93
4.1.4	Estimación de velocidad	94
4.1.5	Estimación de riesgo	96
4.2	Discusión de resultados	98
5	Conclusiones y trabajo futuro	105
5.1	Conclusiones	105
5.2	Trabajo futuro	106
Apéndices	109
5.3	Código estimar distancia	109
5.4	Código estimar velocidad	111
5.5	Código estimar riesgo	114
Referencias	117

Lista de Tablas

3.1	Resultado de la calibración con cámara V2.1 e Infrarroja.	67
3.2	Altura promedio utilizada por clase.	68
3.3	Impacto de un píxel en la estimación de la distancia.	70
4.1	Resultados de la precisión en las detecciones de la prueba 1.	88
4.2	Resultados de la precisión en las detecciones de la prueba 2.	88
4.3	Resultados de la precisión en las detecciones de la prueba 3.	89
4.4	Resultados de la precisión en las detecciones de la prueba 4.	89
4.5	Resultados de la precisión en las detecciones de la prueba 5.	90
4.6	Resultados del seguimiento en la prueba 1.	91
4.7	Resultados del seguimiento en la prueba 2.	91
4.8	Resultados del seguimiento en la prueba 3, AN de Automóvil.	92
4.9	Resultados del seguimiento en la prueba 4.	92
4.10	Resultados del seguimiento en la prueba 5.	93
4.11	Resultados al estimar la distancia en la prueba 1.	93
4.12	Resultados al estimar la distancia en la prueba 2.	94
4.13	Resultados al estimar la distancia en la prueba 3.	94
4.14	Resultados al estimar la velocidad en la prueba 1.	95
4.15	Resultados al estimar la velocidad en la prueba 2, AN de Automóvil.	96
4.16	Resultados al estimar la velocidad en la prueba 3.	96
4.17	Resultados al estimar el nivel de riesgo en la prueba 1.	97
4.18	Resultados al estimar el nivel de riesgo en la prueba 2.	97

4.19 Resultados al estimar el nivel de riesgo en la prueba 3. 98

Lista de Figuras

1.1	Una sola licencia para conducir ambas motocicletas.	16
1.2	Vista por los espejos retrovisores de una motocicleta.	17
2.1	Sistema anti-bloqueo de frenos en funcionamiento.	22
2.2	Sistema de control de tracción en funcionamiento.	22
2.3	Prototipo Varjo 20 20.	24
2.4	Estructura de una red neuronal convolucional.	26
2.5	Ejemplo de convolución.	28
2.6	Faster R-CNN con arquitectura de dos fases.	30
2.7	Arquitectura de una fase vs dos fases.	31
2.8	Fotografía de rosa.	33
2.9	La clasificación de la imagen nos puede ofrecer distintas salidas.	34
2.10	La localización nos da en coordenadas una caja delimitadora sobre el objeto.	34
2.11	La detección ubica en coordenadas una caja delimitadora y lo clasifica.	35
2.12	La segmentación de instancias separa con color los píxeles de cada objeto y los clasifica.	36
2.13	Visión completa del filtro de Kalman, sus ecuaciones y operaciones [1].	37
2.14	Método de evaluación IOU.	38
2.15	Matriz con ejemplo para el algoritmo Húngaro.	38
2.16	Ejemplo de funcionamiento del seguidor SORT.	39
2.17	App de Honda [®] con motocicleta en realidad aumentada[2].	40
2.18	Las Oculus Quest 2 son unas gafas de realidad virtual [3]	41

2.19	El juego Pokémon Go permite visualizar elementos sobre nuestro mundo[4]. .	42
2.20	Las HoloLens son unas gafas de realidad mixta[5].	42
2.21	AutoPilot de Tesla [®]	44
2.22	Sistemas de seguridad en los vehículos modernos.	45
2.23	Sistema de BMW [®] para cambio de carril seguro.	46
2.24	Generador de hipótesis ROI (Regiones de interés).	47
2.25	Diferencia de colores en la carretera al transitar un vehículo.	48
2.26	Detección de vehículos con distancia focal cercana y lejana.	49
2.27	Sistema prototipo montado.	50
3.1	Raspberry Pi 4 [6].	52
3.2	Cámaras utilizadas en el proyecto [7].	53
3.3	TPU Coral USB [8].	54
3.4	Cámara conectada al puerto CSI-2 del RPI4.	55
3.5	Reducción de 640X640 a 320X320.	55
3.6	Detecciones de vehículos en un fotograma tomado por el sistema.	57
3.7	Ejemplo de vehículos detectados.	57
3.8	Obtención de la lista de vehículos filtrados por clase.	58
3.9	Ejemplo de detección doble.	59
3.10	Se puede apreciar en la detección como uno es mas ancho y el otro mas alto.	60
3.11	Ejemplo de estimación de ángulo.	60
3.12	Estimación de tamaño con respecto al tamaño del fotograma y el ángulo.	61
3.13	Ejemplo del rango de filtrado para eliminar detecciones fuera del área de interés.	61
3.14	Sistema de detección funcionando en el CPU versus el TPU.	62
3.15	Seguidor identificando al vehículo número 12 durante 16.83 segundos.	63
3.16	Ejemplo de fiabilidad del seguidor.	63
3.17	Semejanza de triángulos.	64
3.18	Lente y longitud focal.	65

3.19	Captura de calibración con cámara V2.1 infrarroja.	66
3.20	Estimación de distancia en dos vehículos detectados.	68
3.21	Prueba de campo en ciudad.	69
3.22	Prueba de campo en una calle secundaria.	69
3.23	Algoritmo estimación de riesgo.	73
3.24	Función y descripción de los pines del ESP32 Camera [9].	74
3.25	Escáner ELM327 OBDII [10].	76
3.26	Conexión del módulo ELM 327 OBDII [10].	77
3.27	Diagrama de flujo del módulo Escáner.	78
3.28	Función y descripción de los pines del ESP8266 [11].	80
3.29	Pantalla TFT de 1.3" con 240x240 píxeles [12].	82
3.30	Conexión del módulo de realidad aumentada.	83
3.31	Diagrama de flujo del módulo Realidad Aumentada.	84
3.32	Mapa dibujado en la pantalla TFT.	85
4.1	Gráfica mostrando el seguimiento del sistema a un vehículo en las pruebas. .	103
5.1	Valor en pesos mexicanos del costo de las piezas del sistema.	106

Capítulo 1

Introducción

1.1 Planteamiento del problema

La problemática que se abordará es sobre la seguridad vial de los motociclistas en México, existen 4,083,422 de motocicletas registradas por el Instituto Nacional de Estadística y Geografía (INEGI)[13] en el año 2018 en el país mencionado anteriormente, y durante este mismo año se registraron 37,094 colisiones de motocicletas[14] según INEGI sin contar las colisiones producidas en vialidades federales y las motocicletas no registradas o dadas de alta, según el gobierno de México del 2000 al 2016 el número de motociclistas muertos se ha multiplicado por cien[15]. Existe una gran cantidad de razones por las cuales se producen estas colisiones en el país, aquí se enumeran algunas de ellas:

1. Motos de mala calidad, acompañadas de componentes que aumentan la posibilidad de un incidente (como podrían ser llantas con poca adherencia en el asfalto o concreto, sistemas de freno con frenos, balatas o bombas de frenado de mala calidad produciendo un frenado bajo, componentes mecánicos como tornillos, engranes, sellos que se aflojan y sueltan con el motor en funcionamiento pudiendo provocar la abrupta detención de la motocicleta, motores y transmisiones de baja durabilidad que pueden detenerse en funcionamiento dejando al motociclista sin empuje en la vía pública, etc.) que tienen más del 70% del mercado mexicano (por mencionar un ejemplo la marca de motos Italika[®] cuenta con alrededor de 3000[16] puntos de venta en el país, esta compañía no se suele reconocer por su calidad en materiales y durabilidad de sus motocicletas, pues, son muy económicas, mientras que Honda[®] la marca con mayor número de ventas en

el mundo de motocicletas y marca reconocida por su calidad y durabilidad solo cuenta con alrededor de 188 puntos de venta[17]).

2. Existe corrupción en las pruebas de licencias[18], en algunos casos existen ventas de licencias sin nada más que el dinero y un nombre para ponerle, esto produce la posibilidad de que cualquier persona con nulos conocimientos maneje una motocicleta, del mismo modo se debería de aplicar una distinción de licencias de motociclista, pues no es lo mismo manejar una motocicleta de repartidor con 8 caballos de vapor (cv) con peso de 115KG a una motocicleta de carreras con 250 cv y 200KG
3. En México algunas marcas de motocicletas tienen grandes facilidades de pago[19], por lo que permiten sacar una motocicleta de baja calidad con materiales que aportan poca seguridad al conductor pagando una cantidad accesible por semana, lo cual aumenta el número de motociclistas en las vialidades, aumentado así las probabilidades de un incidente vial.
4. Existen marcas de cascos para motociclistas que simplemente no cumplen ninguna certificación[20] y en caso de un accidente no podrían servir más que un gorro para proteger el cráneo del golpe, esto es principalmente por los precios bajos y la nula educación de los motociclistas novatos, ya que no conocen la diferencia entre un casco certificado versus uno que no lo está.
5. La motocicleta es un vehículo motorizado, que debe cumplir con las leyes de tránsito, pagar seguro, portar placas, requiere de mantenimiento y revisiones periódicas, además el motociclista debe contar con licencia de conducir, estas pueden rondar en peso desde 75KG hasta 480KG contando líquidos más aún haría falta sumar el peso del piloto, equipaje y copiloto en caso de llevar uno, por lo que son en toda regla un vehículo automotor capaz de representar un riesgo para el mismo conductor como para los demás y debe conducirse con el debido cuidado y respeto que eso representa.

Para buscar mitigar estos números, el gobierno de México ha implementado campañas

de concientización para los motociclistas y para los automovilistas, de igual manera se exige que se cuente con seguro en la motocicleta para circular, de momento no se tiene en sí una solución para reducir la inseguridad vial de los motociclistas. Algunas posibles soluciones para aplicar:

1. Realizar correctamente las pruebas de manejo a los motociclistas, exigirles que tengan el conocimiento para conducir como es debido, crear una prueba que disponga de varios puntos a evaluar para poder comprobar que el solicitante cumple con lo requerido para obtener dicha licencia.
2. Dividir la licencia de conducir en categorías, ya que como anteriormente se mencionaba, no es lo mismo conducir una motoneta de 75 KG a una motocicleta de viajes con maletas y el equipamiento de 400 KG, por lo que se debería de fraccionar en categorías y realizar distintas pruebas para las mencionadas categorías.
3. Crear un dispositivo que le aporte una mayor protección al piloto, información sobre puntos ciegos e información importante para el conductor que le pueda ayudar a evitar colisiones en la vía o situaciones de riesgo.



(a) Italika® ST90 (77KG)



(b) BMW® R1250 GS Adventure (268KG)

Figure 1.1: Una sola licencia para conducir ambas motocicletas.

Con base en esta problemática se busca disminuir el riesgo que corre el motociclista al conducir la motocicleta en la vialidad.

1.2 Justificación

En México, el uso de la motocicleta ha ido en aumento desde la última década pasando de haber registradas 1,154,145 motocicletas en el 2010 a 5,260,954 motocicletas en el 2020 según INEGI [21] con lo que provoco un aumento en el total de colisiones a nivel nacional pasando del año 2016 de 32,892 colisiones a 42,667 colisiones en el 2019 según INEGI [22] siendo este un aumento del 29.71% en tan solo 3 años, para el año 2019 hubo un total de 608 muertos por accidente y 22,922 heridos en una colisión con motocicleta[23] por lo que se encuentra con un sector de conductores vulnerable a esta situación[15], pues están expuestos al ambiente, algunas de las protecciones que deben llevar sobre su cuerpo para aumentar sus defensas ante una colisión son chamarra, casco, guantes, protecciones en codos, espalda y rodillas, botas y pantalones especiales, y aun con todo ello no se asegura su completa seguridad, más al contar con una visión limitada de sus alrededores en la figura 1.2 podemos ver en el espejo retrovisor de la izquierda se ve la vista trasera obstruida por el hombro del motociclista limitada por el diseño y colocación de los retrovisores y en el retrovisor de la derecha se ve la vista trasera ideal.



Figure 1.2: Vista por los espejos retrovisores de una motocicleta.

Se pretende entonces ayudar a reducir el número de colisiones e incidentes en los que se ven involucrados los motociclistas, concientizar a los conductores en general sobre su vulnerabilidad y crear un dispositivo inteligente de realidad aumentada basado en visión que ayude a estos a evitar verse involucrados en colisiones con otros vehículos, para ello se

recopilara información que permita conocer las limitaciones de los motociclistas en la vía, con lo que se comprenderá cuáles son sus puntos flacos o puntos ciegos al conducir o estar detenidos.

Los resultados obtenidos se repartirán en patrones de distintos tipos de vehículos (motocicletas, automóviles, camiones, camiones articulados), presentaciones en congresos, para obtención de una imagen más clara sobre la inseguridad vial a la cual se enfrentan los motociclistas y para el desarrollo de un prototipo con el cual se eviten estas situaciones, dando como principal beneficiario a los motociclistas e indirectamente la población en general.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema de seguridad activa basado en visión que utiliza realidad aumentada para alertar por medio de avisos visuales al motociclista para apoyo a su seguridad vial, donde este puede detectar y alertar de una situación de riesgo a su alrededor que le pueda afectar directamente en un ambiente de tránsito en la vía.

1.3.2 Objetivos particulares

1. Identificar y seleccionar el tipo de sensores y su posición correcta para determinar las situaciones de riesgo para el motociclista.
2. Definir un subsistema de recopilación de datos de los sensores.
3. Establecer el tipo de manejo que recibirán los datos recopilados.
4. Detectar los obstáculos de los datos recabados a través del sensor óptico.
5. Realizar un seguimiento de los obstáculos detectados.
6. Calcular la distancia, ángulo y velocidad de los objetos seguidos.
7. Mostrar alertas al motociclista de acuerdo a los riesgos detectados.

1.4 Hipótesis

Es posible detectar y advertir las situaciones de riesgo para el motociclista en la vía con un dispositivo inteligente con realidad aumentada capaz de detectar y seguir vehículos en la vía pública.

1.5 Metas

1. Investigar del estado del arte.
2. Especificar los requerimientos que debe cumplir el sistema de seguridad vial.
3. Evaluar y seleccionar
 - Las técnicas de realidad aumentada en seguridad vial a utilizar.
 - Los posibles dispositivos a utilizar.
 - Los sensores a utilizar para la captura de datos en la vía.
 - Las técnicas de detección de objetos que se usarán sobre los datos recopilados.
4. Analizar los requerimientos para diseñar y desarrollar el sistema con inteligencia artificial para la seguridad vial.
5. Verificar el correcto funcionamiento del sistema.
6. Realizar pruebas de campo con el fin de ajustar y validar el sistema creado.

1.6 Metodología

Para lograr crear un dispositivo inteligente para suplir la necesidad en la sociedad sobre obtener mayor seguridad vial al conducir una motocicleta a partir del conocimiento adquirido en el estudio ingenieril. La cual está conformada por los siguientes puntos.

1. Reunir la bibliografía más reciente.
2. Obtener los conocimientos aplicados acerca del estado del arte actual de los sistemas existentes de seguridad activa que implementan soluciones basadas en visión.

3. Analizar el funcionamiento de los distintos métodos de seguridad activa previamente identificados.
4. Evaluar y proponer una solución a la problemática.
5. Crear el diseño conceptual del sistema.
6. Evaluar y seleccionar los componentes a utilizar, las técnicas de realidad aumentada, así como de detección y seguimiento de objetos.
7. Analizar y diseñar el funcionamiento y acoplamiento de los componentes.
8. Desarrollar el sistema e implementar el código necesario para su funcionamiento.
9. Recabar datos con la cámara desde la parte trasera de la motocicleta para su análisis y representación.
10. Detectar los objetos con ayuda de un modelo de inteligencia artificial en *tensorflow lite*.
11. Realizar el seguimiento y la asignación de identificadores de los objetos detectados.
12. Crear la aplicación móvil donde se obtendrá la velocidad GPS (*Global Positioning System*) de la motocicleta.
13. Calcular el ángulo, distancia y velocidad relativa de cada uno de los objetos detectados.
14. Implementar el algoritmo para analizar los datos de los cuales se obtendrán los niveles de riesgo hacia el motociclista.
15. Crear la interfaz de usuario a mostrar.
16. Implementar la realidad aumentada para mostrar la salida de los datos al conductor.
17. Realizar pruebas bajo un ambiente de campo, con el sistema colocado en la parte trasera de la motocicleta para determinar el rango de operación al que trabaja el sistema creado.

Capítulo 2

Marco Teórico

2.1 Antecedentes

2.1.1 Sistemas de seguridad activa

Los avances en tecnología han permitido un gran desarrollo en los sistemas electrónicos y comunicacionales, con lo que se ha podido crear sistemas más complejos que han beneficiado a todas las áreas [24], una de ellas es la industria automotriz [25], dando como ejemplo la automatización de los procesos de fabricación de vehículos [26], los sistemas de entretenimiento, las ayudas a la conducción, los sistemas de frenado de emergencia y los demás sistemas que se pueden encontrar en los distintos modelos disponibles en el mercado.

Todo ello conlleva un gran avance en temas de construcción, confort al conducir y claro, seguridad, pues sin esta última los vehículos aún contarían con parabrisas de vidrio, los cuales al suceder un accidente vial se quebraban y podían provocar daños severos a los ocupantes del vehículo [27], la seguridad que se ha agregado es tanto pasiva como activa, aún ahora se sigue investigando para ofrecer mejoras en los sistemas de seguridad [28]. Esta investigación está incentivada por ofrecer un mejor producto, como por una serie de estrictas normas y estándares impuestos por los gobiernos de cada nación y los mismos consumidores al comprar los vehículos con mejores opciones para proteger a sus tripulantes.

Los sistemas de seguridad activa tienen como función principal apoyar y auxiliar al conductor ante situaciones de riesgo donde el control del vehículo se vea comprometido y esto pueda llevar a un accidente vial [28].

Algunos de los sistemas de seguridad que podemos encontrar en los automóviles son:

Sistema antibloqueo de frenos (ABS)

ABS proviene de *antilock braking system* cuya traducción sería sistema anti-bloqueo de frenos.



Figure 2.1: Sistema anti-bloqueo de frenos en funcionamiento.

Su principal función es evitar que las ruedas se bloqueen al frenar de emergencia. Con lo que se evita la pérdida de control y estabilidad al frenar abruptamente.

Funciona gracias a los sensores de giro colocados en cada rueda, con los que se puede conocer la velocidad y aceleración de cada una de ellas, así como detectar si una de las ruedas está reduciendo su velocidad y se procede a reducir la presión al frenado para así evitar la pérdida de adherencia al suelo, al aumentar su velocidad e igualarla a la de las demás [29].

Sistema de control de tracción (TCS)

TCS proviene de *traction control system* cuya traducción sería sistema de control de tracción.

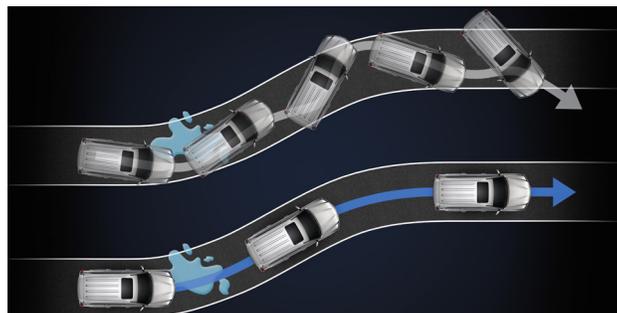


Figure 2.2: Sistema de control de tracción en funcionamiento.

Permite una mejora en la tracción y la seguridad, controlando el par motor y el frenado. Queda aclarar que el par motor es la fuerza necesaria para que el cigüeñal del motor gire, el cigüeñal es el eje que hace subir y bajar a los pistones de un motor.

Cuando el sistema detecta girar sin adherencia una rueda, este reduce el par motor y activa el freno en ella para de este modo controlar la tracción y evitar la pérdida de control [30].

Estos dos sistemas son parte de los sistemas de seguridad activa que por regularizaciones deben traer incorporados los vehículos hoy en día, pero existen algunos que aún son opcionales, ya que su precio es muy elevado, estos sistemas pueden ser: información de punto ciego, luces adaptativas, suspensión activa, aviso de cambio de carril involuntario, adaptativo de estabilidad, entre otros.

Así mismo los vehículos cuentan con una gran cantidad de sistemas enfocados en la seguridad instalados en ellos, los cuales van desde sensores ultrasónicos, de temperatura, de velocidad, de inclinación, de presión, de rotación, de aceleración, de fuerza, de par motor, de proximidad, de luz y aparte distintos tipos de cámaras, y algunos otros dispositivos los cuales utilizan sistemas de seguridad que le aportan al vehículo de una gran información sobre el entorno en el que se encuentra transitando y le permiten decidir de esta manera entre algunas opciones que este sistema tiene, ahora el asunto está en cómo llevar esa información al conductor, pues hasta el momento el vehículo es el que se entera de todo y acciona algún tipo de aviso sonoro o algún indicador, pero no aporta la información completa en un tiempo suficiente para que el conductor reaccione y evada el riesgo por lo que se busca darle una utilidad real a toda esta información que se obtiene por medio de todos estos sistemas, algunas compañías han aportado sus ideas sobre cómo resolver ese problema, la compañía de vehículos Volvo con su prototipo Varjo[®] 20|20 [31] el cual es un dispositivo provisto de unos lentes con una pantalla y dos cámaras, que funcionan como un intermediario entre el mundo real y los ojos del conductor, sobre esta pantalla aparte de mostrar lo que vería el conductor sin ella, se muestran avisos que el automóvil detecta con alguno de sus sistemas de seguridad,

entonces de este modo el conductor podría por ejemplo enterarse por medio de un aviso en pantalla que un animal cruzara la carretera justo enfrente de él, en algunos metros y este poder decidir que acción realizar para evitar dicho riesgo, este dispositivo utiliza realidad mixta y aun cuando es solo un prototipo aporta una utilidad que podría darle el visto bueno para verlo en algunos años como un accesorio de los vehículos.



Figure 2.3: Prototipo Varjo 20|20.

2.1.2 Inteligencia artificial

La inteligencia artificial (IA) vista de manera sencilla son los sistemas computacionales que buscan imitar la inteligencia humana para ejecutar tareas y estos como el humano pueden mejorar con cada iteración ejecutada a partir de los datos que van recopilando o que les entregan al entrenar, por lo que pueden aprender, resolver problemas y reconocer patrones. Debido al avance en esta área se ha llegado a crear dos subcampos, siendo estos el *machine learning* (ML) y el *deep learning* (DL).

La inteligencia artificial mejora su precisión y su umbral de aprendizaje es mayor mientras más datos se le suministren al entrenarla. Hoy en día con la gran aceptación del público por utilizar un dispositivo inteligente día a día con el cual aportan gran cantidad de sus datos, así como el auge del internet de las cosas (*Internet of things* (IoT)) que aumenta el número de sensores conectados a internet proveyendo de aún más datos y las demás tecnologías o empresas tecnológicas que aportan del mismo modo, todo esto añade más datos para analizar y aumentar la inteligencia del *machine learning* y el *deep learning*.

ML y DL se pueden dividir en técnicas de aprendizaje supervisadas y no supervisadas, las supervisadas son las que cuentan en los datos de entrenamiento con la salida deseada y las no supervisadas no [32].

2.1.3 Machine Learning

Machine learning es un derivado del IA, el cual proporciona la posibilidad de aprender a un sistema de los datos, reconocer patrones y tomar decisiones en lugar de aprender con la programación explícita, donde el programador indica paso por paso como debe hacerlo, ya que esta última es demasiado rígida y poco practica. En la programación explícita se busca generar una salida específica a partir de una entrada determinada y en ML se utilizan los datos para generar un modelo que a base de un patrón reconocido por ella a partir de ejemplos anteriores de entradas (y salidas en caso de técnicas supervisadas) generará la salida más adecuada. Mientras más datos se tengan y estos sean de mejor calidad para la tarea que se busca resolver, mejor será la precisión del modelo ML [32].

2.1.4 Deep Learning

Deep Learning se deriva de la IA y del ML, este método agrega capas sucesivas de algoritmos para aprender mejor de los datos. Los algoritmos no lineales de las capas procesan la información en múltiples etapas sucesivas, buscando interacciones complejas entre los datos analizados [33].

Las redes neuronales artificiales son la base del DL, son sistemas que buscan imitar la manera en la que funcionan las neuronas y distintos mecanismos del cerebro humano, por lo que buscan copiar el método de aprendizaje de este, por medio de una estructura flexible, organizada en capas formadas por neuronas las cuales son funciones matemáticas, su salida es una función no lineal dada de sus entradas y pesos [34]. Mientras más capas tenga la red, se consigue mayor complejidad a la hora de analizar los datos y esto ayuda aprender patrones más complejos de un número mayor de datos y con ello reducir el error [35].

2.1.5 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN del inglés *Convolutional Neural Network*) cuentan con un rendimiento superior a las demás redes neuronales cuando se trata de entradas de señales de imagen, video, habla o audio. Estas cuentan con tres tipos de capas, la capa convolucional, la capa de agrupamiento y la capa totalmente conectada.

La primera capa de la red convolucional siempre será la capa convolucional, al final siempre estará la capa completamente conectada y en el medio de ellas habrá una variación de capas de agrupamiento y de convolución. Mientras mayor sea el número de capas, mayor será la complejidad y podrá identificar características más complejas de la imagen que se esté procesando. Las primeras capas comenzarán identificando características más sencillas como podrían ser colores y bordes, mientras que las capas medianas podrán identificar partes más complejas, como podría ser en el caso de identificar características de un rostro humano, la nariz, la boca, los ojos, las cejas y demás partes grandes de un rostro pero sin ser estas el rostro completo, y las capas finales ya podrán identificar el todo del objeto [36]. Por lo tanto, de acuerdo a la distribución de las capas intermedias, el tamaño de cada una de ellas y el número de parámetros, se pueden distinguir distintas arquitecturas CNN, de estas las más conocidas son AlexNet [37], VGGNet [38], ResNet [39] e Inception [40].

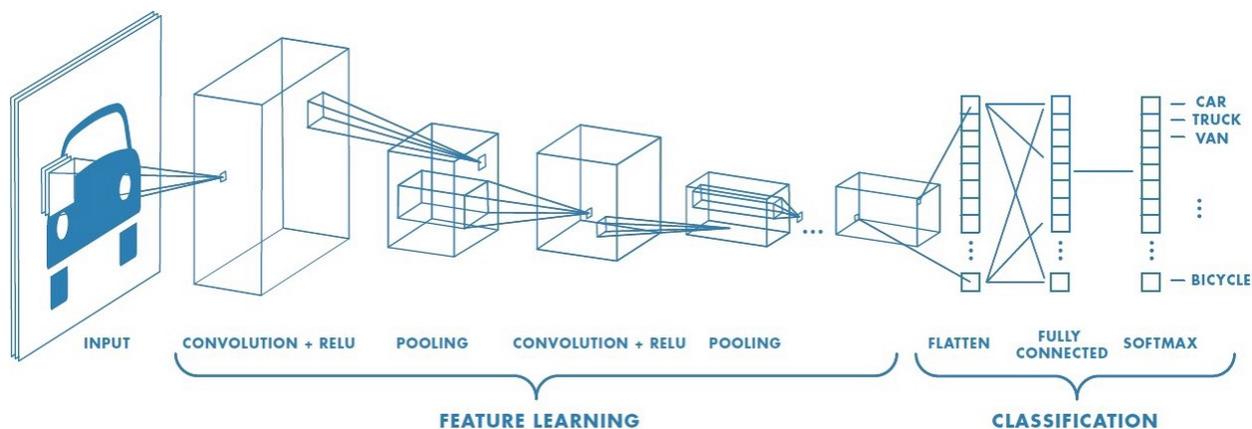


Figure 2.4: Estructura de una red neuronal convolucional.

Las CNN funcionan muy bien en tareas de clasificación, ya que utilizan técnicas para

ello y así consiguen un procesamiento más rápido al requerir de una imagen de entrada con menor resolución, o sea menores datos por procesar, y esto aparte de significar mayor rapidez al utilizar el modelo, también los provee de otro beneficio, tener un entrenamiento más rápido y fácil. Aunque esto tiene la desventaja de no ayudar cuando se necesita detectar objetos, con lo que se crearon nuevas arquitecturas en las que su finalidad es ser mejores en detectar objetos (R-CNNs (*Region Based Convolutional Neuronal Networks*) en 2013, SPP-Net (*Spatial Pyramid Pooling Networks*) en 2015, Fast R-CNN (*Fast Region-based Convolutional Networks*) en 2015 y Faster R-CNN (*Faster Region-based Convolutional Networks*) en 2015), las que interesan en este trabajo son las arquitecturas de una etapa también conocidas como *single-shot*, ya no se generan cuadros candidatos para luego clasificarlos como lo es en una arquitectura de dos fases, sino que el problema de posicionamiento del cuadro delimitador del objeto se convierte en un problema de regresión, o sea buscar que se acerque lo que está dentro del cuadro delimitador a alguno de los conjuntos de características que se tienen como categorías en el clasificador, de estas arquitecturas una de las principales es SSD (*Single-Shot multibox Detection*).

Capa de convolución

La capa convolucional es la más importante de la CNN, ya que es la que realiza la mayor parte del cálculo al identificar características en la imagen. Esta necesita de los datos de entrada, un filtro y un mapa de características, la entrada puede ser un fotograma de un vídeo a color de tres dimensiones (altura, anchura y profundidad) los cuales corresponden con el RGB (*Red, Green and Blue*) de una imagen, puesto que vienen de una matriz de píxeles en 3D, El filtro es un detector de características el cual comprueba que la característica este presente en los campos receptivos de la imagen, a este proceso se le llama convolución.

El filtro es una matriz 2D de pesos, este suele ser una matriz de 3x3, así como el campo receptivo, el filtro se aplica a un área determinada de la imagen y de ahí se calcula un producto de puntos entre los píxeles de la entrada y los del filtro, el resultado se pasa a una matriz de salida, esto se puede observar en la figura 2.5, después el filtro se mueve un paso

y repite el procedimiento anterior hasta que haya finalizado el recorrido por toda la imagen, el resultado del cálculo es el mapa de características.

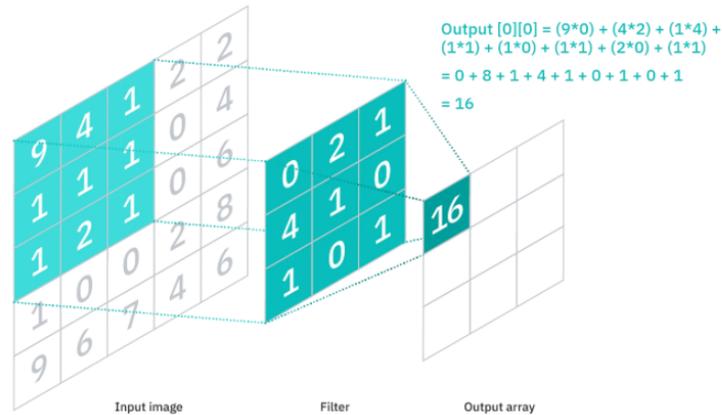


Figure 2.5: Ejemplo de convolución.

Viendo la figura 2.5 se puede deducir que cada valor de salida del mapa de características no tiene una conexión directa con cada valor de píxel de la imagen de entrada, ya que está de intermedio el filtro, con esto se deduce que las capas convolucionales y las de agrupamiento también se pueden denominar capas parcialmente conectadas y esta característica se le suele llamar conectividad local.

Otra característica importante es la compartición de pesos, pues estos permanecen fijos mientras se mueve el detector de características por la imagen, estos se ajustan durante el entrenamiento mediante el proceso de retropropagación y el descenso de gradiente.

Una capa de convolución puede seguir de otra capa de convolución, lo cual genera una estructura jerárquica de características dentro de la CNN, puesto que las capas posteriores pueden ver los valores de los píxeles en el campo receptivo de las capas anteriores, lo cual le permite separar en partes cada vez más complejas o partes distintas entre sí, que cada capa va reconociendo, por ejemplo un vehículo está conformado por ruedas, ventanas, defensa, techo, puertas, espejos y todo esto sumado da un vehículo en un nivel superior.

Capa de agrupamiento

Las capas de agrupamiento (en inglés *pooling*) realizan una reducción de la dimensionalidad al reducir el número de parámetros de la entrada. En esta capa se hace un barrido del campo receptivo de entrada con un filtro sin pesos aplicando una función de agregación a los valores y dando como salida una matriz con los nuevos valores.

Existen dos tipos de *pooling*, la primera es *Max Pooling* la cual barre con el filtro la entrada, seleccionando el píxel con mayor valor para pasarlo a la matriz de salida. Y la segunda es el *Average Pooling* en el cual el filtro se desplaza por la entrada, calculando el valor medio del campo receptivo y enviándolo a la matriz de salida.

En la capa de *pooling* se pierde información, pero así mismo ayuda a reducir la complejidad, mejora la eficiencia y limita el riesgo de sobre ajuste en la CNN.

Capa totalmente conectada

La capa totalmente conectada (en inglés *Fully-Connected* o FC) es la última capa de una CNN, como su nombre lo indica todos los valores de los píxeles de la entrada se conectan con la capa de salida, en esta capa se realiza la clasificación con base en las características obtenidas en las capas anteriores y los diferentes filtros. Las capas convolucionales y las de agrupamiento utilizan funciones ReLU *Rectified Lineal Unit* para sus tareas, la capa FC utiliza una función *SoftMax* para clasificar las entradas en un rango del 0 al 1 y dar de esta manera una salida de probabilidad adecuada.

2.1.6 Arquitecturas de detección de objetos

Como ya se ha mencionado anteriormente, los algoritmos de detección de objetos tienen como objetivo marcar con un cuadro delimitador las coordenadas del objeto detectado y dar la categoría de este, estos algoritmos basados en *machine learning* se dividen en dos categorías, la arquitectura de una fase y la de dos fases.

Arquitectura de una fase

En la arquitectura de una fase se utilizan un gran cantidad de cuadros delimitadores de distintos tamaños y aspectos de relación en la imagen de entrada para abarcar la mayor parte de esta y así tener mayor probabilidad de delimitar al objeto por detectar, con la imagen recortada que representan estos cuadros se utiliza una red completa convolucional para clasificar estas secciones hasta encontrar el que del mejor resultado de detección.

Arquitectura de dos fases

En la arquitectura de dos fases se realiza la primera fase de la misma manera que en la arquitectura de una fase, pero se le agrega una regresión de los cuadros de puntos de anclaje para poder obtener las regiones candidatas y con la regresión obtener las características de esas regiones, de ahí se utiliza una red de clasificación para obtener el resultado final de la detección.

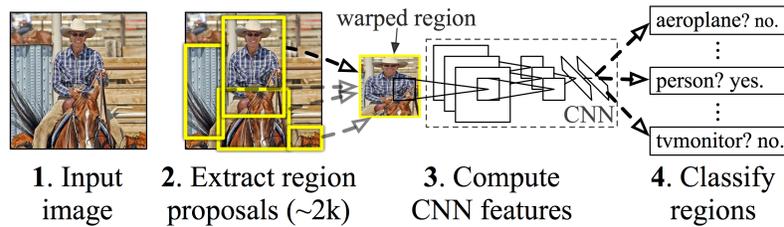


Figure 2.6: Faster R-CNN con arquitectura de dos fases.

Esto genera que al tener menos cálculos por realizar la arquitectura de una fase sea más eficiente en realizar la misma tarea que la de dos fases, pero así mismo tiene en fallo en la calidad de acierto al detectar correctamente a todos los objetos en la imagen, por su parte la arquitectura de dos fases es más lenta por tener que proponer regiones de interés, pero esto la hace más precisa al detectar objetos.

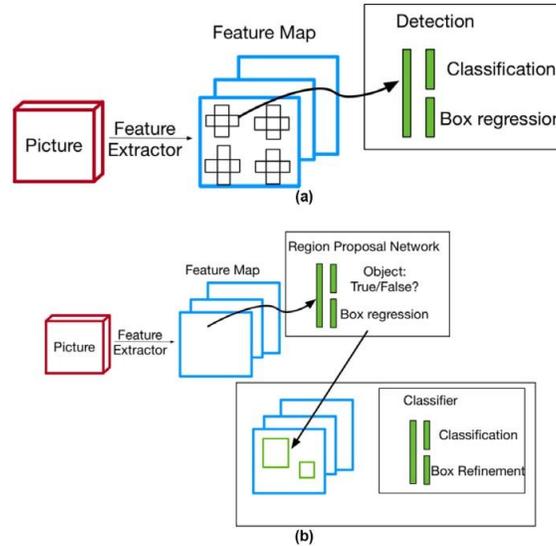


Figure 2.7: Arquitectura de una fase vs dos fases.

En esta investigación se utilizó la arquitectura de una fase por su rendimiento en dispositivos con poder de cómputo reducido, como en este caso al utilizar un Raspberry Pi 4 (RPi4) con una unidad de procesamiento de tensores para ayudarse.

2.1.7 Visión por computadora

¿Qué es la visión por computadora?

La visión por computadora es parte de la inteligencia artificial la cual permite que las computadoras puedan ver como los humanos, o sea, estas pueden llegar a obtener información de vídeos, imágenes u otras entradas digitales que les aportaran la posibilidad de decidir con base en ello o puedan generar información para otro fin. Entonces se tiene que, la visión por computadora les permite ver, observar y comprender la información visual que les llega[41].

Como se menciona anteriormente la visión por computadora es algo similar a la visión del humano, aunque con sus debidas diferencias físicas y algunas ventajas como lo puede ser la longitud del contexto al cual pueden recurrir para interpretar lo que están viendo, la posibilidad casi instantánea y exacta de detectar objetos, clasificarlos, segmentarlos y seguirlos, así como calcular fácilmente la distancia a la que se encuentran, si tienen o no

algún tipo de movimiento y la velocidad estimada de ellos o si algo no cuadra en la imagen, todo ello que al humano no le representa un problema por su experiencia y el poder de su mente es algo más complicado para una computadora con visión, pero aun así no imposible como se podrá ver en este trabajo más adelante.

En cambio, la visión por computadora tiene sus ventajas más enfocadas a la rapidez de la detección, la cantidad de objetos que puede llegar a detectar, sin ningún atisbo de cansancio visual, por realizar la misma tarea 24 horas al día y manteniendo en todo momento la misma exactitud y eficiencia, por lo que rápidamente supera a la visión humana con el debido entrenamiento y los correctos sensores ópticos, datos y algoritmos.

La visión por computadora se utiliza en la industria, automoción, entretenimiento, seguridad, medicina, energía, servicios públicos, fabricación y alimentos, y aun el mercado está en auge. Se estima que para el 2022 el mercado de visión por computadora alcance los 48,600 millones de dólares[42].

¿Cómo funciona la visión por computadora?

Analiza una y otra vez una gran cantidad de datos obtenidos de algún medio visual hasta lograr distinguir entre ellas las distintas variantes y en un punto lograr reconocer imágenes, Por ejemplo, para entrenar una inteligencia artificial a reconocer gatos, se necesita suministrarle miles de imágenes donde aparezcan gatos y distintos elementos relacionados con ellos, con lo que aprenderá la diferencias entre las distintas razas, formas, tamaños y posturas de ellos y lograra reconocer con cierta incertidumbre si hay o no un gato en una imagen.

Para lograr esa tarea requieren del uso del subcampo de *Machine Learning* llamado *Deep Learning* y una red neuronal convolucional.

Tipos de visión por computadora

En el campo de la visión por computadora existen varios tipos de técnicas con distinto objetivo, por ejemplo la clasificación (¿en qué categoría entra el objeto de la imagen?), identificación (¿de qué tipo es el objeto en la fotografía?), verificación (¿está el objeto en

esta foto?), detección (¿dónde está ubicado el objeto en la imagen?), segmentación (¿qué píxeles corresponden al objeto en la imagen?) y reconocimiento (¿qué objetos hay en la imagen y donde están cada uno?), se realiza un repaso de las más destacadas [43].

Clasificación de imágenes

Una de las tareas más comunes que se pueden necesitar al trabajar con imágenes en inteligencia artificial es saber si pertenece a un grupo o categoría, así es clasificarla con base en un conjunto de categorías predefinidas. Puede ser por ejemplo la figura 2.8 en la cual se tiene la fotografía de una rosa en un jardín y se puede identificar rápidamente eso, pero en el caso de la computadora requiere de la ayuda de un clasificador de imágenes al cual se alimenta con la imagen.



Figure 2.8: Fotografía de rosa.

El clasificador creado en este trabajo, de acuerdo al conjunto de imágenes y categorías con el cual se le entrenó, responde que esta imagen pertenece a un grupo de imágenes de flores, como se puede ver en la figura 2.9, si se le entrenara para poder clasificar más a fondo el tipo de flor podría decir que pertenece al grupo de rosas, no quiere decir que identifique o reconozca la imagen sino que más bien ha visto previamente en el conjunto de imágenes que se le dio, fotos de este tipo de flor y que se le dijo como se llamaba. La cercanía a la categoría estará basada en el conjunto de categorías e imágenes que se le dé con anterioridad para que pueda basarse en ellas para realizar su trabajo [44].



Figure 2.9: La clasificación de la imagen nos puede ofrecer distintas salidas.

Localización

La localización en imágenes busca generar una salida con las coordenadas o posiciones donde se encuentra una clase en específico en la imagen o imágenes y así mismo señalarla a través de cajas delimitadoras, como se ve en la figura 2.10 donde se está localizando y señalando el objeto rosa y si hubiese más rosas al ser de la misma clase serían señaladas del mismo modo [45].

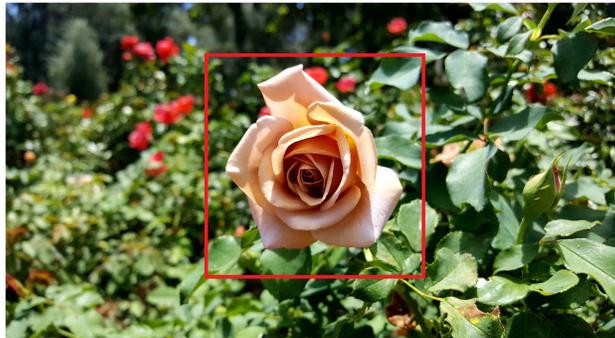


Figure 2.10: La localización nos da en coordenadas una caja delimitadora sobre el objeto.

Esta puede ser de utilidad al ser utilizada en conjunto con la clasificación de imágenes, ya que con la localización se delimita un objeto en la imagen que se entrega, posteriormente se recorta y se pasa a clasificar para obtener más detalles del objeto y poder generar un conjunto de imágenes con etiquetas de clasificación para alguna tarea en concreto.

DetECCIÓN DE OBJETOS

En el caso de requerir localizar los distintos objetos en una imagen y posteriormente recortarlos para enviárselos al clasificador para poder reconocer al conjunto de objetos que pertenecen, es mucho trabajo, por lo que se tiene la detección de objetos en imágenes, la cual aporta simultáneamente la detección y delimitación por coordenadas de todos los objetos presentes en una imagen junto con su etiqueta de la categoría a la que pertenecen sin necesidad de más trabajo.

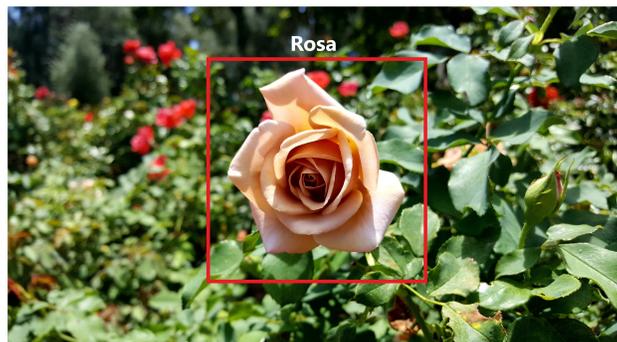


Figure 2.11: La detección ubica en coordenadas una caja delimitadora y lo clasifica.

En la figura 2.11 se ve la posible salida del detector al reconocer el objeto rosa en la imagen, delimitarlo y agregarle su etiqueta, en este caso fue únicamente un objeto, pero los detectores pueden reconocer una gran cantidad de objetos en una misma imagen y los tipos que como podrían detectar pueden ser automóviles, motocicletas, camiones, trenes, perros, gatos, personas, semáforos, señales de tráfico, etc.

El detector tiene una gran utilidad en diversas tareas como podrían ser contar distintos tipos de objetos en distintas situaciones como lo pueden ser personas en un estadio, frutas en un jardín al ser cosechadas o al identificar objetos en tiempo real y de ahí generar avisos como podrían ser vehículos y de ahí calcular su distancia, velocidad, ángulo de aproximación, etc., por lo que son de gran utilidad y en este trabajo se utiliza este tipo de visión por computadora.

Segmentación de instancias

La detección de objetos puede ser de mucha utilidad, pero tiene un problema al determinar el área del objeto, ya que esta remarca un área rectangular alrededor de este lo cual termina abarcando área de otros posibles objetos y puede complicar el trabajo por lo que con ese problema en mente surge la segmentación de instancias en la cual se crea una máscara para cada objeto detectado la cual trata de abarcar únicamente los píxeles que corresponden al objeto así que será lo más precisa posible de acuerdo al detector como se ve en la figura 2.12 en la cual se detecta el objeto rosa y se remarcan solo los píxeles que corresponden a él, pero también se detectan los objetos flor y de igual manera son resaltados en otro color.

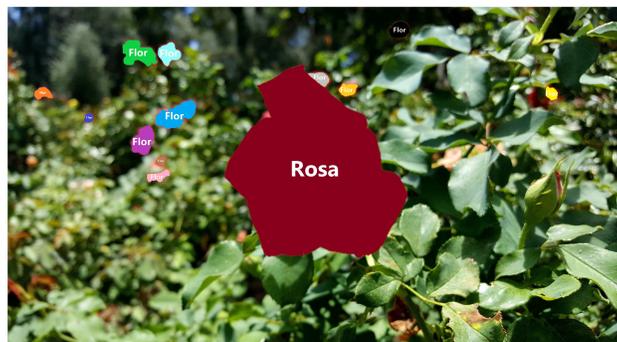


Figure 2.12: La segmentación de instancias separa con color los píxeles de cada objeto y los clasifica.

Este tipo de detector puede ser de utilidad al trabajar con la confidencialidad de las personas que aparecen en imágenes o vídeos y de las cuales se debe censurar su rostro, pero realizar esta tarea fotograma por fotograma podría resultar muy costoso y tardado para un ser humano, pero para una computadora entrenada en ello sería fácil, rápido y eficiente.

2.1.8 Filtro de Kalman

El filtro Kalman [1] ayuda a predecir las posiciones de los vehículos y este mismo realiza una corrección cuando cuenta con el siguiente fotograma para mejorar sus próximas predicciones, este es un algoritmo que estima las variables de un sistema basándose en medidas con ruido,

calcula las distintas posibilidades del siguiente estado (asume que él este está distribuido por una función gaussiana) en el sistema con base en la información disponible en cierto tiempo, estas son las ecuaciones que nos dan la predicción y la estimación de la covarianza del error, posteriormente corrige sus estimaciones con retroalimentación por parte de los datos de entrada y las predicciones hechas anteriormente, en las ecuaciones de corrección tomando en cuenta su componente de ruido añadido, buscando una mejor estimación para el siguiente estado.

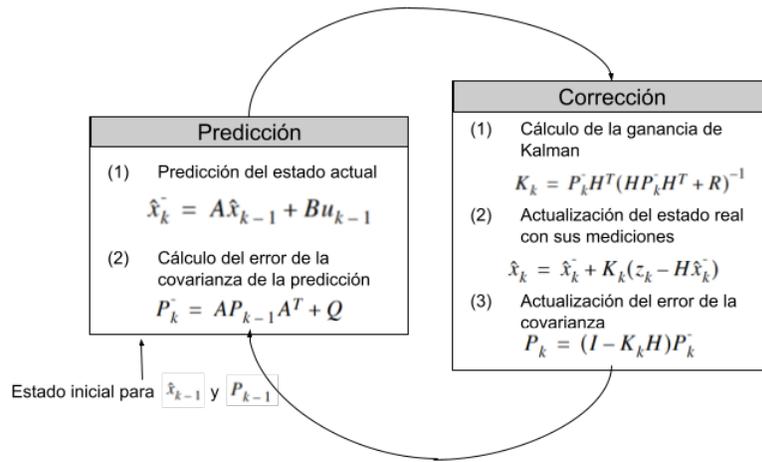


Figure 2.13: Visión completa del filtro de Kalman, sus ecuaciones y operaciones [1].

2.1.9 Método de evaluación IOU

Este método IOU (del ingles *interception over union*) permite medir el porcentaje de la unión entre dos cajas delimitadoras, las cuales son utilizadas por los detectores para remarcar un objeto que ha sido marcado en el fotograma con coordenadas y dimensiones, este método se suele utilizar como métrica para conocer la precisión de un detector comparado con un *dataset* de imágenes, así mismo se puede utilizar para detectar y eliminar detecciones dobles en un fotograma. El método consta de una división entre el área en común sobre el área total de las dos cajas delimitadoras unidas, como se puede ver en la figura 2.14.

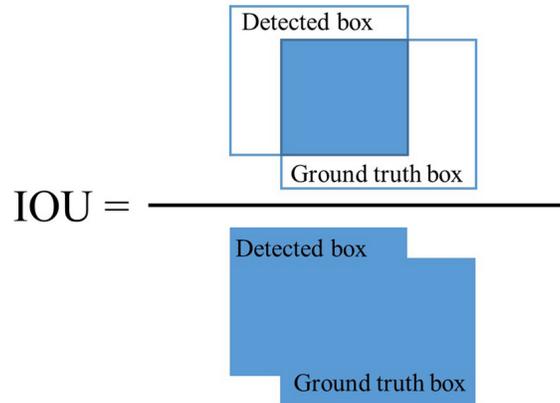


Figure 2.14: Método de evaluación IOU.

2.1.10 Algoritmo Húngaro

El algoritmo Húngaro [46] junto con el método de evaluación IOU ayuda a asignar de la mejor manera los cuadros delimitadores detectados por el detector, con los cuadros delimitadores se predice por el filtro Kalman, ya que este algoritmo pasa el problema de asignación a una matriz de costes $n \times m$, donde se tienen costos por cada elemento asignado a un trabajador por un trabajo, con lo que se divide en dos componentes: los trabajos y los trabajadores. Cualquier trabajador puede realizar cualquier trabajo, pero cada trabajo conlleva un costo diferente, por lo que se requiere asignar cada trabajador a un trabajo y cada tarea a un trabajador con el coste total de asignaciones siendo el mínimo.

	Trabajo1	Trabajo2	Trabajo3
Trabajador1	200	500	400
Trabajador2	500	700	400
Trabajador3	150	500	200

Figure 2.15: Matriz con ejemplo para el algoritmo Húngaro.

2.1.11 Seguidor SORT

El seguidor SORT (del inglés *Simple Online Realtime Tracking*) [47] ayuda a realizar el seguimiento de los distintos vehículos detectados en los fotogramas ayudándose del detector

para obtener los cuadros delimitadores de los objetos, así como del filtro Kalman para predecir la siguiente posición de estos objetos y corregirla posteriormente con ayuda del método Húngaro y del método IOU el cual les dará un costo a la unión de una caja delimitadora con otra, con lo que resolverá cuál caja corresponde de las predicciones y las detectadas por el detector en el nuevo fotograma, para asignarles identificadores con los que se podrán distinguir para posteriormente asignarles atributos.

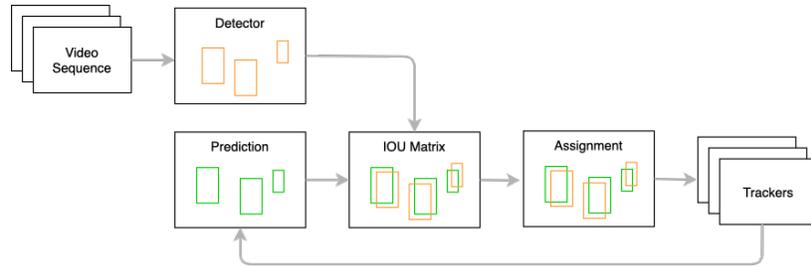


Figure 2.16: Ejemplo de funcionamiento del seguidor SORT.

2.1.12 Realidad aumentada

La realidad aumentada (AR del inglés *augmented reality*) es el conjunto de tecnologías que le permiten a los usuarios visualizar información virtual sobre el mundo real[48], para ello se requiere de un dispositivo inteligente con un sensor óptico capturando imágenes del mundo real y colocando sobre ellas información relevante al contexto para posteriormente mostrarlas en la pantalla del dispositivo, como ejemplo se tiene a la aplicación de Honda *Motorcycles Europe*[®] ver figura 2.17 en la cual el usuario puede seleccionar algunas de las motocicletas del catálogo europeo de Honda[®] para poder simular con ayuda de realidad aumentada y su celular inteligente como sería tener el vehículo de sus sueños estacionado en la sala de su casa[2].

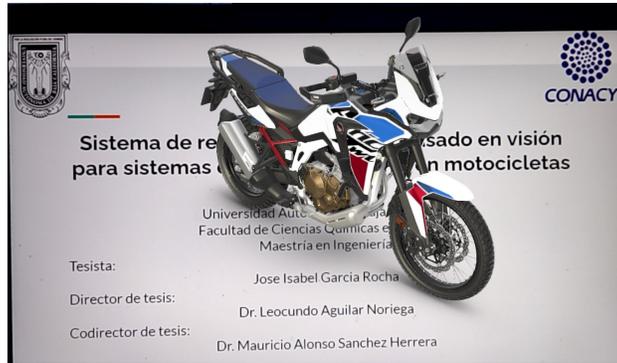


Figure 2.17: App de Honda[®] con motocicleta en realidad aumentada[2].

Del mismo modo existe la realidad virtual (VR del inglés *Virtual Reality*) la cual permite a los usuarios sumergirse en un mundo virtual creado en ordenador con distintos fines como podría ser el entretenimiento en videojuegos o salas de cine virtuales, la capacitación como serian los entrenamientos de los nuevos pilotos y cualquier otro fin que le encuentre uso y utilidad, pero a diferencia de al AR esta información no está relacionada con el mundo en el que se habita y no se pueden utilizar en paralelo, con lo que gracias a la AR se pueden añadir datos importantes al mundo real, para convivir con lo que se ve y de este modo poder facilitar ciertos trabajos o tareas que se realizan y que podrían verse beneficiadas de esta tecnología.

También existe la realidad mixta (MR del inglés *Mixed Reality*) citando la descripción de libro *Tech Trends* “La realidad mixta representa la colisión controlada de las tendencias de AR/VR e IoT. Con la MR, los mundos virtual y real se unen para crear nuevos entornos en los cuales objetos tanto digitales como físicos pueden coexistir e interactuar unos con otros” [49].

VR vs AR vs MR

En la realidad virtual se accede, crea o construye un mundo virtual en el que se aísla al usuario por completo del mundo real para sumergirlo en la simulación y permitirle vivir experiencias distintas a las de su mundo.

Un ejemplo de producto donde se utiliza la VR son las Oculus Quest 2 [3] de la compañía Facebook[®], unas gafas autónomas provistas de todo lo necesario para ofrecer una inmersión casi total al usuario, cuentan con su propio microprocesador, memoria RAM (*Random Access Memory*) y *flash*, pantalla, auriculares, batería y mandos especiales, gracias a ellas el usuario puede sumergirse en mundos virtuales generados por otros usuarios o hasta por el mismo.

Las limitaciones de la VR están en el hecho de obligar al usuario a aislarse del entorno, de su mundo, para poder acceder a ese mundo virtual e interactuar con él sin producir ningún cambio en el mundo real [50].



Figure 2.18: Las Oculus Quest 2 son unas gafas de realidad virtual [3]

Por su parte, la realidad aumentada permite agregar elementos virtuales y mostrar información sobre la pantalla, pero esta no interactúa con el mundo real, ofrece información adicional al usuario que puede sacar beneficio de esto [50].

Un ejemplo podría ser el popular videojuego Pokémon Go[4] de la compañía Nintendo[®], el cual proyecta a través de la pantalla de los teléfonos inteligentes un mundo virtual sobre el nuestro, ya que utiliza la cámara para capturar y posteriormente mostrar en pantalla el entorno del mundo real, pero sobre ella coloca los elementos virtuales para darle la idea al usuario de que se encuentran junto a él.

Las limitaciones de la AR están en el hecho de solo colocar sobre pantalla los elementos virtuales, pero estos no pueden interactuar con el mundo, no le permiten al usuario realizar

algo con esos datos más que mantenerse enterado.



Figure 2.19: El juego Pokémon Go permite visualizar elementos sobre nuestro mundo[4].

Por su parte, la realidad mixta llega a suplir ese vacío entre ambas tecnologías, pues aporta al usuario información extra de la que ya posee en el mundo real y le permite interactuar con ella, hacer algo con esos datos y modificar algo en el mundo real.

Un ejemplo lo podemos encontrar con las HoloLens[5] de la compañía Microsoft[®], las cuales son unas gafas que presentan ante los ojos del usuario un holograma el cual le permite observar elementos imaginarios, pero al mismo tiempo interactuar con ellos, podría ser por ejemplo que un ingeniero eléctrico pueda ver los elementos eléctricos de un circuito en tiempo real y poder hacer cambios en ellos, como cambiando el valor de una resistencia y poder ver como un led aumenta o disminuye su iluminación, estas gafas con esta tecnología son las que más pueden aportar por su versatilidad [51].



Figure 2.20: Las HoloLens son unas gafas de realidad mixta[5].

Esta última (MR) podría resultar más atractiva, ya que une las dos tecnologías anteriores

aunque para este trabajo se logró deducir y comprobar que no hay manera de implementarla en un dispositivo que está enfocado en la seguridad vial del motociclista siendo que para poder utilizar sus beneficios este debe tener una mano libre (para interactuar con el mundo virtual), cosa imposible, puesto que se necesita del uso de las dos manos para conducir como es debido un vehículo y en el caso de las motocicletas no es la excepción, por lo que se eligió por sus ventajas y características la realidad aumentada, pues, esta permitirá mostrar la información de los riesgos alrededor del conductor y este podrá actuar como crea conveniente sin perder control de su vehículo y evitando distraerlo lo más posible.

2.1.13 Detección de vehículos con visión

La detección de vehículos con visión permite con ayuda de sensores ópticos capturar los objetos que se encuentran en el entorno del sistema que los emplea con ayuda de la visión por computadora.

Tesla[®] [52] se enfoca en desarrollar vehículos eléctricos cargados con un nivel de tecnología novedoso, dichos vehículos cuentan con características que los permiten ser autónomos en su conducción, por lo que es de esperar que sus sistemas de seguridad sean de los más avanzados, a diferencia de Volvo[®] esta se enfoca en desarrollar el vehículo autónomo ideal en el que el usuario no tenga que preocuparse de nada más que llegar a su destino en tiempo y forma, a diferencia del vehículo normal estos cuentan con todos esos sensores y sistemas, pero además aportan la capacidad de detección de vehículos, peatones, animales, árboles, señales viales, vías de tránsito, casas y básicamente todo lo que se pueda encontrar en los alrededores de una vía en la cual ellos circulen, con todos estos datos los sistemas del vehículo deciden cómo actuar, podrían decidir si frenar o acelerar, si girar o mantenerse, si cambiar de carril o no, así es sin número de decisiones que se toman al conducir al destino las tomaría sin informarle y sin molestar al usuario. Del mismo modo, sus sistemas de seguridad buscan mantener lo más seguro posible al usuario, evitando los accidentes, pues, al contar con los sistemas de seguridad y con la posibilidad de conducir el vehículo sin la intervención del usuario, puede comprender la naturaleza del riesgo y accionar una maniobra para evitarlo.

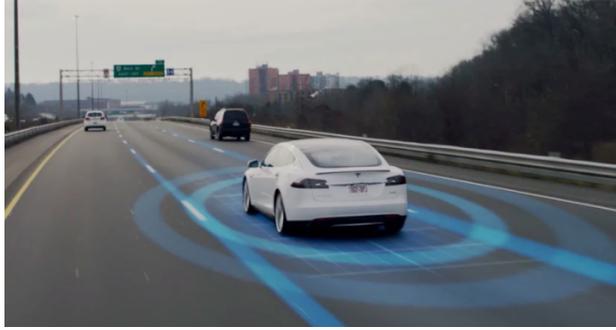


Figure 2.21: AutoPilot de Tesla[®].

2.1.14 Microcontroladores

Como se puede notar, existe la necesidad en la sociedad de obtener mayor seguridad vial y para ello se utiliza una gran cantidad de sensores y sistemas, pero para que esto sea posible se requieren de microcontroladores, los cuales son los que gobiernan todo el sistema, tanto sus entradas como sus salidas, siendo estas la información entrante de los sensores y estas últimas la orden de accionar una medida de emergencia para evitar o intentar evitar el incidente que se registró. Los microcontroladores son dispositivos electrónicos sumamente pequeños, los cuales están presentes en una placa de circuitos en la cual se cuenta con lo necesario para su funcionamiento, siendo esto principalmente memoria SRAM (*Static Random Access Memory*), EEPROM (*Electrically Erasable Programmable Read-Only Memory*), FLASH, puertos de comunicación y puertos de entrada y salida, ellos son los encargados de recibir los datos y procesarlos para realizar lo que se le ha programado por parte de los ingenieros que desarrollaron el sistema que este controla [53]. La variedad de microcontroladores es muy variopinta, pues se encuentran con distintos tipos de potencia de procesamiento, capacidades de almacenamiento, dimensiones físicas, costos, número de puertos de comunicación y tipo de ellos, número y tipo de puertos de entradas y salidas, lenguajes de programación, eficiencia de sus componentes y compatibilidad con otros sistemas o periféricos, por lo que unidos a una gran variedad de herramientas y funcionalidades nos proporcionan un mundo sin igual de oportunidades para el desarrollo de dispositivos del más variado tipo.

2.1.15 Estado del arte

Con ello se visualiza que existe una necesidad de aumentar la seguridad de los conductores en la vía, así como también de que existen dispositivos capaces de capturar las fuerzas físicas ejercidas sobre ellos y transformarlas en señales eléctricas proporcionales y muy exactas, por ende de que existen dispositivos electrónicos que controlan y procesan los datos obtenidos por estos sensores y con ello controlan sistemas de seguridad avanzados y novedosos [54], por último, que existen sistemas de seguridad tanto prototipos como ya funcionales andando en las carreteras y caminos del mundo, aportando una mejora sustancial a los conductores, aumentando la eficiencia de estos al conducir y reduciendo los riesgos para ellos y para los demás.

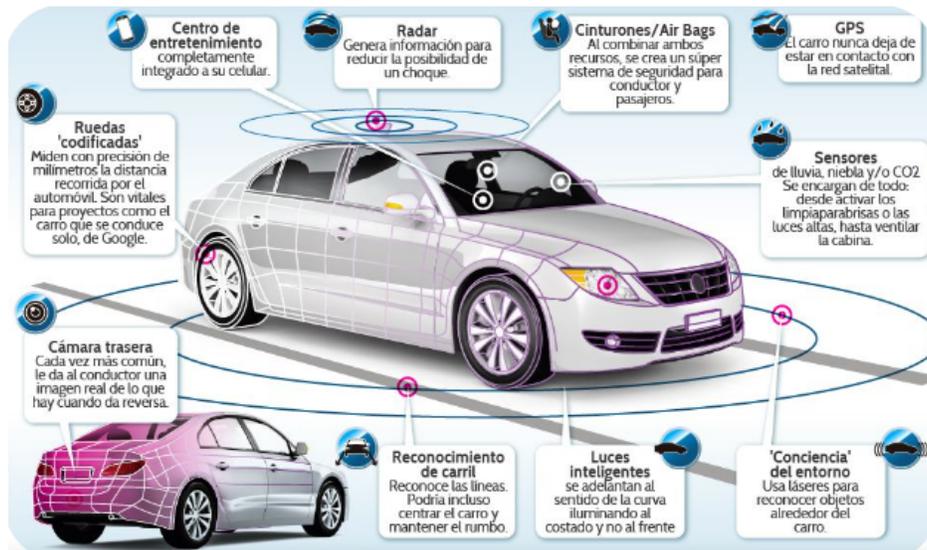


Figure 2.22: Sistemas de seguridad en los vehículos modernos.

Otro desarrollo se puede ver por la filial de motocicletas BMW Motorrad[®] de la compañía BMW[®] que creo un sistema para motocicletas que permite por medio de sensores detectar automóviles en los puntos ciegos del motociclista e informarle por medio de una señal visual en su espejo retrovisor que tiene un vehículo en ese punto ciego, así permitiéndole obtener mayor control y seguridad al conducir en situaciones con tránsito denso [55].



Figure 2.23: Sistema de BMW[®] para cambio de carril seguro.

Byeonghak Lim, Taekang Woo y Hakil Kim del Departamento de Ingeniería de Información y Comunicaciones de la Universidad Inha en Incheon, Corea, en la 3^a Conferencia Internacional sobre Tecnología de Vehículos y Sistemas de Transporte Inteligente (VEHITS 2017) presentaron su artículo “*Integration of Vehicle Detection and Distance Estimation using Stereo Vision for Real-Time AEB System*” en la cual se desarrolló un sistema embebido para detectar los vehículos y medir su distancia utilizando dos cámaras para poder controlar un sistema de frenos autónomos en tiempo real [56], una de las aportaciones más importantes en su investigación sería el hecho de reducir las imágenes tomadas por la cámara a zonas de interés (no importa el área superior de la imagen más allá de la altura del vehículo pues no puede tener contacto con ello, del mismo modo no importa el área del suelo, ya que tampoco encontrara vehículos debajo de él) con ayuda de un algoritmo que ellos diseñaron.

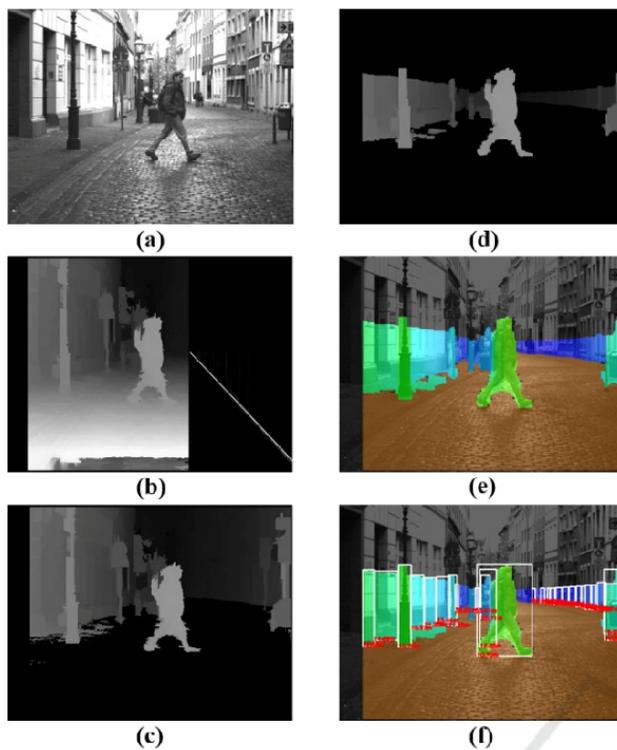


Figure 2.24: Generador de hipótesis ROI (Regiones de interés).

M. Rezaei, M. Terauchi y R. Klette presentaron su artículo "*Robust Vehicle Detection and Distance Estimation Under Challenging Lighting Conditions*," en "*IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2723-2743" en la cual se desarrolla un sistema embebido para detectar vehículos en condiciones con poca visibilidad (lluvia, neblina, polvo, falta de luz, etc.) y con una sola cámara (esto permite un menor tiempo de respuesta del sistema, ya que es la mitad de información a procesar) gracias a la combinación de algoritmos que detectan los bordes y esquinas más un algoritmo que ellos diseñaron en el cual se toma la idea base de que la carretera mantiene un color constante y si este es distinto se puede deber aun vehículo sobre ella [57].

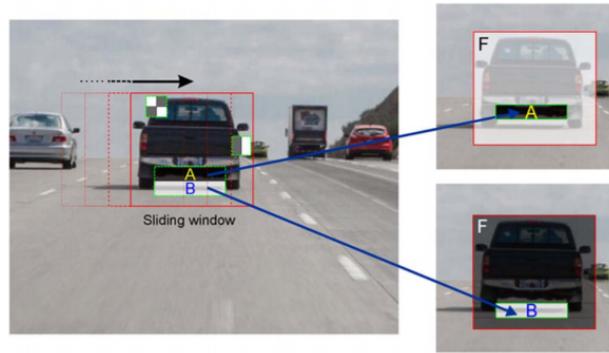


Figure 2.25: Diferencia de colores en la carretera al transitar un vehículo.

Jun Liu y Rui Zhang en su artículo titulado "*Vehicle Detection and Ranging Using Two Different Focal Length Cameras*" toman de base el modelo de detección de objetos de aprendizaje profundo YOLOv3 (*You Only Look Once*) [58] el cual tiene un acierto en la detección casi del cien por ciento, pero en este caso se crea una versión ligera la cual llaman *Lightweight YOLO* en la que se utiliza lo aprendido en YOLOv3 para detectar las matrículas de los automóviles y tomándolo como base (ya se conoce de antemano su ancho) se puede calcular la distancia del vehículo obteniendo un acierto alrededor del noventa y seis por ciento [59].

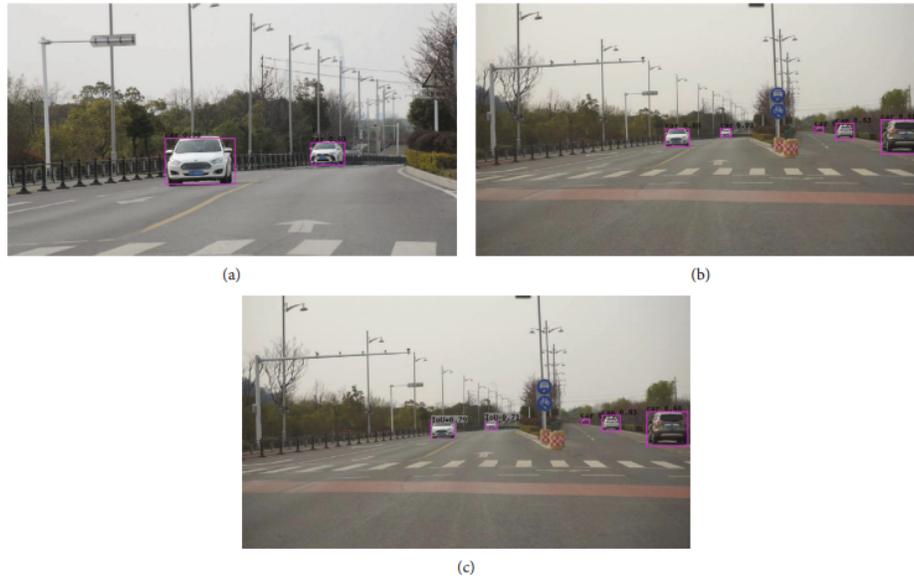


Figure 2.26: Detección de vehículos con distancia focal cercana y lejana.

Gil, G., Savino, G., Piantini, S. y Pierini, M. escribieron su artículo “*Motorcycle That See: Multifocal Stereo Vision Sensor for Advanced Safety Systems in Tilting Vehicles*” en el cual se desarrolla un sistema ADAS (*Advanced driver assistance systems*) pensado para motocicleta, ya que estas son vehículos basculantes (su comportamiento en carretera no es igual al de un vehículo de 4 ruedas) el cual está sujeto al frente de la motocicleta empleando 8 cámaras para obtener 4 pares de ellas y de este modo mantener una mayor fiabilidad de los datos dadas las vibraciones a las cuales se ven sujetas las motocicletas al conducir por su ligereza y su sistema de suspensión más rígido, aunque su sistema requiere de una calibración exhaustiva lo cual viene a quitarle el beneficio de contar con 8 cámaras [60].



Figure 2.27: Sistema prototipo montado.

Capítulo 3

Desarrollo

Para resolver la problemática planteada en este trabajo se desarrolló el sistema dividido en cuatro módulos, siendo estos: modulo principal, módulo escáner, módulo de realidad aumentada y modulo monitor, los cuales se detallan a continuación.

3.1 Módulo principal

La placa Raspberry Pi 4[6] es el centro del sistema, ya que esta va montada en la motocicleta, recibe los datos de la cámara RPI v2, transfiere esta información al módulo TPU Coral USB para procesar las imágenes y detectar los objetos en ellas, recibe la velocidad del módulo escáner, realiza el seguimiento de los vehículos detectados, asigna atributos a ellos, detecta riesgos y envía los datos necesarios para mostrar la información correcta al usuario por medio del módulo de realidad aumentada o el módulo monitor.

3.1.1 Componentes

Placa de cómputo Raspberry Pi 4

La parte central del sistema está en el RPi4, se utilizó la versión de 4 GB de RAM con el sistema operativo Raspbian de 64 bits y con *overclock* a 1.8 GHz por lo que le permite aprovechar de mejor manera el movimiento de datos y esto le aporta un incremento en la velocidad de fotogramas por segundo (fps) al sistema de detección de vehículos de 13.46 fps en la versión de 32 bits a 17.04 fps en la de 64 bits siendo este un aumento del 26.60%, se trabajó con el lenguaje de programación Python 4.5.2 para crear la implementación del

sistema.

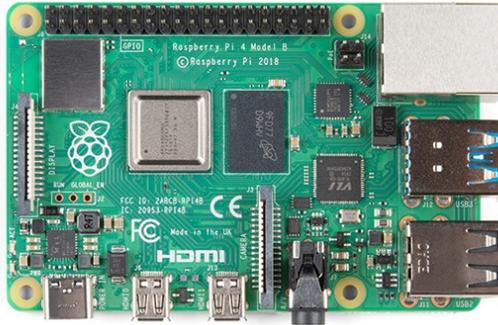


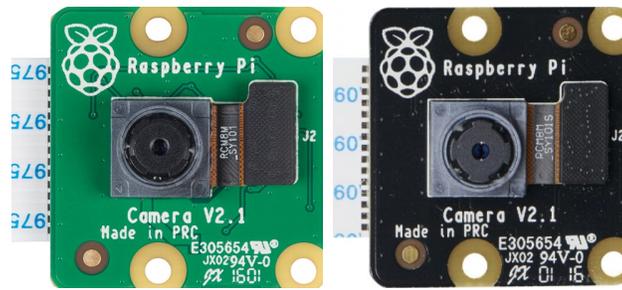
Figure 3.1: Raspberry Pi 4 [6].

Este módulo contiene las siguientes características [61]:

- Procesador Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) de 64-bit SoC @1.8Ghz.
- Memoria 4GB LPDDR4-2400 SDRAM.
- Conectividad WiFi de 2.4 GHz y 5.0 GHz IEEE 802.11b/g/n/ac, LAN, Bluetooth 5.0, BLE, Gigabit Ethernet, USB 3.0, USB 2.0.
- GPIO estándar de 40 pines.
- Puertos para cámara 2-lane MIPI CSI, micro HDMI, para monitor 2-lane MIPI DSI
- Soporte para tarjetas micro SD.

Cámara RPI V2/NOIR

El módulo de cámara oficial de Raspberry Pi[7] permite capturar lo que sucede en la parte trasera del motociclista (nuestra área de interés) y de ahí obtener los posibles riesgos para este, trabajamos con la versión normal y la versión sin filtro infrarrojo.



(a) Módulo de cámara RPI V2.1 (b) Módulo de cámara RPI NOIR

Figure 3.2: Cámaras utilizadas en el proyecto [7].

Este módulo contiene las siguientes características [62]:

- Dimensiones 25 x 24 x 9 mm.
- 8 MegaPíxeles de resolución.
- Tres modos de vídeo: 1080p30, 720p60 y 640x480p60/90.
- Sensor de imagen Sony IMX219 con/sin filtro infrarrojo.
- Resolución del sensor 3280 x 2649 píxeles.
- Tamaño del sensor 3.68 x 2.76 mm (4.6 mm diagonal).
- Campo de visión horizontal de 62.2 grados.
- Peso de 3 gramos.

TPU Coral USB

El acelerador USB Coral [8] añade un coprocesador Edge TPU (*Tensor Processing Unit*) a este sistema, conectado por una toma USB-C (*Universal Serial Bus*) para ayudar al RPI 4 a realizar inferencias de *machine learning* aceleradas.



Figure 3.3: TPU Coral USB [8].

Este módulo contiene las siguientes características [8]:

- Acelerador de ML creado por Google[®].
 - * 4 TOPS (*Trillion Operations Per Seconds* o billones de operaciones por segundo) de rendimiento en modelos int8.
 - * 2 TOPS por watts consumido.
- Conexión USB 3.0 (3.1 Gen 1) tipo C.
- Soportado portado por Linux, Mac y Windows.

3.1.2 Implementación

El módulo principal es el encargado del cómputo de los datos y de la comunicación con los distintos módulos del sistema, por lo que es una parte fundamental de este.

Captura de datos

Para cumplir con la finalidad de este sistema se requiere conocer cuáles vehículos se encuentran detrás del motociclista, en el cual se encuentra su punto ciego, por lo que se requiere de un sensor óptico para reducir el impacto de este, de eso se hace cargo una cámara RPI v2 conectada al RPI 4 por el puerto MIPI CSI-2 (*Mobile Industry Processor Interface Camera Serial Interface 2*), ver la figura 3.4 para referencia.



Figure 3.4: Cámara conectada al puerto CSI-2 del RPI4.

Por parte de software se requiere de OpenCV [63] 4.5.2 para capturar los fotogramas de la cámara a un promedio de 30 fotogramas por segundo (fps), estos son capturados a una resolución de 640X480 píxeles, pero se realiza un recorte que va en el eje y del píxel 95 al 415 y en el eje x del píxel 150 al 470 con lo que obtenemos una imagen de 320X320 píxeles como podemos ver en la figura 3.5, de esta manera se enfoca en los vehículos que pueden representar un riesgo y que nos dan tiempo de avisarlo.



Figure 3.5: Reducción de 640X640 a 320X320.

Con lo que se reduce el tamaño de 307,200 píxeles a 102,400 píxeles, siendo tres veces

menos de píxeles a procesar posteriormente, más aún ayuda para enfocarse en el área de interés que son los vehículos que vienen desde atrás directamente hacia el motociclista, aún se debe reducir más el área de interés de acuerdo a la distancia, pero es algo que se realiza posteriormente.

Detector

Una vez que se tiene los fotogramas capturados por la cámara se debe de detectar los distintos objetos presentes en cada uno de ellos, por lo que se utiliza el *framework tensorflow lite* para ejecutar un modelo de aprendizaje profundo que sea capaz de enmarcar los distintos objetos en los fotogramas como se ve en la figura 3.6, con lo que se utilizó el modelo SSDLite MobileDet el cual está especializado en detectar objetos con una precisión del 32.9 % mAP (del inglés *mean average precision*) evaluado en el *dataset* COCO17 y el cual maneja una resolución de 640X480 píxeles, este detector puede distinguir entre 90 distintos objetos, en este caso lo se ha limitado a 4 clases, siendo éstas automóvil, motocicleta, autobús y camión, ya que son las clases que este modelo detecta y son los principales vehículos presentes en la vía pública que pueden representar un posible riesgo para un motociclista, se pueden ver algunos ejemplos en la figura 3.7.



Figure 3.6: Detecciones de vehículos en un fotograma tomado por el sistema.



(a) Detección de automóvil

(b) Detección de camión

(c) Detección de autobús

Figure 3.7: Ejemplo de vehículos detectados.

Para obtener la lista de vehículos detectados se realizó una serie de pasos que se pueden ver en la figura 3.8

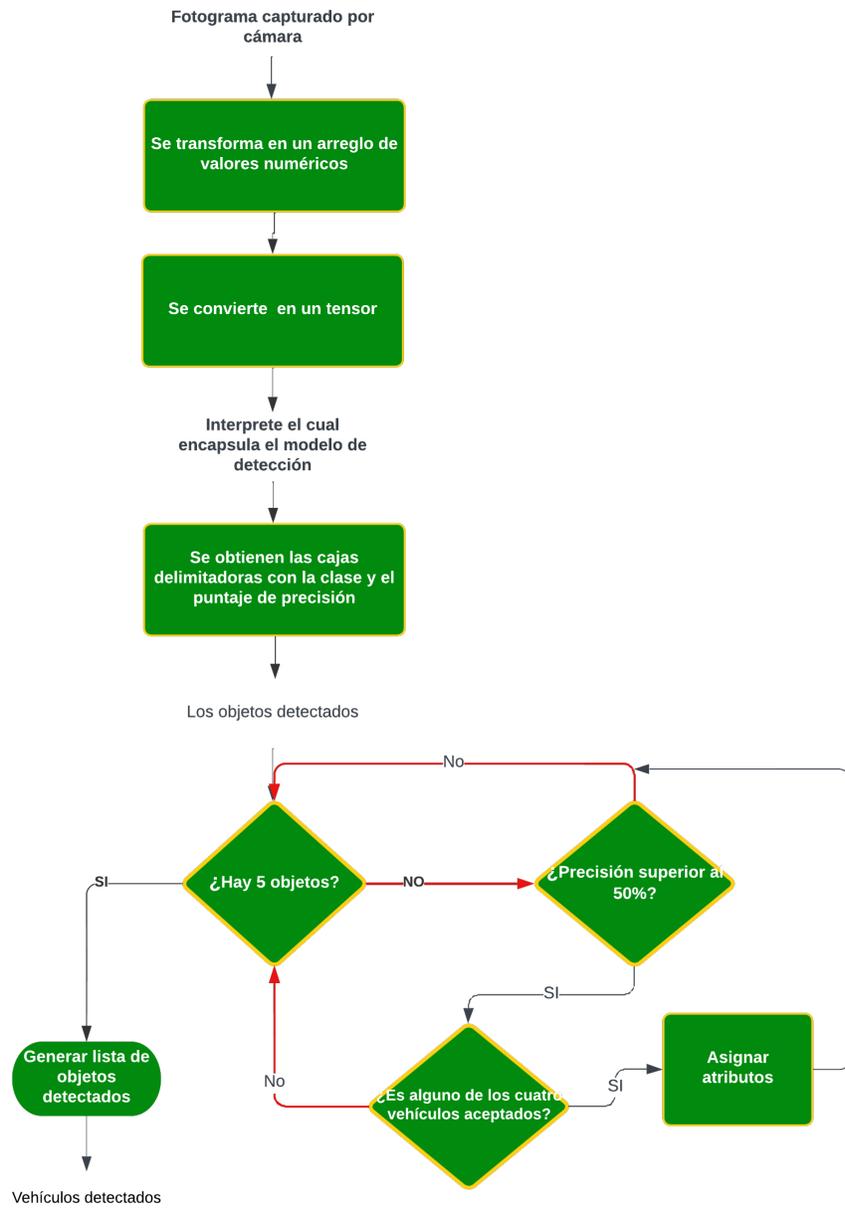


Figure 3.8: Obtención de la lista de vehículos filtrados por clase.

En este punto se tiene los objetos detectados en el fotograma, pero estos aún deben de ser filtrados por duplicados con ayuda del método IOU, como se ve en la figura 3.9 es realmente importante eliminar las detecciones duplicadas, ya que afectan al rendimiento del seguidor que utilizaremos posteriormente.

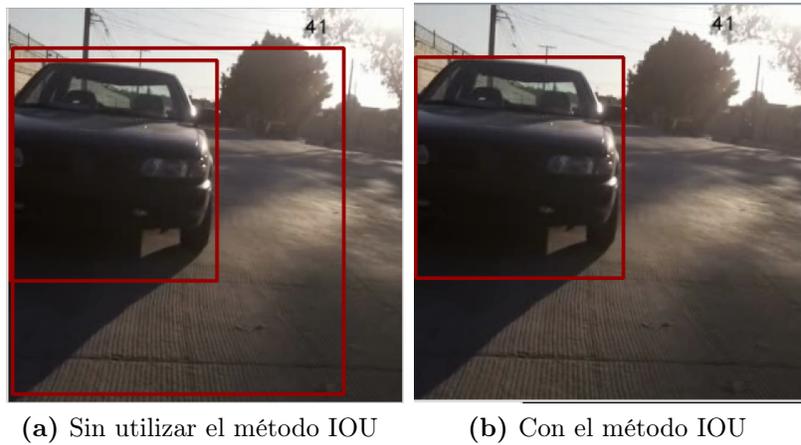


Figure 3.9: Ejemplo de detección doble.

Así mismo se revisa si las clases que se utilizan hasta este momento para detectar los vehículos se visualiza que falta una, esta es la de las camionetas, ya que estás dependiendo su punto de vista pueden ser clasificadas como un automóvil o un camión, el *dataset* COCO no maneja esta etiqueta por lo que lo se soluciona basándose en que los camiones vistos desde la parte de enfrente son más altos que anchos y en el caso de las camionetas son más anchas que altas, con lo que solo con revisar la proporción de estas dos medidas se puede definir si es o no una camioneta y esto es importante, ya que la altura promedio de una de ellas es de 1.85 metros, en cambio, la de un camión es de 3 metros por lo que al basarse en esta altura para obtener una estimación de la distancia a la que se encuentran se encontró con un error al estimar esta distancia, se puede ver un ejemplo en la figura 3.10



(a) Detección de un camión. (b) Detección de un camioneta.

Figure 3.10: Se puede apreciar en la detección como uno es mas ancho y el otro mas alto.

Finalmente, se realiza otro filtrado donde se basa en el ángulo absoluto de los objetos tomado desde el centro de estos hacia el centro inferior del fotograma como se ve en la figura 3.11 y el tamaño proporcional de los objetos en relación con el fotograma, para de este modo filtrar los objetos que estén fuera del área de interés y esta cambia de acuerdo a que tan cerca estén de la cámara y si pueden o no chocar con la motocicleta así que esto se deduce del tamaño de estos, se puede ver un ejemplo en la figura 3.12 y ver la figura 3.13 para comprender cuáles son los valores que toma este filtrado.



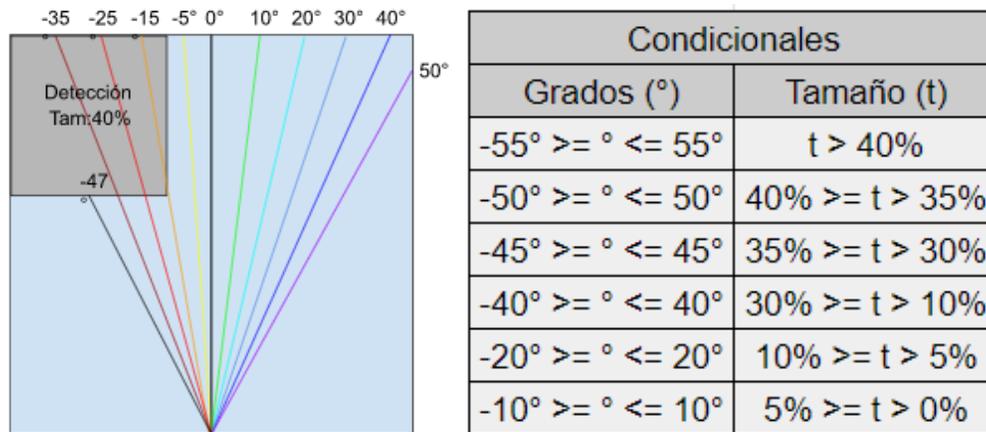
(a) Vehículo a 56 grados. (b) Automóvil a 12 grados.

Figure 3.11: Ejemplo de estimación de ángulo.



(a) Estimación de un vehículo lejano. (b) Estimación de un vehículo cercano.

Figure 3.12: Estimación de tamaño con respecto al tamaño del fotograma y el ángulo.



(a) Ejemplo de ángulos en el mapa y (b) Tabla con rango de condiciones para el detección dentro del rango del filtro.

Figure 3.13: Ejemplo del rango de filtrado para eliminar detecciones fuera del área de interés.

Para reducir el cómputo en el RPI4 se utilizó una unidad de procesamiento tensorial aparte, con la cual el trabajo de inferencia que se realiza con el modelo de detección para obtener los objetos en el fotograma pasa a esta placa y esto nos permite obtener fps con los cuales se midió el rendimiento superior al tiempo real, el cual se marca como superior o igual a 24 fps y limitamos a máximo 30 fps, se puede ver un ejemplo en la figura 3.14.



(a) Sistema en el CPU a 7.42fps. (b) Sistema con ayuda del TPU a 30fps.

Figure 3.14: Sistema de detección funcionando en el CPU versus el TPU.

Seguidor

Una vez que se cuenta con los vehículos detectados, filtrados por clases y eliminados los vehículos duplicados o fuera del área de interés, se puede asignar un identificador único para poder realizar un seguimiento de fotograma a fotograma y de esa manera poder asignarles atributos como lo son velocidad y nivel de riesgo sin que estos se pierdan con el paso de los fotogramas, de esta manera este sistema mantiene un tipo de memoria con la cual puede tomar de manera más inteligente algunas decisiones.

Para ello se implementó el seguidor SORT [47], con el cual se puede identificar el tiempo suficiente a los vehículos como para otorgarles los atributos anteriormente mencionados, se seleccionó SORT, ya que permite el rastreo en tiempo real como podemos ver en la figura 3.15.



(a) Sistema funcionando a 28.91 fps en el fotograma 75. (b) Sistema funcionando a 30 fps en el fotograma 580.

Figure 3.15: Seguidor identificando al vehículo número 12 durante 16.83 segundos.

El seguidor es alimentado por la salida del detector por lo que es importante que este último tenga una precisión correcta aunque podemos ver en la figura 3.16 que aun cuando este genera una salida con cierto margen de error sobre el vehículo se obtiene un seguimiento acertado por parte del *tracker*.



(a) Salida del detector con un margen de error mayor. (b) Salida del seguidor con identificador 183.

Figure 3.16: Ejemplo de fiabilidad del seguidor.

Esa fiabilidad del seguidor nos permite realizar nuestro trabajo sin mayores inconvenientes, permite realizar un seguimiento acertado a los vehículos directamente en la parte trasera de la motocicleta, este podría ser mejor si se contara con una mejor entrada del detector el cual está limitado por el sensor óptico de entrada por lo que el sistema funciona correctamente aun con limitantes de hardware que son fácilmente corregibles con el debido presupuesto y tiempo invertido en realizar las adaptaciones.

Estimación de distancia

Es necesario conocer la distancia a la cual se encuentran los vehículos de la motocicleta donde esta montada la cámara, aunque contamos con la limitante de tener solo un punto de referencia visual, en el caso del ser humano nos favorecemos de tener dos ojos o sea visión estereoscópica con ayuda de nuestra comprensión de la profundidad y la experiencia, calcular o estimar una distancia más o menos certera de un objeto a otro, en el caso de solo contar con una cámara podemos aprovecharnos de la semejanza de triángulos la cual nos dice que si dos triángulos presentan lados proporcionales son triángulos semejantes, esto podemos verlo en la figura 3.17 y en la ecuación 3.1 donde nos indica que hay una correlación entre ambos triángulos, ya que comparten ángulos y lados proporcionales.

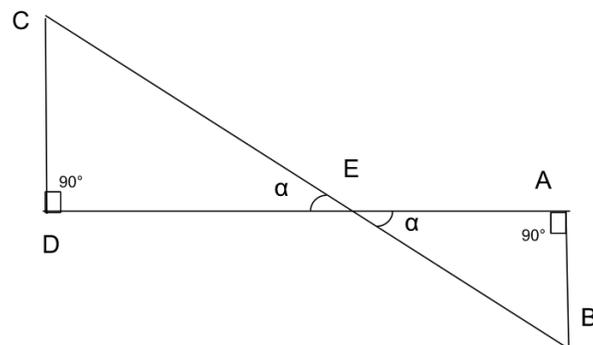


Figure 3.17: Semejanza de triángulos.

$$\frac{\overline{EA}}{\overline{BA}} = \frac{\overline{DE}}{\overline{CD}} \quad (3.1)$$

Basándose en la semejanza de triángulos y pasándolo a la longitud focal obtenemos la figura 3.18 de la cual podemos ver que la longitud focal se puede despejar y obtenemos la ecuación 3.2

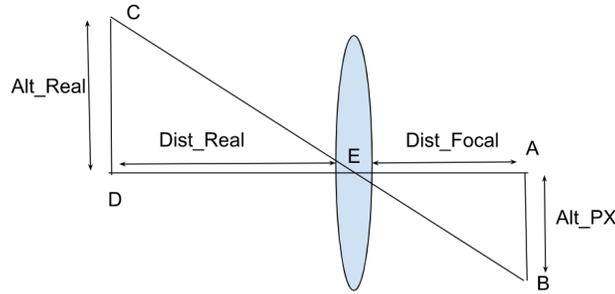


Figure 3.18: Lente y longitud focal.

$$\overline{EA} = \frac{\overline{BA} * \overline{DE}}{\overline{CD}} \quad (3.2)$$

Tomando la ecuación 3.2 y sustituyendo los lados por sus respectivos nombres obtenemos en base a la figura 3.18 obtenemos la ecuación 3.3

$$DistanciaFocal = \frac{AltPX * DistReal}{AltReal} \quad (3.3)$$

De la cual tenemos que AltPX es la altura en píxeles del objeto, DistReal es la distancia real del objeto en metros y AltReal es la altura real del objeto en metros.

Una vez obtenida la ecuación podemos calibrar nuestro sensor, para ello tenemos los siguientes pasos:

- 1 Declarar distancia en metros del vehículo de prueba al lente.
- 2 Seleccionar una clase para calibrar (automóvil, camión, autobús, motocicleta).
- 3 Establecer altura en metros promedio para la clase del vehículo.

- 4 Seleccionar un vehículo de prueba de la clase para la calibración.
- 5 Conectar el sensor óptico al RPI4.
- 6 Colocar el sensor óptico a la distancia establecida anteriormente.
- 7 Capturar y detectar el vehículo de prueba.
- 8 Obtener la altura en píxeles de la detección del vehículo de prueba.
- 9 Utilizar la formula de la focal (ver ecuación 3.3) para obtenerla.
- 10 Repetir con las clases de vehículos faltantes.

Para nuestra calibración se seleccionó un vehículo de la clase automóvil, un Chrysler® Pt Cruiser con una altura promedio de 1.6 metros y a una distancia de 4.5 metros, ver figura 3.19.



Figure 3.19: Captura de calibración con cámara V2.1 infrarroja.

La calibración la realizamos en los dos modelos de cámara v2.1 para RPI4, la versión normal y la versión infrarroja, ya que se realizaron pruebas con ambos modelos, podemos observar en la tabla 3.1 que aun cuando las dos cámaras poseen el mismo sensor óptico, el resultado de la longitud focal es distinta, ya que los lentes no están fijos, estos pueden variar

el enfoque al girarlos.

Altura	Alt_PX_CamV2.1	Alt_PX_CamIR	Dist_Focal_V2.1	Dist_Focal_IR
1.2mt	102.4000px	99.7500px	384	374.0625
1.6mt	128.0000px	133.0000px	384	374.0625
1.85mt	157.8667px	153.7812px	384	374.0625
3.0mt	256.0000px	249.3750px	384	374.0625

Table 3.1: Resultado de la calibración con cámara V2.1 e Infrarroja.

Una vez que se ha calibrado con la ecuación 3.3 la cámara podemos despejar la distancia del lente al objeto para poder estimar esta distancia de una manera rápida utilizando solo un sensor óptico, la ecuación 3.4 muestra esto.

$$DistanciaReal = \frac{DistFocal * AltReal}{AltPX} \quad (3.4)$$

Utilizando esta fórmula, siempre y cuando se mantenga sin alteraciones el lente de la cámara se podrá obtener una estimación de la distancia con ayuda de una altura real promedio por clase establecida por nosotros ver tabla 3.2 con lo que podremos obtener estimaciones como las de las figuras 3.20 de la cual se estima la distancia de dos vehículos detectados a 4.27m y a 42.75m, 3.21 en la cual se realiza una prueba de campo en ciudad, con ayuda de *Google Maps* se obtuvo la distancia a la cual se encuentra el vehículo del lente, dando una distancia de 85.56m y en la estimación promediada del sistema nos dio de 85.56m y en la 3.22 donde se probó en una calle secundaria con un vehículo a 35.2m.

Clase	Altura
Motocicleta	1.2mt
Automóvil	1.6mt
Camioneta	1.85mt
Autobus	3.0mt
Camión	3.0mt

Table 3.2: Altura promedio utilizada por clase.

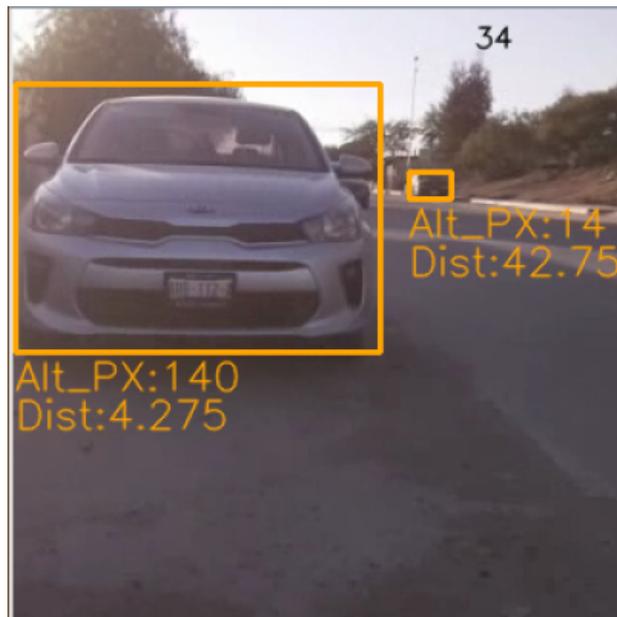


Figure 3.20: Estimación de distancia en dos vehículos detectados.

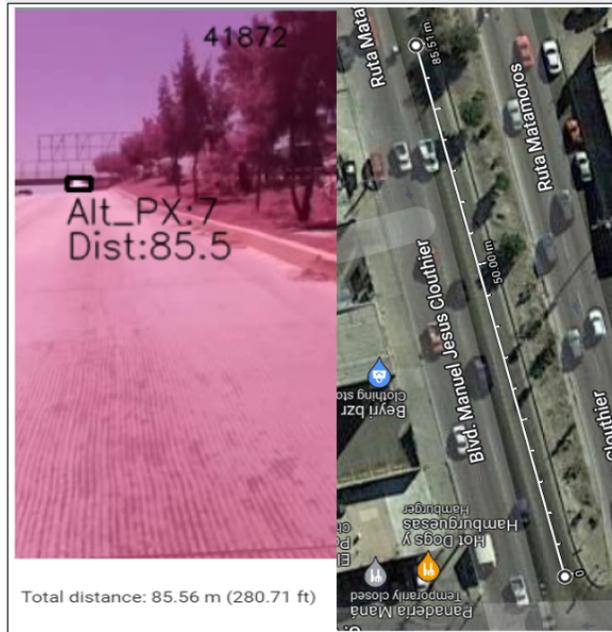


Figure 3.21: Prueba de campo en ciudad.



Figure 3.22: Prueba de campo en una calle secundaria.

Para mejorar las estimaciones de distancia utilizamos un filtro de media móvil con un *buffer* de 30 valores, de esta manera los cambios no son tan abruptos podemos verlo en la

tabla 3.3, ya que se requiere de cambios graduales en la distancia para estimar de una manera más acertada la velocidad con la que se mueven los vehículos.

Alt_PX	Dist_sin_prom	Dist_promediada	Cm por pixel
7	85.500	85.500	22.857
8	74.813	80.156	20.000
9	66.500	75.604	17.778
10	59.850	71.666	16.000
11	54.409	68.214	14.545

Table 3.3: Impacto de un píxel en la estimación de la distancia.

Estimación de velocidad

Tomando la medición de la distancia de los objetos más el tiempo en el que se realizó esta medición se puede obtener la velocidad relativa 3.5 de estos que al sumarle la velocidad de la motocicleta obtenemos una estimación de la velocidad 3.6 de cada uno de estos vehículos.

$$VelocidadRelativa = \frac{Distancia_{nueva} - Distancia_{anterior}}{Tiempo_{nuevo} - Tiempo_{anterior}} \quad (3.5)$$

$$Velocidad = VelocidadRelativa + VelocidadMoto \quad (3.6)$$

Para ello debemos revisar uno por uno, los vehículos seguidos con identificador así mismo con una distancia distinta de cero, si es la primera vez que se sigue este vehículo y no se cuenta con una velocidad anterior se le asignan algunos atributos como lo pueden ser la distancia y el tiempo de ese momento, con ello se podrá utilizar la fórmula 3.5 posteriormente.

Cuando se detecte y siga el vehículo la siguiente vez y este cuente con una distancia actual distinta a la almacenada con anterioridad (lo cual significaría que se ha movido de sitio) se utilizan las ecuaciones 3.5 y 3.6 para estimar su velocidad aunque esta puede contar con cambios muy abruptos o muy poco significativos por lo que se revisa que hayan pasado más de 25 milisegundos (ms) para posteriormente ingresar la medición de la velocidad a un

filtro de media móvil el cual permite suavizar la curva de aceleración de los vehículos y de esta manera obtener mejores resultados posteriormente al estimar el riesgo.

Estimación de nivel de riesgo

La meta es detectar las situaciones de posible riesgo al conducir el motociclista en la vía pública, esto lo hemos ido logrando a través de la recolección de los diversos datos, esta información el motociclista experimentado al obtenerla y analizarla puede deducir si alguno de los vehículos que se encuentra a su alrededor representa un riesgo para él, así que intentamos imitar esto con un algoritmo simple.

Primero debemos definir el nivel de sensibilidad al riesgo, esto se realiza a través de la aplicación móvil con un selector, está predefinido en normal, esta variable lo que modifica es el tiempo máximo desde el cual el impacto del vehículo es relevante con lo que si este es mayor o igual al tiempo de impacto eso representa un riesgo, así mismo es importante modificar el radio de distancia en el cual se considera que el vehículo esta en nuestra área de interés, agrandando esta área al aumentar la sensibilidad al riesgo o disminuyendo en la inversa, también manejamos dos offset uno para la distancia límite de nuestra zona de interés y el otro para la velocidad límite de la motocicleta, estos dos últimos valores entran en juego en caso de que el tiempo de colisión calculado por la ecuación 3.8 y la ecuación 3.7 y la distancia máxima de atención estén fuera de los límites para considerarlo un riesgo.

$$VelocidadRest(m/s) = VelocidadVehiculo(m/s) - VelocidadMoto(m/s) \quad (3.7)$$

$$TiempoColision = \frac{DistanciaAlVehiculo}{VelocidadRest(m/s)} \quad (3.8)$$

Se le asigna el nivel de riesgo más importante si el tiempo de colisión es menor o igual al tiempo de colisión máximo y la distancia que hay entre el vehículo y la motocicleta es menor a la distancia de colisión máxima establecida con anterioridad.

En caso de no entrar en la condición anterior, utilizamos la distancia entre los dos vehículos, la velocidad de la motocicleta, los *offsets* de cada una de ellas y los límites.

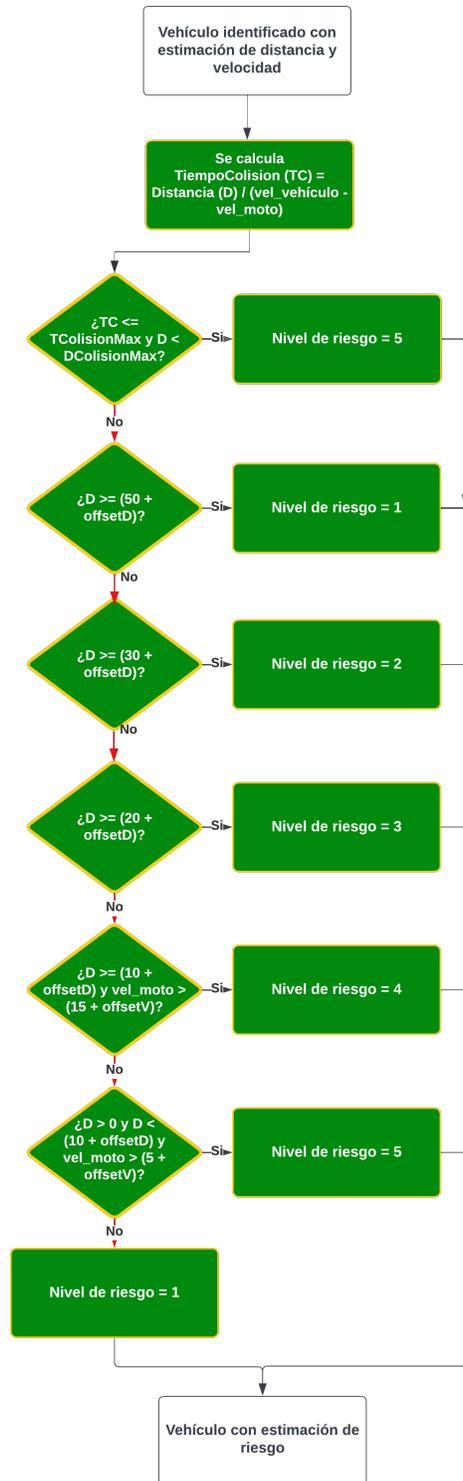


Figure 3.23: Algoritmo estimación de riesgo.

3.2 Módulo escáner

Es necesario contar con la velocidad de la motocicleta con el menor retraso posible y esto lo obtenemos gracias al módulo escáner, el cual consta de un esp32-cam[9] conectado por comunicación serie a un escáner automotor elm327[10].

3.2.1 Componentes principales

3.2.2 Microcontrolador ESP32-CAM

El microcontrolador ESP32 podemos encontrarlo en distintas versiones y tamaños, para este trabajo optamos por la versión cámara por el tamaño reducido de esta e inicialmente por contar con un módulo de cámara con el cual realizamos pruebas para posteriormente terminar utilizando la cámara RPI v2, ya que esta última ofrecía una mayor velocidad y un mejor sensor óptico con lo que nos permite capturar de una mejor manera los datos y obtener mejores resultados con el detector, en este caso nos interesaba el tamaño, puesto que el consumo no era una limitante, pues el módulo escáner obtiene energía de la motocicleta.

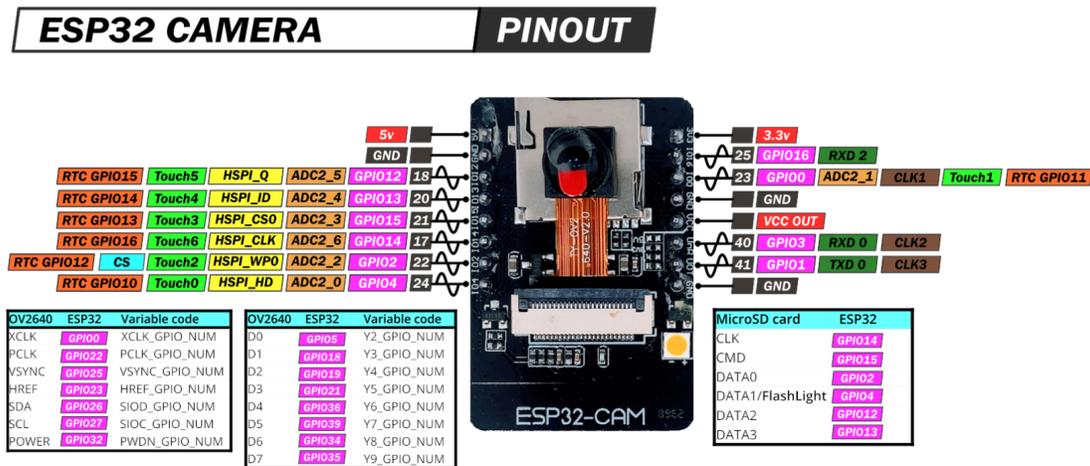


Figure 3.24: Función y descripción de los pines del ESP32 Camera [9].

Este módulo contiene las siguientes características [9]:

- Procesador Dual core Tensilica Xtensa LX6 de 32-bit @160Mhz.
- Memoria 520KB SRAM y externa 4MB SRAM.
- Conectividad WiFi 802.11b/g/n, Bluetooth 4.2.
- Voltaje de alimentación 5VDC.
- Antena PCB y conexión a antena externa.
- Soporte a UART/SPI/I2C/PWM/ADC/DAC.
- Socket para tarjetas micro-SD
- Cámara OV2640
- Dimensiones 27*40.5*6 mm.
- Peso: 20 gramos.

Este dispositivo facilita crear un punto de acceso WI-FI (*Wireless Fidelity*) para comunicar al resto de módulos entre sí de una manera más rápida y poder transferir datos de mayor volumen, así mismo al contar con soporte a la comunicación UART (*Universal Asynchronous Receiver-Transmitter*) podemos aprovecharlo y conectar un escáner automotor para obtener la velocidad de la motocicleta con el menor retraso posible.

3.2.3 OBDII Serial

Como se menciona anteriormente se requiere de la velocidad de la motocicleta para contar con esa base y de ahí estimar la velocidad de los demás vehículos, para ello se utiliza el escáner ELM327 OBDII (*On Board Diagnostics 2*) en una de sus versiones en la que cuenta con los puertos seriales expuestos para conectar el ESP32 y realizar la comunicación para obtener la velocidad en kilómetros por hora.

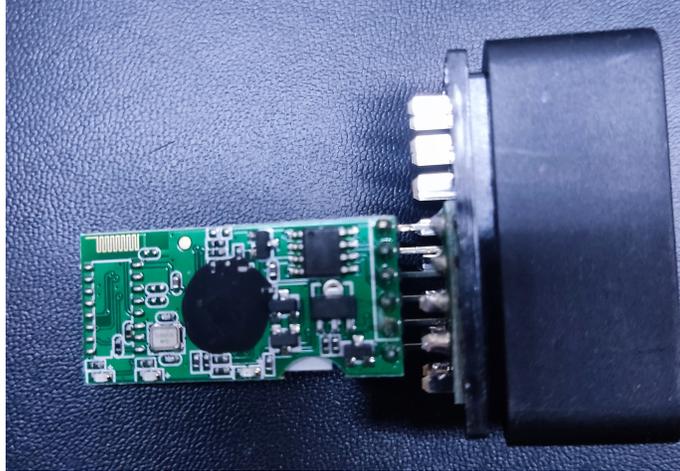


Figure 3.25: Escáner ELM327 OBDII [10].

Este chip contiene las siguientes características [10]:

- Control de la energía y modo en espera.
- Interfaz serie universal (RSR232).
- Busca automáticamente el protocolo correcto de comunicación.
- Completamente configurable con comandos AT (del ingles *Attention*).
- Diseño CMOS de bajo consumo.

3.2.4 Diseño y conexión

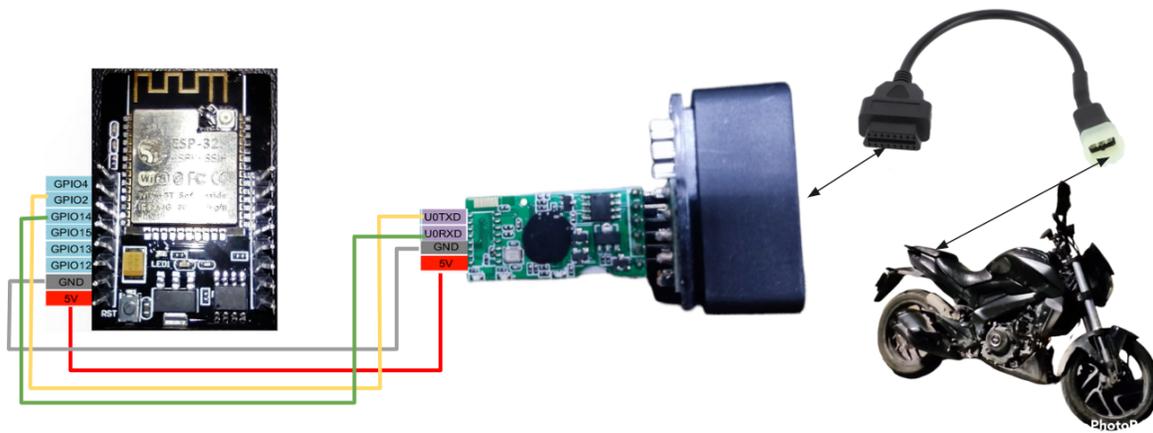


Figure 3.26: Conexión del módulo ELM 327 OBDII [10].

Como se puede visualizar en la figura 3.26 tenemos el microcontrolador conectado por conexión serie al escáner OBDII el cual obtiene energía y la información de la motocicleta por medio de un adaptador y este a su vez se conecta al módulo correspondiente en la motocicleta.

3.2.5 Funcionalidad

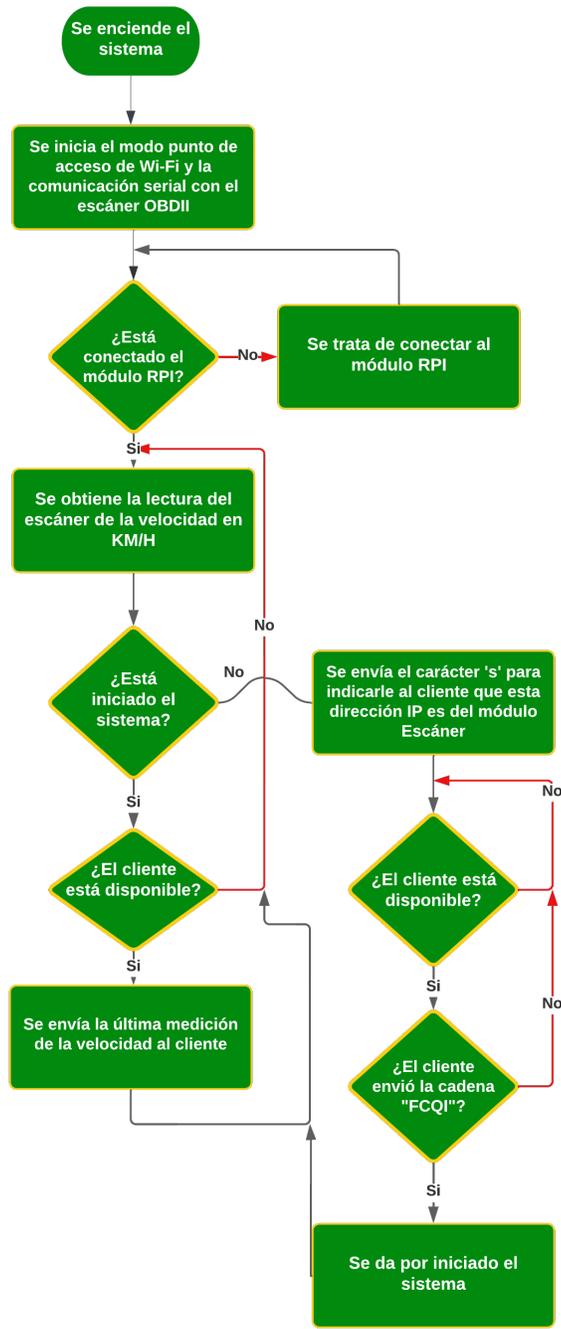


Figure 3.27: Diagrama de flujo del módulo Escáner.

Punto de acceso Wi-Fi

Se optó por utilizar un punto de acceso Wi-Fi para este proyecto con el cual los módulos se puedan comunicar sin problemas, favoreciendo que todos cuenten con módulo Wi-Fi, dadas las características modulares del sistema favorece esta decisión, ya que permite la opción de agregar más módulos lo cual se comenta más a fondo en la sección trabajo futuro.

Captura de velocidad

Como se marca en el algoritmo de la figura 3.31 se obtiene del escáner por comunicación serie la velocidad, el microcontrolador envía el comando para solicitar dicho dato al escáner y este lo retorna al leer la velocidad de la unidad de control del motor ECU (del inglés *Engine Control Unit*) por el puerto OBDII, se realiza esta medición 30 veces por segundo para obtener el valor en tiempo real, la velocidad leída puede ser la misma en dicho tiempo dependiendo el movimiento de la motocicleta en el mundo real, con ello se podrán realizar los cálculos necesarios para el sistema principal con un error reducido en comparación de utilizar la velocidad obtenida del GPS de un teléfono inteligente, ya que hay mucho desfase entre la velocidad real y la estimada.

Transmisión de datos

Es necesario que la velocidad más actual llegue al módulo principal, puesto que es el encargado de utilizarla para estimar la velocidad de los vehículos al sumarla a la velocidad relativa, sin la velocidad real el cálculo estaría propenso a mayores errores por parte del algoritmo de estimación de velocidad relativa, para enviarla nos ayudamos de la comunicación que existe entre el módulo escáner y el cliente (RPI) y enviamos los datos en forma de entero, ya que no necesitamos decimales para este trabajo.

3.3 Módulo realidad aumentada

El sistema busca avisar al conductor de la motocicleta de un posible riesgo que se acerque a él desde un punto ciego, en este caso el trasero, por lo que se requiere mostrarle esta información al usuario de una manera poco intrusiva, pero a la misma vez que esta sea suficiente para servirle de aviso, por lo que se ha optado por ayudarnos de la realidad aumentada para mostrar los objetos cercanos con la estimación de la distancia así mismo el nivel de riesgo con colores para estos en un radar.

3.3.1 Componentes principales

3.3.2 Microcontrolador ESP8266

El microcontrolador ESP8266[11] está disponible en distintas versiones y tamaños, tomando en cuenta que esto reduce o aumenta el número de pines disponibles, para este trabajo se optó por la versión d1 mini dado el tamaño reducido de esta, se buscó que el tamaño sea similar al de la batería utilizada, ya que este sistema debe contar con autonomía por parte de la batería sin necesidad de obtener la energía de forma cableada, puesto que el módulo estará sobre el casco del motociclista y este debe contar con total movilidad.

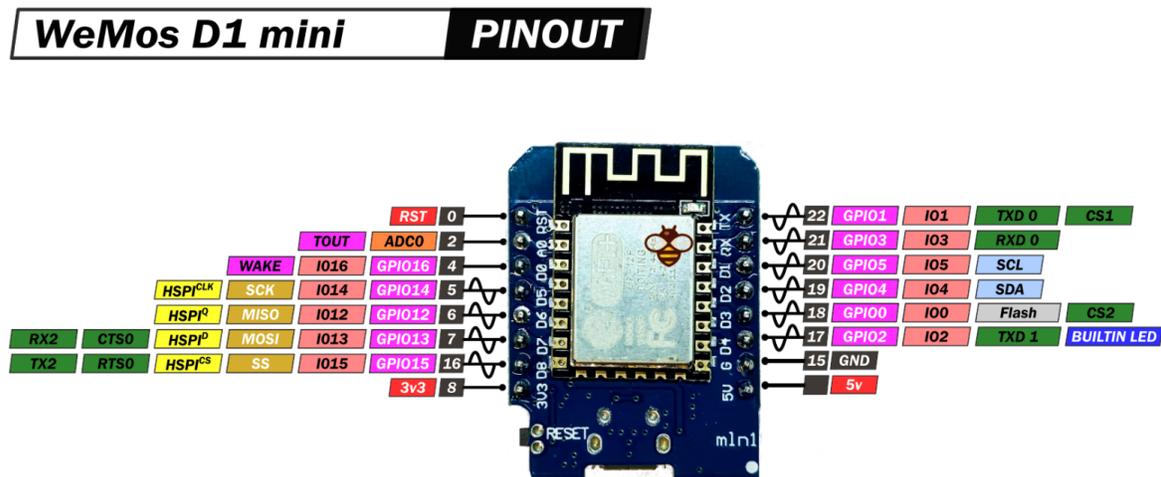


Figure 3.28: Función y descripción de los pines del ESP8266 [11].

Este módulo contiene las siguientes características [11]:

- Procesador Tensilica Xtensa Diamond Standard L106 de 32-bit @80Mhz.
- Memoria 32KB RAM de instrucciones.
- Memoria 32KB RAM para caché de instrucciones.
- Memoria 80KB RAM de datos para el usuario.
- Memoria 16KB RAM de datos del sistema.
- Memoria Flash de 4 MB.
- Conectividad WiFi 802.11b/g/n.
- Voltaje de operacion 3.3V.
- Soporte a UART/SPI/I2C/I2S.
- ADC de 10-bits.
- 16 pines de GPIO.
- Dimensiones 34.2*25.8*4.7 mm.
- Peso: 10 gramos.

3.3.3 Pantalla TFT

Para mostrar la información se requiere de la ayuda de una pantalla de un tamaño reducido más aun así suficiente para dar un panorama de los posibles riesgos, la ubicación y la distancia en la que se encuentran, esto lo realizamos con ayuda de una pantalla TFT[12] (*Thin Film Transistor*) de 1.3 pulgadas la cual se comunica por el protocolo SPI (*Serial Peripheral Interface*) lo cual provee de una mayor velocidad de transferencia de datos y esto permite actualizar la pantalla por lo menos a 30 dibujos por segundo, y aún nos permite trabajar con colores, ya que es RGB así que podemos asignarle los niveles de riesgo basándonos en el color siendo rojo, anaranjado, amarillo, verde y blanco, del mas riesgoso al menos riesgoso.



Figure 3.29: Pantalla TFT de 1.3" con 240x240 píxeles [12].

Este módulo contiene las siguientes características [12]:

- Tamaño del LCD 1.3 pulgadas en diagonal.
- Modo de visualización normalmente negro.
- Resolución de 240x240 píxeles.
- Interfaz de comunicación SPI.
- Microcontrolador ST7789VW
- 250 lúmenes de intensidad.
- Color RGB.
- Dimensiones 26.16*29.22*1.5 mm.
- Peso: 10 gramos.

3.3.5 Funcionalidad

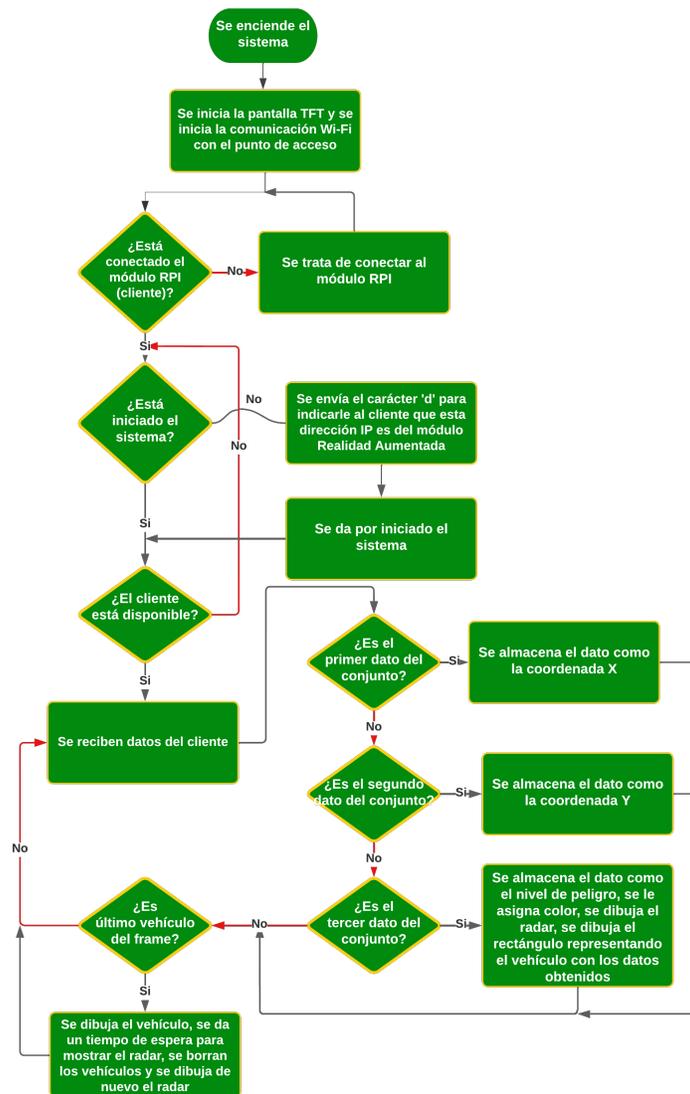


Figure 3.31: Diagrama de flujo del módulo Realidad Aumentada.

Recepción de datos

Este módulo recibe la información en una cadena de texto en la cual se colocan en el siguiente orden los valores $coorX(v1), coorY(v1), pel(v1), \dots, coorX(vN), coorY(vN), pel(vN)$, de los cuales representan los vehículos en el rango del radar, posicionados con coordenadas

en el eje X y en el eje Y, así mismo su nivel de riesgo, si existen más vehículos en el fotograma y en rango se concatenan al arreglo de caracteres separados por una ','. Posteriormente, se recorta en cada ',' y se almacena en la variable correspondiente el valor recibido para poder crear el mapa con los objetos colocados en su posición correcta.

Creación del mapa

La información mostrada en pantalla debe de alguna manera mostrar la distancia a la que se encuentran los vehículos y su posición en el espacio, si se encuentran a la izquierda, al centro o a la derecha del motociclista en su parte posterior y a que distancia se encuentran de él, la manera en la que mostramos esta información con esos requisitos y las limitantes que pueden generarse por utilizar una pantalla con una resolución de 240x240 píxeles es con grupo de medios círculos a los cuales llamamos radar por su similitud a uno, los cuales representan la distancia del vehículo del motociclista que van desde los 5, 10, 20, 30 y 40 metros que son el rango de trabajo del dispositivo y también la zona de riesgo.



Figure 3.32: Mapa dibujado en la pantalla TFT.

Como podemos ver en la figura 3.32 debemos dibujar el radar y los objetos con forma de rectángulo, dada la limitante de no poder mostrar la clase de este, se debe tomar en cuenta que los objetos pueden cambiar de posición o sea alejarse o acercarse, así como pueden cambiar de ángulo en respecto del motociclista por lo que se debe de estar actualizando la imagen en pantalla a nuestro tiempo real que serian 30 dibujos por segundo por lo que se debe dibujar el vehículo en la posición indicada y posteriormente eliminarse para evitar mostrar objetos duplicados.

Desplegado de datos

Teniendo el radar dibujado en la pantalla y la información de los vehículos con sus coordenadas y nivel de riesgo en un arreglo, pasamos a dibujarlos con forma de un rectángulo en su posición correspondiente, dándole al ancho como al alto un cierto tamaño con relación a su nivel de riesgo el cual representa una serie de parámetros como lo es su cercanía, velocidad y clase del vehículo, así mismo como se mencionaba anteriormente para transmitirle mayor información al usuario se le asigna un color con base en el nivel de riesgo y con ello se colocan en pantalla los vehículos existentes en el fotograma.

Capítulo 4

Resultados

4.1 Pruebas

4.1.1 Detección

Para estas pruebas se utilizó la fórmula que podemos ver en la ecuación 4.1 la cual se conoce como el índice de éxito crítico o la puntuación de amenaza, la cual no se ve afectada por la detección de no eventos, los cuales no se utilizan en este trabajo, esta ecuación se utilizó en cinco pruebas las cuales son vídeos tomados desde el sistema funcionando en el campo en cinco situaciones diversas con 900 fotogramas que corresponden a 30 segundos.

$$TS = \frac{TP}{TP + FP + FN}$$

$$TS = \text{Puntuación de amenaza (Threat Score)} \tag{4.1}$$

$$TP = \text{Verdadero positivo (True Positive)}$$

$$FP = \text{Falso positivo (False Positive)}$$

$$FN = \text{Falso negativo (False Negative)}$$

La primer prueba se obtuvo de la motocicleta circulando sobre una vía rápida de subida después de ingresar a esta a una velocidad promedio de 64.52 Km\Hr, esta vía está caracterizada por estar en malas condiciones por lo que en las tomas se puede apreciar como la motocicleta da saltos que alteran la grabación e ingresan ruido al sistema como se puede deducir del TS

promedio de la figura 4.1.

Prueba 1 detección	
Vehículos monitoreados	7
Total de eventos TP	1732
Total de eventos FN	259
Total de eventos FP	24
TS promedio	85.63%

Table 4.1: Resultados de la precisión en las detecciones de la prueba 1.

La segunda prueba se realizó durante la noche, en condiciones no óptimas, tomando en cuenta que se obtienen los datos con un sensor óptico, el cual está pensado para grabar de día, por lo que este es sensible a la falta de luz y a la sobreexposición a ella como puede ser el caso de los vehículos vistos desde el frente en la noche con las luces principales encendidas por lo que se capturan únicamente manchas blancas donde deberían de estar y esto lo podemos ver reflejado en la tabla 4.2.

Prueba 2 detección	
Vehículos monitoreados	8
Total de eventos TP	26
Total de eventos FN	1671
Total de eventos FP	5
TS promedio	1.22%

Table 4.2: Resultados de la precisión en las detecciones de la prueba 2.

La tercer prueba fue grabada en la vía rápida durante un atasco de tráfico, con la motocicleta entrando a esta vía y tomando ventaja de su tamaño reducido, adelantando todos estos vehículos, por lo que la aceleración fue mayor y así mismo el carril por el cual realiza el adelantamiento se encuentra en malas condiciones que producen algunos saltos en las tomas como podemos ver en los resultados de la tabla 4.3.

Prueba 3 detección	
Vehículos monitoreados	31
Total de eventos TP	4136
Total de eventos FN	744
Total de eventos FP	555
TS promedio	77.73%

Table 4.3: Resultados de la precisión en las detecciones de la prueba 3.

La cuarta prueba se realizó ya en la vía rápida en el momento en que una camioneta se pasa del carril izquierdo al carril derecho justo detrás de la motocicleta y se mantiene detrás durante todo el periodo de prueba, por lo que nos sirve para estimar si mantiene o no la detección de esta durante toda la prueba y esto lo vemos en el TS promedio de la tabla 4.4 el cual es mayor al de la tabla 4.3 en la cual no se mantiene mucho tiempo un vehículo en el campo de actuación del detector.

Prueba 4 detección	
Vehículos monitoreados	26
Total de eventos TP	2948
Total de eventos FN	243
Total de eventos FP	366
TS promedio	83.28%

Table 4.4: Resultados de la precisión en las detecciones de la prueba 4.

La quinta y última prueba se realizó en una carretera alterna a la vía rápida en la cual se tenía la luz del sol dando de lleno durante ciertos momentos en la cámara, con lo que se pudo evaluar el desempeño del detector en condiciones no tan favorables que pueden encontrarse en el uso diario del dispositivo, esto lo podemos analizar en la tabla 4.5.

Prueba 5 detección	
Vehículos monitoreados	22
Total de eventos TP	2266
Total de eventos FN	364
Total de eventos FP	120
TS promedio	85.89%

Table 4.5: Resultados de la precisión en las detecciones de la prueba 5.

4.1.2 Seguimiento

Al evaluar el desempeño del sistema en el seguimiento de los vehículos, se tomaron en cuenta los vehículos que podían representar un riesgo para la motocicleta y el usuario de esta, por lo que vehículos que se dirigen en sentido contrario, detenidos a una distancia lejana, detrás de otros vehículos y en ángulos o distancias fuera del alcance de la motocicleta fueron descartados, ya que aunque se les realice un correcto seguimiento no es de importancia para el desarrollo de este trabajo.

Se evaluó el seguimiento sobre las mismas tomas de las pruebas anteriores para llevar una relación en el funcionamiento del sistema.

La manera de evaluar fue la siguiente: desde el primer momento que se le asignó un identificador a un vehículo se contabilizan el número de fotogramas en los que este se mantuvo identificado correctamente y si este cambia de identificador se le restaba una contabilización como amonestación, así mismo si contamos desde el primer fotograma en el que se le asignó identificador y el último fotograma en el que dejó de ser un posible riesgo para el sistema obtenemos la fila de "Deberían de ser" con ello podemos obtener un error y posteriormente promediarlo.

Prueba 1 seguidor				
	Auto 1	Auto 2	Auto 3	Auto 4
Contabilizados	827	511	37	279
Deberían de ser	892	560	50	366
Error (%)	7.3	8.8	26.0	23.8
Error promedio (%)	16.5			

Table 4.6: Resultados del seguimiento en la prueba 1.

De la tabla 4.6 podemos evidenciar que al ser mayor el número de fotogramas en los cuales se realiza la detección del vehículo, el error en el seguimiento del mismo es menor, ya que este se encuentra más tiempo en el campo de funcionamiento del sistema.

En la tabla 4.7 nos confirma que el sistema no funciona bien en la falta de luz, ya que depende de esta para las detecciones y sin las detecciones de vehículos no se puede efectuar un seguimiento a estos, por esto ya no se agregó esta prueba en las secciones siguientes.

Prueba 2 seguidor			
	Auto 1	Auto 2	Auto 3
Contabilizados	4	17	3
Deberían de ser	23	34	10
Error (%)	82.6	50.0	70.0
Error promedio (%)	67.5		

Table 4.7: Resultados del seguimiento en la prueba 2.

El error aumenta en vehículos detectados fuera del ángulo del sistema donde los cataloga como posibles riesgos, así mismo cuando la motocicleta acelera abruptamente, esto genera ruido en el sistema, el cual cambia los identificadores así como los brincos por las malas condiciones de la carretera, esto lo podemos ver en el porcentaje de error del automóvil 13 de la tabla 4.8 donde este error es mayor al promedio.

Prueba 3 seguidor																						
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22
Contabilizados	224	284	699	83	213	76	17	174	59	110	117	43	76	80	47	163	119	66	76	128	58	19
Deberían de ser	226	305	709	88	232	83	17	183	62	110	128	45	93	86	47	184	122	77	85	132	58	19
Error (%)	0.9	6.9	1.4	5.7	8.2	8.4	0.0	4.9	4.8	0.0	8.6	4.4	18.3	7.0	0.0	11.4	2.5	14.3	10.6	3.0	0.0	0.0
Error promedio (%)																						5.5

Table 4.8: Resultados del seguimiento en la prueba 3, AN de Automóvil.

Al tener un vehículo a una corta distancia se puede realizar un seguimiento correcto y en el caso de la prueba 4 en la cual se tenía una camioneta de la marca Nissan[®] a distancia reducida se puede ver que no se cambió de identificador ni una sola vez, así como algunos otros vehículos de la tabla 4.9.

Prueba 4 seguidor							
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5	Auto 6	Auto 7
Contabilizados	900	95	126	573	280	218	42
Deberían de ser	900	114	153	575	280	218	42
Error (%)	0.0	16.7	17.6	0.3	0.0	0.0	0.0
Error promedio (%)							5.0

Table 4.9: Resultados del seguimiento en la prueba 4.

La última prueba fue con la cámara contra la luz, en este caso el error promedio se mantiene como en las demás pruebas, aun cuando se pierde el seguimiento del vehículo principal que representa el posible mayor riesgo para la motocicleta, en esta prueba los vehículos que podían representar un verdadero riesgo eran reducidos, ya que fue grabado en una carretera secundaria a baja velocidad con una cantidad reducida de vehículos en el mismo sentido, por lo que se optó por no realizar las siguientes secciones con esta prueba.

Prueba 5 seguidor					
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5
Contabilizados	183	690	53	46	25
Deberían de ser	215	790	53	46	25
Error (%)	14.9	12.7	0.0	0.0	0.0
Error promedio (%)	5.5				

Table 4.10: Resultados del seguimiento en la prueba 5.

4.1.3 Estimación de distancia

La evaluación de las distancias se realizó de manera cualitativa, tomando en cuenta que no se cuenta con la manera de tener la medición real de la distancia de cada uno de los vehículos en las pruebas, por lo que nos basamos en la calibración del sistema y en el juicio personal para determinar las mediciones que están mal.

La manera de evaluar fue la siguiente: desde el primer momento que se le asignó una medición de distancia a un vehículo se contabilizan el número de fotogramas en los que se mantuvo una medición acercada al promedio y si este obtenía una medición errónea fuera de lugar en la tendencia de ese vehículo o sin sentido al ser exageradamente grande o pequeña se le restaba como amonestación, así mismo si contamos el total de mediciones correctas e incorrectas hasta que el vehículo dejó de ser un posible riesgo para el sistema obtenemos el total de distancias contabilizadas con ello podemos obtener un error y posteriormente promediarlo.

Prueba 1 estimación de distancia					
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5
Distancias estimadas correctas	422	711	17	12	230
Total de distancias contabilizadas	442	718	19	19	233
Error (%)	4.5	1.0	10.5	36.8	1.3
Error promedio (%)	10.8				

Table 4.11: Resultados al estimar la distancia en la prueba 1.

Las detecciones están limitadas por el ángulo y el tamaño de los vehículos detectados

como veíamos en la figura 3.13 por lo que, a causa del movimiento de las malas condiciones de la carretera, inclinaciones de la motocicleta al girar o aceleraciones abruptas de la misma provocan la pérdida del identificador en el vehículo con lo que se pierde el historial del filtro de distancias calculadas y con ello se le pueden asignar mediciones erróneas, como podemos observar en la tabla 4.12 los automóviles con mayor error fue a causa de una de las variables mencionadas anteriormente.

Prueba 2 estimación de distancia														
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5	Auto 6	Auto 7	Auto 8	Auto 9	Auto 10	Auto 11	Auto 12	Auto 13	Auto 14
Distancias estimadas correctas	230	174	601	144	44	25	62	18	27	50	65	65	124	11
Total de distancias contabilizadas	262	179	617	144	44	26	62	23	27	56	77	65	124	11
Error (%)	12.2	2.8	2.6	0.0	0.0	3.8	0.0	21.7	0.0	10.7	15.6	0.0	0.0	0.0
Error promedio (%)	5.0													

Table 4.12: Resultados al estimar la distancia en la prueba 2.

En esta prueba, en la cual se tiene una camioneta cerca de la motocicleta y justo detrás durante todo el periodo de la prueba, nos da como resultado en una estimación de la distancia constante y sin aparentes fallos, los demás vehículos aparecen únicamente cuando adelantan a la camioneta y posteriormente la motocicleta por el carril izquierdo por lo que no obtienen un error mayor al promedio de las pruebas anteriores, esto lo podemos ver en la figura 4.13.

Prueba 3 estimación de distancia							
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5	Auto 6	Auto 7
Distancias estimadas correctas	900	70	98	124	155	108	10
Total de distancias contabilizadas	900	72	102	124	155	108	10
Error (%)	0.0	2.8	3.9	0.0	0.0	0.0	0.0
Error promedio (%)	1.0						

Table 4.13: Resultados al estimar la distancia en la prueba 3.

4.1.4 Estimación de velocidad

La evaluación de las mediciones de velocidad se realizó de manera cualitativa, tomando en cuenta que no se cuenta con la manera de tener la medición real de esta por cada uno de

los vehículos en las pruebas, por lo que nos basamos en la velocidad obtenida del módulo escáner conectado a la motocicleta y en el juicio personal para decidir cuál de las mediciones está fuera del promedio o la tendencia y, por lo tanto, está mal.

La manera de evaluar fue la siguiente: desde el primer momento que se le asignó una medición de velocidad a un vehículo se contabilizan el número de fotogramas en los que se mantuvo una medición acercada al promedio y si este obtenía una medición errónea fuera de lugar en la tendencia de ese vehículo o sin sentido al ser exageradamente grande o pequeña se le restaba como amonestación, así mismo si contamos el total de mediciones correctas e incorrectas hasta que el vehículo dejó de ser un posible riesgo para el sistema obtenemos el total de velocidades contabilizadas con ello podemos obtener un error y posteriormente promediarlo.

Prueba 1 estimación de velocidad						
	Moto	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5
Velocidades estimadas correctas	873	212	302	13	9	132
Total de velocidades contabilizadas	900	227	308	17	13	144
Error (%)	3.0	6.6	1.9	23.5	30.8	8.3
Error promedio (%)	NA	14.24				
Velocidad promedio (KM/HR)	64.5	65.7	67.0	47.4	36.4	63.1

Table 4.14: Resultados al estimar la velocidad en la prueba 1.

En esta prueba el error se mantuvo al margen, ya que la motocicleta iba adelantando a los vehículos quienes estaban detenidos en el tráfico y rápidamente estos pasaban de representar un riesgo a no, porque se quedaban rezagados o detrás de otro vehículo,

Como se puede ver en la tabla 4.15 los vehículos con velocidades muy diferentes en comparación con la motocicleta, como lo es el automóvil 11 obtienen un porcentaje de error mayor, ya que los cambios en ellos son más abruptos.

Prueba 2 estimación de velocidad																
	Moto	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	
Velocidades estimadas correctas	861	141	108	300	137	37	18	54	20	25	35	52	51	87	30	
Total de velocidades contabilizadas	900	146	108	321	138	41	18	54	21	25	39	72	52	87	30	
Error (%)	4.3	3.4	0.0	6.5	0.7	9.8	0.0	0.0	4.8	0.0	10.3	27.8	1.9	0.0	0.0	
Error promedio (%)	NA															4.65
Velocidad promedio (KM/HR)	38.1	31.5	26.8	38.1	38.0	38.5	36.9	39.7	39.1	36.8	35.7	61.5	37.2	37.1	41.4	

Table 4.15: Resultados al estimar la velocidad en la prueba 2, AN de Automóvil.

Prueba 3 estimación de velocidad									
	Moto	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5	Auto 6	Auto 7	
Velocidades estimadas correctas	853	497	24	40	80	88	59	5	
Total de velocidades contabilizadas	900	497	30	42	80	89	59	5	
Error (%)	5.2	0.0	20.0	4.8	0.0	1.1	0.0	0.0	
Error promedio (%)	NA								3.70
Velocidad promedio (KM/HR)	32.6	34.1	36.1	55.1	33.8	18.7	22.2	12.0	

Table 4.16: Resultados al estimar la velocidad en la prueba 3.

4.1.5 Estimación de riesgo

La manera de evaluar fue la siguiente: desde el primer momento que se le asignó una estimación de riesgo a un vehículo se contabilizan el número de fotogramas en los que se mantuvo una estimación acercada y si este obtenía una medición errónea fuera de lugar en la tendencia de ese vehículo o sin sentido al ser de un nivel muy inferior o superior se le restaba como amonestación, así mismo si contamos el total de estimaciones correctas e incorrectas hasta que el vehículo dejo de ser un posible riesgo para el sistema obtenemos el total de estimaciones contabilizadas con ello podemos obtener un error y posteriormente promediario.

La estimación de riesgo general se mantiene alrededor del 13%, en el caso de la prueba 1 que podemos ver en la tabla 4.17 con los vehículos y la motocicleta moviéndose alrededor de 60 kilómetros por hora se obtuvo un 8% de error.

Prueba 1 estimación de riesgo					
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5
Estimaciones de riesgo correctas	196	277	17	13	120
Total de estimaciones contabilizadas	227	308	17	13	144
Error (%)	13.7	10.1	0.0	0.0	16.7
Error promedio (%)	8.08				

Table 4.17: Resultados al estimar el nivel de riesgo en la prueba 1.

En la prueba del adelantamiento a los vehículos se obtiene un error mayor, ya que la estimación acertada del riesgo se obtiene alrededor de 9 fotogramas después de que se le asignara un riesgo y si la estimación se realiza en un número bajo de fotogramas, puede que directamente se le dé una estimación incorrecta en todas sus estimaciones, podemos ver como afecta esta situación en los vehículos con pocas estimaciones en la tabla 4.18.

Prueba 2 estimación de riesgo														
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
Estimaciones de riesgo correctas	124	90	286	118	28	14	46	12	16	30	46	45	75	24
Total de estimaciones contabilizadas	146	108	321	138	41	18	54	21	25	39	72	52	87	30
Error (%)	15.1	16.7	10.9	14.5	31.7	22.2	14.8	42.9	36.0	23.1	36.1	13.5	13.8	20.0
Error promedio (%)	22.23													

Table 4.18: Resultados al estimar el nivel de riesgo en la prueba 2.

La estimación de riesgo de la camioneta Nissan[®] en la tabla 4.19 se realizó la mitad de las detecciones de esta, pero aun así se logró estimar todo el tiempo de la prueba y se le asignó en su gran mayoría una estimación correcta, esta situación representa una situación que en el mundo real se le debería de avisar al motociclista y se logró correctamente.

Prueba 3 estimación de riesgo							
	Auto 1	Auto 2	Auto 3	Auto 4	Auto 5	Auto 6	Auto 7
Estimaciones de riesgo correctas	452	30	42	65	78	43	5
Total de estimaciones contabilizadas	497	30	42	80	89	59	5
Error (%)	9.1	0.0	0.0	18.8	12.4	27.1	0.0
Error promedio (%)							9.61

Table 4.19: Resultados al estimar el nivel de riesgo en la prueba 3.

4.2 Discusión de resultados

En esta sección se pretende analizar los resultados obtenidos de las pruebas en los distintos vídeos.

El promedio del sistema en las pruebas de detección excluyendo la prueba de noche nos da un valor de 83.1%. Este valor está lejos de ser perfecto y más si consideramos la prueba excluida en la cual el sistema simplemente no está preparado para afrontar, lo cual se puede solucionar utilizando la cámara infrarroja con una matriz de leds de gran capacidad lumínica infrarroja para iluminar los vehículos en la noche y así poderlos detectar. Este ayudado de un ángulo distinto de la cámara, ya que el otro problema son los focos frontales de los vehículos, pero al cambiar el ángulo de grabación se corrige este detalle, con ello se podría lograr que el sistema funcione también de noche. Si se usa un sensor óptico de mayor resolución y mejor calidad se lograría detectar a mayor distancia los vehículos y esto es posible con la utilización de otra cámara y aumentando la capacidad de procesamiento de tensores como es el caso del TPU. Del mismo modo, al emplear un modelo de inteligencia artificial con mayor precisión y entrenado con un *dataset* más extenso y específico para vehículos, podría ayudar a acercarse al 100% de detecciones correctas, aun con todo el funcionamiento de este es suficiente para obtener un sistema funcional para el usuario final, ya que este no verá el proceso de detección de los vehículos o el número de fallos.

Por parte del seguidor obtenemos un promedio de error del 8.1% excluyendo la prueba de noche, porque quedo claro que el sistema no está preparado para funcionar correctamente

de noche. Cabe aclarar que esta métrica del error se obtuvo de manera cualitativa, ya que nosotros creamos la tabla y en ella utilizamos como columnas los vehículos identificados por nosotros de manera física basándonos en sus características, el sistema no está capacitado para ello. El seguidor SORT es sumamente rápido en su trabajo y requiere de poco poder de cómputo comparado con los demás seguidores, pero no es el más preciso a la hora de mantener un identificador en el vehículo cuando este está muy lejos de la cámara y está cerca de otros vehículos o cuando la cámara realiza movimientos abruptos como podrían ser huecos en la carretera o cambios de giro de la motocicleta.

Lo anterior se puede solucionar al utilizar un seguidor que utilice características de los objetos como puede ser Deep SORT el cual mantiene de mejor manera el seguimiento aunque requiere más poder de cómputo, por lo que se requiere de otro microprocesador más potente para agilizarlo a tiempo real. Aun con todo ello, el sistema funciona correctamente, ya que se logra mantener la mayor parte del tiempo identificados correctamente a los vehículos. Cuando falla el sistema por algunas de las situaciones anteriormente mencionadas suele ser por pocos fotogramas los cuales el usuario final no notara y al regresar al estado anterior a la falla el identificador suele recuperar el valor anterior correcto. Por lo que para el sistema funciona por encima de lo mínimo aceptable, llegando a ser bueno en ello en el rango de operación no mayor a 40 metros.

El error promedio en las pruebas de estimación de distancia fue del 5.6% para estas se excluye el último vídeo donde se tiene el sol de frente a la cámara, ya que fue grabado en una carretera secundaria de baja velocidad donde no existen muchos vehículos que puedan representar un riesgo. La estimación de distancia depende completamente de la calidad del detector, porque el valor se obtiene con base en los píxeles y a la calibración de la cámara, con un solo fotograma basta para obtener una medición, el algoritmo es simple y eficaz. Las mediciones fueron evaluadas de manera cuantitativa fotograma por fotograma, el usuario final no verá estas mediciones por lo que no le afectara si está a más o menos metros de lo que realmente está, pues esta información se utiliza de manera interna y proporciona la suficiente exactitud para los siguientes algoritmos.

No en todos los fotogramas se le asigna una distancia a los vehículos, ya que estos no son detectados en todos los fotogramas o en algunas situaciones la detección de la cual está sujeta la distancia es errónea, por lo tanto, la medición queda fuera del filtro por superar la relación con base en las mediciones anteriores. Esto podría parecer un error grave, pero como puede verse en las tablas obtenidas en las pruebas de distancia, no existe una afectación al sistema que cause la nula estimación de un vehículo, y estas fallas se pueden solucionar rápidamente al instalar un *gimbal* es cuál estabilizaría la imagen para absorber las vibraciones, inclinaciones o giros de la motocicleta y las imperfecciones de la carretera.

Para aceptar las mediciones nuevas de la distancia de un vehículo primero se comprueba si este lleva en seguimiento más de 5 fotogramas, si es menos se le permite aceptar esa medición, ya que aún es muy pronto para contar con un promedio acertado y al inicio las estimaciones tienen un error mayor. En caso de que este tenga más de 5 fotogramas, se utiliza una relación de $\pm 15\%$ entre el valor promediado anterior de la distancia con el resultado obtenido de simular la inclusión de la nueva estimación de distancia en el filtro con todos los valores anteriormente aceptados.

En el caso de los autobuses o los camiones, el detector suele fallar al asignar correctamente la clase, ya que puede asignarles la clase camioneta o automóvil, los cuales tienen una altura completamente diferente, como podemos ver en la tabla 3.2 por lo que nos puede dar mediciones erróneas de la distancia de acuerdo a la clase. Esto lo podemos ver en la tabla 4.11 con el automóvil 4 el cual es un camión de pasajeros que clasifica mal el detector y posteriormente afecta a sus mediciones de distancia, este es un error importante que se puede evitar utilizando un detector con mayor precisión y una *dataset* que abarque de mejor manera los autobuses y los camiones. Aun en nuestro caso el error de clasificar incorrectamente a un vehículo de estos que suelen ser más altos que los vehículos normales es relevante, pero el usuario no mira estos datos, él únicamente ve los avisos y el sistema sigue realizando mediciones con base en la información que obtiene. Solo en este caso a estos vehículos los toma como si se encontraran a una distancia mucho menor de la que realmente están y les asigna un nivel de riesgo mayor. Es esto último lo que salva el error, ya que estos vehículos

por su tamaño y por la manera en que son conducidos representan un mayor riesgo para el motociclista, por lo que el error no termina afectando en gran medida al sistema.

Por parte de las pruebas de la estimación de velocidad se obtuvo un promedio del 7.5% con lo que el error es reducido, aunque cabe aclarar que aun cuando aparentemente se les estimó la velocidad menos tiempo que la distancia. Lo cual podemos deducirlo de las tablas obtenidas de las pruebas de distancia y las de la velocidad, mirando solo los números se puede llegar a la conclusión de que no se les realizaron las mediciones a los vehículos de las pruebas de velocidad durante el mismo lapso de tiempo que las pruebas de distancia. Esto no fue así, lo que sucede es que existen fotogramas en los cuales no se ejecuta una medición a los vehículos, estos vacíos son a causa de la exclusión del filtro, ya que este utiliza una regla para aceptar o no las nuevas mediciones, por lo que la velocidad se les estima por grupos de fotogramas consecutivos y otros grupos se les deja de estimar siendo estos desde 1 a 6 con un máximo de 15 fotogramas consecutivos. Por lo que hablamos de máximo medio segundo en el cual se les deja de otorgar un aviso de riesgo a esos vehículos. Tomando en cuenta el uso del dispositivo en el cual el usuario está conduciendo la motocicleta en carretera con la mayor parte de su concentración en ello. Y en lapsos de tiempo presta un poco de su atención al módulo de realidad aumentada para estar informado de su entorno y los vehículos en sus puntos ciegos, este vería un vehículo aparecer y desaparecer rápidamente por lo que de manera involuntaria el sistema le estaría llamando la atención hacia esos vehículos.

Para aceptar las mediciones nuevas de la velocidad de un vehículo primero se comprueba si este lleva en seguimiento más de 5 fotogramas, si es menos se le permite aceptar esa velocidad, ya que aún es muy pronto para contar con un promedio acertado y al inicio las estimaciones tienen un error mayor. En caso de que este tenga más de 5 fotogramas, se utiliza una relación $\pm 100\%$ entre el valor promediado anterior de la velocidad con el resultado obtenido de simular la inclusión de la nueva estimación de velocidad en el filtro con todos los valores anteriormente aceptados.

En la primer prueba en la cual la motocicleta va de subida en la vía rápida esta mantiene una velocidad cercana a los 60 kilómetros por hora, así mismo dos de los vehículos detectados

al inicio del vídeo mantienen una velocidad un poco mayor, ya que con el pasar de los fotogramas se acercan a la motocicleta y posteriormente se mantienen detrás de esta hasta finalizar la prueba, la estimación de la velocidad es más susceptible al ruido por lo que como podemos ver en la tabla 4.11 el automóvil 1 paso de 442 mediciones a 227 y el automóvil 2 de 718 mediciones a 308 lo podemos comprobar en la tabla 4.14, esto es debido a un gran número de factores que ya se han mencionado anteriormente que se pueden resumir en que las velocidades estimadas al ser ingresadas al filtro no pasan la relación máxima permitida.

En las últimas pruebas se evaluó el funcionamiento del sistema para dar el nivel de riesgo correcto a los vehículos, de estas pruebas se obtuvo un error del 13.3%. En estas pruebas se culmina la unión de todos los algoritmos anteriores y sus datos para poderle ofrecer al usuario del sistema un aviso de los vehículos que se encuentran en su punto ciego,

A los vehículos se les pueden asignar 5 niveles de riesgo como podemos ver en la figura 3.23 basándose en el posible riesgo que representen para la motocicleta, en estas pruebas estos se evalúan de manera cualitativa con base en su experiencia como motociclista, por lo que se revisó fotograma por fotograma si el nivel asignado de riesgo de las 5 opciones era acertado o si se asignaba un nivel incorrecto.



Figure 4.1: Gráfica mostrando el seguimiento del sistema a un vehículo en las pruebas.

En la gráfica 4.1 podemos ver las mediciones realizadas por el sistema en un periodo de 30 segundos y la salida del nivel de riesgo para un automóvil representado por la línea de color rojo, el eje del lado derecho representa los distintos niveles de riesgo desde el 1 al 4, falta el quinto, pero en esta prueba no se dio la situación que lo amerite asignar. La franja naranja del lado izquierdo representa el lapso en el cual el sistema aún da error al detectar la velocidad del vehículo (esto toma alrededor de 15 fotogramas, o sea medio segundo). La velocidad de la motocicleta está representada por la línea de color verde, debemos notar en ella que esta contiene errores los cuales se obtiene tal cual de la motocicleta, estos errores afectan a los algoritmos utilizados en el sistema, este fallo se puede solucionar agregando un filtrado a la medición de la velocidad de la motocicleta. La línea de color azul muestra las variaciones en la distancia que existen entre el vehículo y la motocicleta. El procesamiento de los valores anteriores nos permite comprender la cuarta línea que es de color negro, está con ayuda de las marcas de riesgo nos detalla el nivel de este basándose en la distancia del vehículo, a la velocidad y al tiempo de la posible colisión. Podemos ver como se va

aumentando de riesgo acorde se va acercando el vehículo, en este caso nunca se tuvo una situación donde el vehículo pudiera colisionar en menos de 2 segundos, por lo que no se entró en dicha condición para dar el 5 nivel de riesgo. Aun cuando rebaso los 5 kilómetros por hora necesarios para darle el 5 nivel no supero los 10 metros de separación, por lo que solo se le asignó el 4 nivel de riesgo, lo cual es correcto, ya que la motocicleta se mantuvo con una velocidad similar al vehículo y la distancia no paso dicho límite.

Capítulo 5

Conclusiones y trabajo futuro

5.1 Conclusiones

Con la realización de este trabajo fue posible crear un dispositivo capaz de detectar posibles riesgos en el punto ciego del motociclista, el cual le proporciona un aviso con realidad aumentada para que este pueda accionar con base en ello, como se logró ver en las pruebas y sus resultados, el sistema es funcional y cumple con sus objetivos.

De acuerdo a las pruebas, el sistema funciona con un error reducido en una distancia no mayor a 40 metros, más allá de ello puede funcionar bajo condiciones ideales. Así mismo se enfrentó a situaciones no ideales con vibraciones, cambios de inclinación, movimientos abruptos causados por las malas condiciones de la vía pública y aun así se mantuvo funcionando y advirtiendo de los posibles riesgos al motociclista. Esto demuestra que es posible la creación de este dispositivo y la mejora de las limitantes vistas en este trabajo fácilmente se pueden corregir como parte del trabajo futuro. Así mismo, se demuestra que esta ayuda aporta información extra y relevante para el motociclista, pues en las pruebas donde este se utilizó fue de verdadera ayuda, aun siendo un prototipo.

Al iniciar este trabajo se realizó una investigación referente a los productos comerciales similares, en ese momento no había implementados sistemas de seguridad activa en motocicletas comerciales, esto después de dos años ha cambiado, pues se han agregado a las motocicletas de mayor cilindrada de las principales marcas del mercado sistemas con detección con ayuda de sensores lidar, estos solo se han implementado en las motocicletas de mayor costo superiores al medio millón de pesos, por lo que queda un gran recorrido para que las motocicletas de precio reducido como pueden ser la gran mayoría de motocicletas en el país puedan contar

con sistemas de este tipo, aunque cabe aclarar que los sistemas de estas motocicletas están enfocados en comodidad, como pueden ser ayudas para mantener una distancia con los vehículos durante largas jornadas de carretera, algunos fabricantes comienzan a agregarles más medidas de seguridad, pero aún tomara un tiempo, en nuestro caso nuestro sistema no controla de ninguna manera el funcionamiento de la motocicleta sino que más bien es un asistente para la conducción y esto ya es una gran ayuda como se logró demostrar y cabe aclarar que todo esto se logró producir a un precio de piezas reducido como se puede ver en la tabla 5.1.

Costo de las piezas por módulo (MXN)					
Realidad aumentada (107Gr)		Escáner		Principal	
Pieza	Costo	Pieza	Costo	Pieza	Costo
Esp8266 D1 mini	99.99	Esp32	85	RPI 4 4GB	2037.39
Pantalla tft 240*240	299.99	Escáner ELM327	200	TPU Coral USB	1300
Batería 1200mAh	153.62	Cable	10	MicroSD 32gb	129
Cable	10	Adaptador OBD2	200	Cámara RPI v2.1	650
Interruptor	15				
Carcasa protectora	34.62				
Plástico reflector	50				
Cargador batería	14.65				
Total por módulo	677.87		495		4116.39
Costo final	5289				

Figure 5.1: Valor en pesos mexicanos del costo de las piezas del sistema.

Con esto quedan sentadas las bases para futuras modificaciones, así mismo la creación de nuevos módulos para este dispositivo o la implementación de nuevas tareas en las cuales pueda aportar asistencia a los motociclistas.

5.2 Trabajo futuro

Esta investigación logró demostrar la posibilidad de crear el dispositivo planteado al inicio de este trabajo de tesis, sin embargo, aún queda trabajo por realizar para complementar aún más este proyecto:

Una de las mejoras al sensor óptico es utilizar una cámara con mayor resolución y mayor calidad de imagen, si se opta por este camino se requerirá agregar mayor cómputo para el procesamiento para lo cual se podría utilizar una de las placas nuevas de la compañía NVIDIA[®][64], para dar una comparación la TPU utilizada en este trabajo tiene una capacidad de inferencia de 4 TOPS la cual es la medida del poder de cómputo de estas placas, en este caso se acaban de anunciar placas que van desde los 20 TOPS por 200, 40 TOPS por 300, 70 TOPS por 400, 100 TOPS por 600 y 275 TOPS por 900 dólares americanos, por lo que las capacidades y posibilidades en este momento son completamente diferentes a cuando se inició este trabajo.

Así mismo, si se cuenta con mayor poder de cómputo, se podría optar por usar un modelo de visión computacional con mayor precisión y con un mejor *dataset* el cual se podría reentrenar para que funcione más específicamente para los vehículos necesarios y pueda distinguirlos de mejor manera.

En cuanto al seguidor SORT funciona correctamente dadas las limitantes, pero se podría optar por uno de mayor precisión, si se cuenta con mayor poder de cómputo una opción podría ser su versión mejorada Deep SORT [65] el cual utiliza la información de la apariencia de los objetos o sea sus características para ayudarse a distinguirlos mejor y realizar de mejor manera el seguimiento, esta es solo una opción haría falta realizar una investigación más a fondo para poder decidir cuáles seguidores probar.

Con respecto a la velocidad se obtiene del módulo escáner, se podría emplear el GPS del celular, lo cual se probó durante este trabajo, pero las mediciones estuvieron muy desplazadas con respecto al mundo real, podría buscarse una manera distinta para no depender de la conexión a la motocicleta, o mejorar la entrega de los datos de esta.

En relación a mejorar la estimación de riesgo, se podrían realizar mas pruebas, agregar más variables a considerar y más sensores para obtener mayor información del entorno, ya que reducir todo a la distancia que hay entre la motocicleta y el vehículo de interés, como su velocidad, ángulo y tamaño para asignarle un nivel de riesgo puede resultar insuficiente.

En cuanto al módulo de realidad aumentada este se puede sustituir por un módulo

aplicación el cual corre sobre un teléfono inteligente, esto se probó durante este trabajo, dándole la capacidad al motociclista de ver lo que captura la cámara trasera en la pantalla del teléfono como si este se tratara de un espejo retrovisor, más aún aportarle las estimaciones de riesgo a los vehículos detectados, se podría trabajar más en este módulo y eliminar la necesidad de los demás módulos, ya que todo el sistema puede correr en los teléfonos inteligentes de ahora, pues la gran mayoría cuenta con suficiente poder de cómputo y algunos con TPU, únicamente se necesitaría el módulo cámara para obtener de manera inalámbrica las imágenes de la parte trasera.

Finalmente se podría utilizar una pantalla microAMOLED (*Active-Matrix Organic Light-Emitting Diode*) con mayor resolución para implementar de mejor manera y con muchas más opciones la realidad aumentada, al iniciar este trabajo no se podían comprar este tipo de pantallas por lo que se utilizó la TFT de 240X240 píxeles de 1.3 pulgadas, hoy en día existen en el mercado pantallas AMOLED de por ejemplo 0.71 pulgadas de diámetro con resolución de 1920X1080 píxeles por lo que al contar con el poder de cómputo necesario para mostrar la información pertinente podrían aumentar las posibilidades del sistema como sistema de seguridad activa a la conducción en gran medida.

Apéndices

5.3 Código estimar distancia

```
def estimar_distancia(objs, count_frames):
    global lista_dist
    global tam_filtro
    relacion = 0
    lado_a = lado_b = long_focal = altura_real = altura_relacional = 0

    for n in range(0, len(objs)):
        centro_x = objs[n].bbox[0].xmin + ((objs[n].bbox[0].xmax -
        objs[n].bbox[0].xmin) / 2)
        lado_b = centro_x - centro_img
        lado_a = imH - objs[n].bbox[0].ymax
        objs[n].angulo[0] = calcular_angulo(lado_a, lado_b)

        for m in range(0, len(lista_dist)):
            if(lista_dist[m][0] == objs[n].id[0]):
                objs[n].distancia[0] = lista_dist[m][2]
                break

    rel_truck = 0
    if(cam_v2_or_v2_or_ir == 1):
        long_focal = 384.0000
    elif(cam_v1_or_v2_or_ir == 2):
```

```

        long_focal = 374.0625
elif (objs[n].clase == [2]):
    alt_real_obj = 1.6
elif (objs[n].clase == [3]):
    alt_real_obj = 1.2
elif (objs[n].clase == [5]):
    alt_real_obj = 3.0
elif (objs[n].clase == [7]):
    rel_truck = (objs[n].bbox[0].xmax - objs[n].bbox[0].xmin)/
    (objs[n].bbox[0].ymax - objs[n].bbox[0].ymin)
    if(rel_truck >= 1.1 and (objs[n].angulo[0] >= -15
    and objs[n].angulo[0] <= 15)):
        alt_real_obj = 1.85
    else:
        alt_real_obj = 3.0

altura_px = objs[n].bbox[0].ymax - objs[n].bbox[0].ymin
distancia = (long_focal * alt_real_obj) / altura_px

if objs[n].id[0] != 0:
    l = index_of(objs[n].id[0], lista_dist)
    if(l != -1):
        relacion = float(lista_dist[l][2] /
        filtro_lista(1, distancia, lista_dist[l], 3))
        if(relacion >= 0.85 and relacion <= 1.15)
        or len(lista_dist[l][3]) <= tam_filtro/3:
            if (distancia !=
            lista_dist[l][3][len(lista_dist[l][3])-1]):

```

```

        objs[n].distancia[0] = lista_dist[1][2] =
        filtro_lista(0, distancia, lista_dist[1], 3)
        lista_dist[1][4] = count_frames

    else:
        lista_tmp = []
        lista_fil_vel = [distancia]
        lista_tmp.append(objs[n].id[0])
        lista_tmp.append(distancia)
        lista_tmp.append(0)
        lista_tmp.append(lista_fil_vel)
        lista_tmp.append(count_frames)
        lista_dist.append(lista_tmp)

    for k in range(0, len(lista_dist)):
        if(count_frames - lista_dist[k][4]) > fps*3:
            lista_dist.pop(k)
            break

    return objs

```

5.4 Código estimar velocidad

```

def estimar_velocidad(objs, count_frames, kph):
    global lista_vel
    freq = cv2.getTickFrequency()
    velocidad = 0

```

```

for n in range(0, len(objs)):
    for m in range(0, len(lista_vel)):
        t = (cv2.getTickCount() - lista_vel[m][5]) / freq
        if(lista_vel[m][0] == objs[n].id[0] and
           lista_vel[m][4] != objs[n].distancia[0]):
            velocidad_relativa = (((float(lista_vel[m][4])
                                   - float(objs[n].distancia[0]))) / float(t)) * 3.6)
            velocidad = velocidad_relativa + kph

    if (objs[n].id[0] != 0 and float(objs[n].distancia[0]) != 0.0):
        l = index_of(objs[n].id[0], lista_vel)
        if(l != -1):
            objs[n].pri_det[0] = lista_vel[l][1]

            t = (cv2.getTickCount() - lista_vel[l][5]) / freq
            if(t >= 0.025 and lista_vel[l][4] !=
               objs[n].distancia[0]):
                val_simulado =
                filtro_lista(1, velocidad, lista_vel[l], 3)
                if(val_simulado == 0):
                    val_simulado = 0.001
                relacion = lista_vel[l][2] / val_simulado

                if(velocidad < 0 and lista_vel[l][6] < 2 and not
                   (relacion >= -2 and relacion <= 2)):
                    lista_vel[l][6] = lista_vel[l][6] + 1

```

```

elif(velocidad > 0 and lista_vel[1][6] > -2
and not (relacion >= -2 and relacion <= 2)):
    lista_vel[1][6] = lista_vel[1][6] - 1

if((relacion >= -2 and relacion <= 2) or
(lista_vel[1][6] == 2) or (lista_vel[1][6] == -2)
or (len(lista_vel[1][3]) <= tam_filtro/2))
and velocidad > 0:
    lista_vel[1][6] = 0
    lista_vel[1][5] = cv2.getTickCount()
    lista_vel[1][4] = objs[n].distancia[0]
    lista_vel[1][2] = objs[n].velocidad[0] =
    filtro_lista(0, velocidad, lista_vel[1], 3)

else:
    if(float(objs[n].distancia[0]) != 0):
        lista_tmp = []
        lista_fil_vel = []
        lista_fil_acel = []
        lista_tmp.append(objs[n].id[0])
        lista_tmp.append(float(count_frames))
        lista_tmp.append(velocidad)
        lista_tmp.append(lista_fil_vel)
        lista_tmp.append(objs[n].distancia[0])
        lista_tmp.append(float(cv2.getTickCount()))
        lista_tmp.append(0)
        lista_vel.append(lista_tmp)
        objs[n].pri_det[0] = count_frames

```

```

for k in range(0, len(lista_vel)):
    if(count_frames - lista_vel[k][1]) > fps*5:
        lista_vel.pop(k)
        break

return objs

```

5.5 Código estimar riesgo

```

def estimar_riesgo(objs, num_frame, vel_moto, n):

    tiempo_choque_max = 2.0
    distancia_choque_max = 50
    offset_sensibilidad_distancia = 0
    offset_sensibilidad_velocidad = 0

    if(modo_sensibilidad_riesgo == 1):
        tiempo_choque_max = 3.0
        distancia_choque_max = 60
        offset_sensibilidad_distancia = +5
        offset_sensibilidad_velocidad = +5
    elif(modo_sensibilidad_riesgo == 3):
        tiempo_choque_max = 1.5
        distancia_choque_max = 40
        offset_sensibilidad_distancia = -5

```

```

offset_sensibilidad_velocidad = -5

if(objs[n].id[0] != 0 and objs[n].distancia[0] != 0)
and (num_frame - objs[n].pri_det[0] >= fps/2):

    vel_carro_m_por_s = objs[n].velocidad[0] / 3.6
    vel_moto_m_por_s = vel_moto / 3.6
    vel_m_por_s = vel_carro_m_por_s - vel_moto_m_por_s

    if(vel_m_por_s != 0):
        tiempo_choque = objs[n].distancia[0] / vel_m_por_s
    else:
        tiempo_choque = 1000

    if(tiempo_choque <= tiempo_choque_max and tiempo_choque > 0
and objs[n].distancia[0] < distacia_choque_max):
        objs[n].riesgo[0] = 5

    elif(objs[n].distancia[0] >= 50 +
offset_sensibilidad_distancia):
        objs[n].riesgo[0] = 1
    elif(objs[n].distancia[0] >= 30 +
offset_sensibilidad_distancia):
        objs[n].riesgo[0] = 2
    elif(objs[n].distancia[0] >= 20 +
offset_sensibilidad_distancia):
        objs[n].riesgo[0] = 3
    elif(objs[n].distancia[0] >= 10 +

```

```
offset_sensibilidad_distancia
and vel_moto > 15 + offset_sensibilidad_velocidad):
    objs[n].riesgo[0] = 4
elif(objs[n].distancia[0] > 0 and objs[n].distancia[0] < 10
+ offset_sensibilidad_distancia and vel_moto > 5
+ offset_sensibilidad_velocidad):
    objs[n].riesgo[0] = 5

return objs
```

Referencias

- [1] G. Welch and G. Bishop, “An introduction to the kalman filter,” *Proc. Siggraph Course*, vol. 8, 01 2006.
- [2] Honda. (2022) Aparca tu honda favorita dentro de tu casa. [Online]. Available: <https://www.honda.es/motorcycles/experience-honda/news-and-events/Aparca-Honda-favorita-dentro-de-casa.html>
- [3] Meta. (2022) Meta quest 2. [Online]. Available: https://store.facebook.com/quest/products/quest-2/?utm_source=www.google.com&utm_medium=oculusredirect
- [4] PokemoGo. (2022) Ultra beast arrival. [Online]. Available: https://pokemongolive.com/es_mx/
- [5] Microsoft. (2022) Hololens2. [Online]. Available: <https://www.microsoft.com/es-es/hololens>
- [6] R. P. Ltd. (2022) Raspberry pi hardware. [Online]. Available: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>
- [7] ——. (2022) Camera. [Online]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html>
- [8] G. LLC. (2022) Usb accelerator datasheet. [Online]. Available: <https://coral.ai/static/files/Coral-USB-Accelerator-datasheet.pdf>
- [9] E. Systems. (2022) Esp32-wroom-32 datasheet. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- [10] ELM. (2016) Elm327 obd to rs232 interpreter. [Online]. Available: <https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>

- [11] Expressif. (2022) Esp8266ex datasheet. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [12] Z. T. Inc. (2022) Tft lcd module. [Online]. Available: <https://www.smart-prototyping.com/image/data/2020/11/102107%201.3%20inch%20TFT%20IPS%20Display%20Module/ZJY133T-IG01.pdf>
- [13] I. Nacional de Estadística y Geografía. (2016) Vehículos registrados en México. [Online]. Available: https://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?#Regreso&c=13158
- [14] S. R. Rosas Osuna and A. Cervantes Trejo. (2015) Los accidentes de motocicleta en México. [Online]. Available: http://conapra.salud.gob.mx/Interior/Documentos/Publicaciones_Especializadas/Los_Accidentes_Motocicleta.pdf
- [15] I. Mexicano del Transporte. (2018) Seguridad vial de los motociclistas en México. [Online]. Available: <https://www.gob.mx/imt/articulos/seguridad-vial-de-los-motociclistas-en-mexico?idiom=es>
- [16] M. Italika. (s.f.) Puntos de venta. [Online]. Available: <https://www.italika.mx/Pagina/Puntos-De-Venta>
- [17] M. Honda. (2020) 5 tecnologías que acompañan las motocicletas Honda. [Online]. Available: <https://www.honda.mx/rp/5-tecnologias-que-acompanan-las-motocicletas-honda/>
- [18] Semovi.cdmx.gob.mx. (2019) Detienen a dos personas acusadas de actos de corrupción en la Semovi. [Online]. Available: <https://semovi.cdmx.gob.mx/comunicacion/nota/detienen-dos-personas-acusadas-de-actos-de-corrupcion-en-la-semovi>
- [19] Yamaha-motor.com.mx. (2021) Financiamiento. [Online]. Available: <https://www.yamaha-motor.com.mx/financiamiento.php>

- [20] P. A. Kuri Morales and A. U. Esteban Marina. (2018) Norma oficial mexicana nom-206-scfi/ssa2-2018, cascos de seguridad para la prevención y atención inmediata de lesiones en la cabeza de motociclistas-acciones de promoción de la salud-especificaciones de seguridad y métodos de prueba, información comercial y etiquetado. [Online]. Available: http://dof.gob.mx/nota_to_doc.php?codnota=5524337
- [21] INEGI. (2021) Vehículos de motor registrados en circulación. [Online]. Available: https://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?#Regreso&c=
- [22] ——. (2020) Accidentes por tipo de accidente. [Online]. Available: https://www.inegi.org.mx/app/tabulados/interactivos/?px=ATUS_4&bd=ATUS&idrt=168&opc=t
- [23] ——. (2020) Accidentes de tránsito terrestre en zonas urbanas y suburbanas según: Categoría y clase de victima. [Online]. Available: https://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?c=13159
- [24] M. Budman, A. Khan, and B. Hurley, *Tendencias tecnológicas 2019: más allá de la frontera digital*. Deloitte Insights, 2019. [Online]. Available: [https://www2.deloitte.com/content/dam/Deloitte/co/Documents/technology/Tendencias_tecnologicas_2019%20\(Reporte%20Completo\).pdf](https://www2.deloitte.com/content/dam/Deloitte/co/Documents/technology/Tendencias_tecnologicas_2019%20(Reporte%20Completo).pdf)
- [25] I. Forbes. (2019) Nueva era disruptiva de la industria automotriz. [Online]. Available: <https://www.forbes.com.mx/nueva-era-disruptiva-de-la-industria-automotriz/>
- [26] C. Sánchez, “Automatización en la industria automotriz: Conceptos y procesos.” vol. 2, pp. 11–15, 12 2014.
- [27] C. N. de Seguridad de Transito. Nueva era disruptiva de la industria automotriz. [Online]. Available: <https://www.conaset.cl/programa/vehiculos/#seguridad>
- [28] O. Flores, M. Fabela, D. Vazquez, M. Martinez, R. Hernandez, L. Sanchez, and M. Molano. (2016) Sistemas de seguridad activa en vehículos: nuevos desarrollos.

- [Online]. Available: <https://www.imt.mx/resumen-boletines.html?IdArticulo=425&IdBoletin=159>
- [29] Toyota. (2020) ¿sabes lo importante que es el abs en tu coche? qué es y cómo te ayuda en tu conducción. [Online]. Available: <https://www.toyota.es/world-of-toyota/articles-news-events/2018/importancia-sistema-frenado-ABS-toyota>
- [30] Mazda. (2020) Sistema de control de tracción (tcs). [Online]. Available: https://owners-manual.mazda.com/gen/es/cx-5/cx-5_8ft3sp17b/contents/05060201.html
- [31] A. Ibañez. (2019) Gafas de realidad aumentada para mejorar la seguridad y crear nuevas experiencias en la conducción. [Online]. Available: <https://tecvolucion.com/gafas-de-realidad-aumentada-seguridad-experiencias-conduccion/>
- [32] IBM. (2021) ¿qué es la inteligencia artificial?. [Online]. Available: <https://aws.amazon.com/es/machine-learning/what-is-ai/>
- [33] ——. (2021) ¿qué es machine learning?. [Online]. Available: <https://www.ibm.com/mx-es/analytics/machine-learning>
- [34] Sonali, B. Maind, and P. Wankar, “Research paper on basic of artificial neural network.” 2014.
- [35] IIC. (2021) Machine learning deep learning. [Online]. Available: <https://www.iic.uam.es/inteligencia-artificial/machine-learning-deep-learning/>
- [36] I. C. Education. (2020) Convolutional neural networks. [Online]. Available: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>
- [37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks.” *Commun. ACM*, vol. 60, no. 6, p. 84–90, may 2017. [Online]. Available: <https://doi.org/10.1145/3065386>

- [38] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition.” *CoRR*, vol. abs/1409.1556, 2015.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*., 2015, pp. 1–9.
- [41] Amazon. (2021) What is computer vision?. [Online]. Available: <https://www.ibm.com/topics/computer-vision>
- [42] B. Marr. (2019) 7 amazing examples of computer and machine vision in practice. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=48ac7bed1018>
- [43] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019. [Online]. Available: <https://books.google.com.mx/books?id=DOamDwAAQBAJ>
- [44] Y. Chen and J. Z. Wang, “Image categorization by learning and reasoning with regions.” *J. Mach. Learn. Res.*, vol. 5, p. 913–939, dec 2004.
- [45] A. D. Rhodes, M. H. Quinn, and M. Mitchell, “Fast on-line kernel density estimation for active object localization.” 2016.
- [46] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109>

- [47] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.
- [48] I. M. Melo, “Realidad aumentada y aplicaciones.” vol. 6(1), pp. 28–35, 6 2018.
- [49] N. Kunkel and S. Soechtig, *Realidad mixta. Las experiencias se vuelven más intuitivas, inmersivas, y empoderadoras.* Deloitte Insights, 2017. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/pa/Documents/technology/2017/2017_TechTrends_Realidad_Mixta.pdf
- [50] B. Mundo. (2016) Qué es la realidad aumentada, cómo se diferencia de la virtual y por qué apple apuesta fuertemente a ella. [Online]. Available: <https://www.bbc.com/mundo/noticias-37678017>
- [51] V. Mollejo. (2018) Realidad mixta, un paso más hacia la tecnología inmersiva definitiva. [Online]. Available: <https://www.redbull.com/mx-es/tecnologia-realidad-mixta>
- [52] I. Tesla. (2020) Autopilot. [Online]. Available: https://www.tesla.com/es_MX/autopilotAI
- [53] B. Brey, *The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit Extensions : Architecture, Programming, and Interfacing.* Pearson Prentice Hall, 2009. [Online]. Available: <https://books.google.com.mx/books?id=gr0OZ3YPTYwC>
- [54] A. González Mateo. (2019) Sensores en los sistemas de seguridad del automóvil. [Online]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/37809/TFG-I-1286.pdf?sequence=1>
- [55] M. BMW. (2015) Side view assist (sva). [Online]. Available: <https://www.bmwmotorcycles.com/en/discover/engineering/technology-detail/comfort-ergonomics/sva.html>

- [56] L. Byeonghak, W. Taekang, and K. Hakil., “Integration of vehicle detection and distance estimation using stereo vision for real-time aeb system.” in *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, INSTICC. SciTePress, 2017, pp. 211–216.
- [57] M. Rezaei, M. Terauchi, and R. Klette, “Robust vehicle detection and distance estimation under challenging lighting conditions,” *IEEE Transactions on Intelligent Transportation Systems.*, vol. 16, no. 5, pp. 2723–2743, 2015.
- [58] J. Redmon and A. Farhadi. (2018) Yolov3: An incremental improvement. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [59] J. Liu and R. Zhang, “Vehicle detection and ranging using two different focal length cameras.” *Journal of Sensors*, vol. 2020, p. 14, 2020.
- [60] G.Gil, G. Savino, S. Piantini, and M. Pierini, “Motorcycle that see: Multifocal stereo vision sensor for advanced safety systems in tilting vehicles.” *Sensors*, vol. 18(1), p. 295, 2018.
- [61] R. P. T. Ltd. (2019) Raspberry pi 4 computer model b. [Online]. Available: https://www.mouser.mx/datasheet/2/630/RIoTboard_07012019_RPI4-MODBP-1GB-1609396.pdf
- [62] RP. (2022) Raspberry pi documentation hardware specification. [Online]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html>
- [63] OpenCV. (2022) Opencv. [Online]. Available: <https://opencv.org/>
- [64] NVIDIA. (2022) Jetson orin modules and developer kit. [Online]. Available: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/>
- [65] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” 2017. [Online]. Available: <https://arxiv.org/abs/1703.07402>