

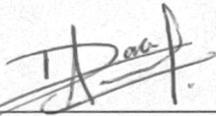
Salamanca, Gto., a 31 de octubre del 2024.

M. en I. HERIBERTO GUTIÉRREZ MARTIN
COORDINADOR DE ASUNTOS ESCOLARES
P R E S E N T E.-

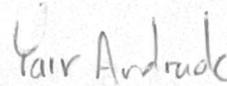
Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) Alfredo Medina García del *Programa de Maestría en Ingeniería Eléctrica (Instrumentación y Sistemas Digitales)* y cuyo número de *NUA* es: 389650 del cual soy director. El título de la tesis es: Estimación de trayectoria de peatones para la navegación autónoma de un vehículo eléctrico.

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

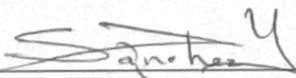
ATENTAMENTE



Dra. Dora Luz Almanza Ojeda
DIRECTOR DE TESIS
SECRETARIO



Dr. Yair Alejandro Andrade Ambriz
DIRECTOR DE TESIS



Dr. Raúl Enrique Sánchez Yañez
PRESIDENTE



Dr. Juan Carlos Gómez Carranza
VOCAL



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO-SALAMANCA
DIVISIÓN DE INGENIERÍAS

*Estimación de trayectoria de peatones para la
navegación autónoma de un vehículo eléctrico*

TESIS

QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN INGENIERÍA ELÉCTRICA

PRESENTA:

Ing. Alfredo Medina García

DIRECTORES:

Dra. Dora Luz Almanza Ojeda

Dr. Yair Alejandro Andrade Ambríz

Agradecimientos Personales

A mi asesora, la Dra. Dora Luz Almanza Ojeda, por su confianza, sus enseñanzas y por darme la oportunidad de vivir experiencias y participar en proyectos que nunca me esperé.

A mi asesor, el Dr. Yair Alejandro Andrade Ambriz, por acompañarme en este trabajo y por transmitirme su seguridad y confianza.

Al Dr. Marco Antonio García Montoya, por las facilidades para usar el prototipo de vehículo eléctrico en la realización de este trabajo.

A mi esposa, Valeria, porque con ella todo es mucho más tranquilo y divertido, gracias por darme paz y estar conmigo siempre.

A mis papás, Martha y Alfredo, por el apoyo que me han dado toda la vida y por sus consejos y enseñanzas que me acompañan a donde sea que vaya.

A mis hermanos, Isa y Chuy, por su apoyo y porque las risas nunca deben de faltar en todo lo que hacemos.

A mi tío Migue, por darme la oportunidad de crecer profesionalmente y como persona, sin duda fue una experiencia que me hizo madurar y me ayudó a tener más confianza en mí.

A mi familia en general, por ser la definición perfecta de unión, gracias por apoyarme en todo lo que hago.

A mi amigo, Jonathan Duarte Jasso, por ser el mejor compañero de trabajo que pude tener estos dos años. Gracias por las enseñanzas y por los buenos momentos.

A mis amigos, Juli y Rodri, porque esto hubiera estado muy aburrido sin los desayunos, el café, las celebraciones, las bromas y las pláticas serias.

Agradecimientos institucionales

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (Conahcyt) por la beca otorgada en la convocatoria “Becas Nacionales 2022” con el número de apoyo 829974.



A la Dirección de Apoyo a la Investigación y el Posgrado (DAIP) de la Universidad de Guanajuato por el apoyo en los proyectos CIIC 059/2023 y CIIC 048/2024: “Implementación de frenado automático de un vehículo eléctrico para personas con discapacidad”, ya que ambos proyectos fueron muy importantes para la realización de este trabajo.



Dirección de Apoyo a la
Investigación y al Posgrado

Índice general

1. Introducción	8
1.1. Justificación	10
1.2. Antecedentes	11
1.3. Objetivos	14
1.3.1. Objetivo general	14
1.3.2. Objetivos específicos	14
1.4. Organización de la tesis	15
2. Estado del arte	16
2.1. Contexto de la investigación	16
2.2. Sensores en los vehículos autónomos	17
2.3. Cámaras en los vehículos autónomos	18
2.3.1. Cámaras RGB convencionales	18

ÍNDICE GENERAL

2.3.2. Cámaras infrarrojas	19
2.3.3. Cámaras de rango controlado	21
2.4. Cámaras de visión estereoscópica	22
2.5. Segmentación de imágenes	24
2.6. Mecanismos de atención	26
2.7. Transformadores de visión	29
2.8. Redes neuronales convolucionales espacio-temporales	30
2.9. Protocolos de transmisión de video	31
3. Metodología	34
3.1. Sistema de visión	36
3.2. Plataforma de pruebas	39
3.3. Base de datos	40
3.3.1. Imágenes de color y profundidad	40
3.3.2. Máscaras de segmentación de entrenamiento	42
3.4. Segmentación y detección de peatones	42
3.4.1. Entrenamiento del modelo ViT	43
3.5. Estimación de niveles de riesgo de colisión	45
3.6. Comunicación entre la unidad de captura y el módulo de procesamiento	46

ÍNDICE GENERAL

4. Pruebas y Resultados	49
4.1. Configuración del sistema de visión	49
4.2. Base de datos	51
4.3. Entrenamiento del modelo de transformadores de visión . . .	53
4.4. Entrenamiento de la Red Neuronal Convolutiva Espacio- Temporal (ST-CNN)	59
4.5. Implementación del módulo	67
5. Conclusiones	72

Resumen

Los accidentes de tráfico han causado grandes pérdidas humanas y materiales durante mucho tiempo. Ante esto, se han propuesto distintas soluciones desde varios ámbitos, y una de las más relevantes está relacionada con los vehículos autónomos. El objetivo principal de la conducción autónoma es garantizar la seguridad de todas las personas que interactúan con los vehículos, haciendo un énfasis en la seguridad de los usuarios más vulnerables, como el caso de los peatones. En este trabajo se presenta la implementación de un módulo basado en visión que ayuda a prevenir colisiones con peatones. Este módulo cuenta con una cámara estéreo, de la cual se obtienen imágenes RGB e imágenes de profundidad. Con las imágenes RGB se realiza una segmentación de peatones usando transformadores de visión. Las máscaras segmentadas obtenidas se utilizan para detectar a los peatones y para aislar la información de las personas en las imágenes de profundidad. La estimación del riesgo de colisiones se obtiene usando una red neuronal convolucional espacio-temporal entrenada con distintas situaciones de interacción entre peatones y un vehículo en un ambiente de exterior controlado. Las pruebas se realizaron usando un prototipo de vehículo eléctrico, el cual fue conducido dentro de las instalaciones de la División de Ingenierías del Campus Irapuato-Salamanca de la Universidad de Guanajuato. Como resultado, se obtuvo un módulo que segmenta peatones con una precisión y exactitud de aproximadamente 90 % y que puede estimar la trayectoria de los peatones detectados para determinar en tiempo real si existe el riesgo de que ocurra una colisión. Si el riesgo existe, lo clasifica en un nivel bajo, medio o alto.

Capítulo 1

Introducción

En términos simples, un vehículo se considera autónomo cuando incorpora dispositivos electrónicos y mecánicos para sustituir la labor humana de la conducción [1]. El sistema autónomo de un vehículo está compuesto principalmente por sensores, unidades de procesamiento, dispositivos de control y actuadores. Con los sensores se percibe el entorno del vehículo y se recopila información para interpretar la situación en la que este se encuentra. En la unidad de procesamiento se analiza la situación y se decide qué acciones se deben realizar, las cuales pueden extenderse desde algo simple como emitir una alarma, hasta acciones fundamentales en un vehículo como frenar, acelerar o girar el volante. Estas últimas son ejecutadas físicamente por los actuadores a través de los dispositivos de control.

Si bien la autonomía no está ligada a un tipo de vehículo en particular, existe una tendencia a que los vehículos autónomos sean eléctricos. La razón principal radica en la facilidad para conectar todos los sistemas del vehículo a una misma fuente de alimentación. Actualmente, los vehículos eléctricos también cuentan con sensores en el motor o la batería, los cuales ayudan a monitorear el consumo de energía. Si esta información se fusiona con los datos del sistema autónomo, se pueden implementar estrategias como la planeación de rutas para optimizar el uso de la energía.

Con el auge de los vehículos eléctricos autónomos surgió la duda sobre qué tipos de vehículos podían considerarse dentro de esta definición. Por esta razón, la Sociedad de Ingenieros Automotrices (SAE, por sus siglas en inglés) publicó en 2014 un estándar que propone una taxonomía con seis niveles de autonomía. Este estándar ha sido modificado y actualizado hasta llegar a la versión más reciente publicada en el 2021 [2], la cual describe los niveles de la siguiente manera:

- Nivel 0 - Sin automatización: el vehículo es operado completamente por una persona.
- Nivel 1 - Conducción asistida: el vehículo está equipado con herramientas que ayudan a mejorar la conducción, aunque sigue siendo una persona quien opera el vehículo.
- Nivel 2 - Automatización parcial: el vehículo realiza tareas simples de movimiento del vehículo, las cuales son supervisadas por una persona que realiza el resto de las tareas.
- Nivel 3 - Automatización condicional: el vehículo se conduce de manera autónoma bajo ciertas condiciones o rutinas; sin embargo, puede requerir la intervención de una persona.
- Nivel 4 - Automatización alta: el vehículo realiza recorridos completos de forma autónoma con rutas predefinidas. En este nivel ya no se requiere la intervención de una persona.
- Nivel 5 - Automatización total: el vehículo es completamente autónomo bajo cualquier condición geográfica o climática en la que pueda transitar un vehículo comúnmente.

De acuerdo con estos niveles, se puede considerar que en la actualidad la mayor parte de los vehículos que circulan en el mundo pertenecen a los niveles 0 y 1. Los modelos de vehículos más recientes pueden alcanzar el nivel 2, e incluso el nivel 3; sin embargo, estos últimos aún no son asequibles para la mayoría de la población. Un ejemplo de un vehículo nivel 4 es el modelo *Waymo One*, un vehículo eléctrico que se usa como transporte privado en algunas ciudades de Estados Unidos [3]. Aunque este vehículo opera de manera completamente autónoma, al ser usado solo en algunas ciudades no puede ser considerado dentro del nivel más alto de autonomía.

1.1 Justificación

La idea es seguir avanzando hasta llegar al nivel 5; sin embargo, la inserción de la autonomía en los vehículos debe ser gradual. Actualmente, una tarea muy importante es mejorar las herramientas de asistencia para la conducción, ya que estas ayudan a que las personas empiecen a familiarizarse con los sistemas autónomos. Además, proporcionan una solución rápida al problema principal que aborda el área de la conducción autónoma: reducir los accidentes de tráfico.

1.1. Justificación

La Organización Mundial de la Salud (OMS) estima que aproximadamente 1.19 millones de personas en todo el mundo pierden la vida cada año a causa de accidentes de tráfico, siendo los peatones, los ciclistas y los motociclistas los usuarios más afectados [4]. En México, el Instituto Nacional de Estadística y Geografía (INEGI) registró en 2023 más de 380 mil accidentes, de los cuales al menos el 96% fueron a causa del conductor. Además, el atropellamiento fue uno de los cinco tipos de accidentes más comunes [5]. Algunos factores como la distracción, los puntos ciegos, la somnolencia, el manejo bajo efectos de sustancias o el exceso de velocidad, influyen en la capacidad de decisión y respuesta de los conductores ante una situación de peligro.

Una solución sencilla para asistir al conductor en situaciones de distracción es la inclusión de sensores de proximidad en las partes delantera y trasera del vehículo. Estos ayudan a emitir una alarma cuando detectan que un objeto o persona está demasiado cerca. En el caso de los puntos ciegos, el ejemplo más común se presenta cuando se realizan maniobras de reversa. Por esta razón, en algunos vehículos se instala una cámara en la parte trasera y una pantalla a la vista del conductor para que este pueda observar lo que hay detrás. De esta forma, puede realizar las maniobras con seguridad. Ambas herramientas son de gran utilidad; sin embargo, su función solo es facilitar algunas tareas al conductor, quien sigue teniendo el control total del vehículo.

1.2 Antecedentes

La verdadera influencia de la conducción autónoma está en la integración de módulos complejos que no solo sirvan como ayuda para el conductor, sino que sustituyan sus tareas parcial o totalmente. La aceptación social de los vehículos autónomos depende en gran medida de este tipo de módulos, principalmente aquellos relacionados con la protección de usuarios vulnerables, como los peatones [6]. Los módulos de prevención de colisión con peatones se centran en cuatro aspectos esenciales: la detección de peatones, los enfoques anticolidión, la eficiencia computacional y los sistemas de acción. Cada uno de ellos es igual de importante; se deben detectar correctamente los peatones y tener una estrategia adecuada para evaluar los riesgos de colisión. El proceso debe ser eficiente para tener una respuesta rápida y esta debe ejecutarse de forma adecuada por el sistema de acción.

Este trabajo se centra en los dos primeros aspectos: la detección de peatones y el enfoque anticolidión. Se presenta una metodología basada en visión con una técnica de detección de peatones por segmentación. El enfoque anticolidión está basado en niveles de riesgo para determinar si existe un riesgo nulo, bajo, medio o alto de que un vehículo colisione con un peatón. Como resultado, se obtendrá un módulo que realice estas tareas y que pueda ser integrado a un sistema autónomo para la navegación autónoma de un vehículo eléctrico.

1.2. Antecedentes

Un módulo para prevenir colisiones con peatones puede construirse desde diferentes enfoques y con distintas técnicas. A pesar de esto, existen etapas que todos tienen en común, como la detección de peatones. Generalmente, esta tarea se realiza usando imágenes, ya que estas ayudan a percibir el entorno de una manera muy similar a como lo haría una persona al manejar. La técnica más utilizada actualmente para detectar peatones en imágenes son las redes neuronales convolucionales (CNNs, por sus siglas en inglés) y las redes neuronales recurrentes (RNNs, por sus siglas en inglés), debido a su capacidad para reconocer patrones complejos.

1.2 Antecedentes

Además de detectar a los peatones, es indispensable saber también a qué distancia se encuentran estos en relación con el vehículo, y este dato se puede obtener de varias formas. Una de las más comunes, es combinar la detección de los peatones en imágenes con datos de otros sensores, como los radares o los sensores de detección y medición por láser (LiDAR, por su nombre en inglés).

De forma muy general, el funcionamiento de un radar consiste en la emisión de ondas de radio que viajan a través del aire, las cuales impactan con los objetos y regresan al dispositivo que los emitió. Tomando como referencia el tiempo que tardaron las ondas en regresar, se puede estimar la distancia a la que se encuentran los objetos. La combinación de cámaras y radares puede usarse para realizar tareas más específicas, como la detección de peatones que están ligeramente ocultos [7] o la detección de peatones en ambientes nocturnos [8], [9].

Los sensores LiDAR tienen un funcionamiento parecido al de los radares, con la diferencia de que estos emiten pulsos de luz láser en lugar de emitir ondas de radio. La ventaja que brindan los pulsos láser es que no solo dan información de la distancia de los objetos, también permiten generar nubes de puntos de los mismos, ayudando a percibir mejor la forma que tienen. La fusión de sensores LiDAR con cámaras, normalmente está enfocado a la detección de objetos tridimensionales [10], [11] y también a la detección de peatones en ambientes nocturnos [12].

La combinación de sensores permite simplificar múltiples tareas; sin embargo, fusionarlos y sincronizarlos para adquirir los datos adecuadamente puede ser un proceso complejo. Una alternativa para evitar la fusión de sensores es el uso de cámaras binoculares, las cuales están formadas por dos lentes que permiten capturar imágenes de forma simultánea desde distintas perspectivas. Al hacer una correspondencia de las imágenes es posible crear un efecto de profundidad, similar al proceso que realizan los ojos y el cerebro en un humano. Este proceso recibe el nombre de fusión estereoscópica.

La fusión estereoscópica de imágenes permite obtener puntos característi-

1.2 Antecedentes

cos o nubes de puntos de los objetos o peatones con sus respectivas coordenadas tridimensionales. Esta información puede ser usada para entrenar clasificadores que ayuden a disminuir la detección de “pseudopeatones” [13], es decir, la detección de objetos o formas que son detectados erróneamente como peatones. Si, además, esta información se procesa de forma adecuada, por ejemplo, usando aprendizaje profundo, se puede hacer detección de objetos en tiempo real e incluso bajo distintas condiciones ambientales y a distintas horas del día [14].

Una vez que se detecta a los peatones y se conoce la distancia a la que están, es importante determinar si existe o no el riesgo de una colisión. En esta parte, los antecedentes sobre enfoques anticolidión sí son muy diferentes, ya que existen muchas variaciones tanto en herramientas como en técnicas. Se puede destacar el uso de enfoques basados en el comportamiento de los peatones en un ambiente urbano para predecir su trayectoria cuando esta es incierta. Estos están basados en el Modelo de Fuerza Social y son combinados con modelos matemáticos, como el modelo de Markov [15], [16] o con estimaciones probabilísticas [17].

Otros enfoques se centran principalmente en la interacción vehículo-peatón. Esta se puede interpretar como una interacción de dos elementos en un mismo sistema y relacionarla con la teoría de la entropía, la cual se determina considerando el espacio de reacción que hay entre el vehículo y el peatón. [18]. Por otro lado, también es posible utilizar el aprendizaje profundo para modelar interacciones complejas en ambientes donde están involucrados muchos peatones [19].

Después de analizar las técnicas y herramientas que se han usado, a continuación se presentan los objetivos de este trabajo como una propuesta de una forma diferente de afrontar el mismo reto de evitar colisiones de vehículos con peatones.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar e implementar una metodología para estimar situaciones de riesgo de colisión con peatones en tiempo real usando un sistema de visión integrado en un carro eléctrico para la navegación en un ambiente de exterior controlado.

1.3.2. Objetivos específicos

- Configurar un sistema de visión para la adquisición y procesamiento de imágenes e integrarlo a un prototipo de vehículo eléctrico.
- Desarrollar una metodología para detectar peatones en tiempo real en un ambiente exterior controlado.
- Desarrollar una metodología para evaluar eventos de riesgo en tiempo real durante la navegación de un prototipo de vehículo eléctrico en un ambiente exterior controlado.
- Integrar las metodologías con el sistema de visión e incorporar el módulo completo en un prototipo de vehículo eléctrico.
- Establecer una comunicación desde el prototipo a una estación de trabajo remota mediante un modelo TCP/IP para la transmisión de los datos.
- Realizar pruebas experimentales en un ambiente exterior controlado y obtener del módulo una señal de salida que indique el nivel de riesgo de colisión en tiempo real.

En cuanto a la metodología, el objetivo del trabajo es comprobar que se puede realizar la tarea de detección de peatones usando un método alternativo a las redes neuronales convolucionales y, adicionalmente, probar el funcionamiento de un modelo de red para clasificar niveles de riesgo con base en el movimiento de los peatones. Este último ha sido usado en otros trabajos para clasificar actividades humanas y en este trabajo se evaluará

1.4 Organización de la tesis

su rendimiento con otro tipo de actividades y bajo distintas circunstancias del ambiente. Al finalizar el trabajo, se obtiene un sistema de detección de peatones y estimación de riesgos que será evaluado para determinar si es óptimo para aplicaciones de navegación autónoma.

1.4. Organización de la tesis

Este documento se divide en 5 capítulos. El presente capítulo ha brindado una breve introducción sobre los vehículos de conducción autónoma y su relación con los vehículos eléctricos; brinda un panorama de la actualidad en el área de la conducción autónoma y da una descripción del trabajo realizado en esta tesis. En el capítulo dos se establecen y describen las bases sobre las cuales se fundamenta este trabajo, así como los conceptos esenciales para seguir la metodología. Esta última se describe en el capítulo tres, dando detalles sobre las técnicas y las consideraciones que se usaron para cumplir con los objetivos planteados al principio de esta tesis. En el capítulo cuatro se mencionan las pruebas realizadas y se muestran los resultados de las mismas. Estos resultados son analizados y en el último capítulo se presentan las conclusiones correspondientes.

Capítulo 2

Estado del arte

En este capítulo, se presentan los conceptos que se utilizaron como fundamento para el desarrollo de este trabajo. Se abordan los temas principales como la visión por computadora y los diferentes sensores utilizados para percepción de ambientes desde vehículos autónomos, haciendo referencia a otros trabajos relevantes en cada una de las áreas.

2.1. Contexto de la investigación

El objetivo de este trabajo es implementar un módulo basado en visión que pueda ser incorporado a un sistema de navegación autónoma y ayude a prevenir colisiones con peatones. El módulo comprende dos tareas fundamentales: la detección de peatones y la estimación del nivel de riesgo de que ocurra una colisión entre el vehículo y el peatón. Aunque existen muchas técnicas y formas de realizar estas dos tareas; la información proporcionada en este capítulo está enfocada directamente a los métodos que se proponen en este trabajo.

2.2 Sensores en los vehículos autónomos

Dado que se trata de un módulo basado en visión, la primera etapa de la investigación se centró en trabajos que utilizan cámaras como sensores y, por ende, imágenes como datos de entrada. La segunda etapa está relacionada con la detección de peatones, la cual se orientó a la segmentación de imágenes y al uso de modelos basados en mecanismos de atención, principalmente en los transformadores de visión.

La tercera etapa, que corresponde a la estimación del nivel de riesgo de que ocurra una colisión, se enfocó en técnicas que estiman el nivel de riesgo basándose en la posible trayectoria de los peatones. De las técnicas que se mencionaron, se resaltó el uso del aprendizaje profundo, específicamente el uso de redes neuronales convolucionales temporales (TCNN, por sus siglas en inglés).

En este trabajo, se consideró la posibilidad de que la captura de imágenes y el procesamiento de las mismas se realice en módulos separados, por lo tanto, la última etapa de la investigación estuvo orientada a los protocolos de comunicación para transmisión de video y la transferencia de datos entre dos dispositivos. Cada uno de los temas abordados en este capítulo se describió primero de manera general y posteriormente se hizo mención a trabajos previos en los que se aplican estos conceptos el área de la conducción autónoma.

2.2. Sensores en los vehículos autónomos

Un sensor es un dispositivo que recibe una señal física o química del entorno y la transforma en una señal eléctrica que pueda ser interpretada por una máquina. Si se explica mediante una analogía, se podría decir que los sensores son para una máquina algo similar a los sentidos en el cuerpo humano. Así como una persona utiliza varios sentidos al conducir un vehículo, los vehículos autónomos también pueden contar con distintos sensores para percibir mejor su entorno. Los sensores que se usan comúnmente en los vehículos autónomos son las cámaras, los sensores de detección y medición

2.3 Cámaras en los vehículos autónomos

por láser (LiDAR, por su nombre en inglés), los radares, los sensores ultrasónicos y los sistemas de posicionamiento global (GPS, por sus siglas en inglés). Cada sensor tiene ventajas y desventajas, las cuales están relacionadas con aspectos como el rango de alcance, la susceptibilidad a condiciones externas (luz, clima o interferencia de otras señales), la velocidad de detección, entre otras [20]. Por esta razón, es esencial conocer las características de cada uno y utilizarlos adecuadamente de acuerdo con sus capacidades.

2.3. Cámaras en los vehículos autónomos

Como se mencionó anteriormente, los vehículos autónomos pueden tener uno o varios sensores y de esto depende su capacidad para realizar ciertas tareas; sin embargo, hay sensores que podrían considerarse como fundamentales en cualquier vehículo autónomo; tal es el caso de los sensores de visión. Los más utilizados en el área de la conducción autónoma son las cámaras RGB convencionales, las cámaras infrarrojas y las cámaras de rango controlado [21], cuya descripción se presenta a continuación.

2.3.1. Cámaras RGB convencionales

Las cámaras RGB convencionales capturan imágenes compuestas por 3 canales: rojo, verde y azul, y su nombre viene de la primera letra de cada color en inglés (R, por *red*, G, por *green* y B, por *blue*). En el diagrama de la Figura 2.1 se ejemplifica la separación de una imagen en estos tres canales y se muestra cómo se ve la imagen original en cada uno. Quizás se esperaría ver, por ejemplo, en la imagen del canal rojo un solo círculo en color rojo; sin embargo, como se observa en imagen, lo que se obtiene es un círculo en color blanco ubicado en la misma posición que tiene el círculo rojo en la imagen original. Esto se debe a que, en los canales, cada píxel solo puede tener un valor entre 0 y 255, donde cero representa el color negro y 255 representa el color blanco. Si un píxel es de color rojo en la imagen RGB, su valor en en

2.3 Cámaras en los vehículos autónomos

el canal rojo será de 255, en cambio, si un píxel es de color azul, este tendrá un valor de cero en el canal rojo, pero un valor de 255 en el canal azul, y se aplica lo mismo para los píxeles verdes en su respectivo canal.

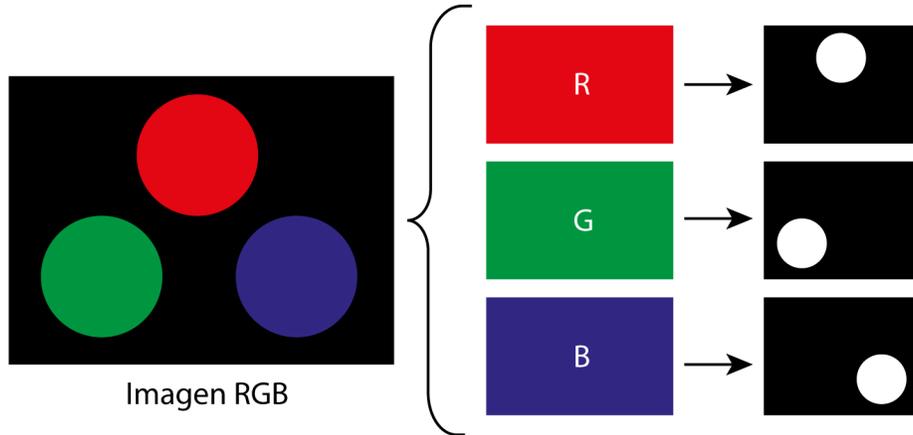


Figura 2.1: Descomposición de una imagen RGB en sus tres capas.

Las cámaras que se encuentran en dispositivos como los celulares o las computadoras capturan imágenes RGB y son el tipo de cámara más común, por eso reciben el nombre de *cámaras convencionales*. En general, son cámaras de fácil acceso, y aunque su costo suele ser relativamente bajo, también pueden tener costos elevados dependiendo de sus características.

2.3.2. Cámaras infrarrojas

Una característica de las cámaras RGB convencionales, es que solo captan la luz visible del espectro electromagnético, cuyos niveles se muestran en la Figura 2.2. Como se observa, la luz visible solo representa una pequeña parte del espectro electromagnético, pero los demás niveles también tienen aplicaciones aunque no puedan ser captados por el ojo humano, y un ejemplo de esto son las cámaras infrarrojas.

Como se observa en la Figura 2.2, la luz infrarroja tiene un rango de longitud de onda que puede ir desde los 700 nm a 1 mm. Dependiendo del rango de longitud de onda con el que trabajan, las cámaras infrarrojas usadas

2.3 Cámaras en los vehículos autónomos

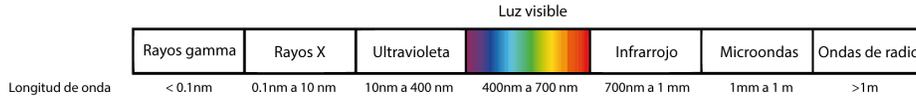


Figura 2.2: Niveles del espectro electromagnético y sus rangos de longitud de onda.

en aplicaciones de conducción autónoma pueden ser de tres tipos: de onda cercana ($0.7\mu m$ a $1.4\mu m$), de onda corta ($1.4\mu m$ a $3\mu m$) o de onda larga ($8\mu m$ a $14\mu m$). Las cámaras de onda cercana y onda corta funcionan similar a una cámara convencional, ya que transforman fotones en señales eléctricas, mientras que las de onda larga, también conocidas como cámaras térmicas, convierten la radiación térmica en calor y posteriormente convierten el calor a señales eléctricas [21]. En la Figura 2.3 se puede observar la diferencia entre una imagen tomada con una cámara RGB convencional y una imagen tomada con una cámara térmica.



Figura 2.3: Diferencia entre una imagen tomada con una cámara RGB convencional (a) y una tomada con una cámara térmica (b). *Adaptada de: Chen et al. (2019) [22].*

Las imágenes de cámaras térmicas pueden representarse en forma parecida a la escala de grises, como se muestra en la Figura 2.3, o también con una escala que va de rojo a azul, en la que normalmente el rojo representa las temperaturas altas y el azul las temperaturas bajas.

2.3.3. Cámaras de rango controlado

Las cámaras de rango controlado cuentan con un módulo llamado iluminador que emite pulsos de luz para iluminar la escena que se desea captar. Dichos pulsos se reflejan en la superficie de los objetos y regresan a la cámara, la cual tiene un sensor receptor que ayuda a generar la imagen. A diferencia de las cámaras RGB convencionales y algunas de las cámaras infrarrojas, las cámaras de rango controlado no cuentan con un sistema que controle la cantidad de luz que llega al sensor receptor. Los fotones que emite el iluminador rebotan en los objetos y el tiempo que tardan en regresar a la cámara depende de la distancia a la que se encuentran. Las cámaras de rango controlado tienen un sistema que funciona como una compuerta que solo deja pasar los fotones que regresan en un intervalo de tiempo programado previamente, lo cual no solo ayuda a generar las imágenes, también permite saber a qué distancia están los objetos en la escena [21].

Gracias al sistema de compuertas de las cámaras de rango se pueden capturar solo los objetos que se encuentran a una cierta distancia, dependiendo del tiempo en el que se abre y se cierra la compuerta. Un primer intervalo puede captar solo los objetos cercanos, otro los objetos a una distancia media y otro los objetos más lejanos. Al final, si se quiere guardar una imagen con la escena completa, lo que se hace es juntar todas las imágenes obtenidas en cada intervalo y unir las en una sola. Este proceso se representa gráficamente en el diagrama de la Figura 2.4.

Una ventaja importante de las cámaras de rango controlado, es que la percepción de los objetos en las escenas depende de los pulsos de luz emitidos por el iluminador y no de la iluminación exterior que recibe la escena, por lo tanto, el uso del iluminador permite que las cámaras de rango controlado puedan ser usadas en ambientes en los que hay deficiencia o exceso de luz, como se muestra en la Figura 2.5.

En aplicaciones como la conducción autónoma es muy importante saber a qué distancia se encuentran los objetos del entorno y las cámaras de rango controlado tienen ese factor a su favor en comparación con las cámaras RGB

2.4 Cámaras de visión estereoscópica

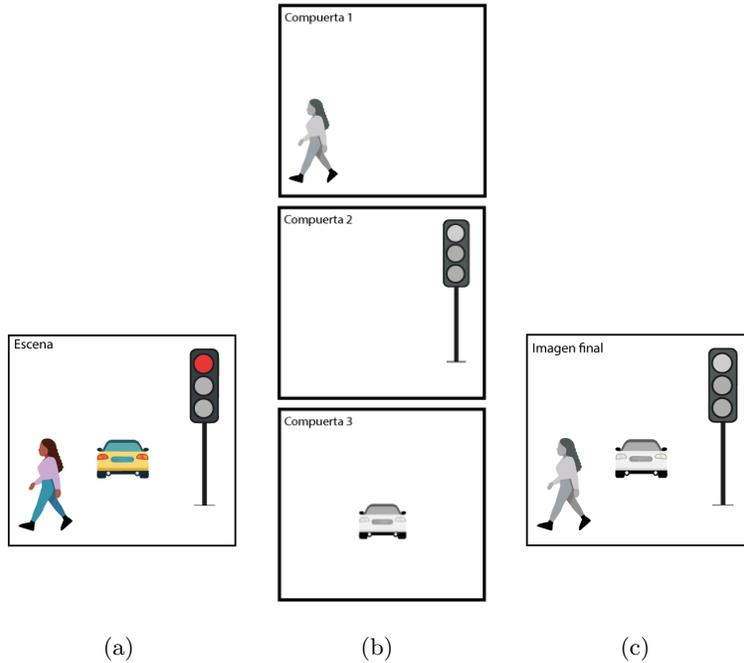


Figura 2.4: Capturas de una escena (a) con compuertas de distintos intervalos (b) y el resultado de la suma de las compuertas (c). (*La imagen incluye elementos generados con la IA de Adobe Illustrator [23]*).

convencionales y las cámaras infrarrojas. La razón por la que no se usan tanto tiene que ver con su precio, ya que este puede ser mucho más elevado que el de las cámaras convencionales; sin embargo, este factor puede ser compensado en las cámaras convencionales usando sensores complementarios o aplicando algunas técnicas como la visión estereoscópica.

2.4. Cámaras de visión estereoscópica

La visión estereoscópica es una técnica que permite extraer información de profundidad de una escena captada desde dos puntos de vista diferentes al mismo tiempo. El proceso para extraer dicha información es similar al que realiza el cerebro humano para dar un sentido de profundidad utilizan-

2.4 Cámaras de visión estereoscópica



Figura 2.5: Comparación de una escena capturada con una cámara RGB convencional (a) y la misma escena con una cámara de rango controlado (b). *Adaptada de Li et al. (2019) [21].*

do la información captada por los ojos. En este caso, la función de los ojos la realizan dos cámaras alineadas horizontalmente con una separación determinada, las cuales captan imágenes de una escena al mismo tiempo. En cada imagen, los objetos de la escena ocupan una posición distinta en cuanto a píxeles y esta disparidad es la que ayuda a indicar la posición, relación y estructura de los objetos [24].

La visión estereoscópica tiene fundamentos matemáticos basados principalmente en operaciones con matrices que permiten calcular de forma correcta la diferencia de posición de un mismo punto en dos imágenes. Esto ayuda a obtener datos de profundidad correctos y sin distorsiones, lo cual hace que esta técnica sea adecuada para usarse en sistemas para vehículos autónomos [25]. En la forma más elemental, en los sistemas de los vehículos autónomos se pueden usar dos cámaras RGB convencionales para capturar las imágenes y, posteriormente, aplicar la técnica tomando en cuenta otros factores como la calibración de las cámaras, principalmente si el sistema va a ser usado bajo distintas condiciones ambientales [14].

Con el avance de la tecnología han surgido las cámaras de visión estereoscópica, las cuales pueden tener un aspecto parecido al que se muestra en la Figura 2.6. Estas cuentan con un arreglo de dos cámaras RGB convencionales horizontalmente alineadas y con distancia entre ellas llamada línea base. Esta juega un papel fundamental, ya que puede influir en la disparidad,

2.5 Segmentación de imágenes

en el ángulo de visión e incluso en la adaptabilidad física del dispositivo.

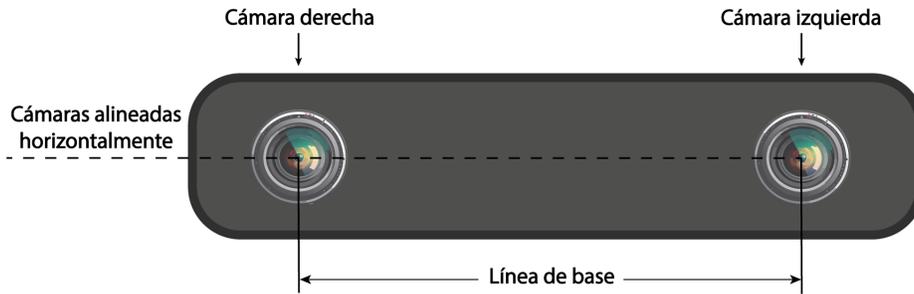


Figura 2.6: Características generales de una cámara de visión estereoscópica. (La imagen incluye elementos generados con la IA de Adobe Illustrator [23]).

La gran ventaja de estas cámaras, es que aplican la técnica de visión estereoscópica de manera automática, incluso sin necesidad de llevar a cabo un proceso de calibración. Esto puede ser muy útil para aplicaciones generales, ya que se pueden obtener fácilmente las imágenes RGB con sus respectivas imágenes de profundidad de forma rápida y eficiente. En aplicaciones con tareas más específicas o cuando no se cuenta con una cámara de este tipo, entonces es necesario aplicar la técnica de visión estereoscópica considerando todos los factores mencionados.

2.5. Segmentación de imágenes

La segmentación de imágenes es una técnica que consiste en dividir una imagen en regiones (como se observa en los ejemplos de Figura 2.7) considerando las características de sus píxeles. La segmentación puede ser de dos tipos: semántica o por instancias. La segmentación semántica realiza un etiquetado de cada píxel de la imagen asignándole una de las clases que se deben establecer previamente. Si se requiere segmentar personas, a cada píxel se le asigna una etiqueta para indicar si este pertenece o no a la clase persona, y al final se puede obtener una imagen que resalte a las personas. Por otro lado, la segmentación por instancias no solo etiqueta los píxeles

2.5 Segmentación de imágenes

para indicar si pertenecen o no a una clase, también es capaz, por ejemplo, de identificar si hay varias personas en la imagen y asignar cuáles píxeles le corresponden a cada persona.

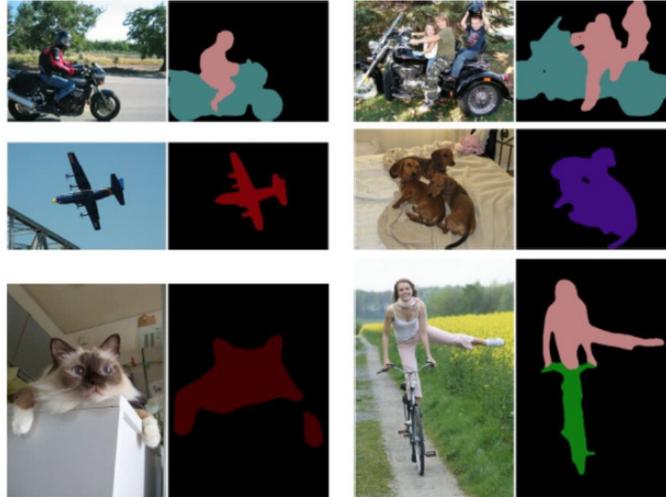


Figura 2.7: Ejemplos de imágenes segmentadas. *Adaptada de Chen et al. (2017) [26].*

Los métodos más básicos para segmentar imágenes son la detección de bordes y la umbralización. La detección de bordes se basa en encontrar cambios abruptos entre píxeles, mientras que la umbralización se basa en encontrar píxeles parecidos entre sí. Este mismo principio de similitud se aplica en métodos más complejos como la teoría de grafos, los contornos activos o el agrupamiento (mejor conocido como *clustering*) [27], los cuales se siguen aplicando y optimizando a pesar de que fueron propuestos hace mucho tiempo. Actualmente, los métodos que más se utilizan para segmentar imágenes están basados en aprendizaje profundo, tales como las redes neuronales convolucionales, las redes neuronales recurrentes, los mecanismos de atención, entre otros [28].

2.6 Mecanismos de atención

La segmentación de imágenes se utiliza en el área de la conducción autónoma para la detección de objetos o peatones [29]. En esta aplicación no solo importan los resultados de la segmentación, también se busca que el proceso de segmentación se realice con baja latencia y con un costo computacional reducido, ya que se tiene la necesidad crucial de tomar decisiones en intervalos muy precisos [30]. Esta es la razón por la cual se siguen optimizando algunos métodos clásicos y se siguen buscando métodos nuevos que realicen esta tarea de forma eficiente.

2.6. Mecanismos de atención

La inteligencia artificial tiene como objetivo desarrollar sistemas que realicen tareas que normalmente son hechas por humanos; incluso, muchas veces estos sistemas están inspirados también en el funcionamiento del cuerpo humano o de sus procesos. El ejemplo más claro son las redes neuronales, las cuales están inspiradas en el funcionamiento del cerebro y pueden realizar tareas como el aprendizaje, la percepción y la toma de decisiones. En su concepto general, una red neuronal está formada por varias capas de neuronas, las cuales están conectadas entre sí. El modelo recibe datos como entrada y en cada capa de neuronas se extrae información importante de ellos que se va transmitiendo capa por capa hasta llegar al final para hacer una predicción basada en la información extraída.

Las redes neuronales han ido evolucionando y actualmente existen muchas variantes dependiendo de su arquitectura o de la tarea para la que fueron diseñadas. Con el tiempo, también se han incluido nuevos conceptos que ayuden a optimizar su funcionamiento, tal es el caso de los mecanismos de atención, los cuales están inspirados en la forma en la que un humano fija su atención en algo. La idea es, que la atención de un humano no es “homogénea”, es decir, no pone atención en todo al mismo tiempo, sino que identifica los elementos más importantes y se enfoca principalmente en ellos. Este principio es una forma de optimizar el proceso cognitivo para el humano, por lo tanto, se consideró que también podía tener un impacto muy

2.6 Mecanismos de atención

favorable en el procesamiento de la información si se aplicaba en las redes neuronales.

La aplicación de un método de atención que fuera similar al humano se propuso en el artículo “*Neural Machine Translation by Jointly Learning to Align and Translate*” de Bahdanau *et. al* [31] como un nuevo enfoque para mejorar la traducción de texto. En este artículo aún no se utilizaba el concepto de “atención”, pero la idea corresponde a la técnica que posteriormente sería nombrada así. Bahdanau *et. al* argumentaron que la forma de los modelos clásicos de traducción, basados en un codificador y un decodificador independientes, generaban un cuello de botella al querer optimizar la arquitectura del modelo. Su propuesta fue implementar un modelo que encontrara de forma automática las partes de mayor relevancia en una oración para realizar las predicciones sin tener que unir estas partes como un segmento completo.

Los mecanismos de atención se usaron mucho tiempo como complemento de las redes neuronales recurrentes para optimizar el proceso de traducción, y fue hasta el año 2017 cuando Vaswani *et. al* formalizaron el concepto de *mecanismos de atención* en el artículo “*Attention is all you need*” [32]. En este artículo se introdujo también el concepto de *transformadores*, proponiendo un modelo basado por completo en mecanismos de atención, prescindiendo del uso de redes neuronales. La arquitectura propuesta para el modelo de transformadores se muestra en la Figura 2.8.

En el diagrama de la Figura 2.8, las representaciones de entrada hacen referencia a la asignación de un vector numérico a cada una de las palabras que se quieren traducir. Estos vectores entran al codificador y pasan por varias capas en las que son transformados por medio de operaciones matemáticas para entender la relación entre las palabras. Al salir del codificador, entran al decodificador junto con los vectores de salida de los pasos previos y ambos pasan por otras capas que ayudan a entender el contexto de la entrada. Este proceso se realiza varias veces dependiendo de la configuración del modelo, por eso en el diagrama se observa un N_x , que se refiere a la cantidad de veces que se repite el proceso. Al final, con cada vector se genera un valor

2.6 Mecanismos de atención

de probabilidad que indica a qué palabra del nuevo idioma se parece y se transforma el vector en esta palabra para realizar la traducción.

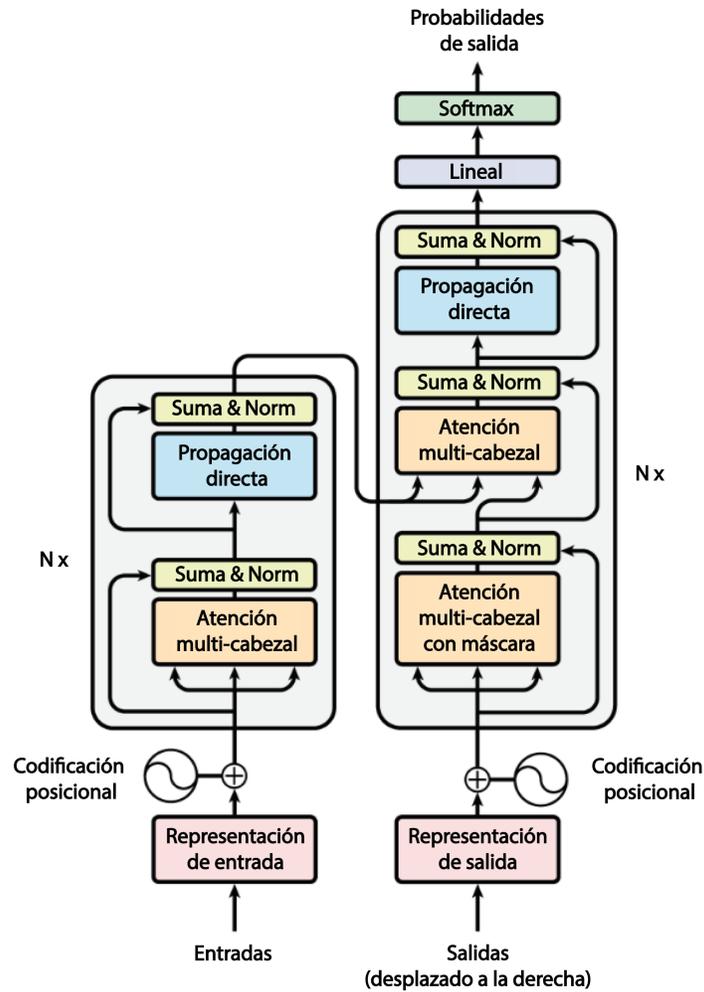


Figura 2.8: Diagrama de la arquitectura del modelo de transformadores para lenguaje natural. *Adaptado de Vaswani et al. (2017) [32].*

2.7. Transformadores de visión

Los transformadores de visión (ViT, por su nombre en inglés) son una arquitectura propuesta en el 2021 por Dosovitskiy *et. al* en el artículo titulado “*An image is worth 16x16 words: transformers for image recognition at scale*” y está basada en la arquitectura de transformadores usada en el lenguaje natural. El diagrama de la arquitectura de los ViT se muestra en la Figura 2.9

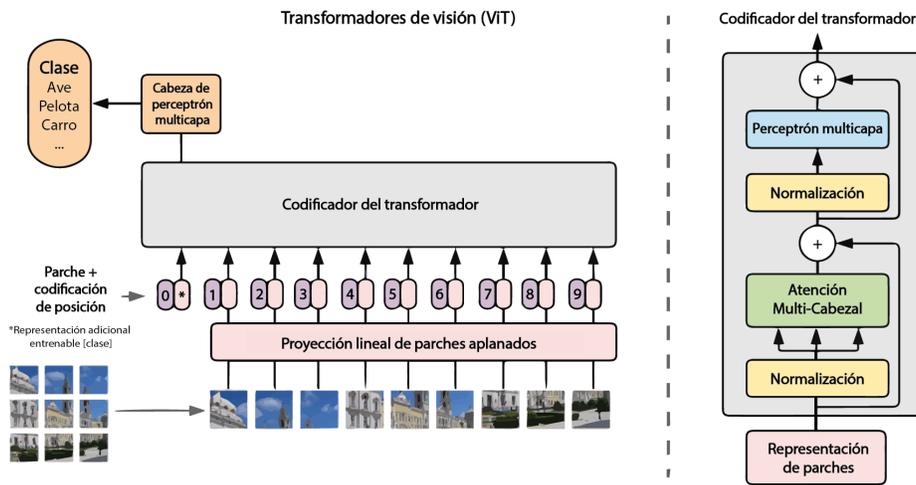


Figura 2.9: Diagrama del modelo de transformadores de visión (ViT).

Adaptado de Dosovitskiy et al. (2021) [33].

Los ViT funcionan de forma similar a la descripción que se hizo en el último párrafo de la sección anterior sobre la traducción de texto, solo que aplicado a imágenes. En este caso, lo equivalente a dividir un texto en palabras es dividir una imagen en parches y, de la misma forma, a cada parche de la imagen se le asigna un vector numérico que se va transformando en las capas del codificador. Cada vector contiene las características de su respectivo parche y, además, se agrega un vector extra en la posición cero de la proyección lineal de los parches. Este vector extra se va modificando en dependencia de los demás, de manera que contiene las características globales de la imagen. Por último, a la salida del codificador se agrega una

2.8 Redes neuronales convolucionales espacio-temporales

cabeza de perceptrón multicapa (MLP, por sus siglas en inglés), la cual puede funcionar como un transformador lineal para combinar las características extraídas, además de realizar la tarea final de la clasificación.

Una de las ventajas que tienen los modelos de ViT sobre las redes neuronales convolucionales (CNN, por sus siglas en inglés), es que estos pueden capturar mejor las características globales de las imágenes, ya que todos los parches están conectados entre sí para encontrar sus similitudes. Sin embargo, esta misma conexión entre todos los parches también puede generar una desventaja, ya que algunos modelos pueden ser más costosos computacionalmente en comparación con una CNN. Otra característica de los modelos ViT, es que se necesitan grandes cantidades de datos para su entrenamiento, lo cual puede parecer otra desventaja en comparación con una CNN; sin embargo, una vez que el modelo de ViT está entrenado puede llegar a dar mejores resultados que una CNN en tareas complejas de visión por computadora [34].

Los modelos de ViT pueden utilizarse en el área de la conducción autónoma para tareas como detección de objetos, detección de carriles, segmentación semántica en 2D [35], segmentación semántica en 3D con nubes de puntos [36], predicción de trayectorias y comportamientos [37]. Incluso también pueden ser usado para clasificar situaciones de riesgo al ir conduciendo y saber si la conducción va normal, si hay probabilidad de que ocurra un accidente, si el accidente está ocurriendo o si ya sucedió [38]. En el caso de este trabajo, la forma en que se usa el modelo ViT es para realizar una segmentación semántica que permita detectar peatones.

2.8. Redes neuronales convolucionales espacio-temporales

En los módulos de prevención de colisiones de los vehículos autónomos, la primera tarea es detectar a los objetos o personas con los que el vehículo

2.9 Protocolos de transmisión de video

puede colisionar. Después, en el caso de los peatones y los objetos móviles, también se debe conocer su trayectoria para poder anticiparse a la situación y avisar al vehículo si existe un riesgo de colisión.

Conocer la trayectoria de peatones u objetos móviles como los vehículos se puede realizar de diferentes formas y desde distintos enfoques. Se pueden usar algoritmos matemáticos como el filtro de Kalman [39], redes neuronales espacio-temporales basadas en grafos [40] (ST-GNN, por sus siglas en inglés), redes neuronales convolucionales espacio-temporales (ST-CNN, por sus siglas en inglés), entre otros.

La razón por la que se usan las ST-CNN en este tipo de aplicaciones, es porque los métodos convencionales, e incluso algunas técnicas de aprendizaje profundo, procesan imágenes de forma independiente sin tomar en cuenta la relación que tienen estas con las imágenes previas. Las ST-CNN combinan redes neuronales convolucionales 3D y 2D, lo cual no solo les permite extraer características en espacio, sino también en tiempo. Esto permite realizar tareas que involucren movimiento, como contar personas en una multitud [41] o en el reconocimiento de acciones o actividades humanas [42].

Si se considera el funcionamiento y las aplicaciones que tienen las ST-CNN, se puede concluir que en el área de conducción autónoma estas pueden ser de mucha ayuda para procesar videos de peatones u objetos móviles interactuando con vehículos autónomos. De esta forma, al tomar en cuenta el espacio y el tiempo, el modelo de la red puede aprender, por ejemplo, cómo es el movimiento de los peatones y si este representa un riesgo para ellos en un ambiente urbano, que es la tarea que se propone en este trabajo.

2.9. Protocolos de transmisión de video

En la transmisión de video en redes modernas se destacan varios protocolos especializados. Algunos de ellos son el Protocolo de Transmisión en Tiempo Real (RTSP, por sus siglas en inglés), la Transmisión en Vivo por

2.9 Protocolos de transmisión de video

Protocolo de Transferencia de Hipertexto (HLS por sus siglas en inglés) y la Transmisión Adaptativa dinámica sobre el Protocolo de Transferencia de Hipertexto (mejor conocida como MPEG-DASH).

Los protocolos mencionados previamente, tienen como diferencia que el protocolo RTSP permite controlar sesiones de transmisión en vivo, mientras que HLS y MPEG-DASH fragmentan el contenido en segmentos pequeños, permitiendo una transmisión adaptativa a través del Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés). Sin embargo, los tres protocolos tienen en común que funcionan sobre la base del Protocolo de Control de Transmisión/Protocolo de Internet, mejor conocido como TCP/IP, por sus siglas en inglés.

En este contexto, TCP/IP se consolida como el protocolo base por sus ventajas clave en aplicaciones cliente-servidor de procesamiento de video. Este asegura la entrega fiable de paquetes mediante un sistema de confirmación, previniendo la pérdida de información durante la transmisión, y reorganizando los paquetes que llegan en desorden. Además, su amplia compatibilidad con dispositivos y redes lo convierte en una opción versátil.

Otro protocolo que se construye sobre TCP/IP es WebSocket, un protocolo muy usado en una gran variedad de aplicaciones que van desde algo trivial como aplicaciones de mensajería o juegos en línea, hasta la conexión de sistemas que funcionan bajo el concepto de Internet de las Cosas (IoT, por sus siglas en inglés) [43].

La ventaja del protocolo WebSocket, es que ofrece control de congestión para mantener la estabilidad en condiciones de red variables, facilitando una comunicación bidireccional confiable. Esto es particularmente útil en sistemas donde el servidor procesa y devuelve datos al cliente, y en transmisiones de video en tiempo real, donde las sesiones persistentes son esenciales.

Con los conceptos abordados en este capítulo, se puede dar una idea preliminar de lo que se realizó en este trabajo. Como sensor, se usó una cámara de visión estereoscópica, con la cual se capturaron imágenes RGB e

2.9 Protocolos de transmisión de video

imágenes de profundidad por separado. Después, se realizó una segmentación semántica para detectar peatones en un ambiente urbano controlado a partir de las imágenes RGB usando un modelo ViT. Posteriormente, se usó una ST-CNN para estimar si existía algún riesgo de atropellamiento considerando a los peatones detectados en la segmentación y la información de su distancia, proporcionada por las imágenes de profundidad. La captura de las imágenes y el procesamiento de las mismas (segmentación y estimación de riesgos) se realizaron en módulos separados, por lo tanto, se utilizó el protocolo TCP/IP para enviar y recibir las imágenes entre ambos módulos.

Una vez explicados todos los conceptos y teniendo una idea de lo que se realizó, en el siguiente capítulo se describe con mayor detalle toda la metodología de este trabajo.

Capítulo 3

Metodología

En este capítulo se describe el proceso y los recursos utilizados en la implementación del módulo de visión para la detección y estimación de trayectoria de peatones, descrito en el capítulo uno. Se presentan los detalles de cada parte de la metodología realizada, así como la configuración de las técnicas usadas con nuestras bases de datos capturadas y en general todos los aspectos que se consideraron para la implementación. El diagrama que describe la metodología de manera general se muestra en la Figura 3.1.

En el diagrama general se pueden observar tres bloques principales: la etapa de captura, la etapa de procesamiento y la etapa de visualización. Los elementos que están involucrados en la etapa de captura y visualización conforman el sistema de visión y están ubicados físicamente en el prototipo de pruebas, el cual se describe más adelante. Por otro lado, la computadora con la que se realiza la etapa de procesamiento se encuentra físicamente fuera del prototipo y es donde se llevan a cabo las tareas de segmentación, detección de peatones y estimación de riesgos.

La cámara utilizada captura imágenes RGB y de profundidad, las cuales son recibidas y codificadas en la unidad de captura. Esta establece una conexión con la computadora a través de un protocolo de comunicación

Metodología

TCP/IP y envía un arreglo de 20 de imágenes de tamaño cada segundo (10 RGB y 10 de profundidad) con un tamaño de 960x600 píxeles.

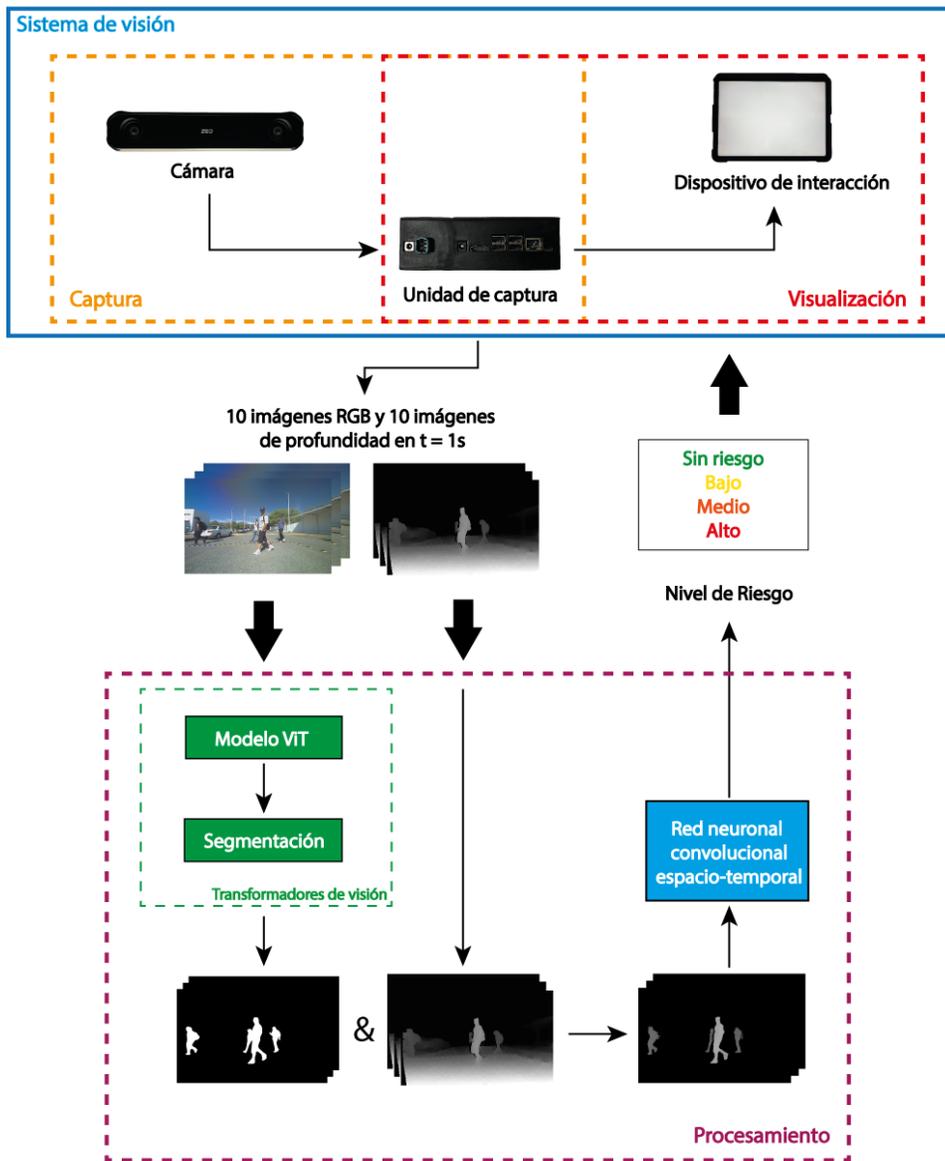


Figura 3.1: Diagrama General de la metodología implementada.

3.1 Sistema de visión

La computadora recibe las imágenes RGB y las procesa con un modelo de transformadores de visión (ViT) para obtener máscaras de segmentación que muestren las siluetas de los peatones. Las máscaras obtenidas se multiplican por sus respectivas imágenes de profundidad; de esta manera, se selecciona solo la información de profundidad que corresponde a los peatones. Este proceso se muestra en el cuadro punteado inferior del diagrama general.

Las imágenes de profundidad segmentadas son la entrada a una red neuronal convolucional espacio-temporal (ST-CNN) previamente entrenada. Esta ayuda a estimar la trayectoria de los peatones y determina si existe un riesgo de colisión entre el carro y los peatones. Si no existe riesgo, lo indica, y si existe riesgo, lo clasifica en uno de tres niveles: bajo, medio y alto.

Al final, el nivel de riesgo determinado es codificado y enviado usando el mismo protocolo de red a la unidad de captura, esta lo recibe, lo decodifica y muestra los resultados en el dispositivo de interacción. Además, pensando en que el módulo sea adaptable a otros sistemas (por ejemplo, sistemas de acción y/o decisión), la unidad de captura también proporciona una salida de voltaje que indica el nivel de riesgo, de esta forma, los sistemas pueden interactuar con una señal que es fácil de procesar.

El dispositivo de interacción funciona como un monitor de la unidad de captura y se controla a través de un protocolo de Computación Virtual en Red (VNC, por sus siglas en inglés), por lo tanto, permite iniciar, finalizar y monitorear todo el proceso de forma visual desde el prototipo de pruebas.

3.1. Sistema de visión

El sistema de visión tiene cuatro componentes principales: un sensor, una tarjeta de captura, una tarjeta de desarrollo y un dispositivo de interacción, los cuales están conectados como se muestra en la Figura 3.2.

El sensor es una cámara estéreo con una línea base de 12 cm y un rango

3.1 Sistema de visión



Figura 3.2: Componentes del sistema de visión.

de profundidad real de 1 a 20 m. Tiene una distancia focal de 2.2 mm, un campo de visión de 110° en horizontal x 80° en vertical x 120° en diagonal y una apertura de la lente $f/2.2$. Permite capturar hasta 120 fotografías por segundo (fps) en resolución 600p (960x600 píxeles) y 60 fps en resoluciones de 1080p (1920x1080 píxeles) y 1200p (1920x1200 píxeles). Por su composición, está diseñada para aplicaciones en el exterior y cuenta con un puerto de conexión serial coaxial GMSL2, lo cual ayuda a transmitir video sin interferencias electromagnéticas a una tasa de transmisión de datos alta y con baja latencia [44].

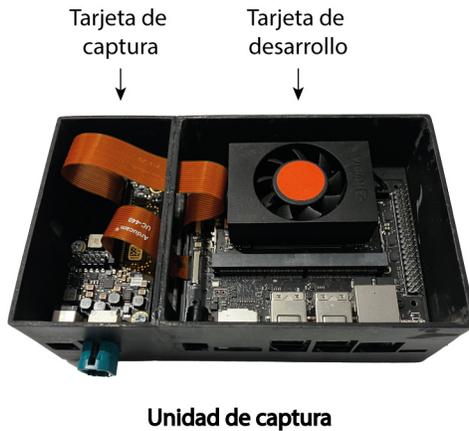


Figura 3.3: Vista interior de la unidad de captura.

La cámara se conectó por medio del puerto GMSL2 a la tarjeta de cap-

3.1 Sistema de visión

tura, la cual se ubica dentro de la unidad de captura, como se muestra en la imagen 3.3. Esta cuenta con un deserializador MAX96712, el cual se encarga de recibir y convertir los datos obtenidos por el sensor de la cámara a un formato que permita procesarlos en las etapas posteriores [45]. Los datos convertidos se transmiten a la tarjeta de desarrollo (que también está dentro de la unidad de captura) a través de un cable flexible de interfaz serial para cámaras (CSI, por sus siglas en inglés). La tarjeta de desarrollo tiene un procesador NVIDIA OrinTM, una memoria RAM de 8 GB, un disco duro de 1 TB y una unidad de procesamiento gráfico de 16 GB.

Ambas tarjetas (de captura y desarrollo) tienen componentes electrónicos expuestos, por lo tanto, se consideró que era necesario protegerlos para evitar daños. Con el uso de *Autodesk Inventor* [46], un software de diseño asistido por computadora, se diseñó una caja de protección que permitiera introducir y extraer los dispositivos fácilmente y que tuviera una ventilación adecuada para transportar y usar las tarjetas de forma segura.

El proceso de captura de las imágenes se monitoreó a través de un dispositivo de interacción, en este caso, una tableta digital. Esta se conectó a la tarjeta de desarrollo a través de un protocolo de Computación Virtual en Red (VNC, por sus siglas en inglés). Debido a la configuración de la tarjeta de desarrollo, no era posible interactuar con el sistema operativo sin la conexión física de un monitor en el puerto de pantalla. En sustitución de este, se conectó al puerto de pantalla un dispositivo para simular las funcionalidades de un monitor conectado físicamente y así poder interactuar con el sistema operativo desde la tableta digital.

Antes de probar el sistema de visión, se aseguró que todos sus componentes fueran adecuados para la aplicación. La cámara está diseñada para uso en exteriores y las tarjetas de la unidad de captura proporcionan la capacidad y velocidad suficiente para el manejo de los datos. El dispositivo de interacción facilita el monitoreo del proceso y en conjunto el sistema es fácil de montar y desmontar en la plataforma de pruebas, la cual se describe a continuación.

3.2. Plataforma de pruebas

Uno de los objetivos de este trabajo es acercarse lo más posible a las condiciones reales de un vehículo en un ambiente de exterior. La División de Ingenierías del Campus Irapuato-Salamanca de la Universidad de Guanajuato cuenta con un prototipo de vehículo eléctrico biplaza, el cual se solicitó para usarlo como plataforma de pruebas e instalar en él el sistema de visión descrito previamente.

Las dimensiones del prototipo son 1.6 m de ancho, 2.8 m de largo y 1.5 m de altura. Tiene una autonomía de 90 km en condiciones normales de manejo y puede alcanzar una velocidad de hasta 75 km/h. Como fuente de alimentación tiene una batería de litio con una capacidad de 150 amperes por hora (Ah) que proporciona un voltaje de 48 volts (V) de corriente directa. Este voltaje entra a un inversor de corriente para convertir la corriente directa (CD) en corriente alterna (CA) y así alimentar el motor trifásico de 60 V con el que opera el prototipo.

Como se mencionó en la introducción, los vehículos autónomos eléctricos tienen la ventaja de poder conectar varios sistemas a una misma fuente de alimentación. En este caso, además de alimentar el motor, la batería también se usó para alimentar la unidad de captura del sistema de visión. Esta requería una fuente de alimentación de entre 12 y 19 V de CD, por lo tanto, fue necesario conectar un reductor de voltaje a la salida de la batería del prototipo. El dispositivo reductor puede operar con un voltaje de entrada de 30 a 60 V de CD y proporcionar a la salida 12 V de CD con una corriente máxima de 30 amperios (A) y una potencia máxima de 360 vatios (W). Además, cuenta con protección contra sobrecorriente, cortocircuito, baja tensión y sobrecalentamiento. El diagrama de conexión entre la alimentación del prototipo y la alimentación del sistema de visión se muestran en la Figura 3.4.

En cuanto a la ubicación del sistema de visión en el prototipo de pruebas, la cámara se situó al frente del prototipo a una altura de 85 cm con respecto

3.3 Base de datos

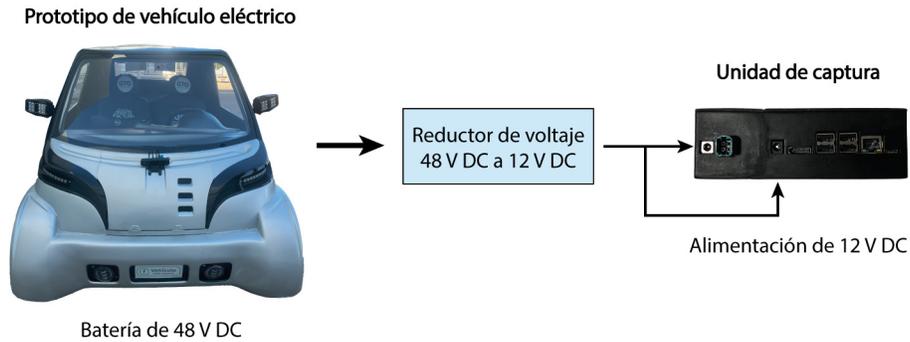


Figura 3.4: Diagrama de alimentación de la unidad de captura desde el prototipo de vehículo eléctrico.

del piso. La unidad de captura se colocó dentro de un pequeño compartimento en el frente del prototipo y el dispositivo de interacción se ubicó en la cabina a la vista del conductor para iniciar y monitorear los procesos que se llevan a cabo en la unidad de captura. Con esta configuración se realizaron las pruebas y se obtuvo la base de datos para entrenar los modelos descritos más adelante.

3.3. Base de datos

Las imágenes que componen la base de datos son de tres tipos: de color, de profundidad y máscaras de segmentación. Las de color y profundidad fueron capturadas con la cámara del sistema de visión, mientras que las máscaras de segmentación fueron obtenidas a partir de las imágenes de color.

3.3.1. Imágenes de color y profundidad

La captura de las imágenes de color y profundidad se realizó por medio de la cámara usando las herramientas del Kit de Desarrollo de Software (SDK, por sus siglas en inglés) que proporciona el fabricante y que permite

3.4 Segmentación y detección de peatones

y vehículos, algo similar a un ambiente urbano; sin embargo, se consideró como un ambiente controlado, pues el tráfico es menor y se conduce a velocidades bajas. Durante las capturas, el prototipo se manejó en un rango de velocidad de 0 a 20 km/h con un modo de conducción normal de un vehículo.

3.3.2. Máscaras de segmentación de entrenamiento

El entrenamiento de un modelo de segmentación, en este caso el modelo ViT, requiere de máscaras binarias de referencia que indiquen las áreas de la imagen que corresponden a los peatones. Estas máscaras se obtuvieron utilizando *PixelLib* [48], una biblioteca para segmentación de imágenes y videos. Esta segmenta más de 80 clases de objetos, pero separa cada uno de los objetos segmentados en máscaras individuales, así que solo se seleccionaron las máscaras que correspondían a peatones.

Algunas de las máscaras resultantes tenían varios errores, por ejemplo, áreas marcadas como peatones que en realidad no lo eran o algunas que no eran marcadas como tal y sí pertenecían a los peatones. Estos detalles se corrigieron en las imágenes usando el software *Adobe Photoshop* [49], con el que se puede trabajar a nivel de píxeles y hace más fácil quitar o corregir las áreas erróneas.

3.4. Segmentación y detección de peatones

La técnica elegida para realizar la segmentación de peatones fue el modelo de transformadores de visión (ViT, por sus siglas en inglés). Se implementó un modelo ViT básico que se construyó a partir del diagrama de la Figura 2.9; sin embargo, se realizaron algunos cambios, ya que el modelo ViT de ese diagrama está diseñado para clasificar imágenes, no para segmentar. Con las modificaciones realizadas, el diagrama que describe el modelo ViT usado en este trabajo se muestra en la Figura 3.6.

3.4 Segmentación y detección de peatones

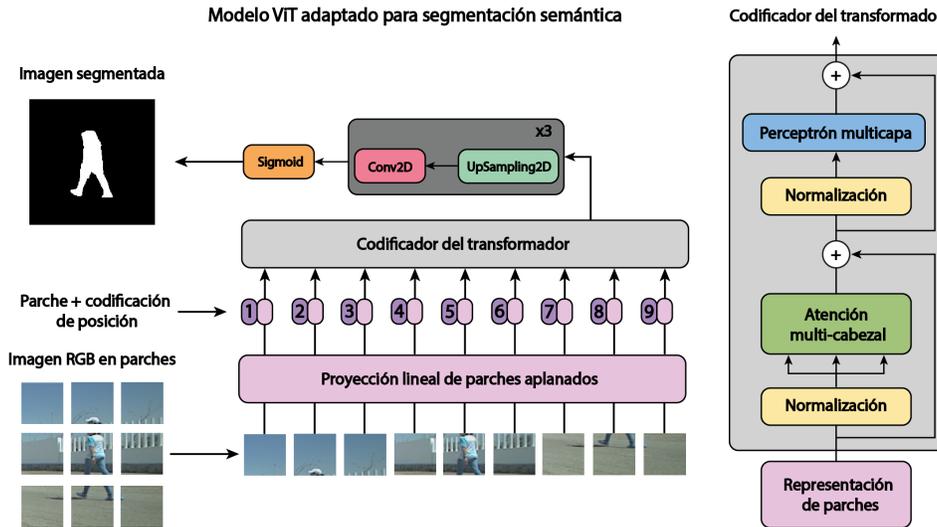


Figura 3.6: Diagrama del modelo adaptado para salida de imágenes segmentadas. *Adaptado de Dosovitskiy et al. (2021) [33].*

El primer cambio que se realizó, es que se quitó la incrustación adicional que estaba al inicio en la proyección lineal de los parches. Esta contenía la información de las características globales de la imagen y se consideró que no era necesario mantenerla, ya que en una segmentación se hace una clasificación píxel a píxel y no de toda la imagen. Otra modificación fueron las capas de salida, ya que el modelo original tenía una cabeza de perceptrón multicapa a la salida del codificador y ésta tuvo que ser cambiada para realizar la segmentación de imágenes. En su lugar se colocaron tres capas *UpSampling2D* alternadas con tres capas *Conv2D* a la salida del decodificador como se muestra en la Figura 3.6, además de una capa de activación *Sigmoid*, y que la clasificación de píxeles es binaria.

3.4.1. Entrenamiento del modelo ViT

Como se observa en el diagrama de la Figura 3.6, el proceso del modelo ViT consiste en generar parches de la imagen y realizar una proyección lineal de los mismos. Si se conservaba el tamaño de los parches del modelo

3.4 Segmentación y detección de peatones

ViT original (16 x 16 píxeles) y se tomaba como entrada la imagen con sus dimensiones originales (600 x 960 píxeles), la proyección lineal que se iba a generar era demasiado grande. Esto significaba un aumento en el costo computacional, tanto en el uso de memoria como en el tiempo de procesamiento. Además, aumentaba el riesgo de que el modelo se sobreentrenara, pues al ser una imagen tan grande y parches tan pequeños, el modelo puede aprender muchos detalles innecesarios. Esta situación tenía dos soluciones: seccionar la imagen en partes más pequeñas o reducir el tamaño de la imagen original.

El método de seccionar la imagen implica recortar la imagen en varias partes y, dependiendo del tamaño de los recortes, se podría requerir adicionalmente un redimensionamiento de cada recorte. Todos los recortes se tomarían como un lote para hacer la predicción con el modelo ViT y, una vez que cada recorte estuviera segmentado, se debían volver a unir para formar la imagen original, cuidando que cada recorte quedara en su lugar correspondiente para no distorsionar la imagen. Tomando en cuenta que el módulo debía procesar varias imágenes por segundo, se consideró que esta opción podía ser demasiado tardada y que existía el riesgo de cometer errores en el acomodo de los recortes.

Por otro lado, el método de redimensionamiento solo requería recibir la imagen y reducir su resolución, de esta forma, podía entrar directamente a la predicción. Con este método, se hizo un solo lote que contenía las 10 imágenes por segundo y se esperaba que eso redujera significativamente el tiempo de procesamiento a diferencia del otro método, con el que se generaba un lote de tamaño 10 multiplicado por la cantidad de recortes obtenidos en cada imagen. Finalmente, se llegó a la conclusión de que el método de redimensionamiento era el más adecuado para entrenar el modelo y se decidió que las imágenes fueran de 224x224 píxeles.

3.5. Estimación de niveles de riesgo de colisión

La estimación de los niveles de riesgo de colisión se realizó con una red neuronal convolucional espacio-temporal (ST-CNN, por sus siglas en inglés), cuya arquitectura se muestra en la Figura 3.7

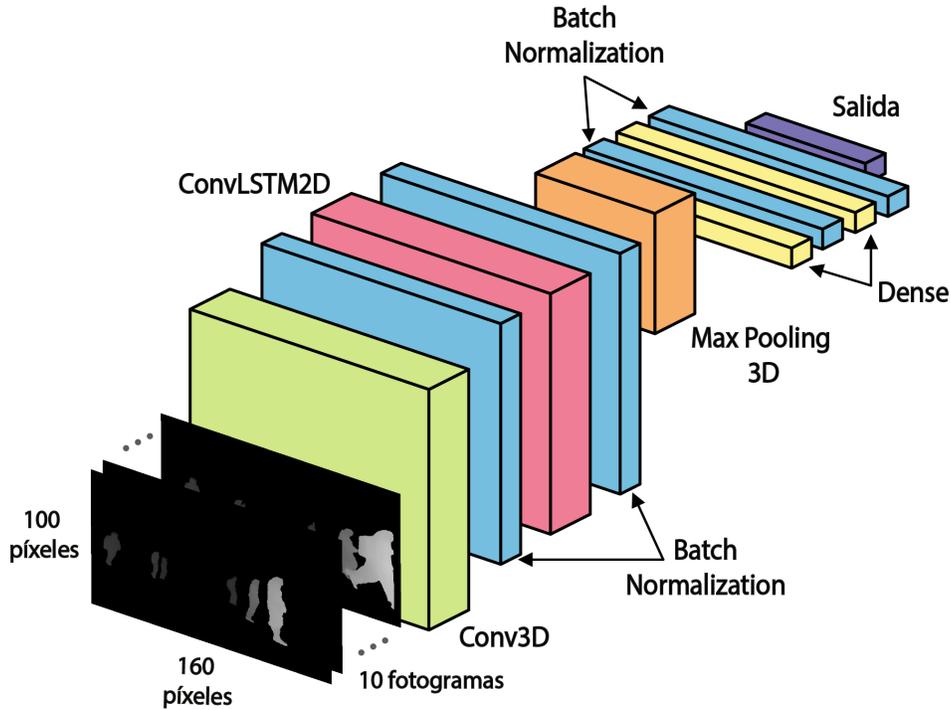


Figura 3.7: Arquitectura de la ST-CNN para la estimación de niveles de riesgo *Adaptado de Andrade et al. (2022) [42]*.

Las imágenes de entrada en la ST-CNN son el resultado de la multiplicación de las máscaras de segmentación por las imágenes de profundidad. La idea de usar estas imágenes para el modelo, es porque una ST-CNN puede extraer características que involucran el movimiento de los peatones. Como el módulo está centrado específicamente en prevenir colisiones con peatones, en la etapa de segmentación se eliminan todos los demás objetos que están en el entorno y se conservan únicamente las siluetas de los peatones. Al multiplicar las máscaras obtenidas por sus respectivas imágenes de profundidad, en las imágenes se quedan solamente las áreas de interés (es decir, los

3.6 Comunicación entre la unidad de captura y el módulo de procesamiento

peatones) con sus datos de profundidad. Esto indica que el modelo ST-CNN va a extraer características que estarán relacionadas directamente con la información de la distancia a la que se encuentran los peatones y así estimar mejor el nivel de riesgo.

Antes de entrar al modelo, las imágenes fueron redimensionadas, quedando de un tamaño de 100x160 píxeles con 3 canales cada una para disminuir el tiempo de procesamiento y el costo computacional. El lote de entrada es de 10 imágenes, que son las 10 imágenes por segundo que se reciben desde el módulo de captura y que ya pasaron previamente por el modelo ViT para realizar la segmentación.

La extracción de características es realizada por una capa convolucional de tres dimensiones y una capa convolucional de dos dimensiones de Memoria a Largo y Corto Plazo (LSTM, por sus siglas en inglés). Estas dos capas juntas pueden extraer características temporales a corto y largo plazo y al mismo tiempo extraer características espaciales, por eso es que el modelo recibe el nombre de red neuronal convolucional espacio-temporal.

Una vez que se extrajeron las características, con la capa *MaxPooling3D* se reducen las dimensiones de los datos y se mantiene solo la información relevante. Después, las características son clasificadas por dos capas totalmente conectadas (*Dense*), ambas con una función de activación ReLU. Al final, para la salida, se incorpora otra capa totalmente conectada con activación *softmax* que contiene 4 salidas, donde cada una corresponde a un nivel de riesgo: sin riesgo, riesgo bajo, riesgo medio y riesgo alto.

3.6. Comunicación entre la unidad de captura y el módulo de procesamiento

La comunicación entre el módulo de captura y el módulo de procesamiento para la transferencia de datos se estableció usando el protocolo WebSoc-

3.6 Comunicación entre la unidad de captura y el módulo de procesamiento

El cual tiene como base el protocolo TCP/IP. Como se mencionó en el capítulo anterior, este tipo de conexión permite enviar y recibir datos entre servidor y cliente en tiempo real, lo cual lo convierte en una muy buena opción para ser usado en aplicaciones del área de la conducción autónoma. La transferencia de datos entre los dos módulos se representa en el diagrama de la Figura 3.8.

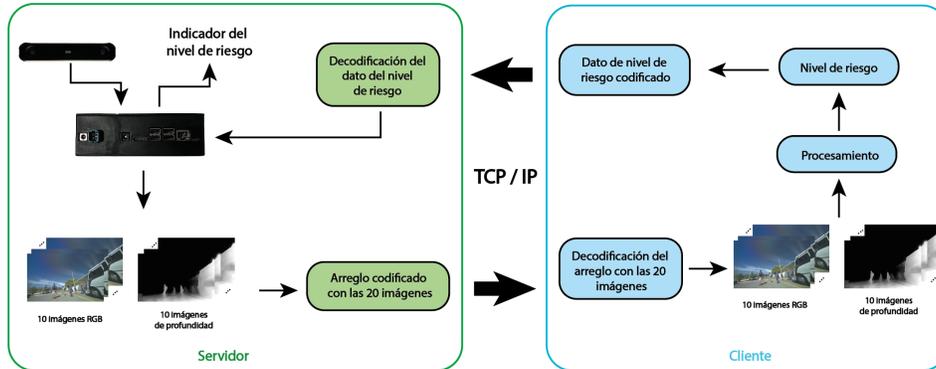


Figura 3.8: Diagrama de la transferencia de datos entre la unidad de captura (servidor) y el módulo de procesamiento (cliente) usando el protocolo TCP/IP.

La cámara captura imágenes a 30 fotogramas por segundo (fps); sin embargo, se observó que en un segundo generalmente no había un gran cambio en la posición de los peatones, por lo tanto, se consideró que no era necesario enviar los 30 fotogramas para su procesamiento, ya que esto podía hacer más tardado el procesamiento de las imágenes. Tomando en cuenta esto, se decidió que solo se mandaran 10 fps, de manera que estas estuvieran distribuidos a lo largo de todo el segundo.

Cada segundo se forma un arreglo con 20 imágenes, 10 imágenes RGB con sus respectivas imágenes de profundidad. El arreglo con las imágenes debe ser codificado en bytes para poder enviarlo desde el cliente (unidad de captura) hacia el servidor (módulo de procesamiento). Una vez que el servidor recibió el arreglo con las imágenes, las decodifica a un formato *uint8*, redimensiona las imágenes RGB a un tamaño de 224x224 píxeles y envía el lote de las 10 imágenes al modelo ViT para la segmentación. Las máscaras

3.6 Comunicación entre la unidad de captura y el módulo de procesamiento

de salida se multiplican por las imágenes de profundidad y el mismo lote de 10 imágenes entra a las ST-CNN.

A la salida de la ST-CNN solo se obtiene un número entre 0 y 3 que indica el nivel de riesgo de que exista una colisión. Este valor se guarda como primer elemento de un arreglo y en el resto del arreglo se guarda la imagen RGB que contiene las cajas delimitadoras que se le asignaron en el proceso de detección de peatones. Una vez completo el arreglo, se codifica a bytes y se envía de regreso a la unidad de captura para mostrar el nivel de riesgo (si es que existe) usando un indicador conectado a una de las terminales de la tarjeta de desarrollo.

Siguiendo la estructura que tiene la metodología se pueden identificar las distintas etapas de prueba que se realizaron. En la primera etapa se comprobó que la configuración del sistema de visión fuera correcta y que todos sus elementos fueran adecuados para la aplicación. En la segunda etapa se realizó la adquisición de las imágenes que conforman la base de datos. Como tercera etapa, se entrenó y probó el modelo ViT usando las imágenes RGB y las máscaras de segmentación de entrenamiento, y en la cuarta etapa, se entrenó y probó la ST-CNN usando las imágenes de profundidad segmentadas.

Finalmente, se unieron los dos modelos entrenados (ViT y ST-CNN) siguiendo la secuencia descrita en la metodología y se realizaron las pruebas de todo el sistema, desde la captura de imágenes hasta la salida del nivel de riesgo entregada por la ST-CNN, enviando las imágenes de entrada y la señal de salida de un módulo a otro a través del protocolo TCP/IP. Con esto, se hicieron las evaluaciones del tiempo de procesamiento, para ver si se podía realizar todo el proceso en tiempo real. Los resultados obtenidos en cada una de las etapas de prueba se muestran en el siguiente capítulo.

Capítulo 4

Pruebas y Resultados

En este capítulo se describen las pruebas experimentales realizadas del sistema de visión para diferentes situaciones con peatones. Además se presentan los resultados obtenidos de acuerdo con la configuración del sistema de visión, los modelos de aprendizaje profundo y el funcionamiento general del sistema en la plataforma experimental.

4.1. Configuración del sistema de visión

Todos los elementos que componen el sistema de visión se utilizaron por primera vez en este trabajo, así que fue necesario configurar todo desde el inicio para adaptar el sistema a lo que se iba a requerir. Lo primero que se hizo fue configurar la tarjeta Nvidia Jetson Orin Nano que forma parte de la unidad de captura. Se le instaló el sistema operativo Ubuntu 20.04, el paquete de desarrollo para usar la cámara, Python 3.7 y los controladores y bibliotecas necesarias para capturar las imágenes con la cámara.

Como primer resultado, se generó un manual que indica paso a paso todo el proceso de instalación y que contiene los enlaces de descarga requeridos,

4.1 Configuración del sistema de visión

así como las versiones compatibles de las bibliotecas que se instalaron. Una vez instalado todo, se desarrolló un código en Python para capturar y guardar imágenes usando la cámara ZED X [44]. El código se puede ejecutar y modificar desde el entorno de desarrollo que se instaló en la tarjeta Nvidia; sin embargo, para hacer el proceso más práctico, se creó un ejecutable con una interfaz gráfica simple que indica si la cámara está conectada y da la opción de iniciar y detener la captura de imágenes sin necesidad de abrir un entorno de desarrollo para compilar el código. La interfaz se muestra en la Figura 4.1.



Figura 4.1: Ventanas de la interfaz para la captura de imágenes con la unidad de captura.

La intención de crear esta interfaz es que los próximos usuarios de la cámara puedan capturar imágenes de forma más simple y más rápida sin tener que entrar al código, a menos que requieran hacer alguna modificación. Si no es así, solo será necesario conectar la cámara, dar clic en el botón de inicio y posteriormente terminar el proceso de captura dando clic en el botón de término.

Como se mencionó en la metodología, fue necesario diseñar una caja

4.2 Base de datos

protectora para las tarjetas y el modelo diseñado se imprimió con una impresora 3D usando resina gris de uso general. El resultado de la impresión de la caja con las tarjetas en su interior se muestra en la Figura 4.2.



Figura 4.2: Caja de resina para las tarjetas del módulo de captura.

La caja fue diseñada para que ambas tarjetas pudieran entrar sin la necesidad de aplicar fuerza, pero a la vez se consideró que los espacios tuvieran la medida justa para evitar que las tarjetas se salieran. En el caso de la tarjeta de captura, sí fue necesario agregar en el diseño de la caja unos orificios para poder sujetarla con tornillos pequeños. Una vez que se protegieron las tarjetas, entonces se continuó con la generación de la base de datos.

4.2. Base de datos

Como se mencionó en la metodología, la cámara se configuró para capturar imágenes RGB e imágenes de profundidad a una tasa de 30 fotogramas por segundo (fps); sin embargo, por el tiempo que tomaba el proceso de guardado de las imágenes, solo se pudieron guardar 15 fps. Al poner en funcionamiento el módulo completo obtenido en este trabajo, esta situación ya no se presentó, ya que los fotogramas no se guardan como un archivo de imagen y el tiempo que toman las imágenes en guardarse ya no afecta. Sin embargo, sí fue necesario considerar esta información en el entrenamiento de la red que clasifica los niveles de riesgo.

4.2 Base de datos

Considerando la tasa de 15 imágenes guardadas por segundo, se obtuvieron en total cinco secuencias con 12248, 12831, 11251, 13216 y 13474 imágenes respectivamente, las cuales fueron tomadas en días y horarios distintos y corresponden a aproximadamente 15 minutos de grabación por secuencia. Cada una de estas está compuesta por el mismo número de imágenes RGB y de profundidad y contiene los cuatro casos posibles de interacción entre el prototipo y el peatón: prototipo y peatón en movimiento, prototipo en movimiento y peatón estático, prototipo estático y peatón en movimiento y prototipo y peatón estáticos. Algunos ejemplos se muestran en la Figura 4.3.



Caso 1. Vehículo y peatón, ambos en movimiento.



Caso 2. Vehículo estático y peatón en movimiento.



Caso 3. Vehículo y peatón, ambos estáticos.



Caso 4. Vehículo en movimiento y peatón estático.

Figura 4.3: Casos de interacción entre el vehículo y los peatones.

4.3. Entrenamiento del modelo de transformadores de visión

En el modelo ViT se usaron las cinco secuencias de la base de datos, que en total sumaron 63022 imágenes RGB y la misma cantidad para las máscaras de segmentación de entrenamiento. Algunas imágenes tenían peatones y otras no, entonces, usando las máscaras de segmentación de entrenamiento se separaron en una carpeta las imágenes con peatones y en otra las que no tenían peatones. En 17692 imágenes aparecían peatones, mientras que en el resto (45330 imágenes) no aparecían, lo cual indica que las imágenes sin peatones eran muchas más; por lo tanto, en el entrenamiento se consideraron solo imágenes con peatones. De estas, el 80 % se utilizó para el entrenamiento del modelo y el 20 % restante se usó para probar el mismo.

Una vez separadas, tanto las imágenes RGB como las máscaras de segmentación de entrenamiento, se redimensionaron y pasaron de tener una resolución de 600 x 960 píxeles a una de 224 x 224 píxeles y con esta medida se usaron para el entrenamiento del modelo de transformadores de visión descrito en la metodología. Los resultados del entrenamiento se muestran en las gráficas de precisión y pérdidas en las Figuras 4.4 y 4.5 respectivamente.

En la gráfica de precisión de la Figura 4.4 se puede observar que el modelo converge relativamente rápido, pues el primer valor de exactitud de validación comienza en aproximadamente 0.980 y alcanza su máximo cerca de 0.991. Al principio, el modelo se entrenó solo con 50 épocas; sin embargo, se observó que al terminar estas épocas, la gráfica de precisión obtenida seguía teniendo una tendencia a subir, por lo tanto, se consideraron 100 épocas para buscar que se estabilizara la gráfica al final. En cuanto a la gráfica de pérdidas de la Figura 4.5, se observa que el descenso de las pérdidas de entrenamiento sigue teniendo una tendencia hacia abajo, mientras que la gráfica que representa la validación parece estabilizarse ligeramente antes de llegar al 0.02.

Otra cosa que se observa, es que entre la época 20 y 40 hay un descenso

4.3 Entrenamiento del modelo de transformadores de visión

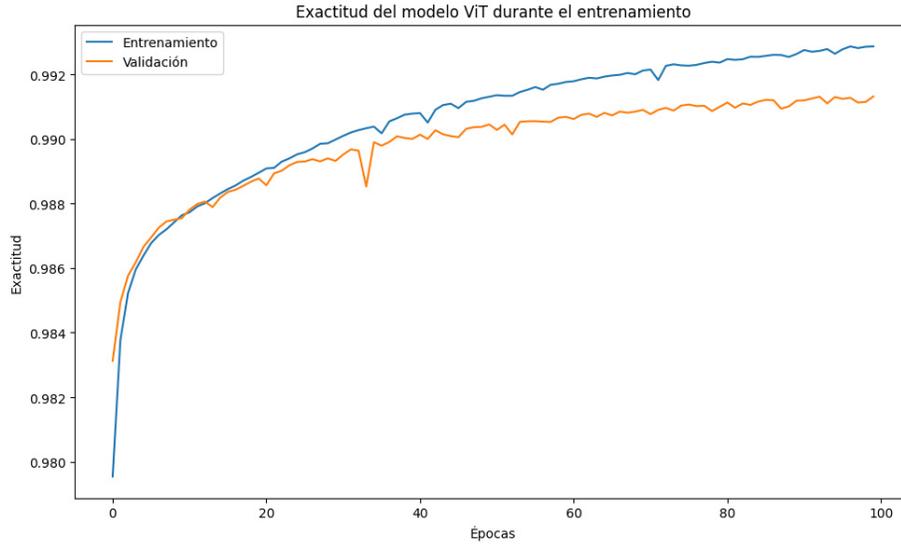


Figura 4.4: Gráfica de la precisión obtenida en el entrenamiento del modelo ViT durante 100 épocas.

en la gráfica que corresponde a la validación. Si se ubica el mismo punto en la gráfica de pérdidas, se observa que en ésta hay un ligero aumento, aunque no tan pronunciado como en la exactitud. En el momento, esto podría indicar que existe un sobreajuste, ya que la gráfica de entrenamiento continúa en ascenso; sin embargo, como la gráfica vuelve a su tendencia inmediatamente, pueden ser otros factores como un ajuste de pesos por parte del optimizador o incluso un posible ruido en los datos lo que cause ese descenso. Al final, como la exactitud de validación se recupera y sigue en aumento, se consideró que el modelo estaba aprendiendo correctamente y que esa caída no tuvo una influencia importante en su desempeño.

Una vez entrenado el modelo, se usaron las imágenes de prueba separadas al inicio (20% de las imágenes totales que corresponden a 3538 imágenes) para validar su funcionamiento haciendo la comparación entre las predicciones y las máscaras originales. De esa evaluación, se obtuvo la matriz de confusión que se muestra en la Figura 4.6 y las métricas que se muestran en la Tabla 4.1.

4.3 Entrenamiento del modelo de transformadores de visión

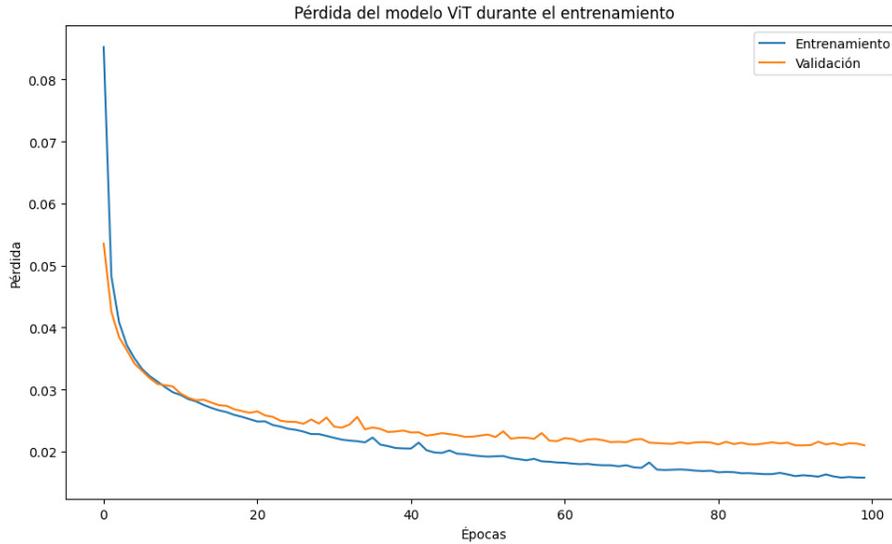


Figura 4.5: Gráfica de las pérdidas en el entrenamiento del modelo ViT durante 100 épocas.

La matriz de confusión contiene la información de los píxeles de las 3538 imágenes, e indica cuántos de estos se clasificaron correctamente y cuántos no. Lo primero que se observa, es que hay una casilla en color azul con un número muy grande, el cual corresponde a la cantidad de píxeles que no pertenecían a los peatones y fueron clasificados como tal. Si se suma la cantidad de píxeles que no pertenecían a los peatones y se compara con los que sí pertenecían, se encuentra que había 50 veces más píxeles que no correspondían a los peatones (píxeles negros) que píxeles que sí pertenecían a los peatones (píxeles blancos). Esta es la razón por la cual este número es tan alto en la matriz de confusión en comparación con los otros, porque la mayor parte de las máscaras eran píxeles en negro.

En cuanto a los píxeles que correspondían a los peatones, el 71 % de ellos se clasificaron correctamente, mientras que el resto se identificaron como que no eran parte de un peatón. De los píxeles que no pertenecían a los peatones, el 99.6 % fueron clasificados correctamente y solo el 0.4 % de estos no fueron clasificados correctamente; sin embargo, al ser tantos píxeles, ese 0.4 % no representa una cantidad considerable de ellos.

4.3 Entrenamiento del modelo de transformadores de visión

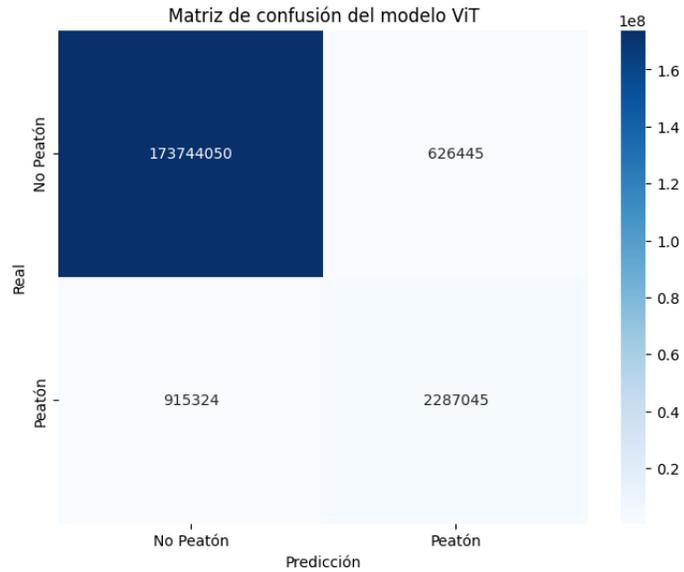


Figura 4.6: Matriz de confusión del modelo ViT.

La información de la matriz de confusión se puede complementar con las métricas de la Tabla 4.1. El porcentaje más alto en la exactitud se debe principalmente a lo que se comentó antes en la matriz de confusión, la cantidad de píxeles que no correspondían a los peatones era mucho mayor y en su mayoría se clasificaron correctamente. Esto se refleja en las métricas de precisión, sensibilidad y F1, las cuales fueron calculadas considerando la diferencia entre las instancias de cada clase.

Métrica	Resultado
Exactitud	0.991
Precisión	0.889
Sensibilidad (Recall)	0.855
Medida F1 (F1 Score)	0.871
IoU	0.597

Tabla 4.1: Métricas del modelo ViT.

4.3 Entrenamiento del modelo de transformadores de visión

Al final de la tabla se agregó una métrica que es comúnmente utilizada para evaluar los modelos de segmentación: la Intersección sobre Unión (IoU, por su nombre en inglés). Básicamente, lo que se hace para obtener esta métrica es sobreponer la máscara resultante de la predicción del modelo en la máscara real, y comparar las áreas que corresponden entre ellas. En este caso, se obtuvo un valor cercano al 0.6, siendo 1 el valor máximo posible, lo cual indica que, aunque sí hay un gran porcentaje de píxeles de personas clasificados correctamente, aunque se identifica que existen algunas deficiencias en el modelo. En este caso en particular, más allá del porcentaje obtenido, lo importante no es que las áreas coincidan por completo, sino que los peatones sean detectados, lo cual se puede analizar haciendo una comparación visual entre las imágenes reales y las predichas por el modelo.

En la Figura 4.7 se muestran algunos ejemplos de comparaciones entre las máscaras segmentadas originales y las predichas por el modelo ViT. La máscara de la izquierda representa la máscara original y la de la derecha representa la imagen obtenida con el modelo ViT entrenado. En los Ejemplos 1 y 2 se puede observar que al modelo le cuesta identificar a los peatones que están lejos y que, por lo tanto, sus siluetas son muy pequeñas. En los ejemplos 3 y 4, se observa que cuando los peatones están relativamente cerca los encuentra a todos, aunque le cuesta obtener una silueta más precisa, lo cual se refuerza en el ejemplo 5. Por último, el Ejemplo 6 representa a un peatón que está demasiado cerca y se observa que la máscara de predicción es prácticamente igual que la real.

De acuerdo con el objetivo que tiene este trabajo, hasta ahora se ha descrito la adquisición y procesamiento de las secuencias de imágenes capturadas para la segmentación de las regiones en la imagen que contienen peatones, basándonos en el modelo ViT. En la siguiente etapa lo más importante va a ser detectar a los peatones que están cerca del vehículo, ya que son los que representan un riesgo de colisión alto. En ese caso, los ejemplos 5 y 6 representan un punto muy positivo del modelo, porque es en estos casos en los que las predicciones dan mejores resultados.

4.3 Entrenamiento del modelo de transformadores de visión

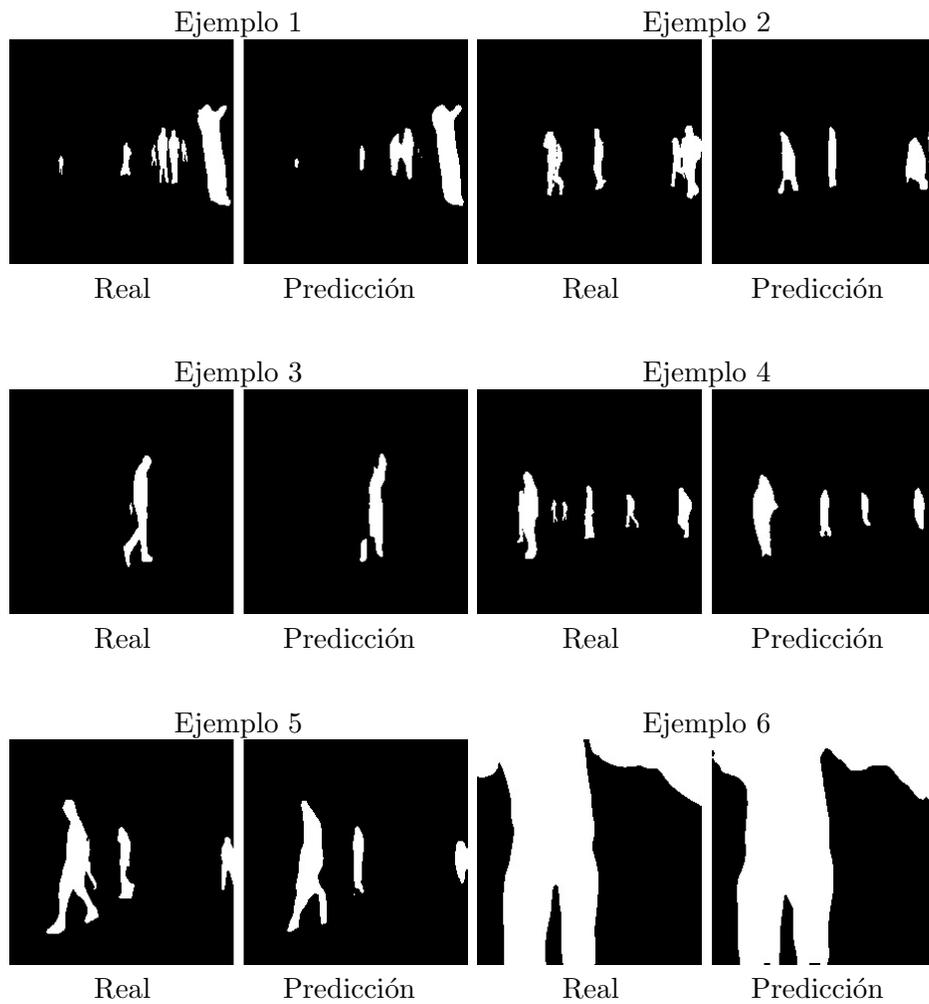


Figura 4.7: Comparación entre las máscaras reales y las máscaras predichas por el modelo ViT entrenado.

4.4. Entrenamiento de la Red Neuronal Convolutiva Espacio-Temporal (ST-CNN)

Al igual que con el modelo ViT, para el entrenamiento de la ST-CNN también se utilizaron las 5 secuencias descritas en la sección de la base de datos, pero en este caso no fueron las RGB. Como se mencionó en la metodología, las imágenes de entrada para la ST-CNN son imágenes de profundidad segmentadas, es decir, las imágenes de profundidad que se capturaron con la cámara multiplicadas por las máscaras de segmentación de peatones.

Las imágenes de la base de datos se separaron en conjuntos de 15 imágenes, las cuales equivalen a un segundo, y de éstas se eligieron solo 10 imágenes distribuidas en todo el segundo, es decir, se seleccionó la primera, la última y 8 más entre las dos. Esto se decidió considerando que en general un peatón avanza poco en un segundo, entonces tomar 15 o 30 imágenes por segundo era como tomar muchas imágenes casi iguales. Así que, si se toman solo 10 imágenes distribuidas, se sigue captando el movimiento de los peatones en un segundo y se disminuye el tiempo de procesamiento. De esta forma, la entrada a la red es una secuencia de 10 imágenes de profundidad segmentadas.

En la clasificación de los niveles de riesgo, un dato muy importante es la distancia a la que se encuentran los peatones en relación con el vehículo. En este caso, esta información la proporcionan las imágenes de profundidad, pero era necesario asegurarse de que los datos fueran reales. Se realizaron pruebas ubicando la cámara en el prototipo de pruebas estacionado y se empezaron a capturar imágenes mientras una persona se situaba frente a la cámara a distancias de 1, 3, 5, 10, 15 y 20 metros. Posteriormente, con las mediciones y las imágenes capturadas se hizo una equivalencia entre distancia y píxeles para saber las distancias reales en las imágenes de la base de datos para el entrenamiento.

Se decidió que la única manera en que se iba a clasificar un evento como *sin riesgo*, era si no se detectaba ningún peatón. Una vez que se detectara

4.4 Entrenamiento de la Red Neuronal Convolutacional Espacio-Temporal (ST-CNN)

a alguno, el evento se iba a clasificar en nivel bajo, medio o alto. El primer parámetro para realizar la clasificación de cada secuencia fue la distancia, considerando los rangos mostrados en la Tabla 4.2.

Nivel	Distancia al vehículo
Alto	1-7 metros
Medio	7-13 metros
Bajo	13-20 metros

Tabla 4.2: Clasificación de los niveles de riesgo por distancia.

Como segundo parámetro se consideró el criterio personal como conductor para analizar también las secuencias por la situación además de la distancia, ya que se podía dar el caso de que por distancia algunos fueran marcados en una clase, pero por contexto se pudieran clasificar en otras. En la Figura 4.8 se muestran algunos ejemplos de secuencias y a qué clase pertenecen y posteriormente se explica por qué se clasificaron de esta forma.

La primera secuencia se considera sin riesgo porque no se detecta ningún peatón, como se mencionó antes. La segunda secuencia se detecta como riesgo bajo en principio por la distancia y también porque el peatón va caminando hacia afuera del carril del vehículo. En la tercera secuencia la primera imagen se puede considerar como nivel medio por distancia, sin embargo, la imagen 10 se podría considerar como nivel alto por la misma razón. Si se analiza la situación, esta corresponde a un peatón que va caminando al lado del carril del vehículo mientras éste está en movimiento. Por el campo de visión de la cámara, al parecer hay un riesgo alto; sin embargo, el vehículo no está pasando cerca del peatón. Al final se clasificó como riesgo medio ya que está cerca y podría atravesarse, pero al menos en la imagen 10 ya el vehículo está pasando casi a su lado.

Por último, en la secuencia de riesgo alto se tiene una situación parecida a la de riesgo bajo, un peatón que pasa frente al vehículo, pero va caminando hacia afuera del carril. Por distancia, en la imagen 1 se podría clasificar como medio; sin embargo, en la imagen 10 ya es nivel alto. Lo que más influyó en

4.4 Entrenamiento de la Red Neuronal Convolutacional Espacio-Temporal (ST-CNN)

que se considerara como nivel alto y no nivel medio es que en la imagen 10 el peatón todavía no sale del carril del vehículo por completo, por lo tanto, existe un riesgo alto de colisión si el vehículo no frena o el peatón no avanza más rápido.

En total, de las cinco secuencias de la base de datos se obtuvieron 4388 secuencias de 15 imágenes cada una, las cuales posteriormente se redujeron a 10 imágenes por secuencia como se comentó previamente. Estas 4388 secuencias se clasificaron en cuatro niveles de riesgo tomando en cuenta los parámetros mencionados previamente (distancia y criterio), obteniendo 3240 secuencias de la clase sin riesgo, 787 de la clase de riesgo de nivel bajo, 266 de la clase de riesgo de nivel medio y 95 de la clase de riesgo de nivel alto.

4.4 Entrenamiento de la Red Neuronal Convolutiva Espacio-Temporal (ST-CNN)

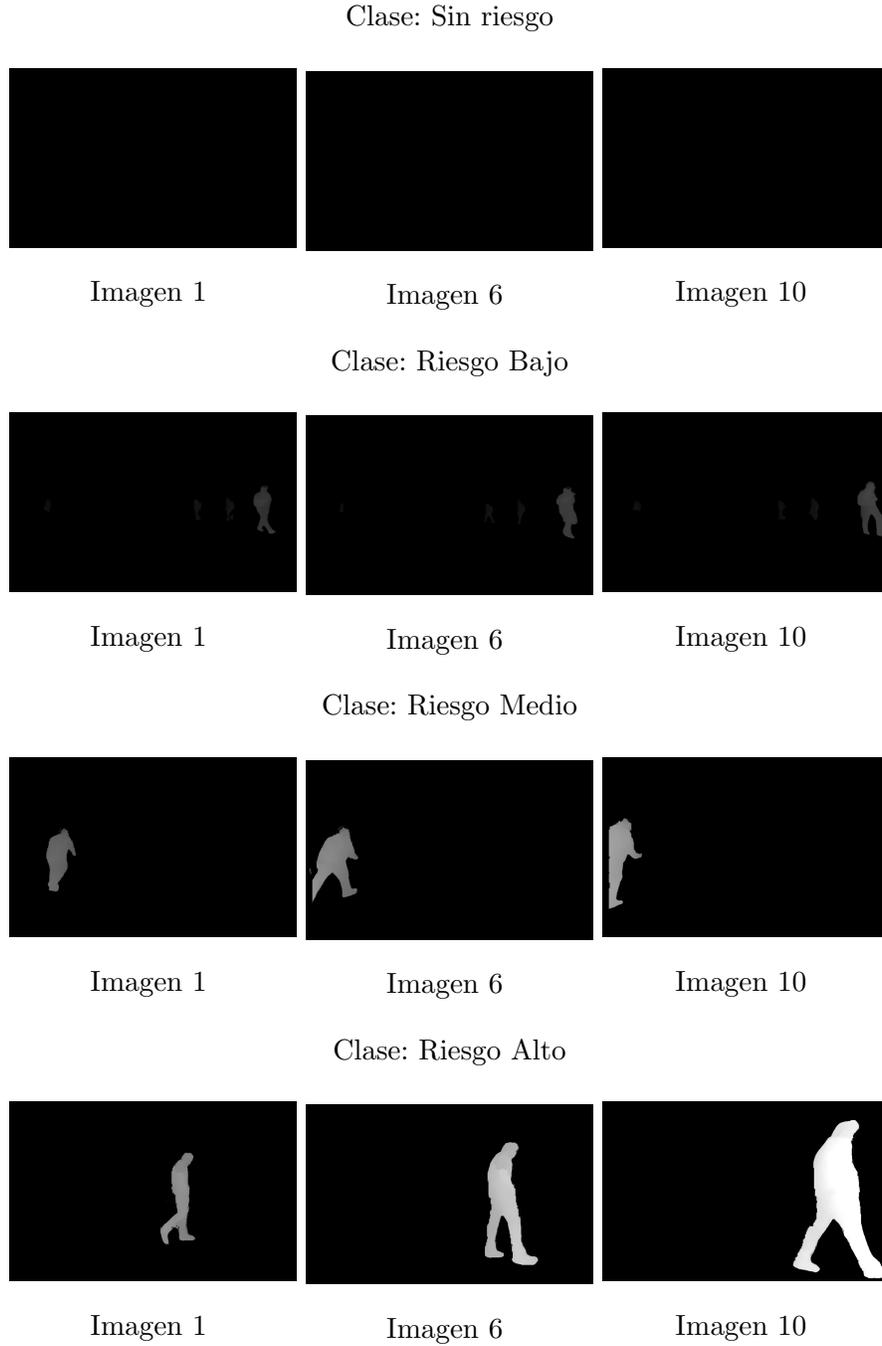


Figura 4.8: Secuencias con casos de clasificación especiales.

4.4 Entrenamiento de la Red Neuronal Convolutiva Espacio-Temporal (ST-CNN)

Como se observa, del nivel alto se obtuvieron muy pocos ejemplos y es por razones obvias, pues al grabar las secuencias no se podían simular por completo estos ejemplos para evitar algún accidente. La forma en que se buscó compensar la baja cantidad de ejemplos de nivel alto fue usando una técnica de aumento de datos, en la cual se generaron las mismas secuencias, pero con un reflejo en horizontal. No se realizó ningún cambio en cuanto a acercamiento o alejamiento de las imágenes o alguna rotación de las mismas, ya que esto alteraría la realidad de las secuencias y los datos serían erróneos. Al final, la cantidad máxima de ejemplos que se pudo obtener fue de 190 secuencias de nivel alto. En el entrenamiento de la red, para evitar el desbalance de clases, las secuencias sin riesgo, de nivel bajo y de nivel medio se ordenaron en cada entrenamiento de forma aleatoria y se tomaron solo 190 ejemplos de cada clase, tomando así todos los ejemplos del nivel alto. Al final, con los 190 ejemplos por clase se obtuvo un total de 760 ejemplos, de los cuales se usó el 80% para el entrenamiento (607 ejemplos) y el 20% para las pruebas (153 ejemplos).

En las Figuras 4.9 y 4.10 se muestran los resultados de exactitud y pérdidas del entrenamiento respectivamente. Se hicieron muchas pruebas con esta red variando los filtros de las capas de convolución, cambiando otros parámetros, se probaron dos optimizadores diferentes, se utilizaron técnicas de regularización como el *dropout* y en todas las pruebas se obtuvo poca estabilidad en el rendimiento de la red. Se estableció un límite de épocas de 50; sin embargo, por la misma inestabilidad se optó por agregar la técnica de detención temprana, y si la red no mejoraba durante 10 épocas, el entrenamiento se detenía.

En la gráfica de la exactitud 4.9 se puede observar que el entrenamiento se detuvo en la época 26. Las primeras tres épocas no aumentó la exactitud y en las siguientes tres creció considerablemente, para mantenerse después entre los valores de 0.8 a 0.9 de exactitud.

4.4 Entrenamiento de la Red Neuronal Convolutacional Espacio-Temporal (ST-CNN)

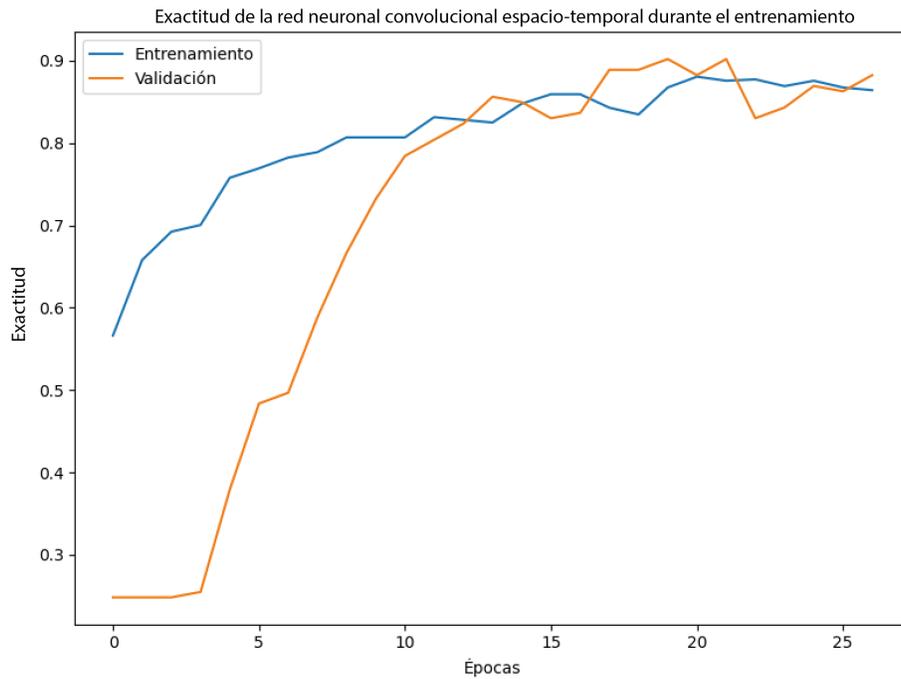


Figura 4.9: Gráfica de la exactitud obtenida en el entrenamiento del modelo ST-CNN.

El poco aumento de la exactitud en las primeras épocas se puede comprender observando la gráfica de las pérdidas 4.10 y cómo estas disminuyen también considerablemente cuando la exactitud sube. En la gráfica de validación se destacan dos picos importantes, uno por la época 3 y otro por la época 22. En esta parte es importante destacar qué es lo que indican los valores de exactitud y pérdida. Por un lado, el valor de exactitud muestra la razón que existe entre los datos bien clasificados y los datos totales, mientras que las pérdidas indican las probabilidades con las que los datos son clasificados en sus respectivas clases. El aumento en las pérdidas no necesariamente representa algo negativo, como es el caso del primer pico de la gráfica, pues si se va a ese mismo punto en la gráfica de exactitud, se observa que es cuando esta comienza a subir. Esto se podría interpretar como que las probabilidades bajaron, pero la clasificación se realizó de mejor forma ayudando a que subiera la exactitud.

4.4 Entrenamiento de la Red Neuronal Convolutacional Espacio-Temporal (ST-CNN)

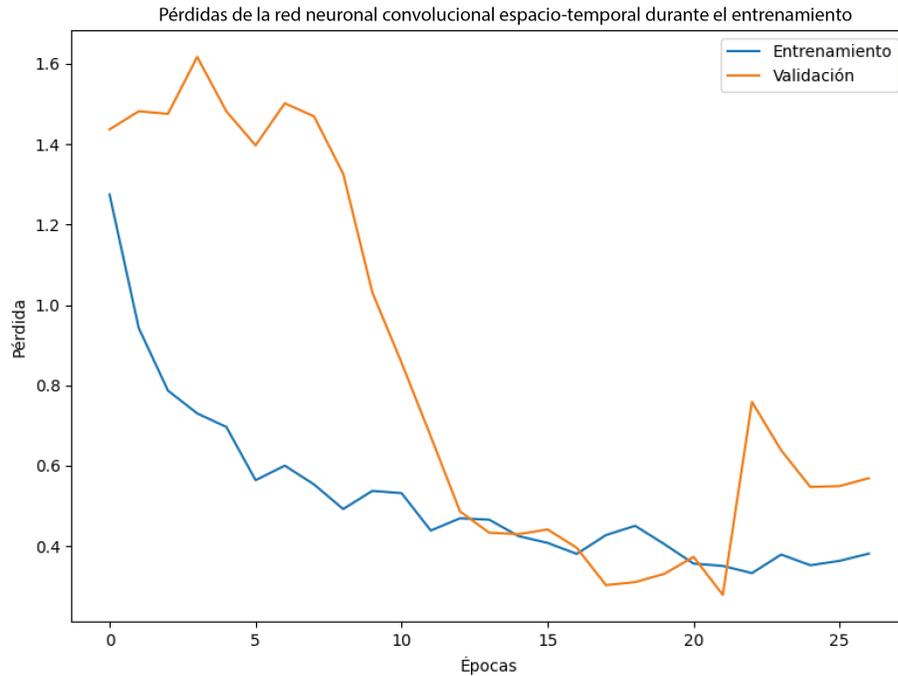


Figura 4.10: Gráfica de las pérdidas en el entrenamiento del modelo ST-CNN.

En general, el comportamiento del modelo, que se ve reflejado en las gráficas, se puede atribuir a que la cantidad de datos por clase era muy baja, y aunque el modelo no cayó en un sobreajuste, sí le costó desde el principio aprender las características de cada clase para identificar bien los datos. Otro aspecto que pudo influir es que la clasificación de las situaciones no solo se basa en la distancia, es decir, en los píxeles, también se basa en el contexto, lo cual hace más complicado el aprendizaje de la red. Esto se puede evaluar desde otro punto obteniendo la matriz de confusión, la cual se obtuvo probando el modelo con 153 ejemplos y se muestra en la Figura 4.11

4.4 Entrenamiento de la Red Neuronal Convolutiva Espacio-Temporal (ST-CNN)

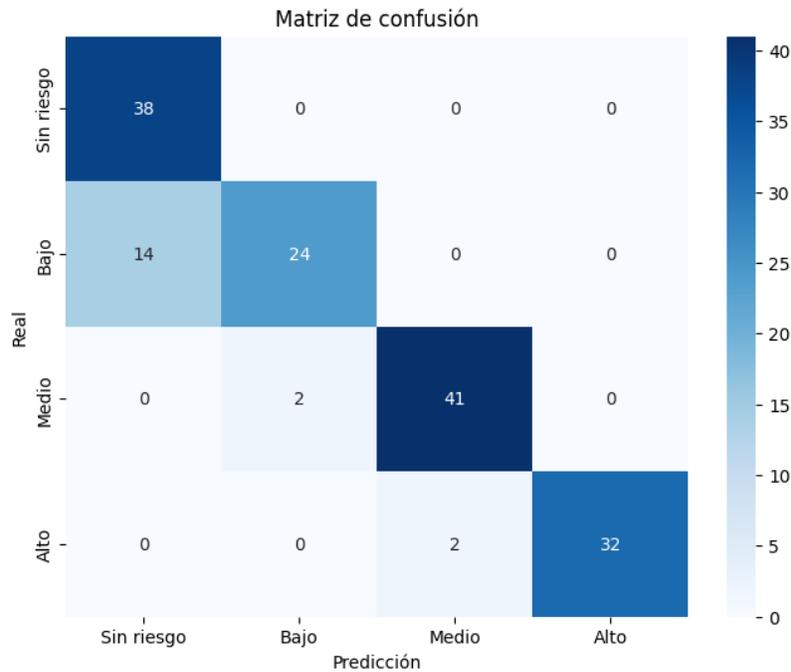


Figura 4.11: Matriz de confusión del modelo ST-CNN.

En la matriz de confusión se puede observar que la parte que le costó más a la red en el aprendizaje fue la diferencia entre el nivel bajo y en el que no hay riesgo, lo cual tiene sentido, porque incluso en algunas situaciones es visualmente complicado diferenciarlos en las imágenes de entrada. La parte positiva del modelo, es que ninguno de los niveles de riesgo más importantes (medio y bajo) fueron clasificados como sin riesgo, pues si esto hubiera pasado representaría un peligro muy importante. Incluso, de las situaciones de nivel alto tampoco se clasificaron como nivel bajo, lo cual es mucho mejor todavía por la misma razón mencionada. Esto nos permite garantizar un buen desempeño del modelo ante los falsos negativos.

A pesar de las variaciones que se ven en las gráficas de exactitud y pérdidas, la matriz de confusión refleja que en la aplicación se tienen muy buenos resultados para los casos críticos, aunque lo óptimo sería mejorar los resultados en las clases bajo y sin riesgo. El análisis de la red ST-CNN se puede complementar con las métricas que se muestran en la Tabla 4.3.

4.5 Implementación del módulo

Métrica	Niveles de riesgo				Promedio
	Sin	Bajo	Medio	Alto	
Precisión	0.730	0.923	0.953	1	0.901
Sensibilidad	1	0.631	0.953	0.941	0.881
Medida F1	0.844	0.749	0.953	0.969	0.878

Tabla 4.3: Métricas por clase con los ejemplos de prueba en la ST-CNN.

Esta tabla muestra la precisión, sensibilidad y F1 para cada una de las clases y ayuda a comprobar que el rendimiento es afectado por la confusión en la predicción de las clases sin riesgo y riesgo bajo. La parte positiva del modelo es que, para las clases con mayor importancia (riesgo alto y riesgo medio) los porcentajes en la predicción son altos, incluso alcanzando el valor de 1 en la precisión de la clase de riesgo alto. Es importante resaltar esta parte, porque si el modelo llega a equivocarse entre las clases de riesgo bajo y sin riesgo, existe un margen de al menos 13 metros de distancia entre el vehículo y la persona, lo cual es aún una distancia considerable, y en cuanto la distancia sea menor a los 13 metros el modelo podrá detectar ahora un caso de riesgo medio y se modifica el error en los niveles previos.

Con esto se comprueba lo que se mencionó antes, que a pesar de que el entrenamiento se llevó a cabo con muchos cambios y poca estabilidad, los resultados de la red no tienen esa poca estabilidad. A excepción de las confusiones entre la clase sin riesgo y la clase baja, la clasificación del modelo no es del todo errónea, y en los puntos que se equivoca no son los más relevantes, lo cual es una ventaja.

4.5. Implementación del módulo

Por último, se unieron todas las partes anteriores; el envío de las imágenes se hizo con el proceso descrito en el capítulo anterior en la sección de

4.5 Implementación del módulo

comunicación entre la unidad de captura y el módulo de procesamiento. En la parte del envío se encontró un inconveniente, ya que, por la forma en la que se configuró el envío, las imágenes se van enviando hasta que se terminan de procesar las anteriores.

Todo el procesamiento, desde que se reciben las imágenes hasta que son procesadas, se realiza en un tiempo promedio de 150 ms por cada secuencia de 10 fotogramas, lo cual se puede considerar como un buen tiempo de procesamiento. Sin embargo, en esos 150 milisegundos la cámara sigue capturando imágenes, pero no las envía hasta que termina el proceso anterior, lo que ocasiona que los fotogramas por segundo ya no correspondan a un mismo segundo y el proceso en total termina tomando de dos a tres segundos, lo cual puede ya no considerarse óptimo, sobre todo ante situaciones críticas.

En cuanto a los resultados del proceso, en las Figuras 4.12, 4.13, 4.14 y 4.15 se muestran algunas secuencias mostrando las imágenes en RGB, los resultados de la segmentación, las imágenes de profundidad segmentadas y el nivel de riesgo en el que se clasificaron. Con estos resultados obtenidos, se confirma que el procesamiento del módulo funciona correctamente, aunque es necesario ajustar la parte del envío de las imágenes para que se pueda considerar que trabaja en tiempo real, de acuerdo con los estándares de los vehículos autónomos.

También, al observar la Figura 4.12 que corresponde a la clase sin riesgo y la Figura 4.13 que corresponde a la clase de riesgo bajo, se entiende por qué el modelo tuvo problemas para distinguir estas dos clases. En esta ocasión la clasificación fue correcta, pero las imágenes de profundidad segmentada, que son las que entran a la red, a simple vista, ambas secuencias se ven completamente negras.

4.5 Implementación del módulo



Figura 4.12: Resultados del procesamiento con el modelo ViT y la ST-CNN. Clase: Sin riesgo.

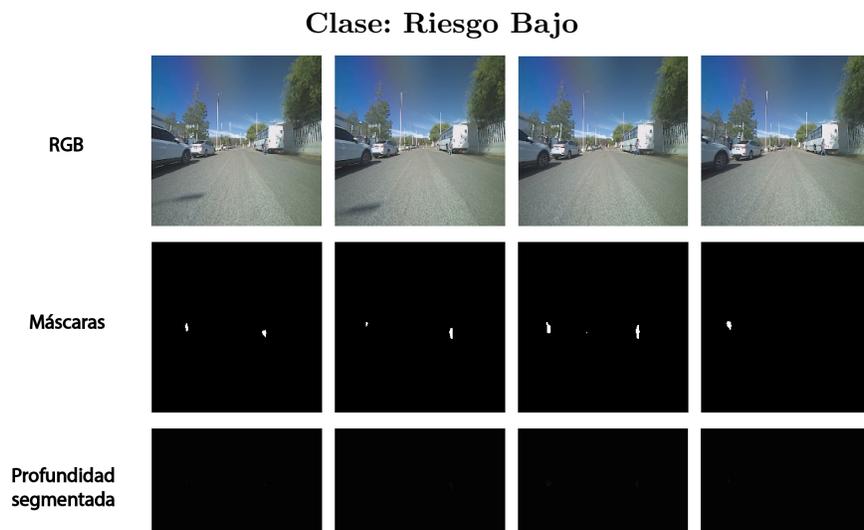


Figura 4.13: Resultados del procesamiento con el modelo ViT y la ST-CNN. Clase: Riesgo Bajo.

4.5 Implementación del módulo

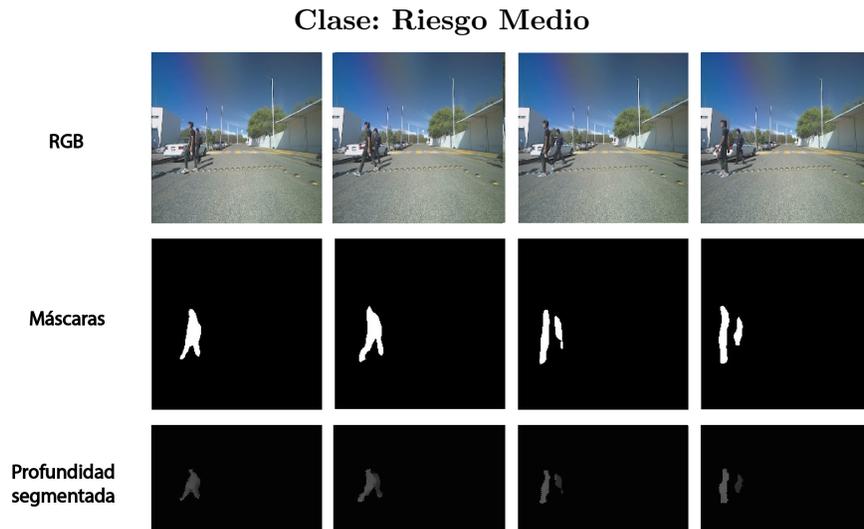


Figura 4.14: Resultados del procesamiento con el modelo ViT y la ST-CNN. Clase: Riesgo Medio.



Figura 4.15: Resultados del procesamiento con el modelo ViT y la ST-CNN. Clase: Riesgo Alto.

4.5 Implementación del módulo

Como último paso, el número de la clase a la que pertenecen las secuencias se regresa desde el módulo de procesamiento hasta la unidad de captura. En esta unidad, y dependiendo de la clase, se utilizan las terminales de la tarjeta de captura para indicar el nivel de riesgo como una salida de voltaje. Se consideraron dos terminales de salida (Pin 11 y Pin 12) para indicar la clase en un formato digital, como se muestra en la Tabla 4.4

Número de clase	Pin 11	Pin 12
0 (Sin riesgo)	Inactivo	Inactivo
1 (Riesgo bajo)	Activo	Inactivo
2 (Riesgo medio)	Inactivo	Activo
3 (Riesgo alto)	Activo	Activo

Tabla 4.4: Salida de los pines para indicar la clase determinada en el procesamiento.

La activación de las terminales se realizó usando la biblioteca *Jeston.GPIO*, la cual permite elegir el pin de salida y configurarlo para que esté activo (salida de 3.3 V) o inactivo (salida de 0 V). Esto se realizó para que el módulo fuera adaptable a algún otro sistema en un vehículo eléctrico. Es decir, como se mencionó al inicio de este trabajo, este módulo solo cubre la parte de detección de peatones y estimación de riesgos. Después de estas dos etapas debería ir un sistema que interprete las señales de salida y tome una decisión con base en los resultados obtenidos con el módulo basado en visión. Considerando esto, se decidió que era más fácil de adaptar una salida de voltaje a otro sistema, que una salida como imagen.

Capítulo 5

Conclusiones

Una vez terminado este trabajo, se ha llegado a la conclusión de que los mecanismos de atención pueden ser una herramienta muy útil en varias áreas, como el lenguaje natural o la visión por computadora. Normalmente se menciona que es un método que requiere muchos datos para el entrenamiento, lo cual es cierto, pero en este trabajo se logró un buen resultado en la segmentación de imágenes sin tener demasiados datos. Esto tomando en cuenta que es un entorno de exterior controlado y que si se quisiera adaptar el módulo realizado para entornos urbanos definitivamente requeriría de más datos de entrenamiento.

El trabajo se realizó probando un modelo básico de transformadores de visión, lo cual fue muy positivo porque se pudo probar con diferentes parámetros, hacer pequeñas modificaciones y esto sin duda ayudó a comprender mejor cómo funciona la arquitectura básica de este modelo. Más allá de los resultados, que para este propósito fueron buenos considerando el alcance del modelo, se queda una base sólida de conocimiento sobre los transformadores de visión que permitirá experimentar en trabajos futuros, permitiendo mejorar la arquitectura una vez que se conocen cada una de sus partes.

Conclusiones

En cuanto al uso de la red neuronal convolucional espacio-temporal, se puede concluir que es una herramienta muy favorable, ya que, en aplicaciones como esta, permite hacer clasificaciones que se acercan más a la realidad, pues no solo se analiza una imagen, sino una situación. En este trabajo quizás fue algo que no se pudo notar demasiado debido a la limitada cantidad de datos para el entrenamiento; sin embargo, la base de datos era bastante variada en cuanto a situaciones y escenarios, lo cual es muy positivo porque el modelo pudo aprender, si no todos, una gran mayoría de los casos que pueden pasar en el entorno donde se obtuvieron las imágenes. Si más adelante se amplía esta base de datos, seguramente los resultados tendrán mejoras significativas.

La parte de la estimación de riesgos es algo que tiene una relevancia muy alta, por eso es que hay tantas técnicas y enfoques para poder realizarla. Como se mencionó en la introducción de este trabajo, de este tipo de módulos depende la aceptación social de los sistemas de conducción autónoma. Al igual que con la segmentación de imágenes, los resultados fueron buenos, aunque hay varias oportunidades de mejora, como incluir datos de velocidad para hacer la estimación o ser más precisos en el cálculo de la distancia. Además, como la mayoría de los trabajos que se estudiaron para realizar tareas de este tipo, una de las partes a mejorar es la realización del proceso en tiempo real, pues en esta área es indispensable tener respuestas rápidas.

En general, se puede decir que fue un trabajo que se realizó con éxito, que entregó buenos resultados y que deja una base importante para trabajos futuros. Se quedan también algunas herramientas como el manual de configuración para la tarjeta de desarrollo, la protección para las tarjetas y la interfaz de captura de imágenes que, aunque parecen algo simple, pueden ahorrar tiempo en otros trabajos. Al final, se obtuvo un gran aprendizaje no solo en la implementación del módulo, sino también en investigación, ya que se realizaron dos publicaciones directamente relacionadas con este trabajo:

- Medina-Garcia, A.; Duarte-Jasso, J.; Cardenas-Cornejo, J.-J.; Andrade-Ambriz, Y.A.; Garcia-Montoya, M.-A.; Ibarra-Manzano, M.-A.; Almanza-Ojeda, D.-L. Vision-Based Object Localization and

Classification for Electric Vehicle Driving Assistance. *Smart Cities* 2024, 7, 33-50. IF:7, Q1, <https://doi.org/10.3390/smartcities7010002>.

- Duarte-Jasso, J.; Medina-Garcia, A.; et al, “Reconstruction of an outdoor environment during navigation of an electric vehicle prototype” International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE 2023), presentado el 14 de diciembre 2023.

Además, este trabajo de tesis forma parte de diferentes proyectos de investigación apoyados por DAIP y por IDEA GTO:

- Proyecto DAIP con número de propuesta CIIC 048/2024: “Implementación de frenado automático de un vehículo eléctrico para personas con discapacidad”, Vigencia: 1 de Enero al 31 de Diciembre de 2024.
- Proyectos Estratégicos de Ciencia, Tecnología, Innovación e Impulso de la Propiedad Industrial para el Valle de la Mentefactura, con el proyecto: SOL-1249 “Implementación de vehículo eléctrico biplaza para transporte de personas con discapacidad”, Vigencia: 10 de enero al 7 de octubre 2024.
- CIIC 059/2023: “Detección de obstáculos para la conducción autónoma de un vehículo eléctrico”, Vigencia: 1 de Enero al 31 de Diciembre de 2023.

Toda la experiencia conseguida en el desarrollo de proyectos de investigación, configuración de cámaras, diseño y prueba de modelos de redes convolucionales, me han permitido abrir el panorama sobre el uso y aplicación del cómputo a los sistemas inteligentes para el desarrollo de tecnologías en favor de la sociedad. En ese contexto, y como perspectiva personal, me decido por continuar mi formación profesional en esta área del diseño de sistemas inteligentes a través de un doctorado.

Bibliografía

- [1] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, and G. Currie, “Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy,” *Journal of Transport and Land Use*, vol. 12, Jan. 2019.
- [2] S. International, *Surface Vehicle Recommended Practice: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2021.
- [3] Waymo, “Tecnología de conducción autónoma - más información sobre nosotros.” <https://waymo.com/intl/es/about/>, Apr 2024. Accessed: 2024-04-29.
- [4] W. H. Organization, “Road traffic injuries.” <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>, Dec 2023. Accessed: 2024-04-08.
- [5] I. N. de Estadística y Geografía (INEGI), “Accidentes por tipo de accidente.” https://www.inegi.org.mx/app/tabulados/interactivos/?pxq=ATUS_ATUS_2_eee02f47-cbdf-4b2e-b11c-e85b678223a4, 2023. Accessed: 2024-08-19.
- [6] T. Verstraete and N. Muhammad, “Pedestrian collision avoidance in autonomous vehicles: A review,” *Computers*, vol. 13, no. 3, 2024.
- [7] A. Palffy, J. F. P. Kooij, and D. M. Gavrilă, “Detecting darting out pedestrians with occlusion aware sensor fusion of radar and stereo came-

- ra,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1459–1472, 2023.
- [8] W. Zhao, T. Wang, A. Tan, and C. Ren, “Nighttime pedestrian detection based on a fusion of visual information and millimeter-wave radar,” *IEEE Access*, vol. 11, pp. 68439–68451, 2023.
- [9] H. Kulhandjian, J. Barron, M. Tamiyasu, M. Thompson, and M. Kulhandjian, “Ai-based pedestrian detection and avoidance at night using multiple sensors,” *Journal of Sensor and Actuator Networks*, vol. 13, no. 3, 2024.
- [10] K. S. Arikumar, A. Deepak Kumar, T. R. Gadekallu, S. B. Prathiba, and K. Tamilarasi, “Real-time 3d object detection and classification in autonomous driving environment using 3d lidar and camera sensors,” *Electronics*, vol. 11, no. 24, 2022.
- [11] Z. Peng, Z. Xiong, Y. Zhao, and L. Zhang, “3-d objects detection and tracking using solid-state lidar and rgb camera,” *IEEE Sensors Journal*, vol. 23, no. 13, pp. 14795–14808, 2023.
- [12] H. Xu, S. Huang, Y. Yang, X. Chen, and S. Hu, “Deep learning-based pedestrian detection using rgb images and sparse lidar point clouds,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 5, pp. 7149–7161, 2024.
- [13] X. Song, G. Li, L. Yang, L. Zhu, C. Hou, and Z. Xiong, “Real and pseudo pedestrian detection method with ca-yolov5s based on stereo image fusion,” *Entropy*, vol. 24, no. 8, 2022.
- [14] Q. Xie, Q. Long, J. Li, L. Zhang, and X. Hu, “Application of intelligence binocular vision sensor: Mobility solutions for automotive perception system,” *IEEE Sensors Journal*, vol. 24, no. 5, pp. 5578–5592, 2024.
- [15] Y. Zhang, X. Zhang, Y. Fujinami, and P. Raksincharoensak, “Social force model-based adaptive parameters collision avoidance method considering motion uncertainty of the pedestrian,” *IEEE Access*, vol. 12, pp. 794–809, 2024.

BIBLIOGRAFÍA

- [16] B. Tang, Z. Yang, H. Jiang, and Z. Hu, “Development of pedestrian collision avoidance strategy based on the fusion of markov and social force models,” *Mechanical Sciences*, vol. 15, no. 1, pp. 17–30, 2024.
- [17] Y. Zhang, X. Shen, and P. Raksincharoensak, “Study on collision avoidance strategies based on social force model considering stochastic motion of pedestrians in mixed traffic scenario,” *Journal of Robotics and Mechatronics*, vol. 35, no. 2, pp. 240–254, 2023.
- [18] Y. Ding, W. Zhang, X. Wu, J. Xu, and J. Gong, “A collision avoidance strategy based on entropy-increasing risk perception in a vehicle–pedestrian-integrated reaction space,” *World Electric Vehicle Journal*, vol. 15, no. 5, 2024.
- [19] M. Everett, Y. F. Chen, and J. P. How, “Collision avoidance in pedestrian-rich environments with deep reinforcement learning,” *IEEE Access*, vol. 9, pp. 10357–10377, 2021.
- [20] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, “An overview of autonomous vehicles sensors and their vulnerability to weather conditions,” *Sensors*, vol. 21, no. 16, 2021.
- [21] Y. Li, J. Moreau, and J. Ibanez-Guzman, “Emergent visual sensors for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 4716–4737, 2023.
- [22] Z. Chen and X. Huang, “Pedestrian detection for autonomous vehicle using multi-spectral cameras,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 211–219, 2019.
- [23] A. Inc., “Adobe illustrator.” Software, 2024.
- [24] A. Zaarane, I. Slimani, W. Al Okaishi, I. Atouf, and A. Hamdoun, “Distance measurement system for autonomous vehicles using stereo camera,” *Array*, vol. 5, p. 100016, 2020.
- [25] R. Fan, L. Wang, M. Junaid Bocus, and I. Pitas, *Computer Stereo Vision for Autonomous Driving: Theory and Algorithms*, pp. 41–70. Cham: Springer International Publishing, 2023.

BIBLIOGRAFÍA

- [26] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017.
- [27] Y. Yu, C. Wang, Q. Fu, R. Kou, F. Huang, B. Yang, T. Yang, and M. Gao, “Techniques and challenges of image segmentation: A review,” *Electronics*, vol. 12, no. 5, 2023.
- [28] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022.
- [29] M. M. Islam, v. A. R. Newaz, and A. Karimodini, “Pedestrian detection for autonomous cars: Inference fusion of deep neural networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23358–23368, 2022.
- [30] I. Papadeas, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, “Real-time semantic image segmentation with deep learning for autonomous driving: A survey,” *Applied Sciences*, vol. 11, no. 19, 2021.
- [31] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2014.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.
- [33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020.
- [34] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, “A survey on vision transformer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.
- [35] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, “Semantic segmentation using vision transformers:

- A survey,” *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106669, 2023.
- [36] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, “Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5240–5250, June 2023.
- [37] Q.-V. Lai-Dang, “A survey of vision transformers in autonomous driving: Current trends and future directions,” 2024.
- [38] M. Kang, W. Lee, K. Hwang, and Y. Yoon, “Vision transformer for detecting critical situations and extracting functional scenario for automated vehicle safety assessment,” *Sustainability*, vol. 14, no. 15, 2022.
- [39] F. Cui, Q. Zhang, J. Wu, Y. Song, Z. Xie, C. Song, and Z. Xu, “Online multipedestrian tracking based on fused detections of millimeter wave radar and vision,” *IEEE Sensors Journal*, vol. 23, no. 14, pp. 15702–15712, 2023.
- [40] H. Li, Y. Ren, K. Li, and W. Chao, “Trajectory prediction with attention-based spatial-temporal graph convolutional networks for autonomous driving,” *Applied Sciences*, vol. 13, no. 23, 2023.
- [41] Y. Miao, J. Han, Y. Gao, and B. Zhang, “St-cnn: Spatial-temporal convolutional neural network for crowd counting in videos,” *Pattern Recognition Letters*, vol. 125, pp. 113–118, 2019.
- [42] Y. A. Andrade-Ambriz, S. Ledesma, M.-A. Ibarra-Manzano, M. I. Oros-Flores, and D.-L. Almanza-Ojeda, “Human activity recognition using temporal convolutional neural network architecture,” *Expert Systems with Applications*, vol. 191, p. 116287, 2022.
- [43] N. Mitrović, M. Đorđević, S. Veljković, and D. Danković, “Implementation and testing of websocket protocol in esp32 based iot systems,” *Facta universitatis - series: Electronics and Energetics*, vol. 36, pp. 267–274, 2023.

BIBLIOGRAFÍA

- [44] Stereolabs, *ZED X and ZED X Mini Datasheet*, 2024. Accessed: 2024-05-07.
- [45] Stereolabs, *ZED Link Duo Capture Card Datasheet*, 2024. Accessed: 2024-05-07.
- [46] I. Autodesk, “Autodesk inventor.” Software, 2024.
- [47] I. The MathWorks, “Matlab.” Software, 2023.
- [48] A. Soni, “Pixellib,” 2021. GitHub repository.
- [49] A. Inc., “Adobe photoshop.” Software, 2024.