2020-06

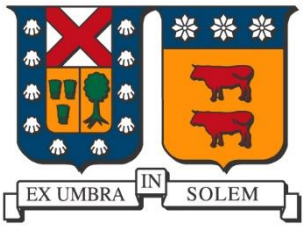
SISTEMA DE GESTION AUTOMATICA DE IMAGENES PARA MEDICIONES OBSERVACIONALES SOBRE EL USO DEL CINTURON DE SEGURIDAD EN VEHICULOS LIVIANOS

VERGARA ZAPATA, SERGIO OSVALDO

Universidad Técnica Federico Santa María

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO - CHILE



"SISTEMA DE GESTIÓN AUTOMÁTICA DE IMÁGENES PARA MEDICIONES OBSERVACIONALES SOBRE EL USO DEL CINTURÓN DE SEGURIDAD EN VEHÍCULOS LIVIANOS"

SERGIO OSVALDO NICOLÁS VERGARA ZAPATA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA:

Marcos Zúñiga Barraza

PROFESOR CO-REFERENTE:

Werner Creixell Fuentes

JUNIO 2020

"Dedicado a mis profesores de la vida, mis amados padres"

Agradecimientos

Francamente soy un agradecido de mi etapa universitaria en todos sus más diversos aspectos.

Primero quiero agradecer a la universidad por presentar en mi vida esta mezcla rara entre una vida social diversa con personas de todo el país y una vida académica de la más alta exigencia. Aquello me permitió poner a prueba aspectos personales no solo en lo académico, sino también en lo emocional. Son experiencias que me acompañarán durante toda la vida.

Segundo quiero agradecer a todos los grupos humanos que me permitieron disfrutar de su compañía, desde mi queridísimo equipo de básquetbol, con el que representamos durante muchos años a nuestra casa de estudios en varias competencias regionales y nacionales, hasta al grupo de trabajo SEAD con quienes compartí maravillosos momentos durante las jornadas de apoyo docente, junto con nuestros carretes llenos de histrionismo en su estado más puro.

Tercero quiero agradecer a los grandes amigos que hice en esta etapa, con quienes viví intensamente toda esta travesía desde lágrimas de dicha y felicidad por obstáculos superados hasta lágrimas de grandes penas y frustraciones de la vida misma.

Finalmente, quiero agradecer infinitamente a quienes hicieron posible todo lo anteriormente mencionado, mi motor de vida, muchas gracias a mi familia.

Resumen

El objetivo de este trabajo es generar un sistema automático, confiable y de bajo costo, que sea capaz de facilitar las mediciones de tránsito sobre el uso del cinturón de seguridad. Así como también, tiene por objetivo influenciar el desarrollo de nuevos sistemas tecnológicos en materia de seguridad vial en Chile. Para el desarrollo de este proyecto se implementaron algoritmos de visión por computador, junto con el desarrollo de una plataforma web, integrada con una tarjeta de desarrollo Raspberry Pi y una cámara. Las funciones de estos elementos son: la Raspberry Pi conectada con la cámara realizan la captura de imágenes de vehículos desde un sector urbano de interés, las cuales son procesadas para extraer imágenes de los vehículos de forma individual y se envían; por otro lado, el servidor web, recibe y almacena estas imágenes, que eventualmente un usuario puede acceder a ellas a través de la plataforma web, con la finalidad de medir el uso del cinturón de seguridad, observando las imágenes una a una. El acceso a la plataforma es a través de una cuenta Google para la autenticación del usuario, quien cuenta con una interfaz que le permite realizar mediciones sobre imágenes de vehículos y llevar un registro histórico de estas mediciones. El desarrollo de este trabajo se lleva a cabo a través del programa de Memorias Multidisciplinarias de la UTFSM, donde confluyen perspectivas de distintas carreras para la resolución del problema.

Palabras clave: Cinturón de seguridad, seguridad vial, visión por computador, procesamiento de imágenes, Raspberry Pi.

Abstract

The aim of this work is to generate an automatic, reliable and low-cost system that is capable of facilitating traffic measurements on the use of seat belts. As well as, it aims to influence the development of new technological systems for road safety in Chile. For the development of this project, computer vision algorithms were implemented, along with the development of a web platform, integrated with a Raspberry Pi development card and a camera. The functions of these elements are: the Raspberry Pi's camera capture images of vehicles from an urban sector of interest, which are processed to extract images of vehicles individually and are sent; On the other hand, the web server receives and stores these images, which eventually a user can access through the web platform, in order to measure the use of the seat belt, observing the images one by one. Access to the platform is through a Google account for user authentication, which has an interface that allows you to make measurements on vehicle images and keep a historical record of these measurements. The development of this work is carried out through the UTFSM Multidisciplinary Thesis program, where perspectives from different careers come together to solve the problem.

Keywords: Seat belt, road safety, computer vision, image processing, Raspberry Pi.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Glosario

SIGLA	Significado de la sigla
CRUD	Create, Read, Update and Delete
YOLO	You only look once
CONASET	Comisión Nacional de Seguridad y Tránsito

Capítulo 1

1 Introducción

A través del programa de Memorias Multidisciplinarias [1], se desarrolló el proyecto propuesto por la Comisión Nacional de Seguridad y Tránsito (CONASET) [2]. Para llevar a cabo este proyecto se formó un equipo multidisciplinario, con especialidades en el aspecto comercial por Romina Morales y técnica por Sergio Vergara. El presente documento abarca solamente el aspecto técnico del problema.

1.1 Propósito

El propósito de esta memoria es desarrollar un sistema de base tecnológica que colabore con las mediciones sobre el uso del cinturón de seguridad que se debe efectuar por parte de CONASET cada año. Este sistema debe cumplir 3 características principales: que realice una gestión automática de los datos; que su rendimiento sea confiable; y que los resultados de las mediciones se puedan obtener de manera oportuna. Además de estos aspectos, el enfoque que se abarca contribuye a que su implementación sea de bajo costo.

1.2 Alcance

El alcance del proyecto es implementar una prueba de concepto de un sistema consistente en una plataforma web, de fácil acceso y uso intuitivo para los usuarios, que les permita realizar mediciones observacionales sobre el uso del cinturón de seguridad. Estas mediciones se hacen sobre imágenes capturadas por un dispositivo en terreno controlado de forma remota, donde se muestran vehículos livianos en tránsito. Además de la plataforma web, se desarrolla el dispositivo de muestreo que captura las imágenes de los vehículos, consistente en una tarjeta de desarrollo Raspberry Pi, conectada a una cámara. Este dispositivo se comunica con un servidor web para almacenar las imágenes, las que pueden ser visualizadas a través de la plataforma web.

1.3 Contexto

En el desafío propuesto por parte de CONASET en el programa de Memorias Multidisciplinarias de la UTFSM, se enuncia la necesidad de contar con un sistema que sea capaz de detectar el uso del cinturón de seguridad al interior de los vehículos livianos que circulan a través de los principales focos urbanos de nuestro país. Los requisitos principales con los que debería contar el sistema de detección para satisfacer las necesidades de CONASET, corresponden a realizar una detección del uso del cinturón de seguridad de forma automática, remota y confiable.

Actualmente, este organismo de gobierno envía a licitación los proyectos relacionados a la medición de estadísticas de tránsito. Para el levantamiento de

información estadística en estos proyectos se realizan mediciones observacionales con personas en terreno, es decir, registros visuales sobre el uso de variables como: cantidad de pasajeros en los vehículos, uso de sistemas de retención infantil, uso de teléfonos celulares en la conducción, uso del cinturón de seguridad de los ocupantes del vehículo, entre otros. Los resultados de estos procedimientos son entregados a CONASET en forma de reportes estadísticos.

1.3.1 Definición del problema

En relación con la medición de estas variables de tránsito, CONASET presenta dificultades en la trazabilidad de la información y en la obtención de resultados de forma oportuna. Cada año cuentan con un presupuesto diferente para presentar estos proyectos, teniendo que adaptarse a los recursos presentes. Los resultados de estas mediciones cada año tienen diferentes características y son poco comparables entre sí. En este contexto, el problema abarcado en esta memoria, es el de la imposibilidad de realizar mediciones sobre el uso del cinturón de seguridad que sean comparables en el tiempo.

1.4 Objetivos generales

- Generar un sistema que gestione imágenes de vehículos livianos en tránsito, de forma automática, para la realización de estudios observacionales del uso del cinturón de seguridad.
- Mejorar el sistema actual con que CONASET realiza los estudios observacionales.

1.5 Objetivos específicos

- Implementar un dispositivo capaz de capturar fotografías de vehículos livianos en tránsito y enviarlas por internet a un servidor web.
- Desarrollar una plataforma web que permita revisar las imágenes de los vehículos y registrar variables de interés como: cantidad de pasajeros por vehículos, cantidad de ocupantes que hacen uso del cinturón de seguridad, entre otras.
- Ofrecer una captura visual lo más nítida posible de los vehículos en las imágenes.
- Permitir el acceso a la plataforma a través de cuentas de Google.
- Evaluar y seleccionar las tecnologías en función de una rápida implementación y facilitar actividades de mantención posteriores.
- Permitir que los datos obtenidos en las mediciones realizadas por el usuario puedan ser descargados en un formato de salida Excel.

1.6 Estructura del documento

Este trabajo de memoria se encuentra organizado de la siguiente manera: El capítulo 2 corresponde al Estado del Arte. En él se habla sobre distintas técnicas y tecnologías que existen actualmente en el área del procesamiento de imágenes, además de posibles implementaciones que apoyen la detección del cinturón de seguridad en los vehículos. En el capítulo 3 trata sobre los distintos requerimientos que posee el trabajo propuesto, tanto funcionales como no funcionales. El capítulo 4 abarca los temas de análisis y diseño del sistema a implementar, la arquitectura del sistema y descripción de sus componentes. El capítulo 5 detalla los posibles riesgos que podría tener el sistema y aquellas técnicas usadas para su prevención y mitigación. El capítulo 6 trata principalmente en el proceso de implementación del sistema, especificando el flujo de la información a través de cada funcionalidad y cómo colaboran en conjunto para dar solución el problema propuesto. En el capítulo 7 se detalla los procedimientos realizados para obtener una prueba de concepto del sistema implementado, que permita validar su funcionamiento. El capítulo 8 presenta las posibles mejoras al proyecto. Finalmente, el capítulo 9 muestra las conclusiones obtenidas a través del trabajo realizado.

Capítulo 2

2 Estado del arte

En este capítulo se dará a conocer distintas técnicas y procedimientos existentes en materia de procesamiento de imágenes para la detección de los vehículos en tránsito y aquellas soluciones que apuntan a detectar directamente el uso del cinturón de seguridad.

2.1 Detección del cinturón de seguridad desde el interior

En el año 2004, los inventores George E. Hypke y Robert R. Walker con su patente "*Apparatus and methods for monitoring and detecting seat belt usage*" [3], proponen un sistema de detección del uso del cinturón a través de una pistola radar. Este sistema contempla un dispositivo transmisor y receptor alojado en el vehículo, el cual se comunica a través de ondas de radio con la pistola radar al exterior, que también posee un transmisor y receptor. Esta comunicación se realiza preferentemente en la banda de 418 MHz, a través de 8 canales, los cuales envían los datos desde el vehículo al ser solicitados por la pistola radar, cuando se "dispara" la señal al automóvil. La información de los canales contempla el estado del uso del cinturón, es decir, si está siendo utilizado o no, además del código del vehículo en particular para ser identificado.

En el sistema anterior, se hace indispensable la instalación de los sensores y el dispositivo de comunicación en el vehículo para lograr la detección del uso del cinturón, por lo que presenta un alto grado de intervención en los automóviles, sin mencionar el hecho que aquello requiere una política de seguridad, que establezca normas legales que permitan su instalación. Dicho sistema se encuentra dentro de las categorías de detección interior-exterior, ya que se detecta el uso del cinturón al interior del vehículo y la comunicación con el exterior hacia la pistola radar permite monitorearlo. También se puede catalogar como detección manual, puesto que requiere intervención humana en el momento y lugar que se requiere monitorear, junto con el uso de la pistola radar. Finalmente se puede decir que esta detección se hace con conocimiento de los ocupantes, por lo que el comportamiento de estos está condicionado al tener conciencia de que pueden ser monitoreados sobre el uso del cinturón de seguridad al transportarse. Otro sistema es el propuesto en la patente "*Seat belt status external monitoring apparatus and method*" [4], publicada el año 2010 por el inventor William D. Cotter, cuya propuesta consiste en un sistema que monitorea el uso del cinturón de seguridad al interior de un vehículo, utilizando sensores infrarrojos y de peso en los asientos. Estos sensores tienen el propósito específico de detectar la presencia de una persona que ocupe un asiento, donde se tiene la capacidad de calibrar el sensor de peso para establecer un rango de pesos que aporta información acerca de lo que se encuentra sobre el asiento, en cuyo caso satisfacer la condición de este rango, el sensor infrarrojo procede a detectar si es una persona o no lo que se encuentra sobre el asiento. El sistema cuenta además con un controlador central que recibe las señales de estos

sensores, junto con otros sensores posicionados en las hebillas de los cinturones de seguridad, que detectan si estos se encuentran abrochados. Con estos datos sobre la presencia de ocupantes en los asientos y el uso del cinturón en cada uno, se tiene la capacidad de tomar una decisión y comunicar el estado de los asientos hacia el exterior del vehículo con una señal lumínica, facilitando el monitoreo desde el exterior.

La patente anterior presenta similitudes con la propuesta de George E. Hypke y Robert R. Walker, en el sentido de que requiere la intervención del vehículo con instalación de sensores y otros dispositivos. También se puede categorizar como detección interior-exterior y detección manual, ya que requiere que un observador externo visualice la señal lumínica para realizar la medición. En este último aspecto, la presente patente puede representar una ventaja sobre utilizar la pistola radar de la patente anterior en circunstancias donde se hace complejo focalizar la pistola radar apuntando al vehículo en movimiento, mientras que representa una desventaja en situaciones donde no se logre distinguir la señal lumínica con facilidad o que se confunda con otras luces, por ejemplo. Cabe mencionar que esta última patente también realiza la detección con conocimiento de los ocupantes del vehículo, por lo que el comportamiento de estos se ve condicionado por este factor.

2.2 Detección del cinturón de seguridad desde el exterior

Un tercer caso de estudio de presenta en el artículo *"Image-based seat belt detection"*, publicado el año 2011 por los autores Huiwen Guo, Hui Lin y Shaohua Zhang [5]. Este trabajo propone un método de detección del uso del cinturón mediante una cámara externa y ajena al vehículo, desde donde se capturan imágenes y se realiza un posterior procesamiento de las mismas para determinar la posición del conductor. Se proponen parámetros a considerar para encontrar dicha zona de interés con respecto a la patente del automóvil. En dicha zona de interés donde se encuentra el conductor, se aplican técnicas de detección de bordes y se analizan para determinar regiones candidatas que satisfagan las características de un cinturón de seguridad. Finalmente se analizan los resultados para tomar la decisión acerca de si existe uso o no del cinturón de seguridad por parte del conductor.

Este último trabajo se puede categorizar como detección automática, exterior y con desconocimiento de los ocupantes del vehículo, por el hecho que no fue necesaria ningún tipo de instalación al interior del vehículo, además de que las cámaras son elementos habituales en las rutas, por lo que no necesariamente se puede sospechar de una detección del uso del cinturón al notar su presencia.

2.3 Segmentación en procesamiento de imágenes

La segmentación, en el campo de la visión por computador, es el proceso de dividir la imagen digital en varios grupos de píxeles u objetos. El objetivo de esta técnica es poder cambiar la representación de la imagen a otra que resulte más significativa. Existen numerosos algoritmos que permiten dividir la imagen en regiones de interés, asignando etiquetas a estas zonas y extraer información útil.

En el trabajo realizado por Prem Kumar Bhaskar y Suet-Peng Yong titulado *"Image Processing Based Vehicle Detection and Tracking Method"* [6], se utiliza el modelo de mezcla de Gaussianas (GMM: Gaussian Mixture Model) para separar el primer plano del fondo de la imagen, aprendiendo las características del fondo a través de una sucesión de cuadros de video. De esta forma logra la detección y conteo de los vehículos que pasan por la calzada.

GMM permite identificar aquellas regiones de la imagen que se encuentran con gran variabilidad, lo que ayuda a detectar el movimiento de los vehículos. Una vez segmentadas las regiones en movimiento, se utilizan métodos de procesamiento para eliminar el ruido existente. Posteriormente, se calculan las áreas y centroides de las regiones en movimientos y se le realiza el seguimiento a través de la imagen. Cuando las regiones sobrepasan una barrera establecida en la imagen, se considera que corresponde a un vehículo que ha pasado por la carretera y aumenta el contador.

Capítulo 3

3 Análisis de Requerimientos

En este capítulo se detallan los requerimientos funcionales, de interfaces y de ambiente que presenta el sistema. Esta información nace desde la perspectiva técnica, así como desde la del usuario final.

3.1 Requisitos del sistema

A continuación, se describen requisitos funcionales del sistema, interfaces externas y eventos.

3.1.1 Requisitos funcionales

RF1 El sistema debe ser capaz de obtener imágenes de vehículos en tránsito.

RF2 El sistema debe ser de carácter No Invasivo (que no requiera instalar algún dispositivo adicional al vehículo)

RF3 El sistema debe ser capaz de ayudar a realizar mediciones del uso del cinturón de seguridad de forma remota.

RF4 El sistema debe ser capaz de almacenar la información sobre el uso del cinturón de seguridad en un formato que permita la trazabilidad de las mediciones.

3.1.2 Requisitos de interfaces

RI1 El dispositivo de muestreo, una vez inicializado, debe ser capaz de establecer conexión con el servidor.

RI2 El dispositivo de muestreo debe ser capaz de enviar las imágenes de los vehículos individuales al servidor.

3.2 Requisitos del ambiente

3.2.1 Descripción del Hardware de desarrollo

Para el desarrollo del dispositivo de muestreo, se requiere alguna placa de desarrollo a la que se le pueda conectar una cámara para obtener las imágenes. Este dispositivo debe ser capaz de conectarse a Internet y ejecutar programas de visión por computador.

3.2.2 Descripción del Software de desarrollo

Para el procesamiento de imágenes se utilizará la biblioteca de OpenCV. Considerando algunos factores, como bibliotecas a utilizar y el tiempo de desarrollo con que se cuenta, el lenguaje de programación a utilizar es Python, con Tornado como *framework* web y comunicación a través del protocolo de WebSockets con el servidor.

En el servidor se utiliza Ubuntu 14.04LTS (Trusty Tahr) como sistema operativo, con Nginx como servidor web y Tornado como servidor de aplicación,

tanto para servir la plataforma del usuario como para la comunicación con el dispositivo de muestreo.

3.3 Requisitos de usuario

Dentro de la metodología de trabajo como memorias multidisciplinarias, se considera también los requisitos que puedan tener los usuarios finales, quienes en este caso particular corresponden a miembros de CONASET. Para obtener los requisitos más relevantes para los usuarios, se realiza un ejercicio de priorización en conjunto con ellos.

Figura 3-1: Conjunto de requisitos presentados a los usuarios finales para su priorización.

Para este ejercicio, se confeccionan tarjetas de papel, cada una con un requisito de la Figura 3-1 y se les pide que las ordenen de mayor a menor relevancia sobre si el sistema debe poseer aquella característica. Paralelamente, los memoristas realizamos la misma actividad de priorización. Después de 5 minutos, tiempo límite para ordenar los requisitos, se pueden contrastar la visión distinta entre los memoristas y los miembros de CONASET, sobre la prioridad de las características. La Figura 3-2 muestra las 5 primeras opciones de cada listado.

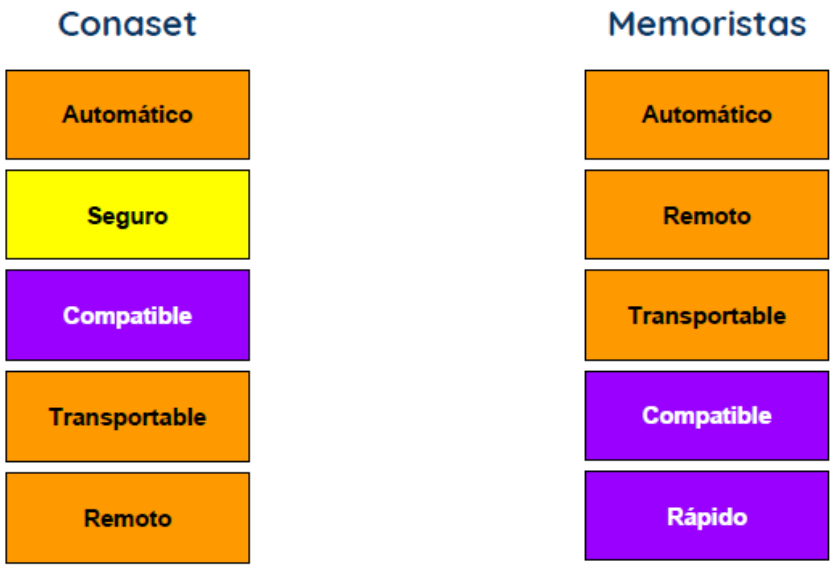


Figura 3-2: Contraste de priorización de requisitos.

A pesar de lo simple que parece este ejercicio, es relevante en el aspecto de diseño e implementación del sistema, puesto que permite mantener un enfoque cercano a lo que esperan los usuarios finales. Existen coincidencias entre las opciones escogidas por ambos grupos, pero el aspecto que el sistema sea controlado de forma remota, no es tan relevante para CONASET como nosotros lo habíamos considerado. En su lugar, prefieren que sea seguro, en el sentido que no pueda ser intervenido por agentes externos, tanto informáticamente como físicamente, con el robo del dispositivo, por ejemplo.

Capítulo 4

4 Análisis y Diseño

4.1 Arquitectura del Sistema

En la Figura 4-1 se muestra el principal flujo de información entre los distintos componentes del sistema.

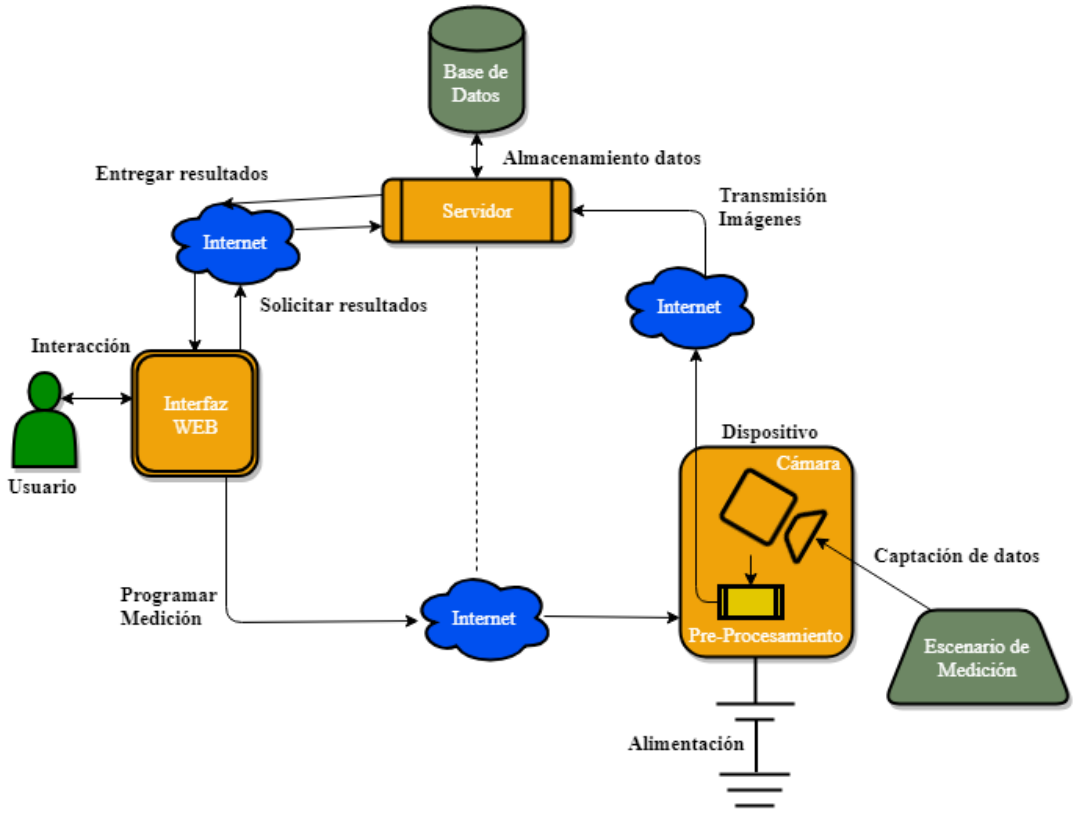


Figura 4-1: Infraestructura del sistema. Fuente: Elaboración propia.

4.2 Diagrama de flujos

En la Figura 4-2 se muestra el principal flujo de información y acciones que realiza tanto el usuario como el sistema para la obtención de resultados de medición.

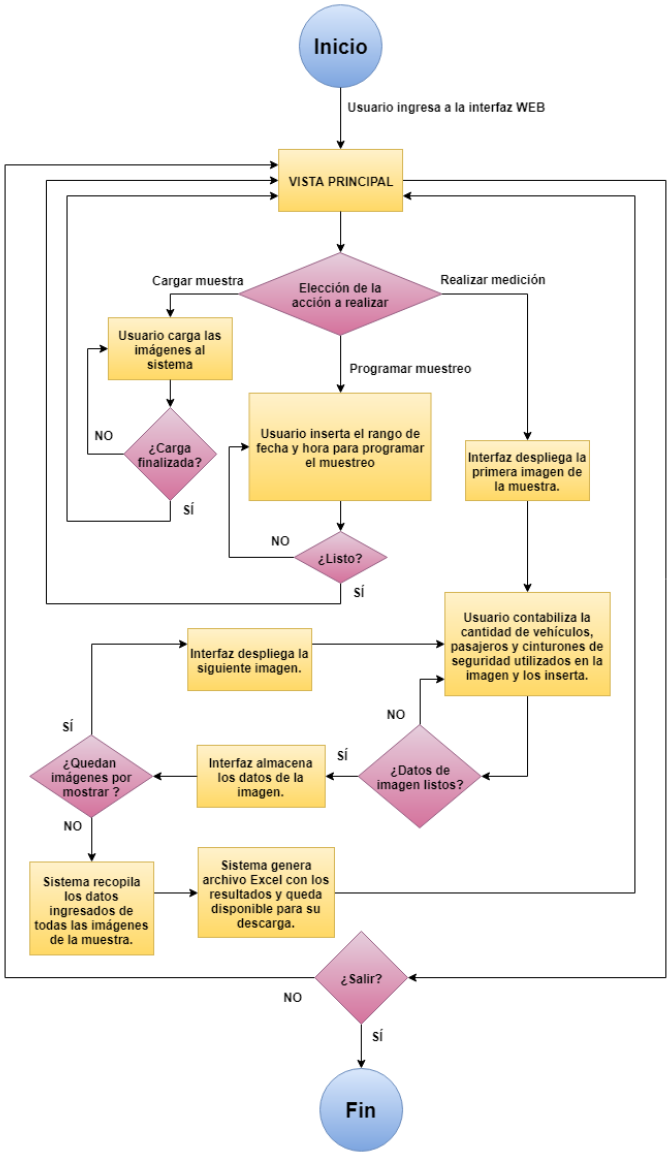


Figura 4-2: Diagrama de flujo en la obtención de resultados de medición.

Elaboración propia.

4.3 Descripción de componentes

El sistema en general puede verse compuesto por 3 componentes principales, contextualizados en el esquema de infraestructura de la Figura 4-1, en color anaranjado como son el Dispositivo, la Interfaz WEB y el Servidor. A continuación, se profundiza en algunos aspectos funcionales de cada uno y de los subcomponentes presentes.

4.3.1 Componente Dispositivo (CD)

Este componente, posicionado en terreno apuntando a la calzada, tiene la capacidad de obtener las imágenes de los vehículos que se encuentran transitando, realizar un preprocesamiento de ellas y entregarlas al Componente Servidor (CS).

4.3.2 Componente Interfaz WEB (CIW)

La función principal de este componente es interactuar con el usuario y ofrecerle de forma intuitiva y amigable, las distintas funcionalidades que el sistema posee.

4.3.3 Componente Servidor (CS)

En la infraestructura del sistema se observa como este componente es el encargado de interconectar al usuario y el CIW con el CD y el almacenamiento con la base de datos.

Capítulo 5

5 Gestión de Riesgos

En este capítulo se estudian los riesgos asociados al desarrollo del sistema de gestión de imágenes para la detección del uso del cinturón de seguridad. Además de las estrategias y mitigaciones para prevenir dichos riesgos.

5.1 Supuestos

1. Los usuarios del sistema poseen un nivel de conocimiento suficiente que les permita hace un correcto uso de las funcionalidades que ofrece el sistema.
2. Del lado del dispositivo de muestreo, se cuenta con una conexión a internet estable y que permita persistencia en la conexión con el servidor del sistema.

5.2 Dependencias

1. El dispositivo de muestreo debe apuntar a la calzada en un cierto ángulo dentro de un intervalo que permita visualizar el uso del cinturón de seguridad al interior de los vehículos que se encuentran en tránsito.
2. El dispositivo de muestreo debe mantener una posición estable (evitar en lo posible las vibraciones), con el fin de no obtener falsos positivos en la detección de movimiento en los algoritmos de sustracción de fondo.

5.3 Restricciones

1. Limitaciones en las horas de desarrollo.
2. Limitaciones en el presupuesto disponible para el desarrollo. Esto es relevante al momento de decidir acerca del hardware a utilizar para el dispositivo de muestreo.

5.4 Riesgos

5.4.1 Pérdida de conexión del dispositivo de muestreo con el servidor.

Bajo el supuesto de que existe una conexión estable a internet por parte del dispositivo de muestreo, aún hay probabilidades de que se pierda la conexión con el servidor. Además, depende del protocolo de comunicación que se utilice y del tiempo de *Timeout* si la conexión persiste o no.

5.4.2 Vulneración del sistema por ataques informáticos.

En todo sistema informático, existe una probabilidad de que sea vulnerado por ataques informáticos externos, ya sea para acceder a los datos privados como para dañar el funcionamiento del sistema.

5.4.3 Suplantación de identidad de un usuario del sistema.

En sistemas que poseen un proceso de identificación del usuario, existe el riesgo de que el usuario pueda ser suplantado por un agente externo y se realicen operaciones dentro del sistema en nombre de este usuario.

5.4.4 Robo o destrucción del dispositivo.

En la infraestructura del sistema, el dispositivo de muestreo se encuentra en terreno (en alguna pasarela o poste enfocando hacia la calzada) a la espera de que se programe algún muestreo y realizar la captura de las imágenes de los vehículos. Dado este escenario, existe la probabilidad que se robe o destruya el dispositivo por vandalismo.

5.5 Mitigaciones

5.5.1 Pérdida de conexión del dispositivo de muestreo con el servidor

Un factor que ayuda a mitigar este riesgo es el protocolo de comunicación utilizado. Para comunicar el dispositivo con el servidor del sistema se utiliza el protocolo WebSockets, que provee de una conexión persistente bidireccional. Es decir, luego de realizar el *handshake*, tanto el cliente como el servidor puede enviar mensajes a su contraparte cuando lo estimen conveniente. A diferencia de HTTP, cuando existe un intercambio de mensajes entre ambas partes en WebSockets, el canal de comunicación se mantiene abierto para futuros mensajes, por lo que la conexión persiste. En el caso de que la conexión se perdiera por desconexión a internet, la biblioteca de Python de WebSocket intentará reconectarse con el servidor hasta que exista conexión a internet.

5.5.2 Vulneración del sistema por ataques informáticos

En este aspecto existen múltiples opciones para intentar mitigar el riesgo, aunque ninguna es infalible. Con respecto al dispositivo de muestreo, una vez

que se enciende y conecta a internet, se conecta al servidor del sistema como cliente WebSocket y no establece un servidor web. Esto es importante debido a que un buen número de los ataques informáticos van dirigidos a buscar aquellos puertos que se encuentren escuchando, como el puerto 80 (HTTP) y puerto 22 (SSH). Con respecto a la comunicación, se realiza encriptada con el protocolo WSS (WebSocket Secure), por lo que un tercero no puede leer los mensajes que se transfieren entre dispositivo y servidor. El servicio de *hosting* en el que se encuentra alojado el servidor del sistema corresponde al portal DigitalOcean.com [7], que cuenta con infraestructura para prevenir ataques y mantener un nivel de disponibilidad del servicio aceptable. A la máquina virtual del servidor del sistema, se accede mediante una llave SSH privada.

5.5.3 Suplantación de identidad de un usuario del sistema

Una buena práctica en sistemas de autenticación e identificación del usuario, es la tercerización del proceso. En este caso, el usuario debe utilizar su cuenta de Google para tener acceso a la plataforma, delegando la seguridad de autenticar al usuario a dicha entidad.

5.5.4 Robo o destrucción del dispositivo

Para mitigar esto, se debe ubicar el dispositivo lejos del alcance de cualquier transeúnte, en la medida de lo posible. En el caso que el dispositivo sufra daños de terceros o, en su defecto, alguien lo robe, la solución implementada posee la característica que es de bajo costo, por lo que se reduce el costo económico de reponer el dispositivo.

Capítulo 6

6 Implementación del sistema

En la Figura 4-1 se presentó la infraestructura del sistema, donde existen 3 elementos principales: Interfaz WEB, Servidor y Dispositivo. En este capítulo se detalla la implementación de estos elementos y se habla acerca de su estructura modular.

6.1 Servidor de Aplicación en Tornado

Con el propósito de coordinar la comunicación entre los distintos participantes del sistema, se implementa un servidor con Tornado. Este *framework web* de Python entrega herramientas de conectividad asincrónica en su ciclo de ejecución, permitiendo hacer más eficiente la ejecución de ciertas tareas. Una de las ventajas que presenta esta tecnología es que utiliza operaciones no bloqueantes de entrada y salida, lo que permite escalabilidad de miles de conexiones abiertas de forma simultánea. Lo anterior representa un escenario ideal para implementarlo junto con protocolos de comunicación como WebSockets.

6.2 Accesibilidad del sistema

El sistema propuesto es implementado con el objetivo que sea utilizado por personal de CONASET desde cualquier dispositivo. Para ofrecer un acceso libre

al sistema, se monta la infraestructura en un *hosting* de la empresa DigitalOcean, permitiendo obtener una IP pública de la máquina.

6.3 Rutas y Handlers

La biblioteca *web* de Tornado tiene tres clases relevantes en la implementación de este proyecto: la clase ***Application***, la clase ***RequestHandler*** y la clase ***WebSocketHandler***.

La clase ***Application*** se utiliza para generar una instancia de todo el servidor de aplicación y de su configuración, siendo esta clase la encargada de gestionar las rutas para acceder al servicio. Las rutas son gestionadas por clases también, denominadas como clases manejadoras o "*handlers*". Estas clases *handlers* cumplen el rol de ejecutar acciones a partir de algún evento. Tornado cuenta con múltiples clases *handlers* de las que se puede heredar y reutilizar funcionalidades en la implementación del sistema. Los *handlers* utilizados en este proyecto basan su implementación en 2 *handlers* fundamentales: *RequestHandler* y *WebSocketHandler*.

La clase ***RequestHandler*** está implementada sobre el protocolo HTTP con operaciones del tipo REST, por lo que al implementar una clase que herede de *RequestHandler*, se pueden tomar acciones frente a mensajes recibidos por operaciones GET, POST, DELETE, PUT, PATCH entre otras. Esta clase también ofrece métodos de redireccionamiento, renderización de contenido web y gestión de *cookies* en el navegador.

La clase ***WebSocketHandler*** está implementada sobre el protocolo de comunicación *WebSockets* que utiliza conexiones persistentes. Esta clase ofrece métodos para gestionar la apertura y cierre de las conexiones *WebSockets*, la recepción y envío de mensajes, y verificación del origen de estos, entre otras funciones.

A partir de las clases anteriormente mencionadas, se observan 2 protocolos de comunicación que enlazan la interfaz WEB y el servidor de aplicación: El protocolo HTTP y el protocolo WebSocket. Para HTTP (*RequestHandler*), se implementa una clase base, llamada *BaseHandler*, que será utilizada de padre para el resto de los *handlers* HTTP. En esta clase se definen métodos comunes entre los *handlers* de comunicación HTTP, como la verificación de si la sesión del usuario que se conecta con el servidor se encuentra activa.

En la implementación del sistema se definen 4 rutas, cada una con su *handler*. Esta información es presentada en la Tabla 6-1.

Tabla 6-1: Rutas y handlers del servidor de aplicación.

Ruta	Clase Handler(Padre)	Descripción breve
Sin ruta	class BaseHandler(RequestHandler)	Corresponde a la clase base para la implementación de funciones comunes entre los *handlers* HTTP. Hereda de la clase *RequestHandler* e introduce validación de sesión de usuario, entre otras funciones.

/	class IndexHandler(BaseHandler)	Corresponde a la ruta raíz. Es el acceso principal al servidor y verifica que existe alguna sesión del usuario activa. De no ser así redirecciona a la página de *login*.
/oauth	class OauthHandler(BaseHandler)	Corresponde a la autenticación con el protocolo OAuth 2.0. La implementación permite una vía única de ingreso a través de una cuenta Google.
/logout	class LogoutHandler(BaseHandler)	Corresponde a la ejecución de un cierre de sesión. En esta función se elimina la *cookie* del navegador y se redirecciona a *login*.
/ws	class SocketHandler(WebSocketHandler)	Corresponde al canal de comunicación a través del protocolo de *WebSockets*. Esta clase gestiona las conexiones y los mensajes entre la interfaz WEB de los usuarios con el servidor de aplicación.

6.4 Autenticación en Google con Oauth 2.0

La autenticación del usuario en la interfaz Web es del tipo *Third-Party Authentication* o autenticación vía terceros. La tendencia tecnológica ha impulsado a delegar la autenticación de los usuarios a protocolos y plataformas que ya han demostrado confiabilidad en términos de seguridad, además de simplificar el ingreso desde el punto de vista del usuario, que ya no debe recordar múltiples cuentas de ingreso y puede utilizar las que ya posee. En este proyecto la autenticación y autorización para el ingreso es a través del protocolo OAuth2.0 con las cuentas de Google.

En la sección anterior se menciona el *handler* implementado para esta autenticación, que corresponde a la clase *OauthHandler*. Antes de la implementación, se debe contar con las credenciales de Google que permiten utilizar el protocolo OAuth2.0 en el desarrollo, obtenidas de forma gratuita ingresando con la cuenta de Google en la plataforma https://console.developers.google.com. Una vez dentro, se crea una organización y un proyecto, donde se especifica qué tipo de proyecto corresponde, siendo este caso un proyecto de tipo web. En esta sección se configura la URL de redirección autorizada a la que Google redirigirá al usuario una vez sea se verifique que las credenciales son correctas.

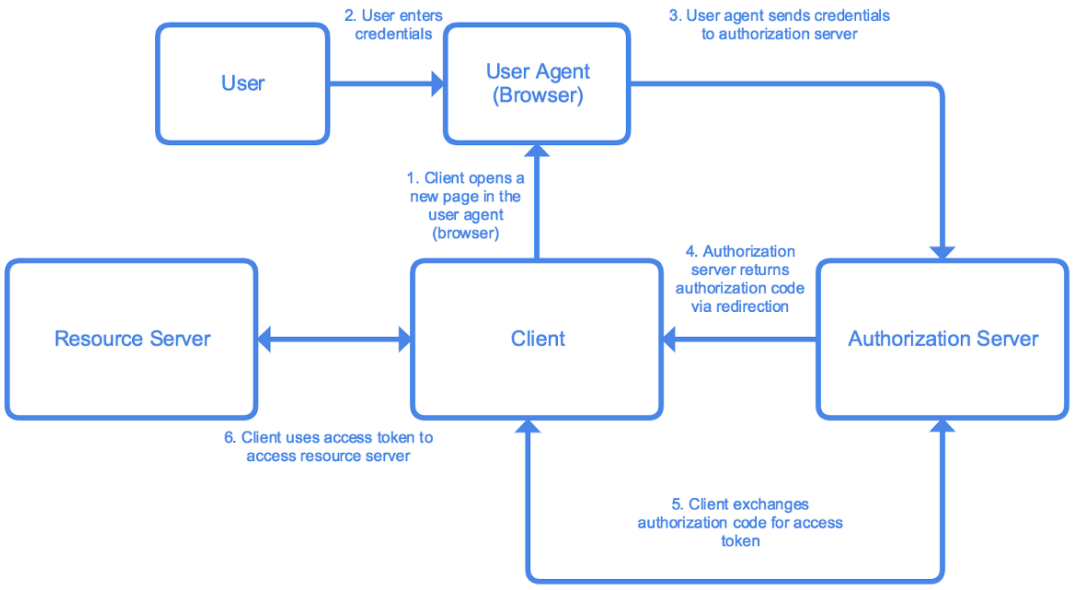


Figura 6-1: Flujo de trabajo en la autenticación con OAuth 2.0.

Al tener el proyecto registrado y configurado, se obtienen 2 parámetros fundamentales: el ID de cliente y el secreto de cliente. Estas cadenas de caracteres son utilizadas en el código de *OauthHandler*. En la Figura 6-1 se puede observar el flujo que sigue la autenticación con Google y todos los componentes y acciones involucradas.

El proceso de autenticación es fundamental en los paradigmas actuales de diseño de software, debido a los riesgos en la seguridad de los sistemas, potenciales ataques y suplantación de identidad. Por esta razón es que la tendencia corresponde a delegar estos procesos a entidades que ya cuentan con mayor experiencia en el área. Para el caso específico de este proyecto se utiliza a Google como entidad verificadora de la identidad del usuario que ingresa a la plataforma.

6.5 Estructura modular y WebSockets

Figura 6-2: Módulo base en el desarrollo del sistema.

La Figura 6-2 muestra el módulo base para todo el desarrollo de *software* en el sistema del presente proyecto. Este se basa en la programación por capas o niveles, dividiéndose en 3 partes: Capa de presentación (CP), Capa de lógica de negocio (CLN) y Capa de datos (CD). Esta estructura modular facilita el orden y mantención del código, ya que cada módulo gestiona sus propias capas.

La Capa de presentación es servida al Front-end en la Interfaz WEB para que el usuario pueda interactuar con el sistema. La Capa de negocio y Capa de datos se encuentran en el Back-end, principalmente en el Servidor, además de estar distribuidas en el lado del Dispositivo.

La comunicación entre las capas de presentación que están en el Front-end y la capa de lógica de negocios que se encuentra en el Back-end, se realiza principalmente con WebSockets, siguiendo el patrón Publicación – Suscripción. Cada módulo de funcionalidad presenta un inicializador, donde se instancian los objetos de las distintas clases que posee el módulo. Al iniciar el funcionamiento de la plataforma e instanciar el sistema, cada módulo subscribe

sus mensajes WebSockets en el *Router*. De esta forma, la capa de presentación de cada módulo envía desde el Front-end mensajes con el nombre del mensaje suscrito, para que sea recibido por el módulo y función respectiva de la capa de lógica de negocios.

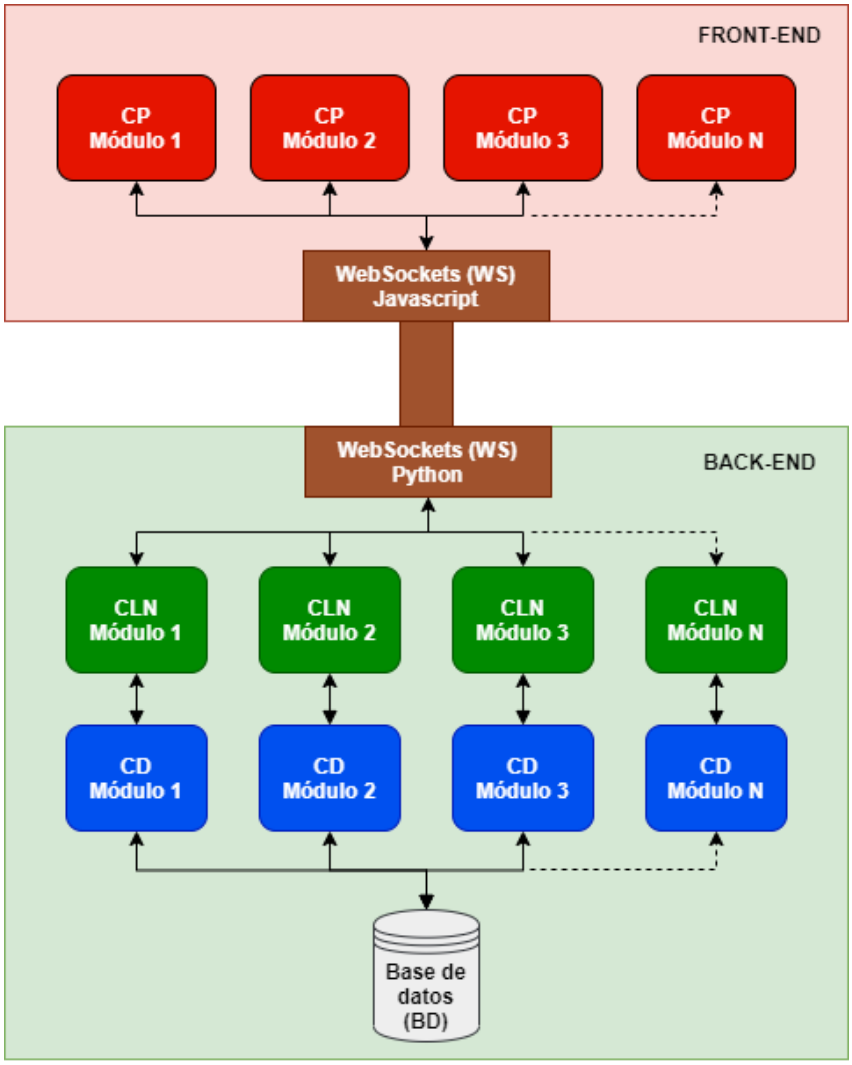


Figura 6-3: Canal de distribución de mensajes en patrón PubSub en WebSockets.

6.6 Implementación de módulos

En esta sección se describen las distintas estrategias utilizadas para abordar cada uno de los módulos de funcionalidad.

Antes de explicar cada uno de los módulos, se hace necesario entregar el contexto de la terminología que se va a utilizar.

6.6.1 Contexto y semántica

Como se ha explicado antes, este proyecto tiene por objetivo facilitar las mediciones observacionales del uso del cinturón de seguridad de vehículos en tránsito. La forma en cómo se estaban abordando estas mediciones correspondía a disponer de personas en terreno que observaran el uso del cinturón y registraran sus mediciones. El enfoque que presenta el actual proyecto es reemplazar a la persona en terreno por un dispositivo de muestreo. En este contexto, se entiende por una *muestra* a un conjunto de imágenes que contienen vehículos en tránsito. Este sistema ofrece 2 formas de cargar una muestra: desde el dispositivo de muestreo o que el usuario suba a la red este conjunto de imágenes

En el sistema propuesto existen 4 funcionalidades principales: operaciones CRUD de la programación de muestreos, operaciones CRUD con las muestras, operaciones CRUD de mediciones sobre las muestras y finalmente operaciones CRUD sobre los resultados de las mediciones. Cada una de estas

funcionalidades estarán expresadas como un módulo de funcionalidad, los que se muestran en la Figura 6-4.

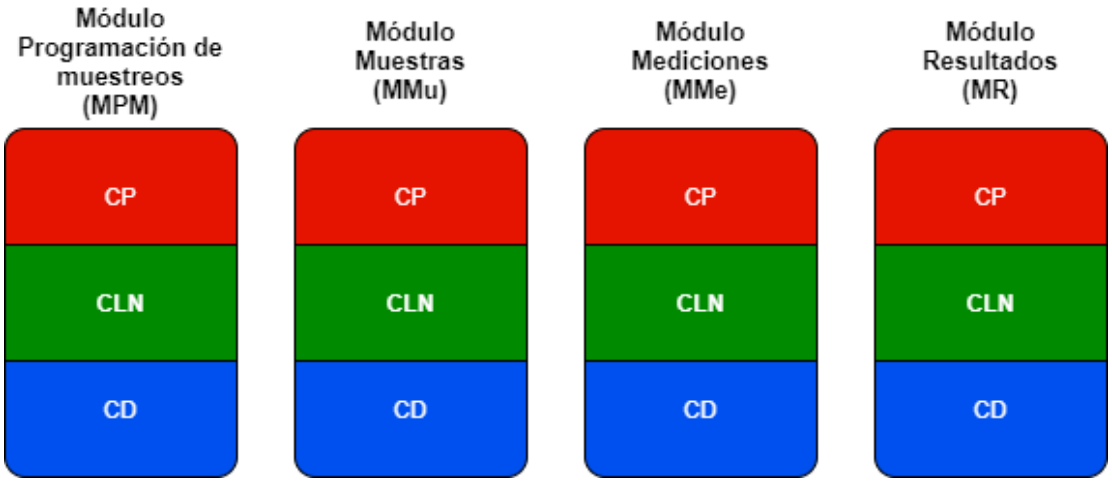


Figura 6-4: Esquema general de módulos de funcionalidad en la implementación

6.6.2 Módulo de Programación de muestreos (MPM)

El objetivo de este módulo es que el usuario pueda agendar la creación de una muestra. En el contexto de este sistema, una muestra corresponde a un conjunto de imágenes que contienen vehículos en tránsito, de la cual se pretende realizar la medición del uso del cinturón de seguridad en sus ocupantes.

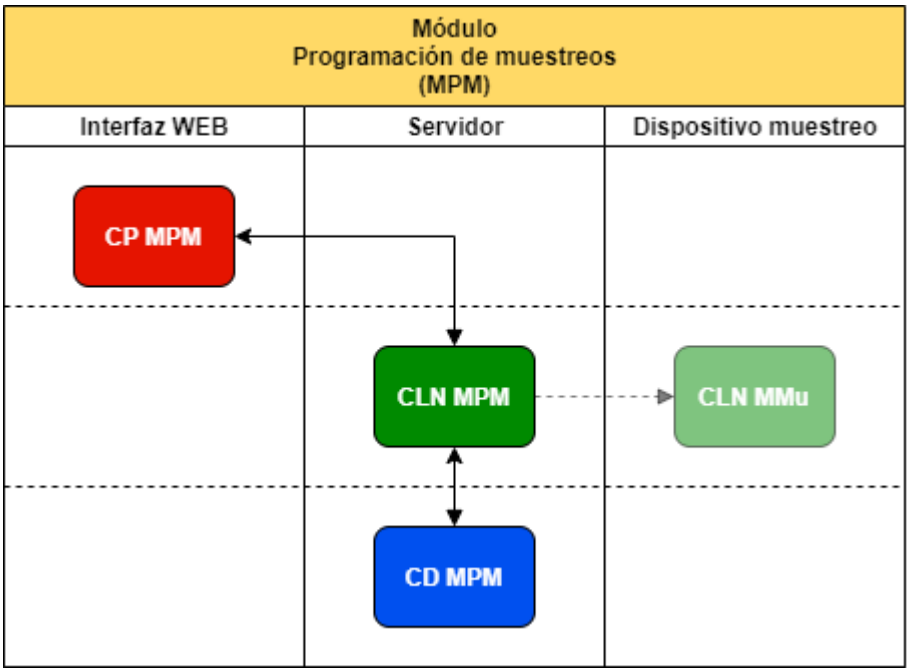


Figura 6-5: Distribución de las capas del módulo de programación de muestreos en los distintos componentes.

La Figura 6-5 muestra cómo se distribuyen las capas del módulo en los componentes del sistema. En la Interfaz WEB se encuentra la Capa de Presentación de este módulo, desde donde el usuario puede establecer los parámetros para agendar la creación de una muestra. Estos parámetros son principalmente la ubicación de medición, seleccionando alguna localización de uno de los dispositivos que puedan estar desplegados para realizar muestreos, junto con el rango de tiempo del muestreo.

Esta información es enviada a la Capa de negocio presente en el Servidor, siendo almacenada a través de la Capa de datos en la Base de Datos de MongoDB como un documento. Además de aquello, se envía un mensaje a una parte de la Capa de negocio del módulo de Muestras, presente en el dispositivo en terreno

para agendar la captura de imágenes en el rango de fechas propuesto por el usuario para la creación de la muestra.

El motivo para que estas funcionalidades fuesen parte de un módulo separado al módulo de Muestras es que de esta forma se puede tener una gestión más organizada en cuánto al agendamiento de creación de muestras, puesto que estas pueden ser eliminadas antes de llevarse a cabo, pueden actualizarse, entre otras operaciones. Haberlas posicionado junto con la gestión de las muestras en el módulo de Muestras habría complejizado el problema, que se puede mantener simple separando en módulos aparte.

6.6.3 Módulo de Muestras (MMu)

El objetivo de este módulo representa el desafío principal del proyecto, debido a que involucra el procesamiento de las imágenes para la detección de vehículos y generar la muestra de las imágenes.

Es el único de los módulos que sus capas están presentes en todos los componentes. La Capa de Presentación está presente en la interfaz WEB para que el usuario pueda gestionar las muestras ya creadas o crear una muestra a partir de imágenes que cargue desde su propio dispositivo local. La Capa de negocio y la Capa de datos se encuentran divididas cuyas partes se distribuyen entre el Servidor y el Dispositivo de muestreo, lo que se refleja en la Figura 6-6.

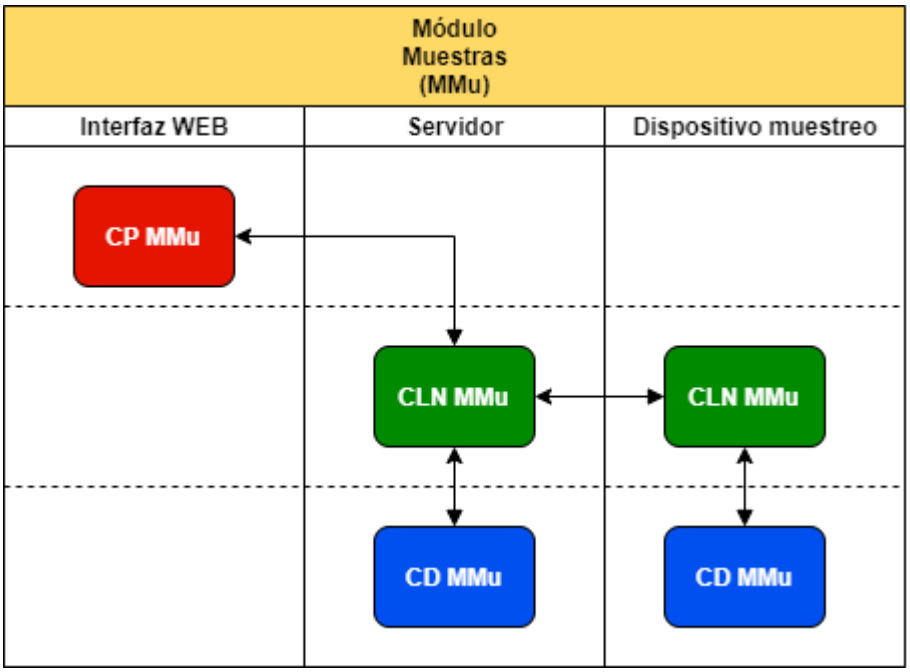


Figura 6-6: Distribución de las capas del módulo de muestras en los distintos componentes.

En la imagen de la sección anterior (Figura 6-5), se señala que existe una conexión que corresponde a un mensaje que es enviado desde la Capa de negocio del módulo de Programación de muestreos a la misma capa, pero del módulo de Muestras en el dispositivo. Este mensaje contiene el rango de tiempo en que se pretende medir, lo que incluye fecha y hora de inicio y fin, expresadas según la norma ISO 8601 (YYYY-MM-DDThh:mm:ss.sssZ) [8].

El dispositivo de muestreo, cuenta con una parte de la Capa de negocio y una parte de la Capa de Datos del módulo de muestras. Esto supone que, tal como se ha explicado, tiene la capacidad de comunicarse por WebSockets con su contraparte de Capa de Negocio presente en el Servidor para enviar los datos sobre las imágenes capturadas en terreno sobre vehículo en tránsito.

El dispositivo corresponde a una Raspberry Pi 3 Model B, ubicada en terreno para realizar la captura de imágenes. En esta tarjeta se encuentra en ejecución un proceso Python con la Capa de Negocio del módulo de Muestras, que espera la llegada de algún mensaje WebSocket por parte de la Capa de Negocio del módulo de programación de muestreo. Con la llegada de un mensaje, se agenda una captura de imágenes para el rango de tiempo que contempla la muestra a generar, a través de la biblioteca *Crontab* de Python. La tarea agendada a ejecutar por *Crontab* corresponde al algoritmo de procesamiento de imágenes, el cual es explicado a continuación.

6.6.3.1 Algoritmo de procesamiento de imágenes

Este algoritmo realiza todo lo relativo con la detección de vehículos en tránsito y la captura de las imágenes para la generación de una muestra.

Este proceso lo ejecuta la Capa de Negocio presente en el dispositivo de muestreo en el momento en que se inicia el rango de tiempo de muestreo. Para realizar una captura efectiva de los vehículos de tal manera que se pueda distinguir el uso del cinturón de seguridad observado por un ojo humano, el dispositivo debe estar ubicado de tal forma que la cámara apunte a la calzada con un ángulo favorable para observar a los ocupantes del vehículo que está pasando por el lugar.

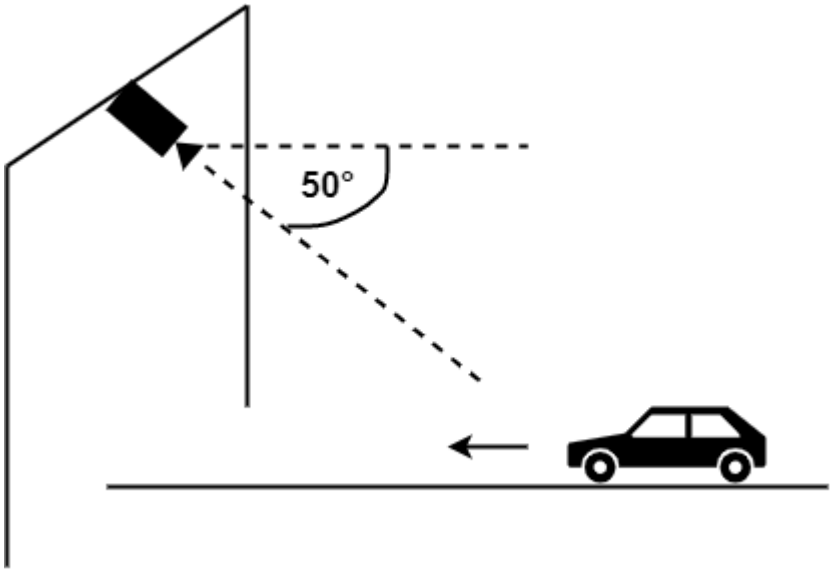


Figura 6-7: Inclinación utilizada de la cámara presente en el dispositivo de muestreo con respecto a la calzada.

Una vez ubicada la cámara, se configura en la imagen las regiones de interés (ROI, *Region of interest*), correspondiente a las zonas de la imagen que serán procesadas por el algoritmo. En el ejercicio de este proyecto, el dispositivo se ubica en la pasarela frente a la Universidad Técnica Federico Santa María, apuntando hacia la calzada de Av. España en Valparaíso, por lo que las pistas vehiculares disponibles en un mismo sentido son 3, donde se ubicada cada ROI.

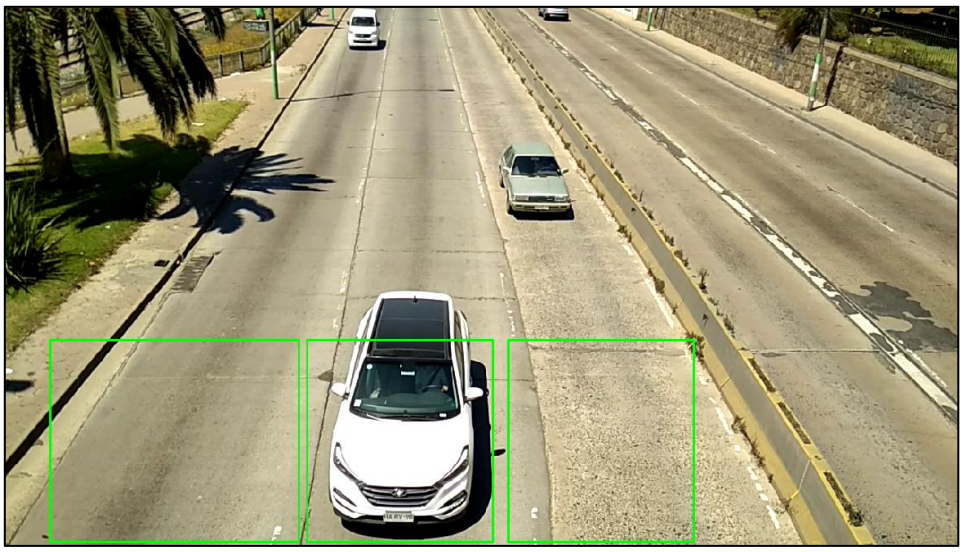


Figura 6-8: Disposición de cada ROI en cada pista de la calzada a medir.

Como se muestra en la Figura 6-8, cada ROI (rectángulos con marco de color verde) está ubicada en la sección inferior de la imagen de cada pista, debido a que espacialmente es la zona más cercana a la cámara y con una mejor probabilidad de observación sobre si los ocupantes están utilizando el cinturón de seguridad.

Para llegar a obtener las imágenes de cada vehículo que transita por la calzada se requiere detectar el movimiento de estos. A medida que la cámara va capturando los cuadros de video, en cada ROI se realiza una sustracción de fondo utilizando el método de mezcla de Gaussianas que ofrece la biblioteca de OpenCV en Python. Específicamente se utiliza el método MOG2, que ofrece ventajas comparativas frente a otros métodos. Este método se va adaptando a medida que transcurre el tiempo respondiendo bien ante cambios de iluminación en el ambiente. Es el segundo mejor método, justo detrás de KNN (*K-Nearest Neighbors*) [9]. Se ha utilizado MOG2 porque es más rápido que KNN,

característica importante si se considera que el dispositivo funcionaría procesando imágenes en tiempo real, por lo que el resultado que entrega MOG2 en la segmentación es suficiente para cumplir el objetivo de detección de vehículos.

Antes de obtener la imagen de sustracción de fondo, cada ROI debe pasar por un breve procesamiento.

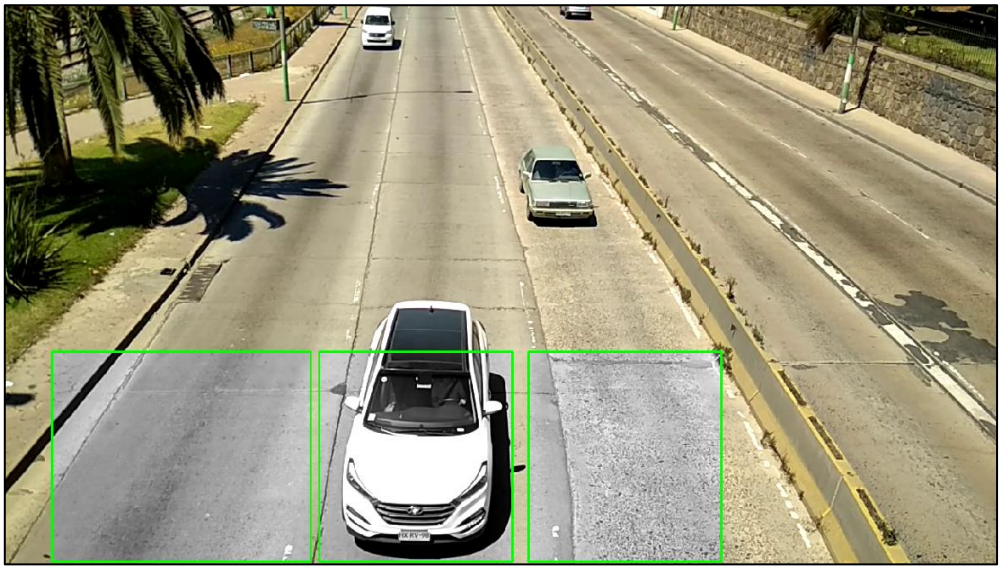


Figura 6-9: Cada ROI en una escala de grises.

La Figura 6-9 corresponde al primer paso. Cada una de las ROI sufre una transformación de colores a una escala de grises. Esto permite que cada píxel se represente por un valor en un único canal, ofreciendo alternativas para operar sobre sus valores, cómo por ejemplo comparar las diferencias del canal con los de un cuadro anterior, que es justamente en lo que consiste técnica de sustracción de fondo.

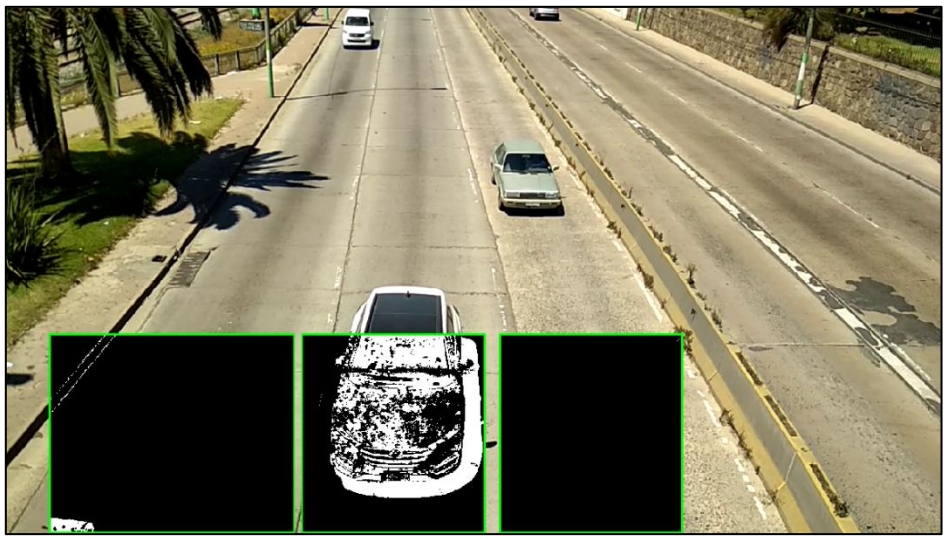


Figura 6-10: Aplicación de sustracción de fondo con el método MOG2 en cada una de las ROI.

La Figura 6-10 refleja el resultado de aplicar la técnica de sustracción de fondo a través del método MOG2, donde se observa imágenes binarias en cada una de las ROI. Las diferencias entre cuadros de video son comparadas con un umbral establecido en el programa. Los píxeles cuya diferencia superen este umbral, son marcos con color blanco y considerados como píxeles "en movimiento". Por otro lado, aquellos píxeles que no superen este umbral, son considerados "estáticos" y parte del fondo de la imagen.

El método de sustracción de fondo puede entregar píxeles considerados "en movimiento" que realmente no corresponden a lo que queremos capturar, incluso dentro de las regiones de interés. Estos píxeles de color blanco pueden ser resultado de ruido blanco presente en el sistema o por vibraciones en la cámara, como es probable que esté ocurriendo con los píxeles que se encuentran

en la ROI de la izquierda en la Figura 6-10, marcando con píxeles blancos el borde de la calzada.

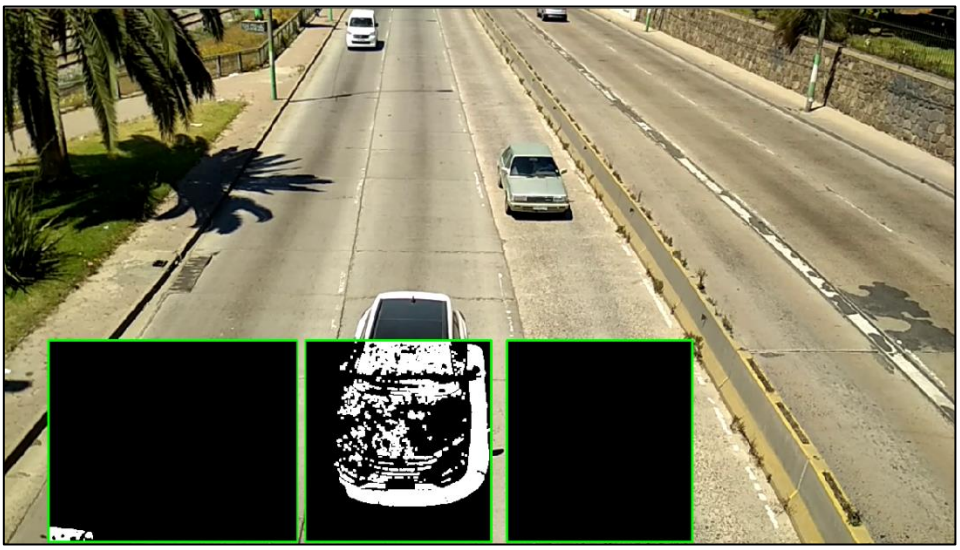


Figura 6-11: Operación de apertura en cada una de las ROI con elemento estructurante cuadrado de 3x3.

Para filtrar aquellos píxeles producidos por vibraciones leves o por ruido blanco, se utiliza en este proyecto una operación de *opening* (apertura) con un elemento estructurante cuadrado de 3x3 píxeles. En este proceso se produce una erosión y posterior dilatación morfológica de los píxeles en blanco dentro de las ROI. De esta forma, aquellas áreas de color blanco que no sean significativas, simplemente desaparecen. Lo anterior es justamente lo que se puede observar en la Figura 6-11 donde aquellos píxeles blancos que señalaban el borde de la calzada en el ROI de la izquierda en la Figura 6-10, son eliminados como resultado de este proceso.

Posteriormente al *opening* se realiza un *closing* (cierre), que tal y cómo se infiere de los nombres, corresponde al proceso morfológico inverso, realizándose

primero una dilatación de los píxeles, seguido de una erosión. La diferencia con el proceso anterior es que en esta ocasión se utiliza un elemento estructurante cuadrado de 11x11 píxeles. Lo que se obtiene es un rellenado de los espacios que existen entre las regiones blancas.

En la Figura 6-12 se muestra el resultado de esta operación. Nótese que la cantidad de regiones blancas disminuyen considerablemente, lo que permite resaltar la región donde se encuentra el vehículo en realidad.

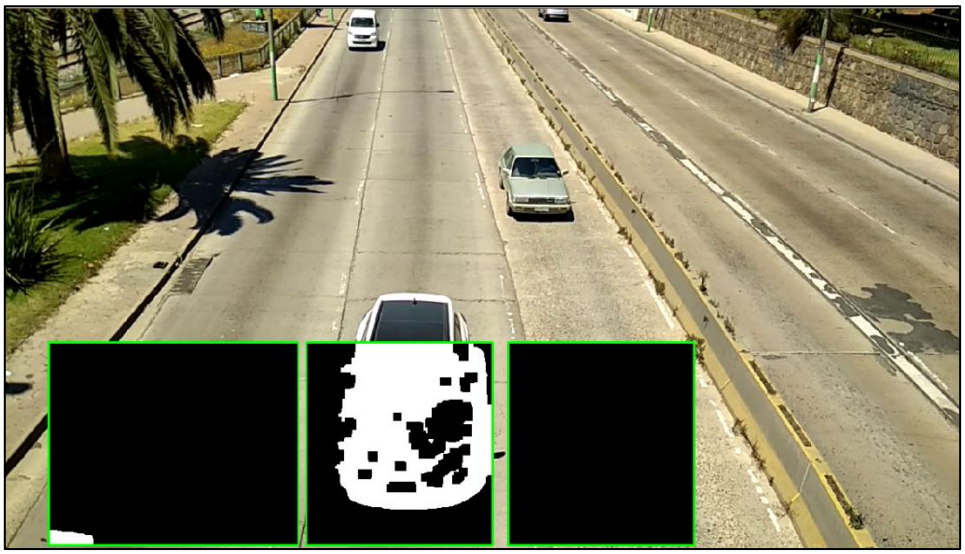


Figura 6-12: Operación de cierre (closing) con elemento estructurante de 11x11 píxeles.

El objetivo central de este algoritmo es hacer una captura del vehículo. El criterio que se utiliza para hacer la captura será la ubicación del centroide de la región blanca, que corresponde a la detección del vehículo en movimiento. Por otro lado, se pretende obtener imágenes de vehículos livianos en su mayoría, por lo que no sirve cualquier región blanca. Según el contexto propio de la

medición realizada en estas imágenes, se determina de forma arbitraria que el área mínima para considerar una captura será de 10.000 píxeles.

$$Area_{mín} = 10000\ [píxeles] \quad (1)$$

Obtener información sobre el contorno de una región ayuda a determinar tanto el área como el centroide. Para realizar esto se ocupa el método de la biblioteca OpenCV de Python, *cv2.findContours.* Un contorno en este escenario, supone la separación entre píxeles de color blanco y los de color negro, por lo que las regiones negras al interior de la región blanca también tienen contornos. El método *cv2.findContours* entrega una lista con estructuras que contienen las coordenadas del contorno de cada región en la imagen. Esto se hizo para cada una de las ROI.

Existe otro método de OpenCV, *cv2.contourArea* [10], que permite calcular el área de una región basado en las coordenadas del contorno. Con esto se procede a calcular las áreas de cada uno de los contornos que entrega el método anterior y utilizar el área mayor en la evaluación del criterio en la obtención de la captura de imagen.

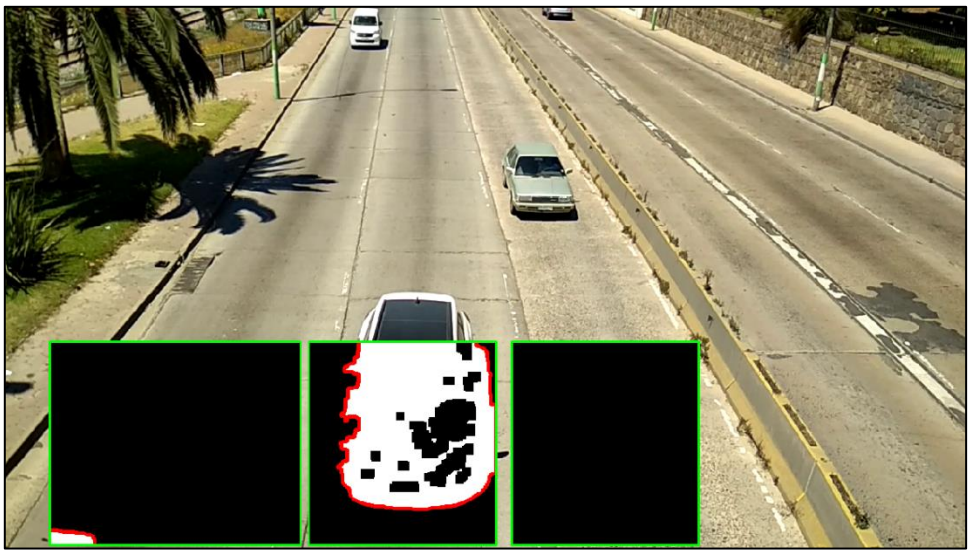


Figura 6-13: Contorno mayor existente en las regiones de cada ROI (contorno de color rojo).

En la Figura 6-13 se dibuja el contorno mayor encontrado en cada una de las ROI, el cual será utilizado para encontrar el centroide de la región.

Las irregularidades que pueda presentar la región en sus bordes podrían afectar el eventual cálculo del centroide. Para mejorar esto, se aplica lo que se conoce como una envoltura convexa (Convex Hull) en el contorno. La Figura 6-14 muestra la aplicación del método de OpenCV en Python para la envoltura convexa *cv2.convexHull* [11]. La envoltura convexa en ciertos casos puede entregar el mismo contorno del área y en otros casos, envuelve aquellos bordes que presentan entradas, correspondiente a lo que sucede en el ROI central de la Figura 6-14 en el borde izquierdo de la región blanca.

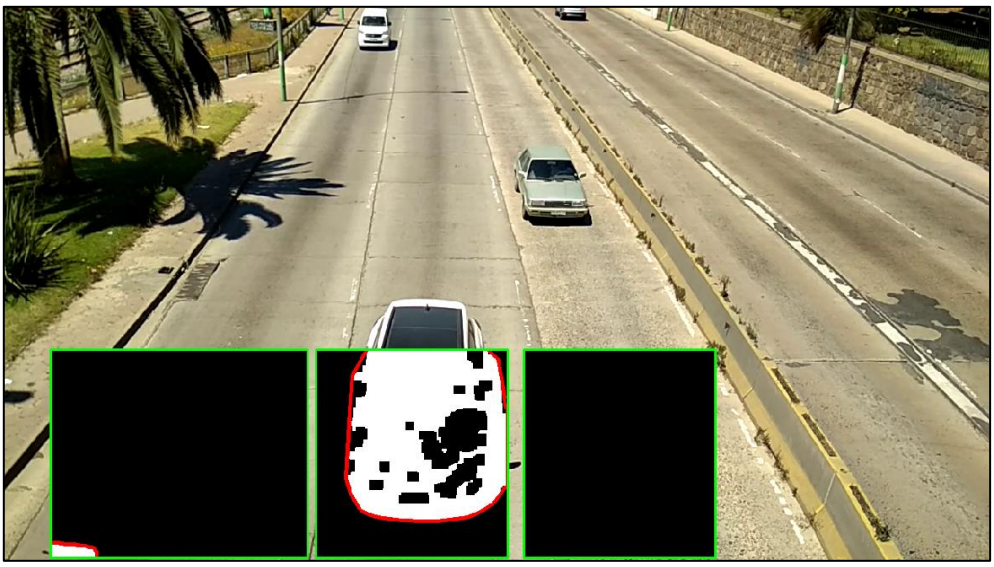


Figura 6-14: Envoltura convexa (Convex Hull) en las regiones mayores en las ROI (contorno de color rojo).

Uno de los métodos matemáticos para encontrar el centroide de una región es a través de la obtención de los momentos geométricos [12]. Los momentos geométricos proporcionan información útil para la representación de formas de objetos. OpenCV tiene un método para poder calcular los distintos momentos geométricos espaciales con los que cuenta una imagen (*cv2.moments()*) [13]. Si el objeto M almacena el resultado de este método en Python, el cálculo del centroide está dado por la expresión 2.

$$Centroide = (X_c, Y_c) = \left(redon\left(\frac{M['m10']}{M['m00']}\right), redon\left(\frac{M['m01']}{M['m00']}\right)\right) \quad (2)$$

En esta expresión 2, X_c y Y_c corresponden a las coordenadas del centroide en los ejes x e y respectivamente en la imagen. La función $redon(x)$ redondea el valor a un entero.

Se realiza el respectivo cálculo de los momentos geométricos a partir de cada envoltura convexa presente en las ROI. Posterior a esto se procede al cálculo del centroide en cada ROI utilizando la fórmula en la expresión 2. El resultado de este proceso puede ser visualizado en la Figura 6-15, donde cada centroide es dibujado como un círculo de color amarillo.

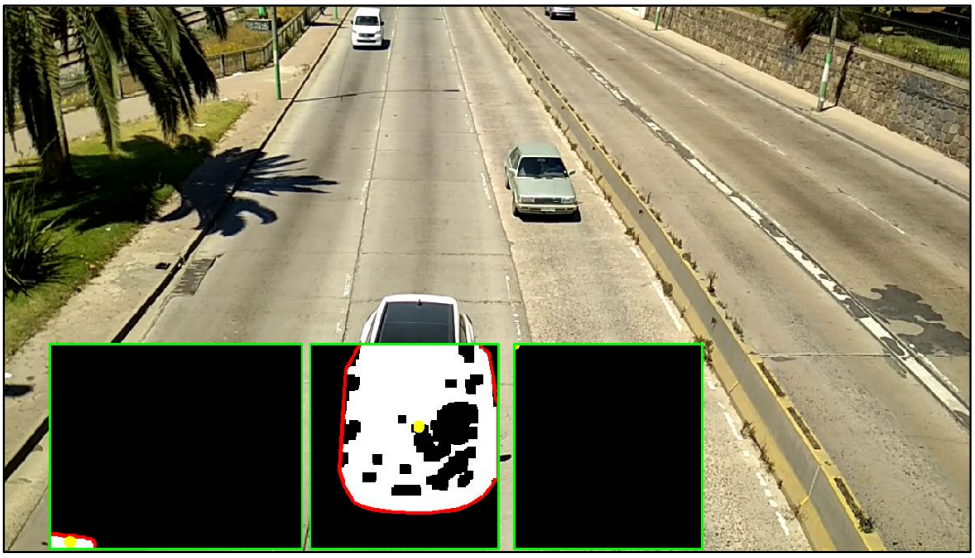


Figura 6-15: Centroide del contorno mayor de cada ROI (punto de color amarillo).

Como el objetivo final de este algoritmo es conseguir una captura del vehículo, en especial una captura del parabrisas para observar si se utiliza el cinturón de seguridad, la disposición del centroide en la imagen es de utilidad para tomar decisiones con respecto a su posición.

El criterio de decisión sobre la captura de la imagen en este caso fue diseñado a partir de una cuadrícula en el ROI. Se divide cada ROI en 3 columnas de iguales dimensiones y 2 filas. La fila superior contiene 1/3 de la altitud del ROI

y la segunda fila contiene los 2/3 restantes. La Figura 6-16 muestra las cuadrículas en cada una de las ROI, delineadas en color violeta.

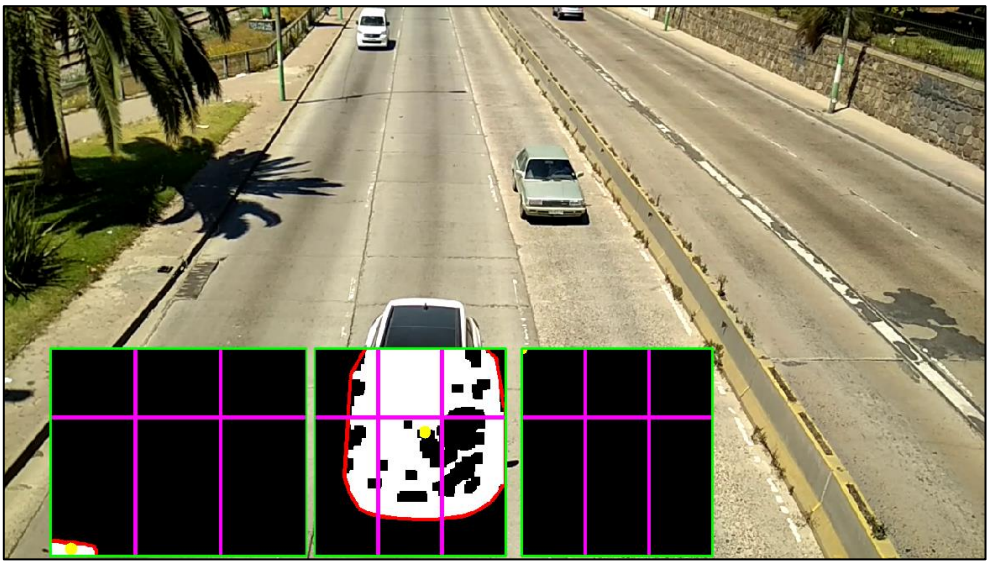


Figura 6-16: Cuadrícula para seleccionar el momento de la captura de imagen.

Finalmente, la captura de la imagen del vehículo a partir de la información obtenida hasta ahora se realiza con el cumplimiento de dos condiciones:

1. El área de la región en movimiento debe ser superior al valor establecido en la expresión 1 de 10.000 [píxeles].
2. El centroide de la región en movimiento debe estar posicionado al interior de la sección central-inferior de la cuadrícula.

Ambas condiciones se cumplen en el instante que muestra la Figura 6-16 en la ROI central, captura que corresponde a la imagen de la Figura 6-17. Para mejorar la captura, se transforma la imagen al modelo de color HSV y se realiza

un leve aumento del brillo para mejorar la visualización del cinturón de seguridad. [14]

Figura 6-17: Imagen capturada al cumplirse los criterios establecidos.

Una vez realizada la captura, los cuadros siguientes de video siguen cumpliendo las 2 condiciones definidas. Para evitar las múltiples capturas de un mismo vehículo, se establece un parámetro de cantidad de cuadros en que se ignora el cumplimiento de las condiciones de captura y no se realiza. Este parámetro depende de la tasa de cuadros por segundo con que se realiza la captura de imágenes, que a su vez debe ir acorde a las velocidades alcanzadas por los vehículos.

6.6.4 Módulo de Mediciones (MMe)

El objetivo principal de este módulo es disponibilizar las capturas de vehículos obtenidas por el Módulo de Muestras.

La Figura 6-18 muestra la distribución de las distintas capas de este módulo a través de los componentes presentes en el sistema.

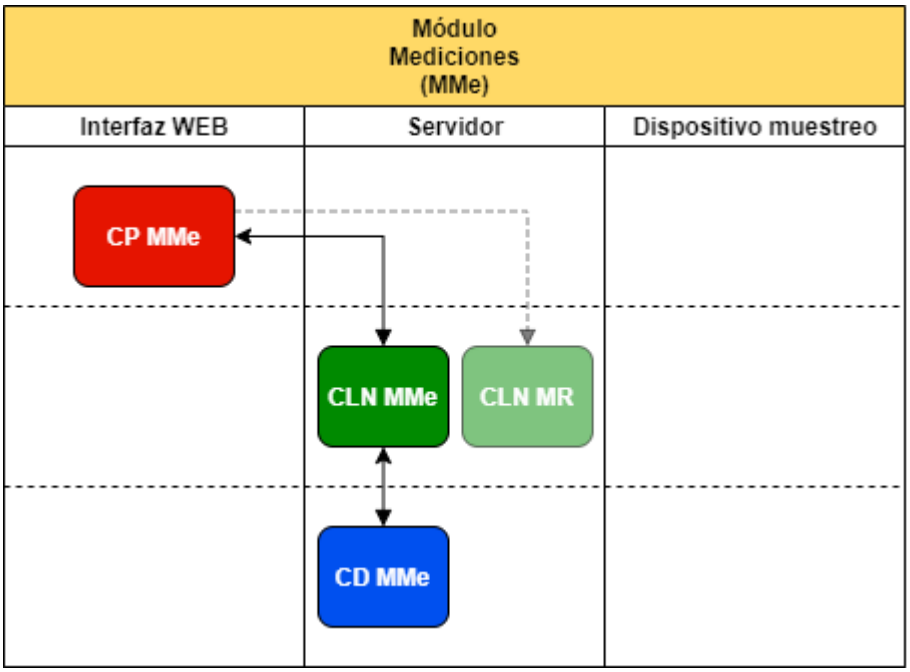


Figura 6-18: Distribución de las capas del módulo de mediciones en los distintos componentes.

EL usuario de sistema interactúa con este módulo a través de la capa de presentación en el componente de interfaz web. Cuando se solicita realizar una medición a una muestra específica, la capa de presentación se comunica con la capa de lógica de negocio en el servidor, quien gestiona la consulta a la capa de datos para obtener las imágenes de la muestra y son desplegadas en la interfaz web.

La imagen de la Figura 6-19 muestra el Módulo de Muestras donde el usuario del sistema puede ver los datos de las distintas muestras que ya ha realizado, tanto por cargar manualmente imágenes, así como las muestras que se hayan

obtenido desde el dispositivo de muestreo, donde aplica el algoritmo de procesamiento de imágenes explicado en la sección anterior.

Figura 6-19: Módulo de muestras en la interfaz web desde donde se puede ingresar a realizar una medición.

Antes de realizar la medición respectiva, el usuario debe antes indicar las características a medir, información que será almacenada junto con los resultados de la medición. Esta selección es la que se muestra en la Figura 6-20. El conjunto base de datos que el usuario puede seleccionar son tres: cantidad de vehículos que se observan en la imagen; cantidad de pasajeros que se observan los vehículos de la imagen; cantidad de cinturones de seguridad utilizados que se observan en la imagen.

Figura 6-20: Selección de características a medir por parte del usuario.

Al empezar la medición, aparece en la interfaz cada una de las imágenes de la muestra redimensionadas a un tamaño que permita una mejor visualización para el observador en lado izquierdo de la interfaz, mientras que en el lado derecho se encuentra un formulario donde el usuario completa los datos observados en la imagen según las características que se está midiendo, como se puede apreciar en la Figura 6-21.

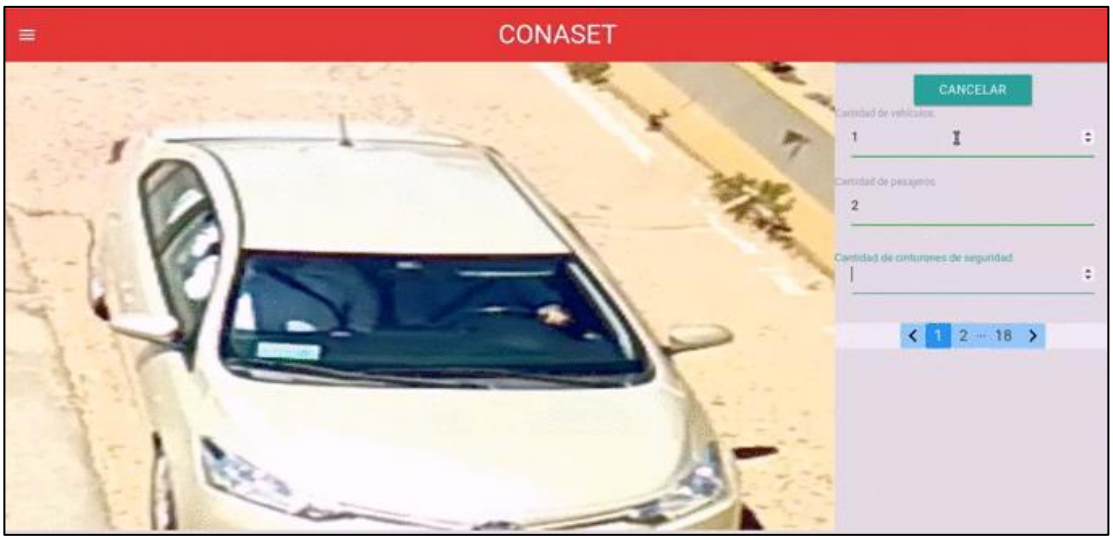


Figura 6-21: Interfaz de medición donde el usuario completa los datos a partir de la observación de la imagen.

Una vez que el usuario realizar la medición en cada una de las imágenes de la muestra, puede dar finalizada la medición, entregando la información a la capa de lógica de negocio del Módulo de Resultados.

6.6.5 Módulo de Resultados (MR)

El objetivo de este módulo es habilitar la gestión de los resultados obtenidos en las mediciones realizadas por el usuario.

La distribución de sus distintas capas a través de los componentes del sistema se puede observar en la Figura 6-22.

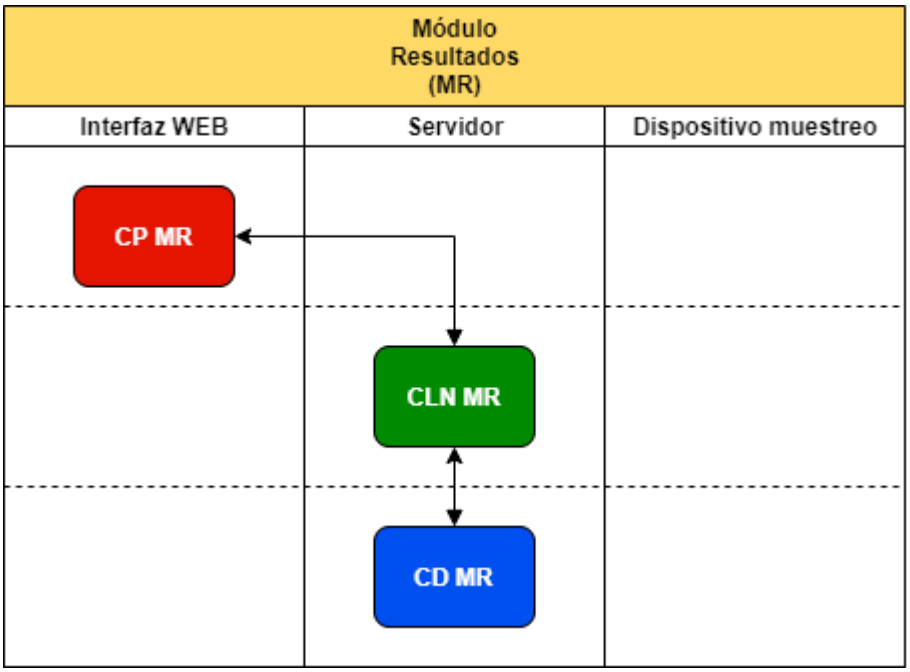


Figura 6-22: Distribución de las capas del módulo de resultados en los distintos componentes.

En la interfaz web, la capa de presentación disponibiza al usuario un listado con todos los resultados que ha obtenido y le entrega la posibilidad de descargarlos en formato Excel.

Estos archivos Excel con los resultados contienen información sobre la fecha y hora de la muestra, el lugar donde se obtuvo la captura de las imágenes de la muestra, las características medidas y los datos de cada medición que son ingresados por el usuario. Esta información entrega la posibilidad de realizar comparaciones entre distintos resultados con características similares de medición y generar trazabilidad del uso del cinturón de seguridad por parte de los pasajeros de los vehículos.

Capítulo 7

7 Prueba de concepto

En este capítulo se explica la implementación del sistema como una prueba de concepto y el procedimiento de la realización de una prueba en terreno para validar su funcionamiento.

7.1 Implementaciones faltantes

El alcance principal de la implementación realizada en este proyecto corresponde a una prueba de concepto del sistema. Para cumplir con aquello en los tiempos establecidos del trabajo y alcanzar a realizar un *testing* de validación, es que existen ciertos módulos del capítulo "Implementación de módulos" que no se desarrollaron de forma completa, sino que lograron un estado de avance suficiente para realizar la prueba de concepto.

Principalmente la implementación faltante en el proyecto consiste en la interconexión de los componentes del sistema. A continuación, se expone una breve explicación de aquellos elementos en los componentes y módulos que se encuentran incompletos.

7.1.1 Conexión entre servidor y dispositivo de muestreo

El sistema consta de 3 componentes principales: interfaz web, servidor y dispositivo de muestreo. En las explicaciones de capítulos anteriores se describen mensajes entre las capas de los módulos distribuidos en el servidor y

el dispositivo de muestreo. En la implementación de la prueba de concepto, el servidor y el dispositivo de muestreo no se conectan entre sí y se someten a validaciones de funcionamiento por separado.

7.1.2 Módulo de programación de muestreos

A causa de que en realidad el servidor y el dispositivo de muestreo estarán desconectados entre sí para la prueba de concepto, se omite la implementación de este módulo.

7.2 *Testing*

La metodología utilizada para la validación del funcionamiento en un contexto de prueba de concepto del sistema fue basada en dividir el sistema en 2 etapas: prueba del dispositivo de muestreo en terreno y prueba de la interfaz web en conjunto con el servidor para la gestión de la muestra y medición.

7.2.1 Dispositivo de muestreo en terreno

Para realizar la prueba del dispositivo del muestreo, se utiliza la pasarela de Av. España frente a la Universidad Técnica Federico Santa María en Valparaíso. Como se muestra en la Figura 7-1, la cámara del dispositivo (Raspberry Pi 3 Modelo B), se encuentra con una inclinación hacia la calzada para realizar las capturas.

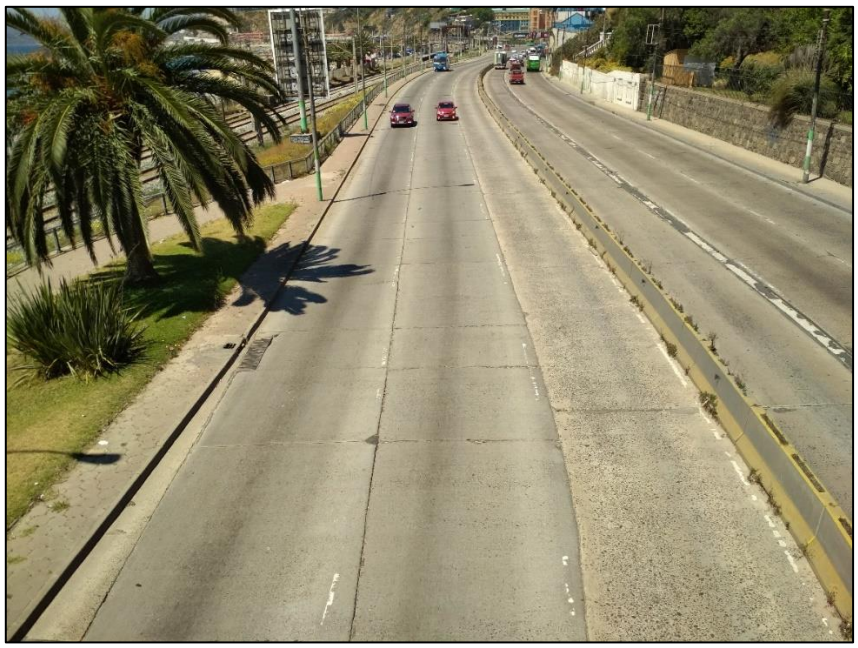


Figura 7-1: Visual del dispositivo de muestreo hacia la calzada de Av. España, Valparaíso.

La prueba del dispositivo consiste en que la cámara capte un breve video con circulación regular de vehículos y lo procese, obteniendo las capturas de los vehículos que transitan, utilizando el algoritmo descrito en el capítulo 6 sobre procesamiento de imagen. La visual que tiene el dispositivo de la calzada inclinada en un ángulo de 50° con respecto a la horizontal puede verse en la imagen de la Figura 7-1.

La prueba se considera exitosa si las imágenes almacenadas posterior al procesamiento corresponden efectivamente a los vehículos y son adecuadas para su posterior medición observacional.

7.2.2 Interfaz web y servidor

Estos componentes se pueden probar por separado independiente de la obtención de las imágenes capturadas por el dispositivo de muestreo. El módulo de muestras permite la creación de una muestra de forma manual, cargando imágenes al sistema desde cualquier dispositivo con acceso a un navegador web.

Una vez cargada las imágenes de una muestra, la prueba de concepto en este caso consiste en realizar una medición observacional de una muestra, ingresando las características medidas de la imagen.

La prueba se considera exitosa si la medición es almacenada correctamente y el usuario puede extraer la información en formato Excel.

7.3 Resultados del *testing*

Figura 7-2: Muestra de las 50 capturas obtenidas desde el dispositivo de muestreo.

En el *testing* realizado para el dispositivo de muestreo, se procesaron imágenes de un video breve con una duración de 70 segundos. En ese intervalo de tiempo se logró la captura efectiva de 50 vehículos de un total de 71 que circularon en ese instante.

La Figura 7-2 refleja los 50 vehículos capturados en el *testing*. De este resultado se desprende la siguiente información en la Tabla 7-1.

Tabla 7-1: Datos obtenidos en la prueba del dispositivo de muestreo.

Descripción dato	**Valor**
Duración total en segundos de la secuencia de *testing*	70 segundos

Cantidad total de imágenes capturadas por el dispositivo	50 imágenes
Cantidad total de vehículos capturados por el dispositivo	50 vehículos
Cantidad total de parabrisas capturados completamente	47 parabrisas
Cantidad total de parabrisas donde es posible realizar la medición observacional del uso del cinturón de seguridad	35 parabrisas

Además de los datos en la tabla, hay información anexa que puede ser útil a la hora de evaluar las mejoras al sistema:

1. De los 15 parabrisas que no se puede realizar la medición, 3 de ellos son vehículos de mayor tamaño que un vehículo liviano. Aquellos 3 mismos vehículos fueron captados en el ROI de la izquierda del procesamiento y el parabrisas no logra verse completo en el encuadre de la imagen.
2. De los 15 parabrisas que no se puede realizar la medición, 11 de ellos corresponden a parabrisas donde se refleja directamente la luz del sol y se encuentran ubicados en el ROI central en la imagen.

Con respecto al proceso de *testing* en la interfaz web y servidor, se utilizaron las 50 imágenes obtenidas en la prueba del dispositivo, las que fueron cargadas de forma manual. No hubo inconvenientes en realizar las mediciones correspondientes en cada imagen, almacenándose los datos en el servidor y posteriormente quedando disponible para descarga el Excel con los resultados.

Capítulo 8

8 Posibles mejoras

En este capítulo se describen algunas mejoras que se pueden introducir al sistema expuesto para mejorar su rendimiento, a partir de los resultados obtenidos en la fase de pruebas capítulo anterior.

8.1 Configuración dinámica de las ROI

Actualmente la posición de las ROI en la imagen se encuentra establecido por parámetros estáticos. Esto dificulta el proceso de dejar operativo el dispositivo en su instalación inicial, además de ser poco flexible ante eventuales cambios de ubicación.

Una alternativa para establecer la disposición de las ROI en la imagen de forma dinámica es utilizar la transformada de Hough para la detección de las líneas rectas que separan las pistas de la calzada [15].

Esta alternativa también podría mejorar la ubicación de las ROI laterales, debido a que se podría conocer la inclinación de las líneas diagonales de los extremos con respecto a la visual del dispositivo y establecer las ROI en los casos extremos de manera que la diagonal pase por el interior de la región, de esta forma lograr visualizar mejor el parabrisas en esos casos.

La transformada de Hough de todas formas representa incrementar el coste computacional, lo que va en desmedro de procesamiento en tiempo real que pueda requerir la obtención de muestras.

8.2 *Upgrade* del dispositivo de muestreo

Una de las barreras del actual dispositivo de muestreo es que fue implementado a manera de abaratar costos en el proyecto, lo que limita las capacidades del *hardware*.

De contar con mayor presupuesto se podría considerar la utilización de tarjetas gráficas especializadas en el procesamiento de imágenes, como por ejemplo la línea de mini computadoras Jetson de NVIDIA [16], lo que entregaría recursos computacionales superiores para procesamiento en tiempo real.

8.3 YOLO

Otro desafío para el proyecto actual es lograr la detección de vehículos livianos dentro de las ROI establecidas en la imagen. Si el proyecto contara con un *upgrade* en el *hardware* del dispositivo, esto habilitaría el uso de YOLO para la detección de los vehículos [17].

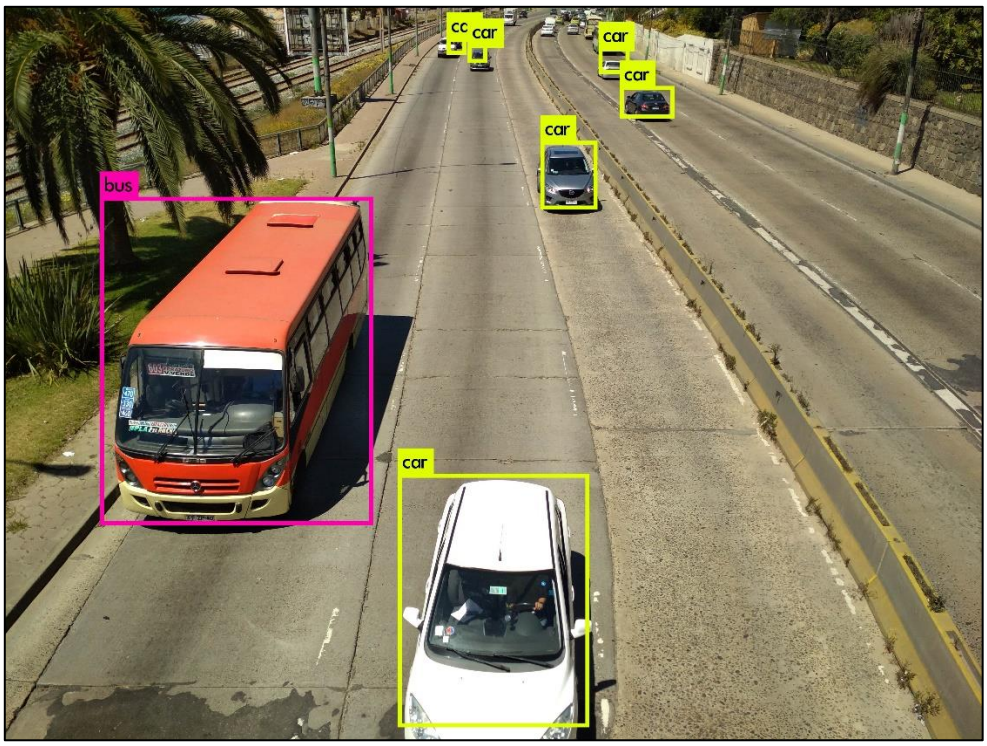


Figura 8-1: Detección de vehículos con YOLO en una imagen captada por el dispositivo de muestreo.

YOLO (You only look once) es una red neuronal capaz de detectar múltiples objetos en una imagen y reconocer su ubicación. YOLOv3 presenta diversas ventajas con respecto a otros métodos de detección y se caracteriza por su buen rendimiento [18].

En la Figura 8-1 se muestra YOLO aplicado a una imagen captada por el dispositivo de muestreo en Av. España. En ella se puede apreciar no solo la correcta detección de los vehículos dentro de la imagen, sino que además la diferenciación que realiza del tipo de vehículo con las distintas etiquetas. Esto permitiría abrir las posibilidades en la caracterización de las mediciones sobre el uso del cinturón de seguridad.

Capítulo 9

9 Conclusiones

Este capítulo presenta las conclusiones obtenidas a través del desarrollo del proyecto.

9.1 Conclusiones Generales

El objetivo del proyecto de generar un sistema viable para la gestión de imágenes de vehículos en tránsito se cumple con los resultados obtenidos en las pruebas realizadas a la implementación de prueba de concepto.

Este proyecto fue desarrollado a partir de una necesidad por parte de CONASET, quienes requerían lograr trazabilidad en las mediciones que realizan de manera observacional al uso del cinturón de seguridad. La propuesta fue el desarrollo del sistema descrito con el menor costo posible y que permitiera facilitar el proceso de mediciones sobre el uso del cinturón de seguridad. Una vez finalizado el proyecto, se presentaron los resultados que se muestran en este escrito y quedaron bastante satisfechos con el resultado alcanzado, en relación a las expectativas que tenían, pudiendo comprobar que sí es posible conseguir un proceso más automatizado y con menor costo a lo que realizan actualmente para las mediciones.

9.2 Trabajo futuro

El resultado deseado sobre las mediciones del uso del cinturón de seguridad en vehículos en tránsito, equivalen a que el sistema logre la detección del uso del cinturón de forma automática y de forma no invasiva para los ocupantes del vehículo.

Con respecto a esto, CONASET valora el avance conseguido en este proyecto y señalaron que apunta en la dirección correcta para automatizar el proceso de medición que realizan actualmente de forma observacional. Esta institución valida el resultado obtenido, comprendiendo que para lograr el objetivo final se requiere de más tiempo de desarrollo.

En la presentación final de estos resultados a CONASET, se manifiesta por parte de ellos la satisfacción de que el proyecto puede ejecutarse con costos reducidos, ya que los presupuestos que manejan año a año son bastante variables, un aspecto clave que ha dificultado la trazabilidad de las mediciones.

Los proyectos de CONASET en esta materia funcionan en base a licitaciones, por lo que ven una dificultad si tuviesen que ejecutarlo ellos mismos contratando al personal indicado. Sin embargo, reconocen el valor presente en esta memoria como una fase exploratoria de lo que se puede conseguir con las tecnologías existentes a bajo costo y con resultados prometedores.

El enfoque modular del proyecto actual permite añadir en un futuro capas de automatización, delegando la medición del uso del cinturón de seguridad al Módulo de Mediciones, por ejemplo. Para conseguir esto se debe desarrollar un

algoritmo de detección lo suficientemente robusto para superar cambios en el entorno, como reducción de la visibilidad o reflexiones por la luz solar, entre muchos otros factores.

10 Bibliografía

[1] «Memorias Multidisciplinarias. Fortalecimiento del Desarrollo de Competencias Transversales en la Formación del Profesional USM - Foco Q1 Q2,» [En línea]. Available: <http://competenciastransversales.usm.cl/>. [Último acceso: 5 de Julio del 2018].

[2] «Comisión Nacional de Seguridad y Tránsito. CONASET,» [En línea]. Available: <https://www.conaset.cl/>. [Último acceso: 5 de Julio del 2018].

[3] R. R. W. George E. Hypke, «Apparatus and methods for monitoring and detecting seat». 2004.

[4] W. D. Cotter, «Seat belt status external monitoring apparatus and method». 2010.

[5] H. L. S. Z. Huiwen Guo. [En línea]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5983807.

[6] S.-P. Y. Prem Kumar Bhaskar. [En línea]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6868357.

[7] DigitalOcean, «Digital Ocean,» [En línea]. Available: https://www.digitalocean.com/. [Último acceso: 19 02 2019].

[8] ISO, [En línea]. Available: https://www.iso.org/iso-8601-date-and-time-format.html. [Último acceso: 11 01 2020].

[9] P. S. R. H. Tibor Trnovszkýa*, «ScienceDirect,» [En línea]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817327005. [Último acceso: 18 Abril 2020].

[10] «OpenCV-Python Tutorials,» [En línea]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html#contour-area.

[11] [En línea]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html#convex-hull.

[12] L. V. P. C. P. C. LOURENA ROCHA. [En línea]. Available: https://www.researchgate.net/publication/221337276_Image_Moments-Based_Structuring_and_Tracking_of_Objects. [Último acceso: 4 Junio 2020].

[13] «OpenCV-Python Tutorials,» [En línea]. Available: https://opencv-python-

tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html#moments.

[14] G. G. Li Shuhua. [En línea]. Available: https://www.researchgate.net/publication/251934488_The_application_of_improved_HSV_color_space_model_in_image_processing.

[15] J. L. M. L. N. D. Luis Canul Arceo, «ResearchGate,» [En línea]. Available: https://www.researchgate.net/publication/287818230_Un_algoritmo_rapido_de_la_transformada_de_Hough_para_la_deteccion_de_lineas_rectas_en_una_imagen.

[16] [En línea]. Available: https://www.nvidia.com/es-la/autonomous-machines/jetson-store/.

[17] [En línea]. Available: https://pjreddie.com/darknet/yolo/. [Último acceso: 7 Junio 2020].

[18] A. F. Joseph Redmon, «YOLOv3: An Incremental Improvement,» [En línea]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf.