



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA Y
CIENCIAS DE LA COMPUTACIÓN



Desarrollo de un sistema de pesaje en movimiento no intrusivo para el monitoreo de activos críticos

POR

Richard González Lara

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para
optar al título profesional de Ingeniero Civil Informático

Profesor Guía

Gonzalo Rojas Duran

2024

Concepción (Chile)

© 2024 Richard Alejandro González Lara

© 2024 Richard Alejandro González Lara

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Índice

1. Introducción.....	4
1.1. Objetivos.....	5
1.1.1. Objetivo general.....	5
1.2. Metodología de trabajo.....	5
2. Marco teórico.....	7
2.1. Infraestructura Vial y su importancia en el contexto nacional.....	7
2.2. Técnicas de Pesaje de Vehículos.....	8
2.3. Interacción puente-vehículo.....	10
2.4. Redes neuronales convolucionales (CNN).....	10
2.5.1. Implementación de YOLO.....	14
3. Obtención y exploración de datos.....	16
3.1. Recopilación de datos.....	16
3.1.1 Primera recolección de datos.....	16
3.1.2 Segunda recolección de datos.....	17
3.2. Limpieza de datos.....	22
3.3. Exploración de datos.....	23
3.3.1 Ejemplo 1: Auto con varios vehículos alrededor.....	23
3.3.2 Ejemplo 2: Van con un auto detrás.....	25
3.3.3 Ejemplo 3: Auto con un auto detrás.....	26
3.3.4 Ejemplo 4: Micro.....	27
3.3.5 Ejemplo 5: Autobús más grande.....	28
3.3.6 Ejemplo 6: Auto con una moto detrás.....	29
3.3.7 Ejemplo 7: Moto.....	30
3.3.8 Ejemplo 8: Conclusiones.....	31
4. Arquitectura del Sistema.....	32
4.1. Arquitectura de procesamiento de datos.....	38
4.2. Arquitectura CNN.....	39
5. Implementación de módulos del sistema.....	41
5.1. Red neuronal convolucional (CNN).....	41
5.2. Detección de vehículos.....	42
5.3. Selección de zona de interés.....	45
6. Resultados.....	48
7. Sigüientes pasos.....	52
8. Conclusión.....	53
9. Referencias.....	54
10. Anexos.....	58

1. Introducción

El mantenimiento de las infraestructuras viales consiste en una serie de actividades, como la detección y reparación de daños en el pavimento (grietas, baches, etc), la revisión y reforzamiento constante de estructuras (puentes, túneles), y la conservación de los diferentes elementos que tiene el camino (drenaje, señalización, pintura y repintado de líneas de demarcación). Estas tareas requieren de un gran esfuerzo para llevarse a cabo con éxito, ya que se deben tener muchas variables en cuenta, como el clima, el tráfico, los materiales y técnica con que se construyó la infraestructura, etc. El mal funcionamiento de una estructura vial puede producir problemas que van desde la necesidad de cortar la ruta, hasta generar siniestros que pueden afectar a muchas personas.

Una de las variables importantes en este caso es justamente el tráfico, para eso hay diferentes normas, entre ellas el límite de peso en vehículos dependiendo del lugar por el que transitan. Para cumplir con esto, se suele hacer uso de dos técnicas generales utilizadas para pesar los vehículos, existe el pesaje estático, que ofrece una medición precisa, pero tiene la desventaja de necesitar la detención de los vehículos, lo que implica cortar el flujo del tráfico, por otro lado, están las técnicas de pesaje en movimiento (WIM, por sus siglas en inglés Weigh in motion), las que, con ayuda de la instrumentación de las estructuras viales, permite estimar el peso de los vehículos en movimiento.

La presente memoria utiliza un enfoque basado en redes neuronales convolucionales [1], que estima el peso de los vehículos pasantes en base a una ventana de tiempo de aceleraciones verticales captadas por acelerómetros. Para identificar el momento en el que pasan los vehículos por el acelerómetro, se hace uso de un módulo de visión artificial, que detecta el paso de los vehículos por una zona de interés.

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar un sistema de pesaje en movimiento no intrusivo para el monitoreo de activos críticos, principalmente puentes, que permita estimar el peso de los vehículos sin necesidad de alterar la estructura.

1.1.2. Objetivos específicos

- Diseñar un sistema de visión artificial que permita detectar a los vehículos que transitan en el puente, específicamente el momento en que pasan por un acelerómetro.
- Diseñar un sistema que permita calcular el peso de los vehículos que transitan por el puente, en base al sistema de visión artificial y acelerómetros.
- Implementar un proceso de calibración y ajuste único para cada puente.
- Modelar la arquitectura para utilizar el módulo de pesaje en movimiento no intrusivo en la plataforma LIND.
- Validar el desempeño de las partes que conforman el sistema y el sistema completo, para evaluar la precisión.

1.2. Metodología de trabajo

Se llevó a cabo un marco de trabajo OSEMN [24] (Obtain, Scrub, Explore, Model, Interpret) como marco de referencia para la creación y entrenamiento de la red neuronal convolucional utilizada, por otro lado, se siguió un proceso iterativo para el desarrollo incremental de las funcionalidades en los diferentes módulos del sistema. Se utilizaron estas metodologías para enfrentar de buena forma las diferentes funcionalidades y también, para explicar de forma más ordenada los módulos desarrollados.

1.3. Estructura del informe

Este informe está organizado en diferentes secciones que explican los fundamentos teóricos, el desarrollo técnico y los resultados del proyecto. A continuación se describe brevemente el contenido de cada sección:

2. Marco teórico: Aborda los conceptos clave para el desarrollo del sistema, explicando conceptos civiles y tecnológicos.
3. Obtención y exploración de datos: Detalla el proceso de recopilación, limpieza y análisis de los datos utilizados para el entrenamiento y validación del sistema basado en el marco de trabajo OSEMN.
4. Arquitectura del sistema: Describe la arquitectura general del sistema, y los módulos que la conforman.
5. Implementación de módulos del sistema: Explica cómo se desarrollan e integran los diferentes módulos y su funcionamiento.
6. Resultados: Presenta los resultados obtenidos de la CNN implementada.
7. Sigüientes pasos: Propone posibles mejoras y futuras líneas de trabajo para el proyecto.

2. Marco teórico

En este capítulo se abordan los conceptos básicos del monitoreo de la salud estructural (SHM, por sus siglas en inglés: Structural Health Monitoring) [25-26], y herramientas tecnológicas que ayudan a este monitoreo, enfocándose principalmente en aplicaciones de pesaje en movimiento (WIM) [1-4], funcionamiento de las redes neuronales convolucionales [12-16], interacción dinámica entre un puente y un vehículo [27], para caracterizar las señales, y también, la visión artificial [19-20] y su uso práctico en esta memoria.

2.1. Infraestructura Vial y su importancia en el contexto nacional

Los puentes son piezas clave en las redes de infraestructura en todo el mundo, conectando comunidades y haciendo posible que bienes y personas se trasladen de forma eficiente. Muchos de los puentes que usamos hoy se construyeron hace muchas décadas y se han deteriorado bastante por el aumento del tráfico y el castigo del clima con el paso del tiempo.

En Chile, los camiones de carga pesada son fundamentales para nuestra economía, que sigue creciendo con el comercio y la industria. Esto obliga a buscar formas de garantizar que el tráfico sea seguro y fluido en las carreteras más usadas. Cada vez circulan camiones más pesados por las rutas de Chile, y esto ha mostrado que varios puentes tienen problemas estructurales, lo que genera preocupación respecto a la seguridad. Hay tramos especialmente delicados para los camiones grandes, ya sea porque son muy importantes para el comercio o por la geografía complicada del terreno.

Hay varios factores que pueden causar problemas en estas estructuras, es necesario medir qué tan graves son los daños y predecir posibles problemas antes de que sea demasiado tarde. La tecnología ofrece muchas soluciones; una de las más útiles es el monitoreo en tiempo real [29], que con la ayuda de diferentes instrumentos con sus respectivos análisis numéricos y computacionales, generan diagnósticos sobre el estado de las estructuras, permitiendo la identificación y corrección de fallas en estas, evitando futuros problemas de mayor magnitud.

Un ejemplo claro de puente con problemas es el Cau Cau en Valdivia, que se construyó como el primer puente levadizo de Chile. Desde que lo inauguraron en 2015, ha tenido muchos problemas técnicos, sobre todo con su sistema para levantarse y bajarse, que hasta el día de hoy necesita

arreglos continuamente. También hay otros casos como el puente Malleco, uno de los puentes ferroviarios más viejos del país, o el Loncomilla, que tienen daños visibles como grietas y óxido, y necesitan mantenimiento a fondo para evitar problemas graves.

Una gran cantidad de camiones recorren las carreteras chilenas, formando la espina dorsal del sistema de distribución. Estos camiones grandes transportan el 95% de las mercancías del país según estimaciones de la Confederación Nacional de Dueños de Camiones de Chile [22]. Las carreteras principales, como la Ruta 5 y los caminos que van de las zonas productivas a los puertos, aguantan cada día el paso constante de estos vehículos, que hacen que los puentes sufran un desgaste más rápido. Esto es especialmente importante porque muchos de estos puentes se diseñaron cuando nadie imaginaba la cantidad y el peso de los camiones que circulan hoy, lo que hace muy necesario implementar sistemas para vigilarlos y mantenerlos antes de que fallen.

2.2. Técnicas de Pesaje de Vehículos

El WIM es la forma más útil para mejorar y optimizar la monitorización del tráfico y el control en el peso de los vehículos, ya que obviamos la necesidad de detenerlos. Hay dos enfoques utilizados principalmente, y estos se dividen en WIM basado en sensores sobre la calzada y B-WIM (por sus siglas en inglés: Bridge Weigh In Motion) que se apoyan de la respuesta estructural del puente [3].

- **Sensores en la calzada:** Para estos se utiliza un principio en común, el cual es medir la deformación y/o la fuerza de los vehículos que pasan sobre los diferentes sensores, se pueden destacar las placas de carga, que se encargan de medir la fuerza aplicada por los neumáticos, y son altamente precisos, los sensores piezoeléctricos, que detectan la fuerza aplicada deformandose, y otros sensores como los de fibra óptica e inductivos. El problema principal de este enfoque es que se deben instalar los diferentes sistemas en la calzada, lo que implica, interrupción del tráfico, constante desgaste y mantenimiento de los sensores, lo que conlleva a una posible reducción de la precisión al pasar del tiempo, ya sea por el estado del pavimento o la distribución constante de carga por parte de los vehículos.
- **B-WIM:** Es una alternativa más duradera y eficiente, aprovechando la estructura del puente para crear una suerte de balance estructural. En lugar de instalar sensores en la calzada, se emplean dispositivos en la superestructura del puente para medir la deformación inducida por los vehículos en movimiento. Hay dos grandes grupos cuando hablamos de sistemas B-WIM.

- Algoritmos Estáticos: La base de esto es el algoritmo de Moses que fue el que introdujo el concepto de B-WIM en 1979, para puentes hechos de vigas y losa y lo que busca es minimizar la diferencia entre la respuesta medida del puente, esto es, la deformación registrada por sensores y la respuesta teórica, en base a la línea de influencia y una estimación inicial del peso del vehículo, para esto se sigue una serie de pasos, desde la instalación de diferentes sensores para la respuesta medida, y la definición de variables para la respuesta teórica, el cálculo del error y la posterior minimización de este error, en la figura 1 se muestra la formulación matemática propuesta por Moses, que dice que el momento flector medido (es el esfuerzo interno que experimenta un elemento estructural cuando se somete a cargas que lo flexionan o cargan) [30] en un tiempo k puede obtenerse sumando el momento flector individual de cada viga.

$$M_k^m = \sum_i^G ES_i \varepsilon_i$$

Figura 1: Momento flector medido para B-WIM.

Donde G es el número total de vigas, E es el módulo de elasticidad, S_i es el módulo de sección de la i -ésima viga y ε_i es la deformación medida de la i -ésima viga. Junto a esto también establece una fórmula para establecer una predicción del momento flector y otra para definir el error entre el momento medido y el previsto, y luego minimizando el error, se puede calcular el peso por eje y el peso bruto del vehículo.

Este es un método ampliamente utilizado, pero con muchos factores de mejora, los efectos dinámicos del vehículo pueden ser especialmente variables, la posición del vehículo también puede afectar la distribución de la carga en el puente, lo que puede afectar la medición, y también el ruido que se puede generar en las mediciones puede potenciar el error. Este algoritmo se ha ido iterando con el pasar de los años, calibrando diferentes factores, como las líneas de influencia, utilizando superficies de influencia, o apoyándose con inteligencia artificial para adaptarse a todos estos factores dinámicos.

- Algoritmos dinámicos: De esta forma se proponen algoritmos que puedan adaptarse al cambio constante de la estructura, el movimiento y posición de los vehículos, utilizando diferentes métodos, como la identificación en el dominio del tiempo, que analiza la respuesta estructural a lo largo del tiempo, para estimar la evolución de la carga móvil, utilizar transformadas de Fourier para separar la contribución dinámica del puente y del vehículo, y métodos que utilizan la inteligencia artificial, como redes neuronales o algoritmos de optimización para mejorar la precisión en condiciones de tráfico complejo.

2.3. Interacción puente-vehículo

La respuesta estructural de un puente depende de las cargas móviles que son inducidas por los vehículos [28], esto es, cada vehículo varía su posición, velocidad, intensidad, etc. mientras pasa por el puente, y esto genera esfuerzos dinámicos en la infraestructura, a su vez, las características del puente afectan el comportamiento del vehículo. Hay una serie de variables que pueden ser claves al momento de generar una buena estimación de peso, por un lado está la velocidad del vehículo, el peso por eje, la distribución de carga en los carriles, y la rugosidad de la superficie del puente, ya que estas afectan las vibraciones y la respuesta estructural, y por su parte el puente también tiene factores relevantes, como su frecuencia natural, la rigidez de la superestructura, su capacidad de amortiguamiento, etc. Para estimar de forma precisa el peso con el enfoque B-WIM, se requieren de algoritmos o formas de compensar estos efectos dinámicos.

2.4. Redes neuronales convolucionales (CNN)

Las Redes Neuronales Convolucionales [12] (CNN por sus siglas en inglés, Convolutional Neural Network) son un tipo de red neuronal artificial (véase anexo 2) diseñada para procesar datos con estructura de grilla, principalmente con imágenes, aunque también se ve en el procesamiento de señales y clasificación de datos. Su arquitectura se compone de múltiples capas que permiten aprender características complejas de los datos.

2.4.1. Estructura de una CNN

Una CNN típica consta de las siguientes capas:

1. Capa de convolución: La convolución [14] es un proceso donde se combinan dos funciones que producen una nueva función como resultado. La capa de convolución se crea pasando la información de entrada, generalmente imágenes, por un filtro que no es más que un vector generado de forma aleatoria, generando pesos y sesgos que formarán un mapa de

características, en la figura 2, se observa como a una entrada de 4x4 se le aplica un filtro de 3x3, generando una salida de 2x2 que encapsula características particulares, notar que el filtro o kernel, recorre la matriz moviéndose por las columnas, hasta llegar al final, luego pasa a la siguiente fila y recorre las columnas hasta cubrir toda la matriz de entrada.

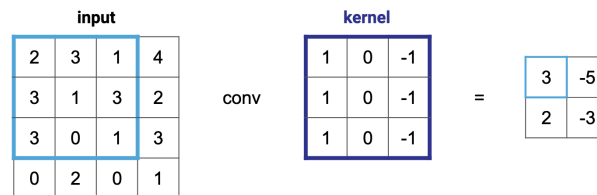


Figura 2: proceso de convolución.

2. Capa de activación ReLU: Es una función de activación que se computa luego de la convolución y sirve para generar no linealidad, de esta forma se pueden encontrar relaciones más complejas y así, mejorar el resultado, funciona asignando todos los valores negativos como 0, y manteniendo los demás valores. También se puede utilizar otro tipo de función de activación como es sigmoid, pero ReLU suele ser la más típica.
3. Capa de Pooling: Es una capa que opera sobre todos los mapas de características [13], reduciendo el tamaño de estos, pero manteniendo las características más importantes, los tipos más usados son:
 - Max Pooling: selecciona el valor máximo.
 - Average Pooling: calcula el promedio.

Estas capas pueden tener diferentes dimensiones en base a cuanto se busque reducir y mantener las características. En la figura 3 se observa la aplicación de max pooling y average pooling, donde se puede observar la característica extraída de la capa principal en base a los colores, por ejemplo max pooling verde toma el máximo valor entre 32, 10, 4 y 14, mientras que average pooling toma su promedio, luego se puede ver en el mapa de características resultante, como disminuye su dimensión.

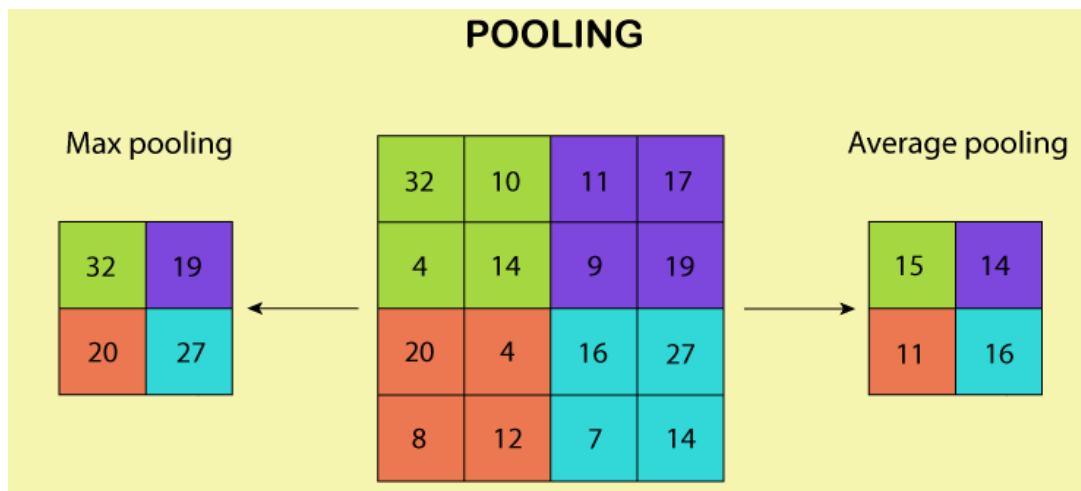


Figura 3: ejemplo de max pooling y average pooling.

4. Capa Fully Connected: Esta capa se encarga de conectar todas las neuronas de las capas anteriores [17], como se puede ver en la figura 4, donde se ve una conexión completa, sirve para preparar la salida utilizando toda la información y características adquiridas, se pueden utilizar técnicas como Dropout para prevenir el sobreajuste.

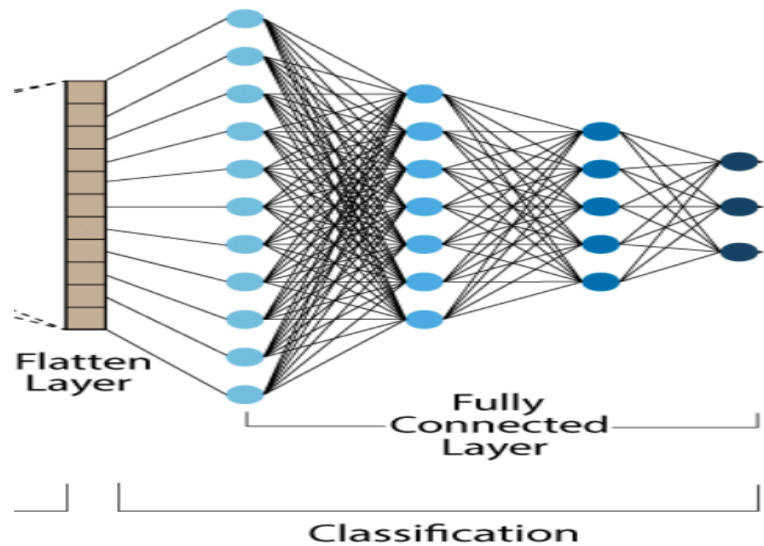


Figura 4: Capa Fully Connected.

5. Capa Softmax: Esta es una capa de activación que se suele aplicar a la última capa de la red y es útil principalmente para las redes neuronales convolucionales de clasificación, atribuyendo una probabilidad a las diferentes salidas, como en la figura 5, donde se

representan las diferentes frutas en las que se puede clasificar la entrada, teniendo mayor peso la pera, y siendo el valor elegido.

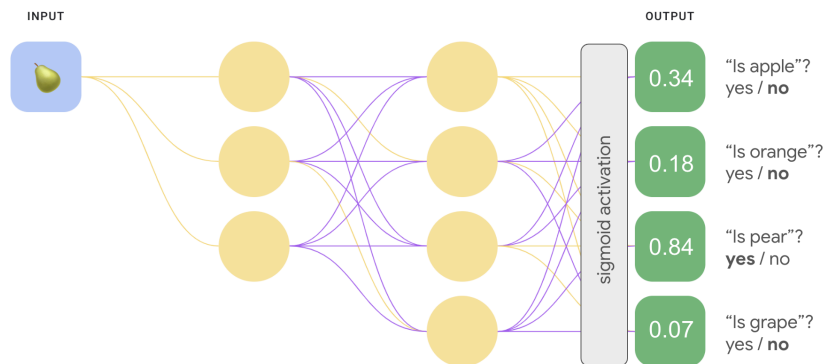


Figura 5: aplicación de función softmax para la clasificación de imágenes.

Las CNN son herramientas poderosas y versátiles debido a su capacidad para aprender automáticamente características jerárquicas a partir de datos estructurados. Su arquitectura ayuda para la extracción de información relevante, tanto en imágenes como en relaciones que no son lineales. Además, su capacidad para capturar patrones locales y globales las convierte en una herramienta idónea para tareas como reconocimiento de imágenes, pero también, como el procesamiento de señales, pensando en un vector de una dimensión.

2.5. You only look once (YOLO)

YOLO es un modelo de detección de objetos basado en redes neuronales convolucionales [20], este modelo tiene un enfoque particular si lo comparamos con otros modelos como R-CNN o Faster R-CNN [21], los que dividen la detección en múltiples pasos, por su parte, YOLO aborda el problema como una tarea de regresión única.

El modelo toma una imagen completa como entrada y predice simultáneamente múltiples cajas delimitadoras (bounding boxes) junto a las probabilidades de la clase que se le asocia. En la figura 6, se puede observar un ejemplo del funcionamiento de este modelo, creando una grilla, generando cajas delimitadoras que se encargan de identificar los posibles objetos, con sus coordenadas y un nivel de confianza, lo que se traduce a un mapa de probabilidades de cada clase y termina entregando una detección con sus respectivas cajas delimitadoras. Este enfoque permite mayor rapidez y eficiencia en la detección de objetos en tiempo real.

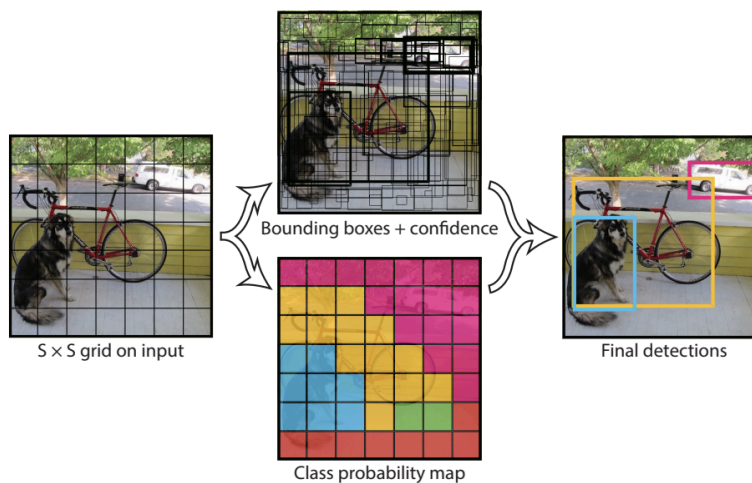


Figura 6: detección de bounding boxes y asociación de clases de YOLO.

2.5.1. Implementación de YOLO

Una implementación de YOLO sigue en general los siguientes pasos:

1. Se divide la imagen de entrada en una cuadrícula de $S \times S$.
2. Cada celda predice B bounding boxes y su confianza, además de C probabilidades de clase.
3. Se aplican funciones de activación para calcular la mejor combinación de predicciones.
4. Se eliminan detecciones redundantes con Non-Maximum Suppression (NMS).

5. Se obtiene la salida final con las cajas delimitadoras y las clases detectadas.

La arquitectura de YOLO está basada en una CNN profunda que incluye capas convolucionales y completamente conectadas, inspirada en modelos como GoogLeNet o DarkNet, modelos de clasificación de imágenes con diferentes cantidad de capas, filtros y funciones de activación.

A lo largo del tiempo, YOLO ha experimentado diversas mejoras en sus diferentes versiones, cada una enfocados en optimizar aspectos específicos del modelo original, por ejemplo:

- Basarse en diferentes arquitecturas, como GoogLeNet [32], Darknet-19 [33] , CSPDarknet53 [34], etc.
- Mejoras en la normalización de los datos.
- Mejoras en detección de objetos pequeños.
- Nuevas técnicas de aumento de datos (data augmentation).
- Mejoras en velocidad y precisión.

Con esta base teórica se establecen los fundamentos necesarios para desarrollar el sistema de pesaje en movimiento, utilizando los acelerómetros y la visión artificial para identificar la interacción dinámica puente-vehículo, y la utilización de una red neuronal convolucional para caracterizar estas señales y converger en la estimación del peso.

3. Obtención y exploración de datos

En este capítulo se siguen los primeros tres pasos del framework OSEMN, que corresponden a obtención, limpieza y exploración de los datos utilizados.

3.1. Recopilación de datos

La obtención de datos se llevó a cabo en un puente ubicado en Víctor Lamas con Arturo Prat en Concepción, en el Parque Ecuador (-36.83515357780239, -73.05583047037), donde se instaló un acelerómetro en la banqueta de un lado del puente, para obtener la interacción en una de las vigas, además, se instaló una cámara para detectar los vehículos y el momento en el que pasen por el acelerómetro. En total se recopilaron datos en dos ocasiones.

3.1.1 Primera recolección de datos

En la primera ocasión se utilizó un acelerómetro y una cámara ubicada en lo alto de un edificio, la toma y la sincronización con los datos se vio bastante favorecida, pero faltó una forma de estimar el peso de los vehículos que pasaban, de todas formas sirvió como un primer acercamiento a relacionar el peso que puede llegar a tener un vehículo respecto a su interacción con el acelerómetro, en la figura 7 se puede ver la señal vertical, correspondiente al eje Z del acelerómetro, donde se puede ver la aceleración en términos de gravedad, esto quiere decir que 1 corresponde a 9.81 m/s^2 aproximadamente. En este gráfico se ven tres líneas inusuales, que son fuerzas aplicadas a propósito, para poder sincronizar el tiempo de inicio de las mediciones con los videos, y también se ven dos valles entre los minutos 16:15 - 16:30 y 17:00 y 17:15, que son momentos en que se cayó la medición de la señal, por lo que es información que no sirve.

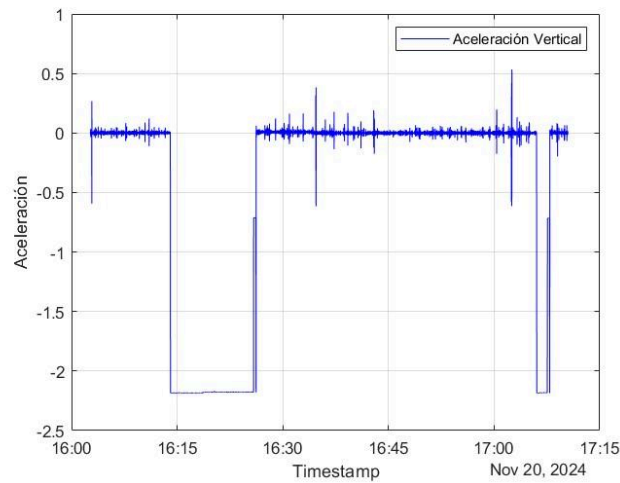


Figura 7: Datos capturados por el acelerómetro.

En la figura 8 se ve la toma que se hizo en lo alto de un edificio.

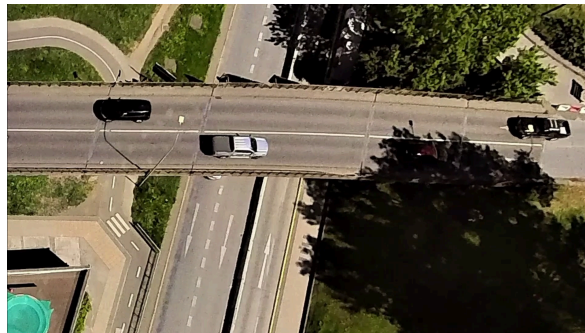


Figura 8: Toma de la cámara.

Con el fin de detectar y clasificar con éxito los vehículos, dada la resolución y ángulo del video, se tuvo que hacer un entrenamiento del modelo YOLO con imágenes y etiquetas personalizadas, la implementación de este modelo se comenta en el capítulo 6.

3.1.2 Segunda recolección de datos

En la segunda ocasión se utilizaron dos acelerómetros, instalados en un tramo, uno instalado al inicio del vano y otro en un tercio del vano (en el contexto de puentes, un vano es el espacio libre que hay entre dos pilares o estribos, como se puede observar en la figura 9), estos acelerómetros también miden la aceleración vertical correspondiente al eje Z, como se puede ver en las figuras 12

y 13, particularmente en la figura 13 se puede ver una línea al inicio que sirvió para sincronizar el video con las señales, de forma similar a lo hecho en la primera oportunidad, además se utilizaron dos cámaras, una instalada en lo alto del edificio como en la primera ocasión (figura 10), y otro ubicado en una posición estratégica para poder recuperar el modelo del vehículo (figura 11).

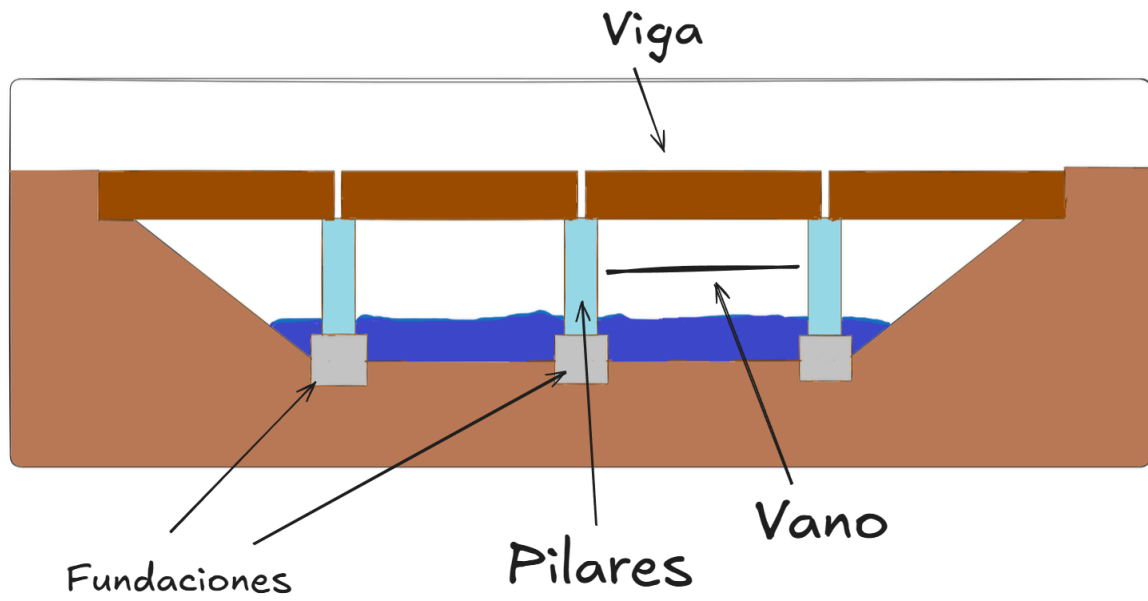


Figura 9: Partes del puente.



Figura 10: Toma de la primera cámara.



Figura 11: Toma de la segunda cámara.

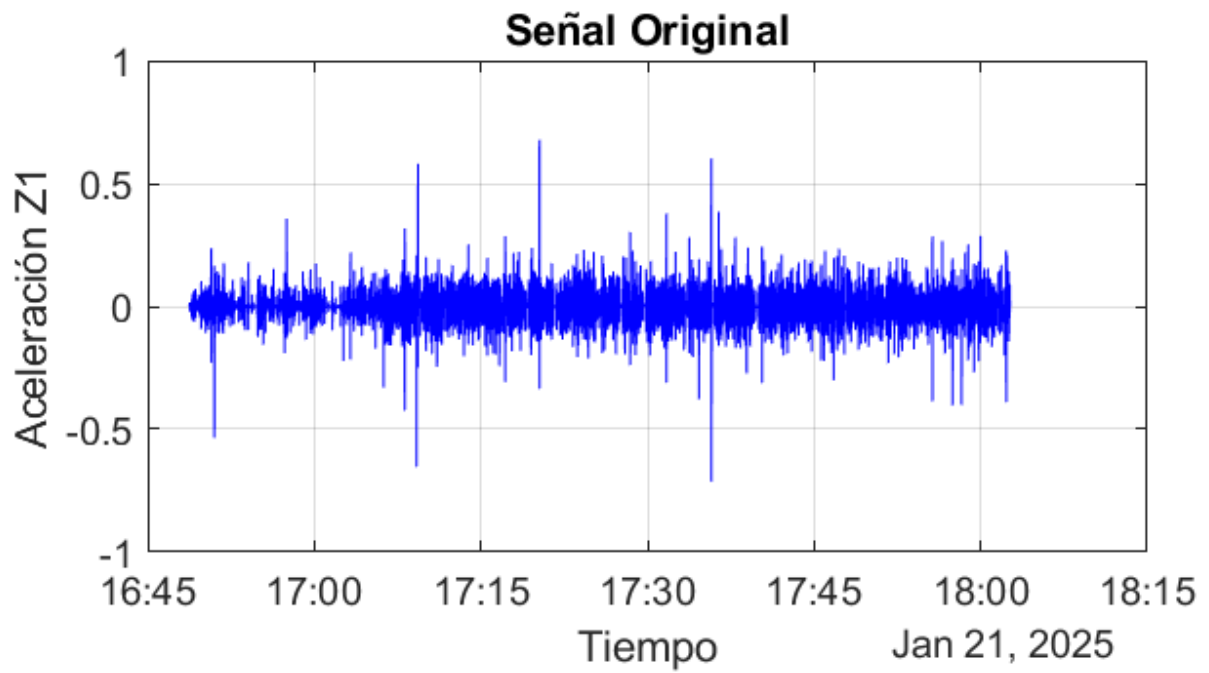


Figura 12: Acelerómetro a $\frac{1}{3}$ del vano.

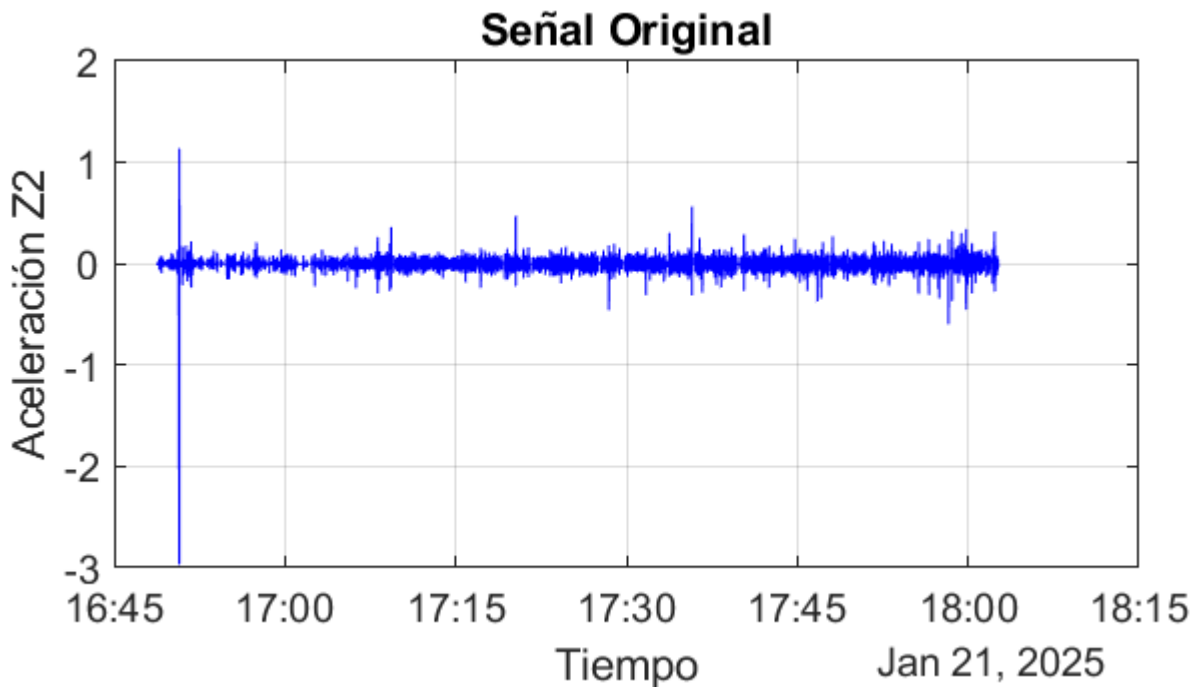


Figura 13: Acelerómetro a inicio de vano.

Con respecto a esta segunda recolección de datos surgió un inconveniente, y es que los picos más altos detectados del acelerómetro se consiguen en los momentos que pasan autobuses por el carril contrario, pero al pasar por el otro extremo del vano, esto se podría explicar por la propagación de las ondas, por ejemplo que la micro excite una vibración al otro lado del vano y este se refleje en los apoyos, de forma que se vean las mediciones en el acelerómetro propiamente tal. En la figura 14 se muestra un ejemplo, donde un bus pasa, y saliendo del tramo, se detecta una señal muy alta, específicamente la mostrada en la figura 15, estas son las señales más grandes detectadas y son muy diferentes a todo el resto de las mediciones, lo que entorpece el análisis de los datos.



Figura 14: Detección de uno de los picos del acelerómetro.

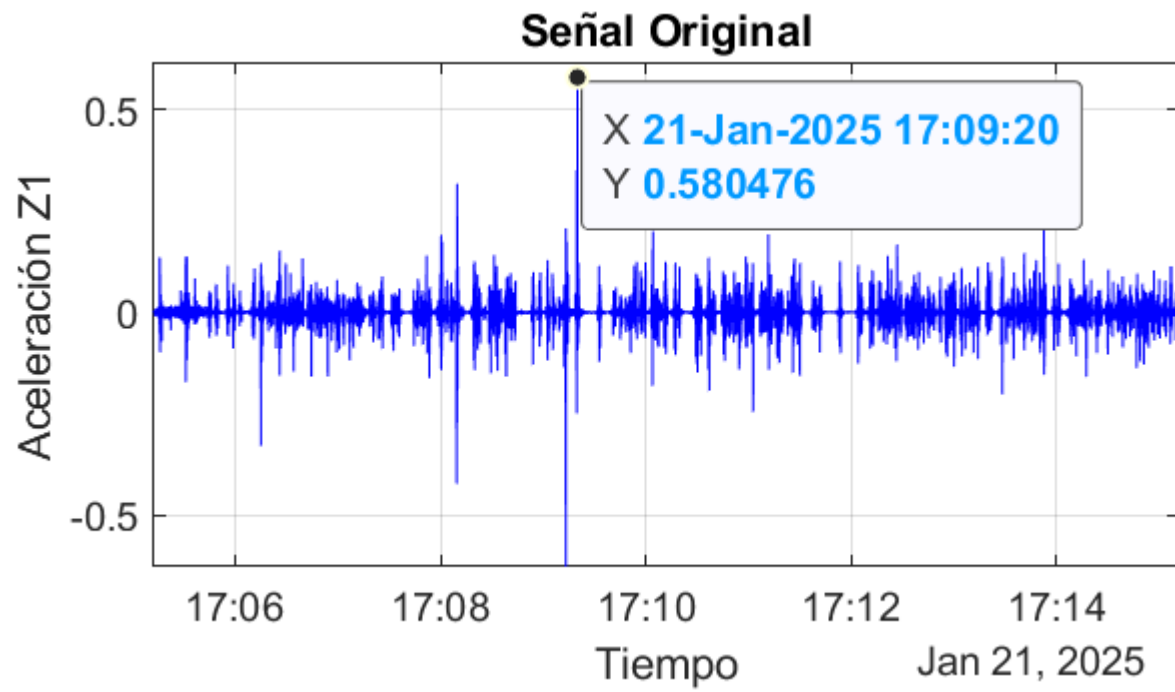


Figura 15: Dato con la hora correspondiente.

3.2. Limpieza de datos

Finalmente se utilizaron solo los datos de la primera recolección, dado el problema que surgió en la toma de datos de la segunda recolección, sin embargo, se utilizó la grabación cercana de vehículos de la segunda recolección, para estimar los pesos en vacío de los vehículos, y así lograr una estimación y entrenamiento más realista en el primer set de datos.

En cuanto a la cantidad de vehículos estudiados, en un inicio se tenían unos 1000, pero entre vehículos que se detectaban durante la caída de los sensores, y vehículos a los que no se les pudo estimar un rango de peso, finalmente se utilizaron 332 vehículos para el entrenamiento.

En cuanto al acelerómetro de la primera recolección, como se puede observar, hubo un momento en que se cayó la conexión, generando dos espacios de tiempo, donde hay datos basura, por lo tanto se obviaron esos minutos para el momento de la detección de vehículos.

La frecuencia de muestreo del acelerómetro estaba configurada a 500 hz, por lo que se hizo uso de una función de interpolación, para reducir el tamaño de la muestra a 100 hz, esto con tal de alinearse con lo propuesto en el artículo de base.

Para estimar el peso de los vehículos se utilizó Google Lens, el que tiene una funcionalidad de buscar por imagen, de esta forma utilizando la grabación que tenía una vista más cercana a los vehículos, se hicieron capturas de cada vehículo detectado, al pasarlo a Google Lens se podía obtener el modelo de los vehículos. Para el caso de las micros, con ayuda de la línea y el número se podía encontrar el modelo y chasis del vehículo, con esto se pudo hacer una estimación más cercana a la realidad, por lo menos para mantener un rango de pesos. Se usó el peso en vacío de los vehículos que contempla el peso del vehículo con todo el equipamiento y combustible lleno, pero sin pasajeros ni carga. En la figura 16 se pueden ver algunos pesos estimados en base al modelo de vehículo. Como se vio anteriormente, no se pudieron utilizar estos pesos ya que las detecciones de la segunda recolección presentan mediciones extrañas que dificultaron el análisis y entrenamiento, pero sí sirvieron para generar un rango de pesos en los vehículos detectados en la primera recolección.

TrackerID, Timestamp, Class, Weight	Modelo	Peso (KG)
6,2025-01-21 17:11:10.965517,cars	MAZDA/CX-30 I (2019-)	1,400
13,2025-01-21 17:11:16.034483,cars	Peugeot/2008 I facelift (2016-2019)	1,200
16,2025-01-21 17:11:17.896552,cars	Peugeot/208 I facelift (2015-2019)	1,050
24,2025-01-21 17:11:24.655172,cars	MG/ZS II facelift (2019-)	1,300
26,2025-01-21 17:11:26.724138,cars	KIA/Picanto II (2011-2015)	900
45,2025-01-21 17:11:52.827586,cars	Mazda CX5 2021	1,550
72,2025-01-21 17:12:12.17,van	Hyundai Porter h100	1,800
78,2025-01-21 17:12:18.965517,cars	Peugeot/Partner I facelift (2002-2012)	1,400
83,2025-01-21 17:12:21.275862,cars	Mitsubishi/Pajero IV (2006-2011)	2,100
86,2025-01-21 17:12:23.310345,cars	Citroen/C3 III facelift (2020-)	1,100
89,2025-01-21 17:12:25.103448,cars	Nissan/Versa III facelift (2022-)	1,100
91,2025-01-21 17:12:26.931034,van	Ford/Ranger IV (2015-)	2,000
97,2025-01-21 17:12:29.482759,van	Zotye/T600 2013 - Now	1,600
107,2025-01-21 17:12:33,cars	KIA/Rio IV (2017-2020)	1,100
110,2025-01-21 17:12:34.103448,cars	Nissan/Qashqai II (J11) facelift (2017-)	1,400
113,2025-01-21 17:12:36.482759,cars	Hyundai/Solaris I (2010-2014)	1,100
115,2025-01-21 17:12:38.068966,cars	Volvo/V40 II facelift (2016-)	1,400
118,2025-01-21 17:12:40.551724,van	Nissan/Navara (Frontier) IV (D23) (2014-)	1,900
124,2025-01-21 17:12:43.103448,van	Mitsubishi/L200 IV (2006-2014)	1,900
135,2025-01-21 17:12:46.034483,cars	Suzuki/Grand Vitara III (JT) (2005-2008)	1,500
140,2025-01-21 17:12:49.689655,cars	Peugeot/3008 II facelift (2020-)	1,400
151,2025-01-21 17:12:53.551724,cars	Nissan/Almera Classic I (B10) (2006-2013)	1,100
155,2025-01-21 17:12:57.241379,cars	Dodge/Journey I facelift (2011-2020)	1,800
161,2025-01-21 17:13:01.241379,cars	Subaru/XV II (2017-)	1,400
166,2025-01-21 17:13:06.275862,van	Lexus/ES VII (2018-)	1,600
185,2025-01-21 17:13:21.931034,cars	ZAZ/Ida 2012 - 2016	1,000

Figura 16: Identificación de modelos y peso de vehículos.

3.3. Exploración de datos

Para la exploración de los datos, se identificó principalmente, como se comportan las señales con respecto al paso de los diferentes tipos de vehículos, a la vez de la cantidad de vehículos simultáneos.

La figura 17 muestra un lapso de aproximadamente un minuto de aceleraciones, donde se pueden ver líneas punteadas de diferentes colores, cada línea identifica el momento en que se detectó el vehículo y el color identifica el tipo de vehículo, puede ser un auto, una van, un bus o una moto. En este caso se puede observar como tienden a verse picos en la aceleración en los momentos cercanos a la detección de buses, esto se puede explicar por el aumento en la vibración generado por autos más pesados.

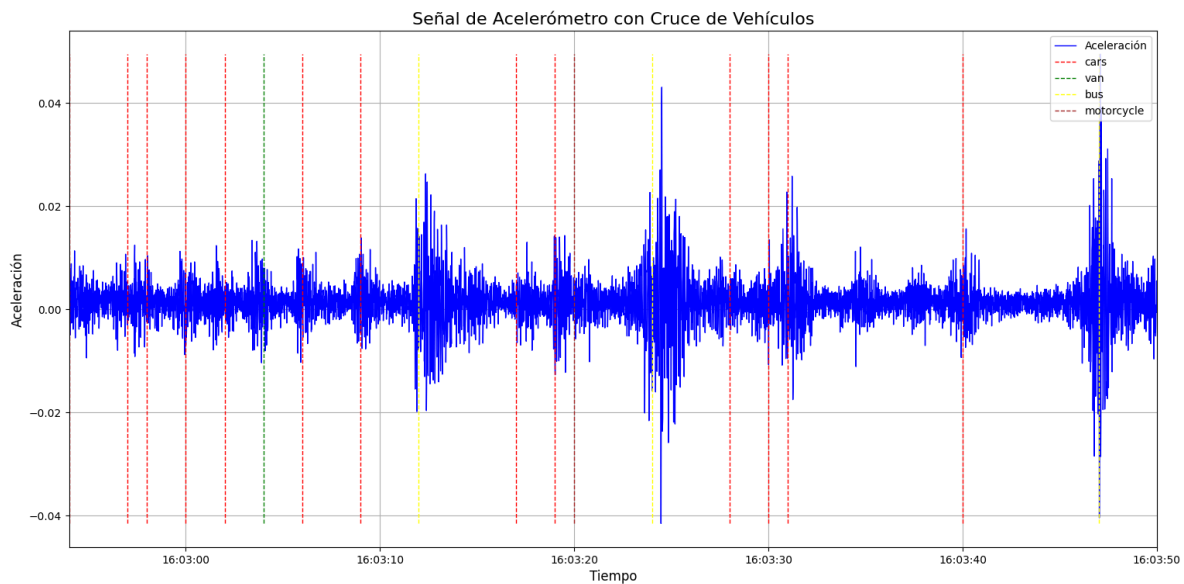


Figura 17: Cruce detección de vehículos vs señal de acelerómetro.

3.3.1 Ejemplo 1: Auto con varios vehículos alrededor

Se pueden ver ejemplos de diferentes vehículos y las respectivas señales detectadas por el acelerómetro, en estos primeros dos ejemplos, podemos ver un auto pasando por el puente junto a más vehículos en la figura 19 y se puede ver una medición con pequeños picos en la figura 18, desde el segundo 1 al 2 se pueden ver una oscilación más constante, que podría deberse a la vibración natural que se produce en el puente, seguido de otros picos que pueden explicarse por el auto que viene a la cola.

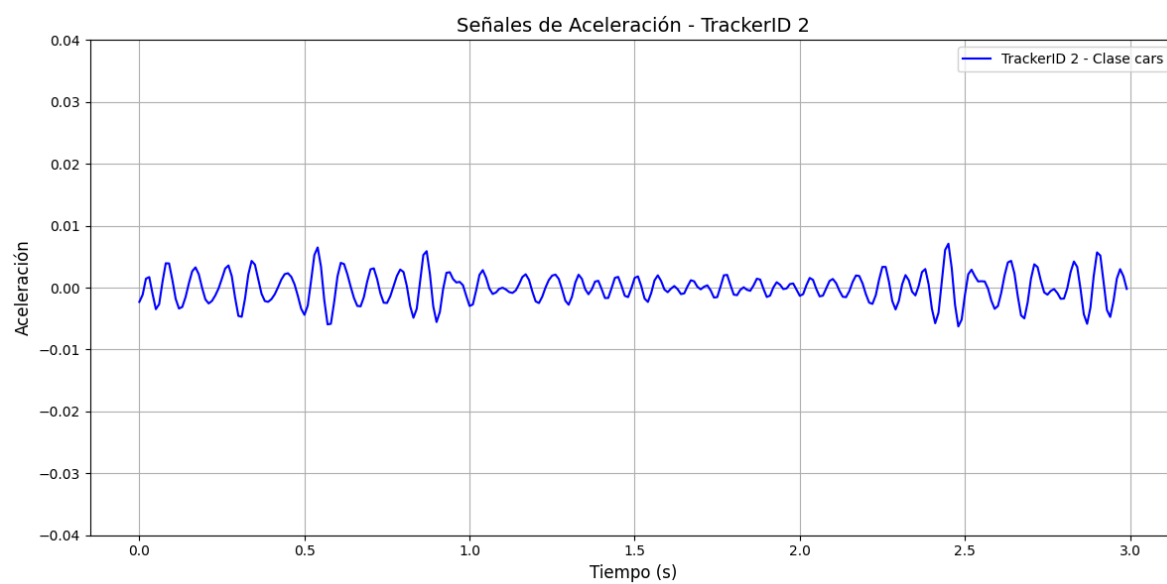


Figura 18: Ventana de tiempo de auto con id 2.

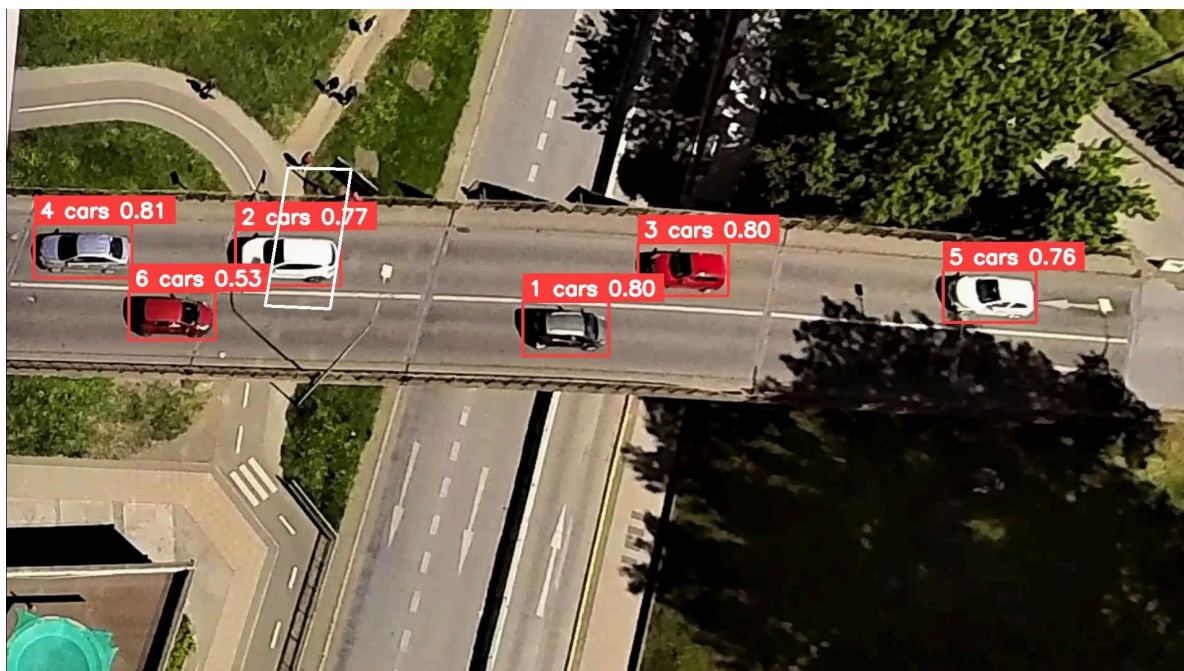


Figura 19: auto con id 2, 16:02:54.

3.3.2 Ejemplo 2: Van con un auto detrás

En este caso se ve una van, que tiene a la cola un auto, de aquí se desprende un comportamiento similar, aún con menos tráfico, también se aprecian oscilaciones repetitivas que se ven desde el segundo 0.5 al 1.5 en la figura 21, luego se ve otro pico que se podría explicar por el paso del auto 45 que va a la cola del van 27 en la figura 20.

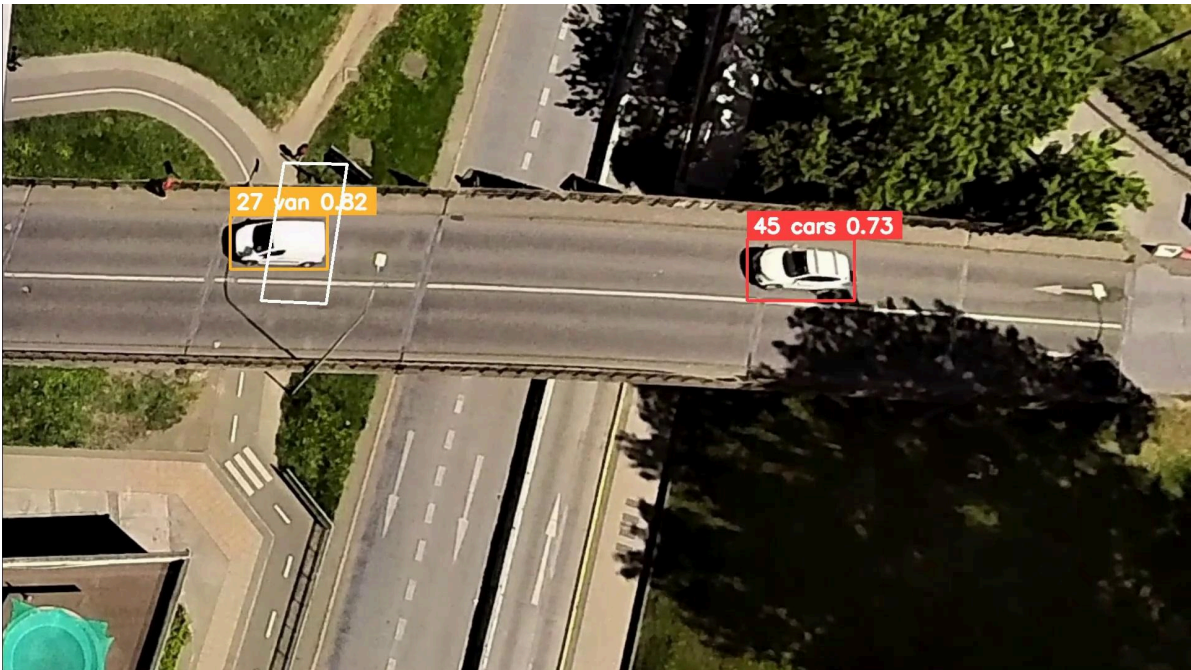


Figura 20: van con id 27, 16:03:04.

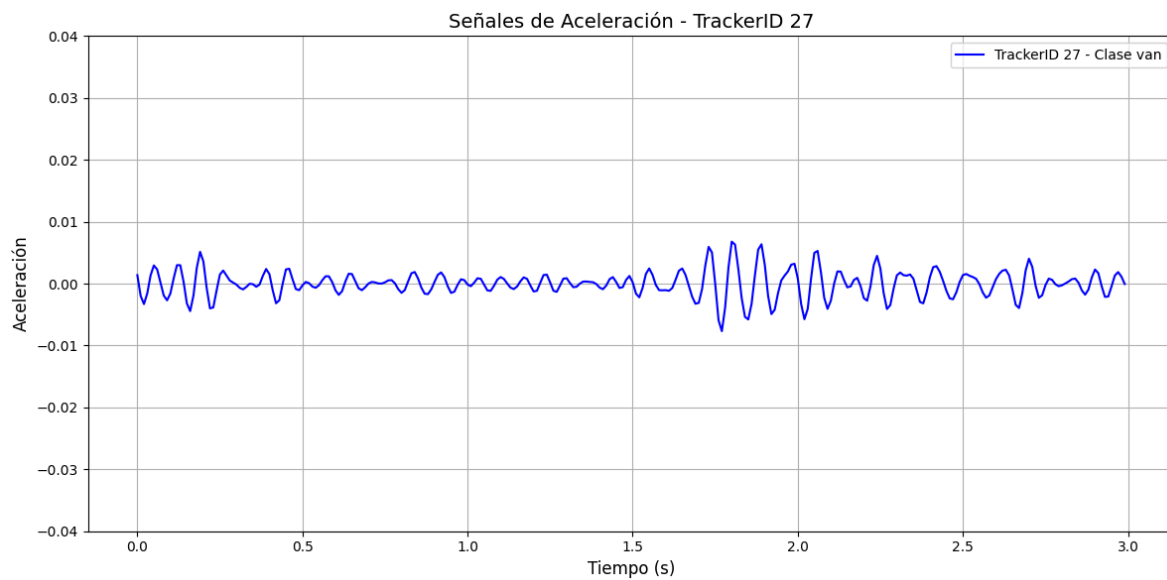


Figura 21: ventana de tiempo de van con id 27.

3.3.3 Ejemplo 3: Auto con un auto detrás

Nuevamente se ve un comportamiento similar, un estado parecido al de la van con el auto, pero esta vez con dos autos en la figura 22, de la figura 23 se puede comentar el impacto que ejerce el primer vehículo, y parece mostrarse al final de la señal la aceleración ejercida por el vehículo que viene a la cola, con vibraciones más bajas en la mayoría de la medición.

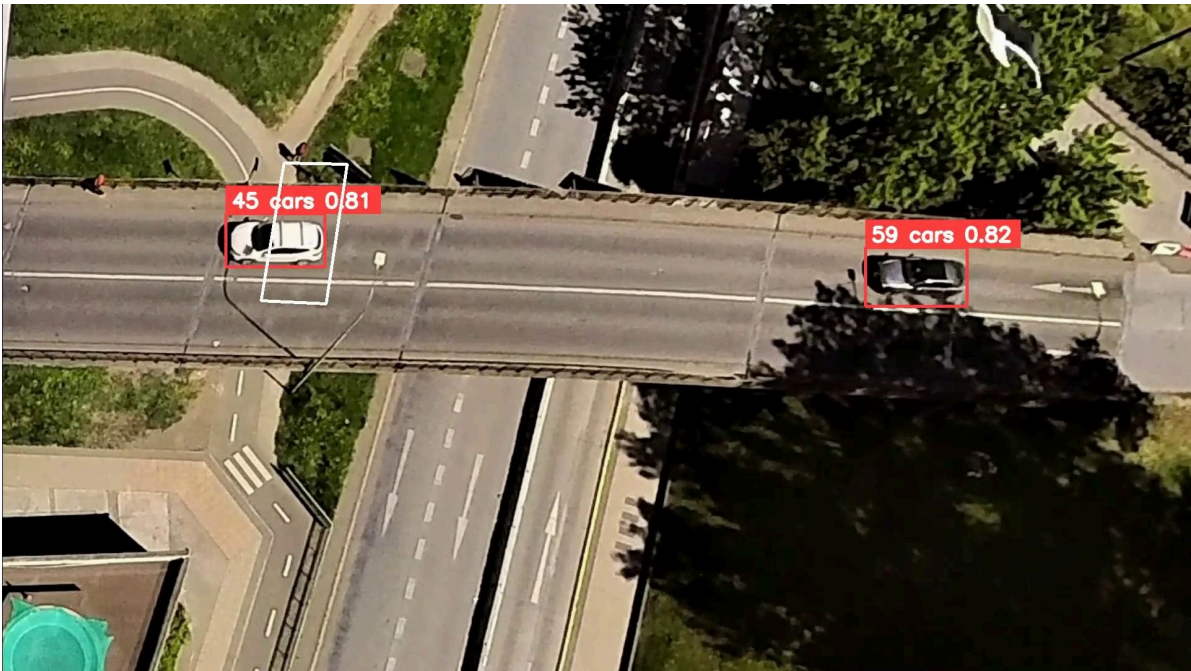


Figura 22: auto con id 45, 16:03:06.

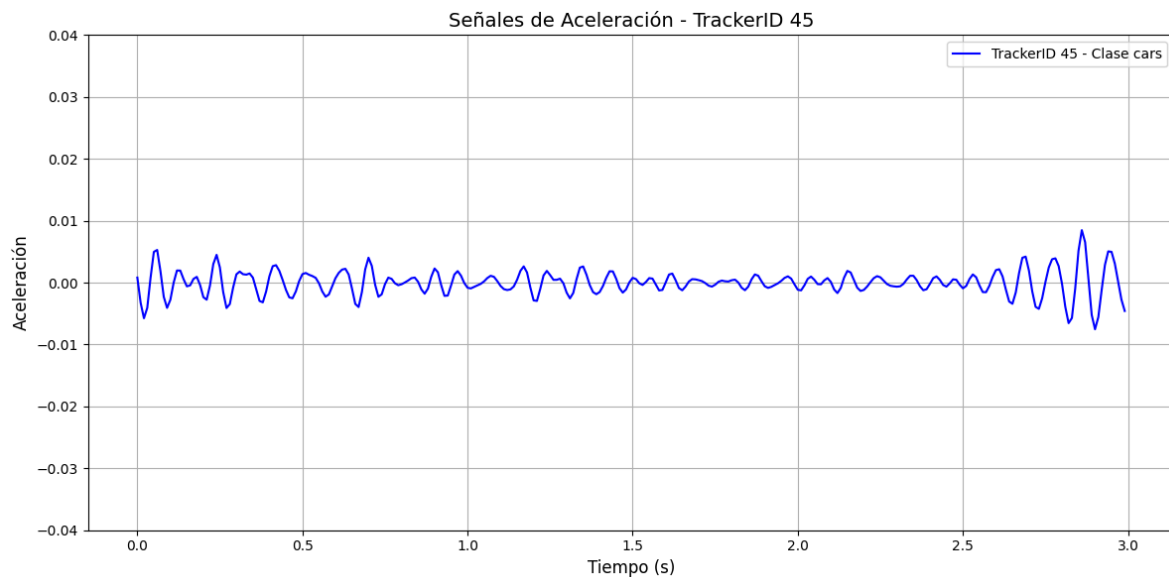


Figura 23: ventana de tiempo de auto con id 45.

3.3.4 Ejemplo 4: Micro

En la figura 24 se ve el caso de un bus, en este caso una micro, la que ya influye de manera más significativa como se puede ver en la figura 25, donde se aprecia un pico y en general, oscilaciones más pronunciadas, se evidencia como en vehículos pesados, se deja una vibración notable y que parece estar por encima de la vibración natural del puente.

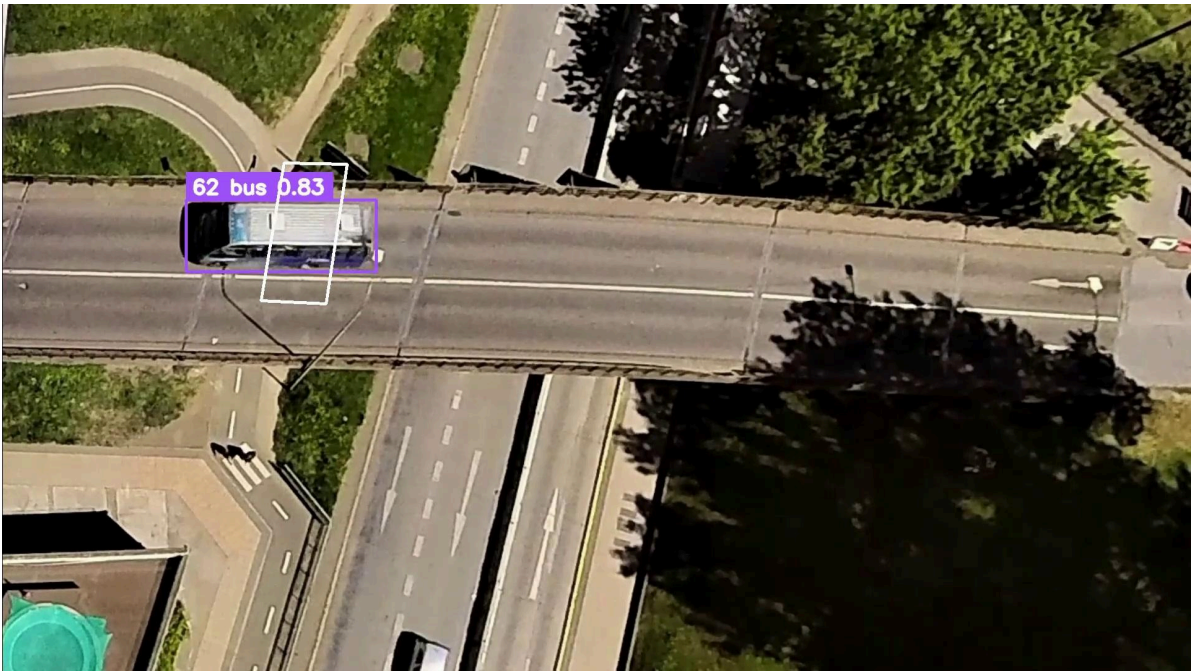


Figura 24: bus con id 62, 16:03:12.

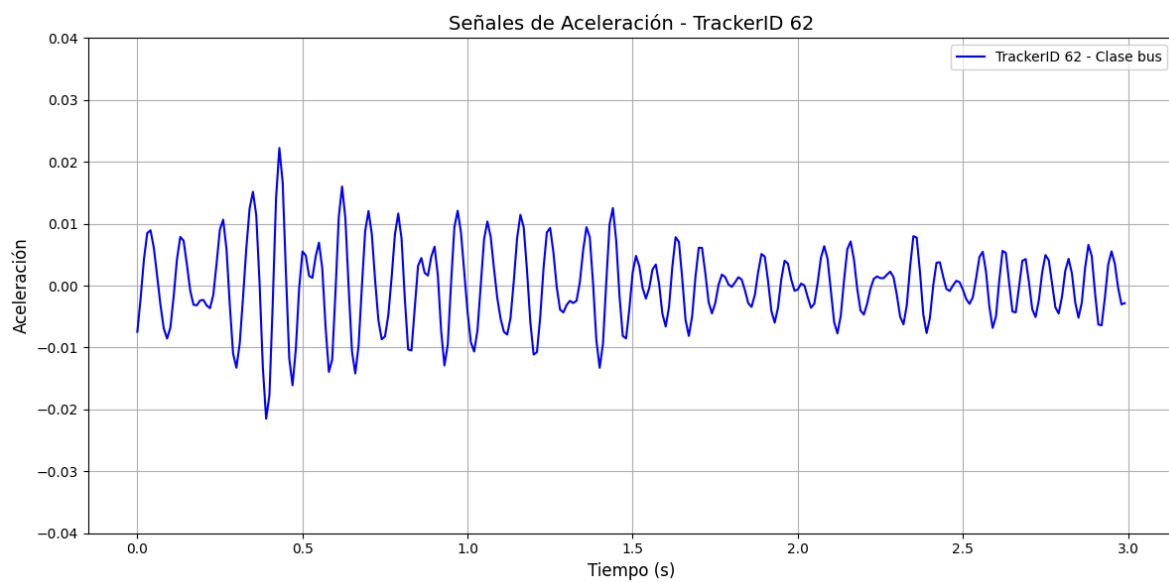


Figura 25: ventana de tiempo de bus con id 62.

3.3.5 Ejemplo 5: Autobús más grande

En la figura 26 se puede ver un bus más grande y pesado que la micro anterior, la que también deja en evidencia en la figura 27 unos picos altos y unas oscilaciones en general muy pronunciadas.

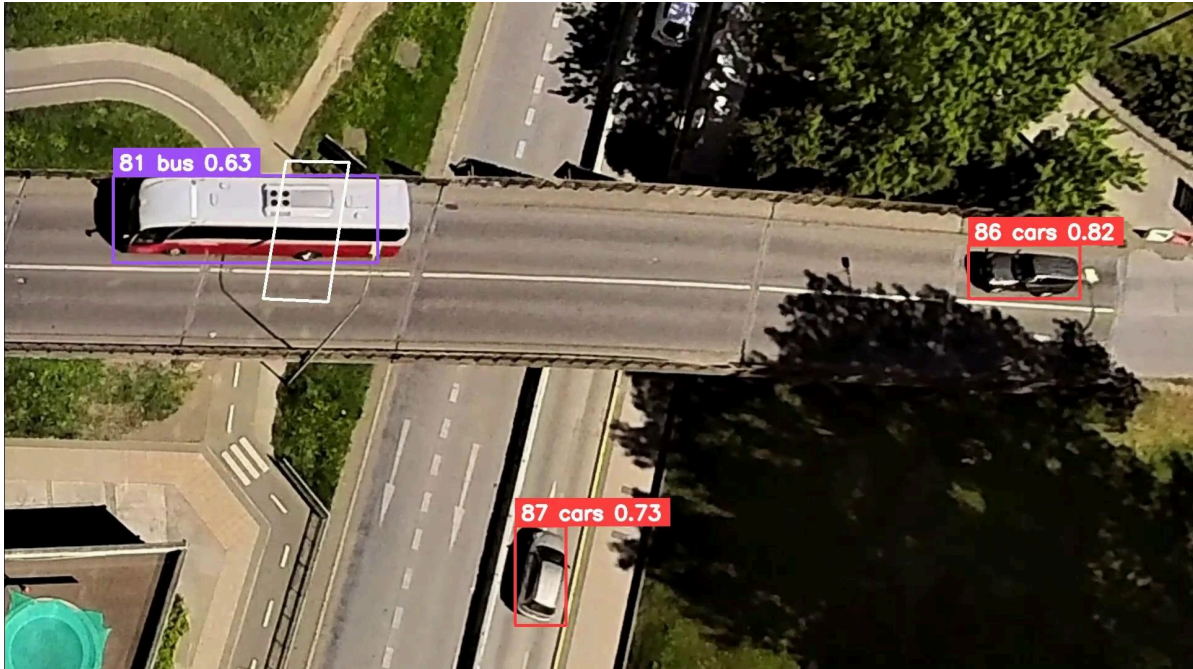


Figura 26: bus con id 81, 16:03:24.

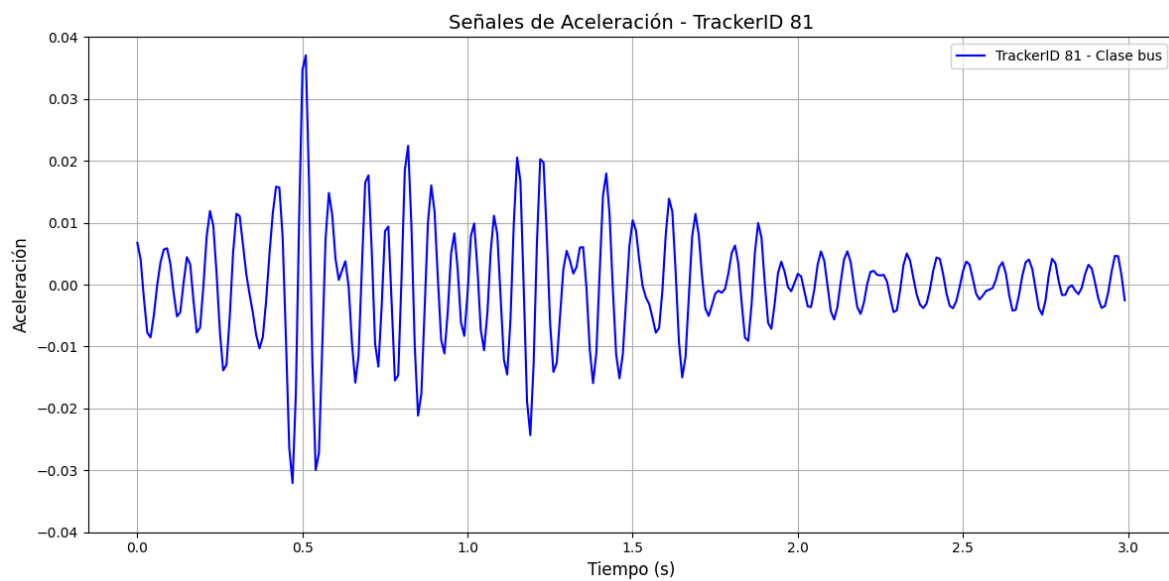


Figura 27: ventana de tiempo de bus con id 81.

3.3.6 Ejemplo 6: Auto con una moto detrás

En la figura 28 podemos ver un auto que tiene a la cola una moto, bien cerca, en las aceleraciones de la figura 29 se puede ver unos picos pronunciados para ser un auto, lo que podría indicar la presencia de un auto pesado, notar que esta vibración que deja, podría dificultar la detección de la aceleración que aplica la moto.

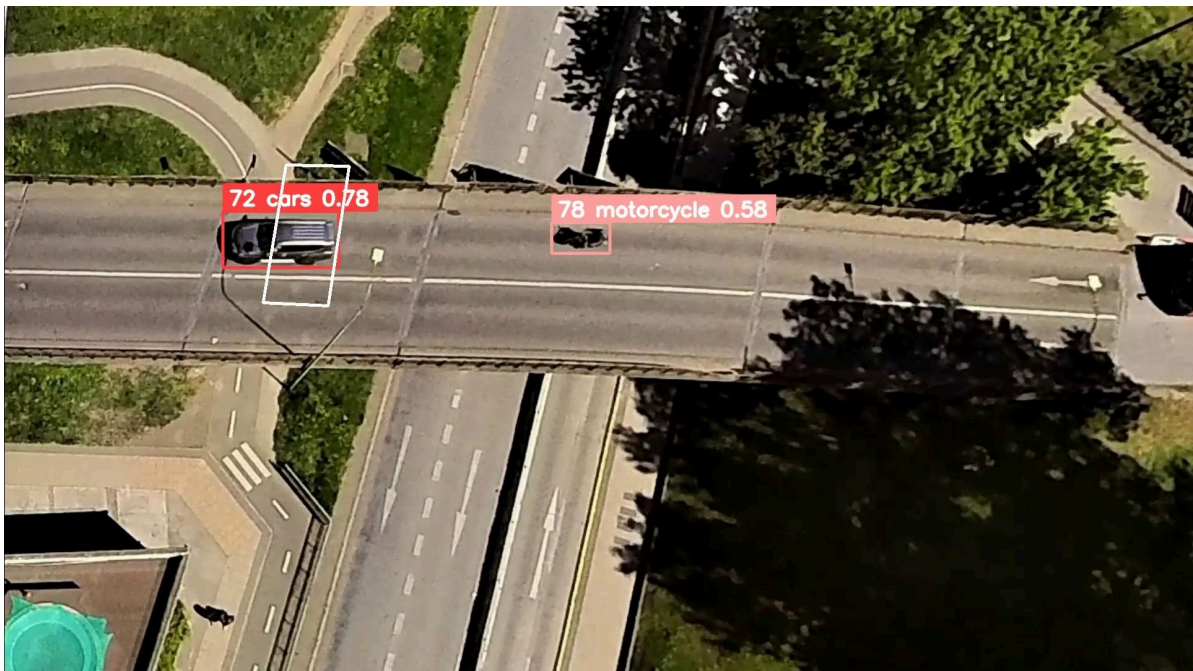


Figura 28: auto con id 72, 16:03:19.

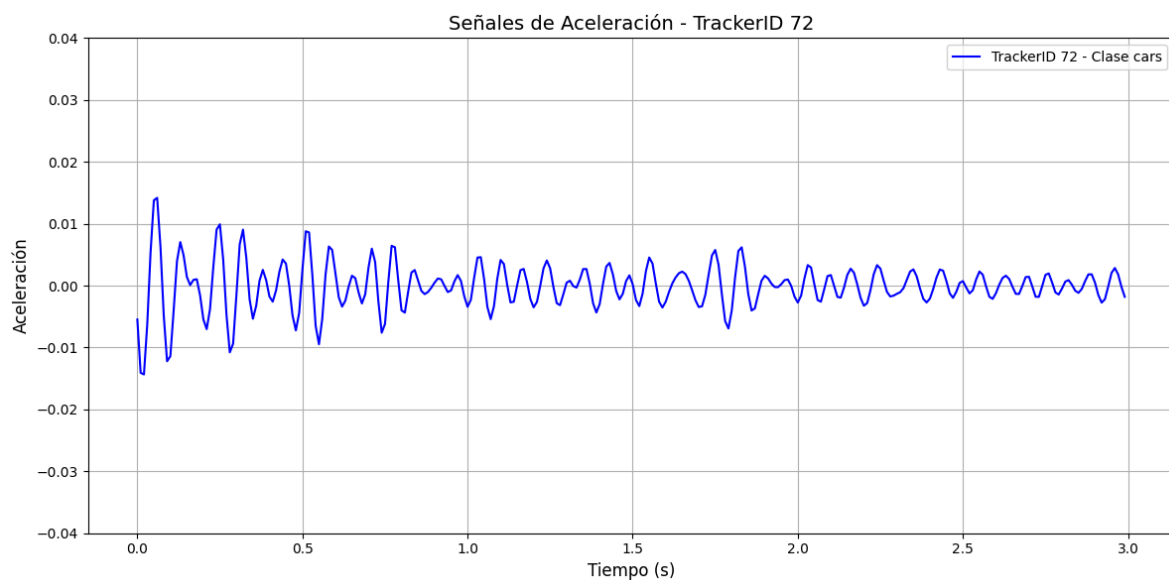


Figura 29: ventana de tiempo de auto con id 72.

3.3.7 Ejemplo 7: Moto

En la figura 30 se ve el paso de la moto que iba a la cola del vehículo analizado anteriormente, esta moto es detectada un segundo después del paso del auto, en las señales de aceleración de la figura 31 se ven oscilaciones que parecen estar más relacionados con la vibración dejada por el vehículo que pasó antes, dificultando la interpretación de la aceleración que aplica la moto por si misma, la que parece no ser muy influyente.

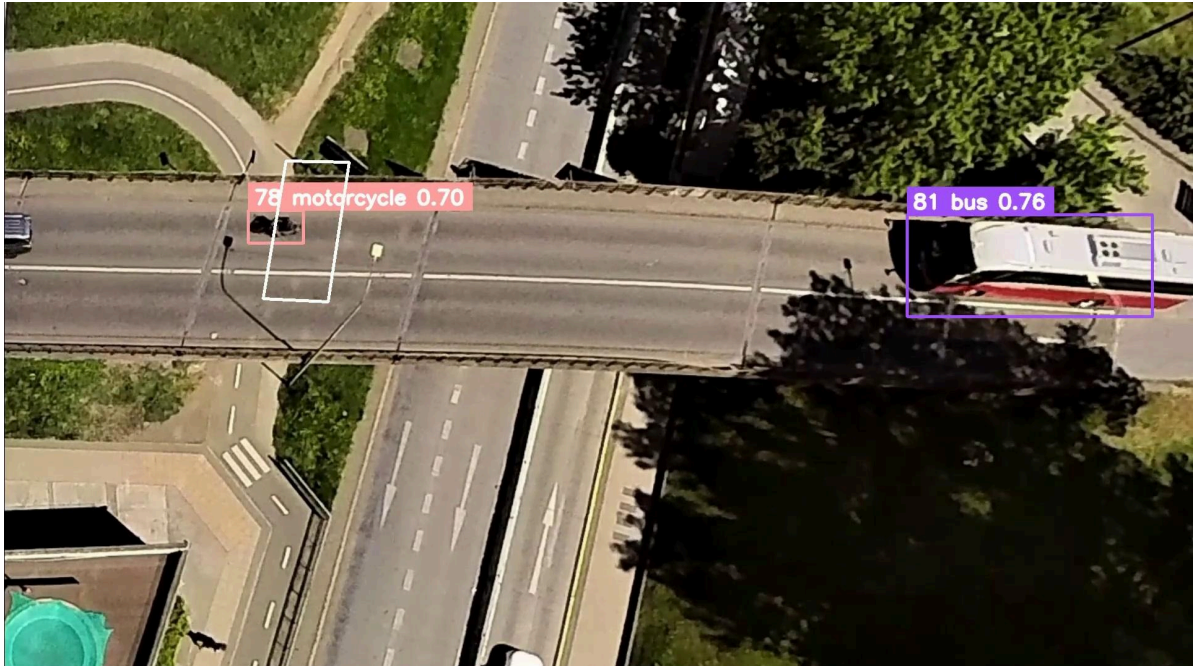


Figura 30: moto con id 78, 16:03:20.

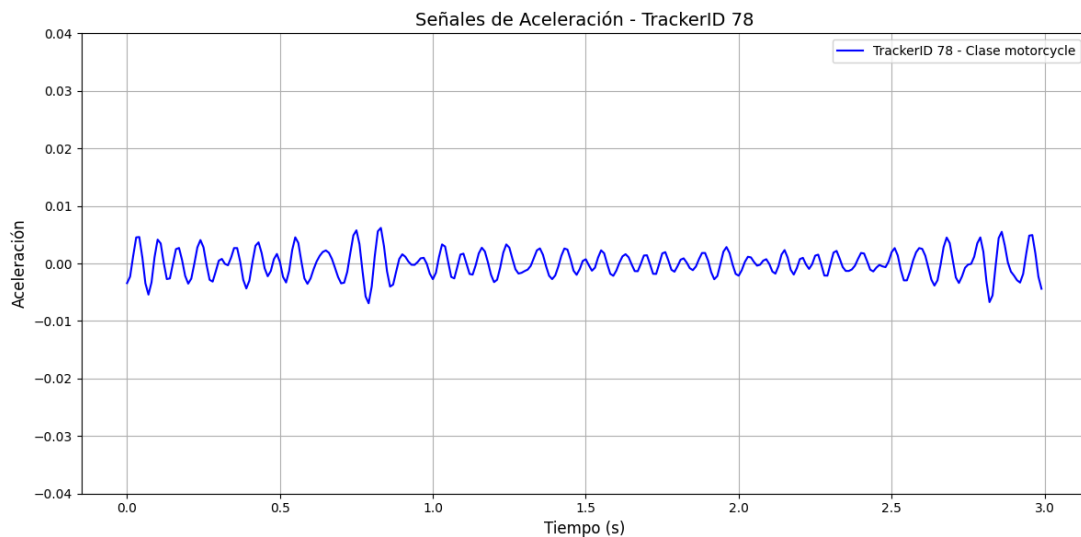


Figura 31: ventana de tiempo de moto con id 78.

3.3.8 Ejemplo 8: Conclusiones

En base a lo visto en los diferentes casos analizados, se puede observar que el peso de los vehículos más pesados facilita mucho más el análisis de las aceleraciones, ya que permite ver con mayor claridad los cambios en la señal, mientras que vehículos más livianos podrían perder características debido a la confusión que pueden dejar las vibraciones ejercidas por vehículos anteriores o vibraciones naturales del puente, como se pudo apreciar en el caso de la moto o los primeros vehículos. También se pudo ver que no hay gran diferencia en las señales de aceleración cuando pasa un vehículo con otros de peso similar cerca, comparado a cuando pasa solo.

4. Arquitectura del Sistema

Para la estructura de la arquitectura del sistema se utilizó un modelo C4, en este caso se mostrarán las 2 primeras capas, correspondientes a contexto y contenedores.

Antes de mostrar las capas es importante saber los componentes principales de esta arquitectura, en este caso está compuesto por un Frontend, el que contiene la interfaz de usuario, la visualización de los activos, en este caso puentes, el acceso a las cámaras pertenecientes a los activos y los videos procesados, junto con esto, una tabla que contiene el vehículo identificado junto a su peso estimado.

El Backend se encargará de gestionar la disponibilización de información y procesar las consultas.

La Base de datos almacenará los datos procesados, la información de las cámaras, los videos, datos de la empresa y la información de los activos.

Hay un sistema de procesamiento de datos que contiene un módulo de visión artificial, un módulo de integración y un módulo final de procesamiento para utilizar en la CNN, esto se explica en la sección 4.1.

Finalmente la CNN que recibe vectores de aceleración de parte del módulo de procesamiento, estima el peso de vehículos y entrega las predicciones.

Para la capa 1, se presenta en la figura 32 un sistema que contiene al usuario, el que puede tener un rol de usuario o administrador, donde se le disponibiliza la información en el Frontend mediante el Backend, a su vez el Backend toma la información desde la Base de datos y toda la información a la base de datos es entregada por el sistema de procesamiento y la CNN.

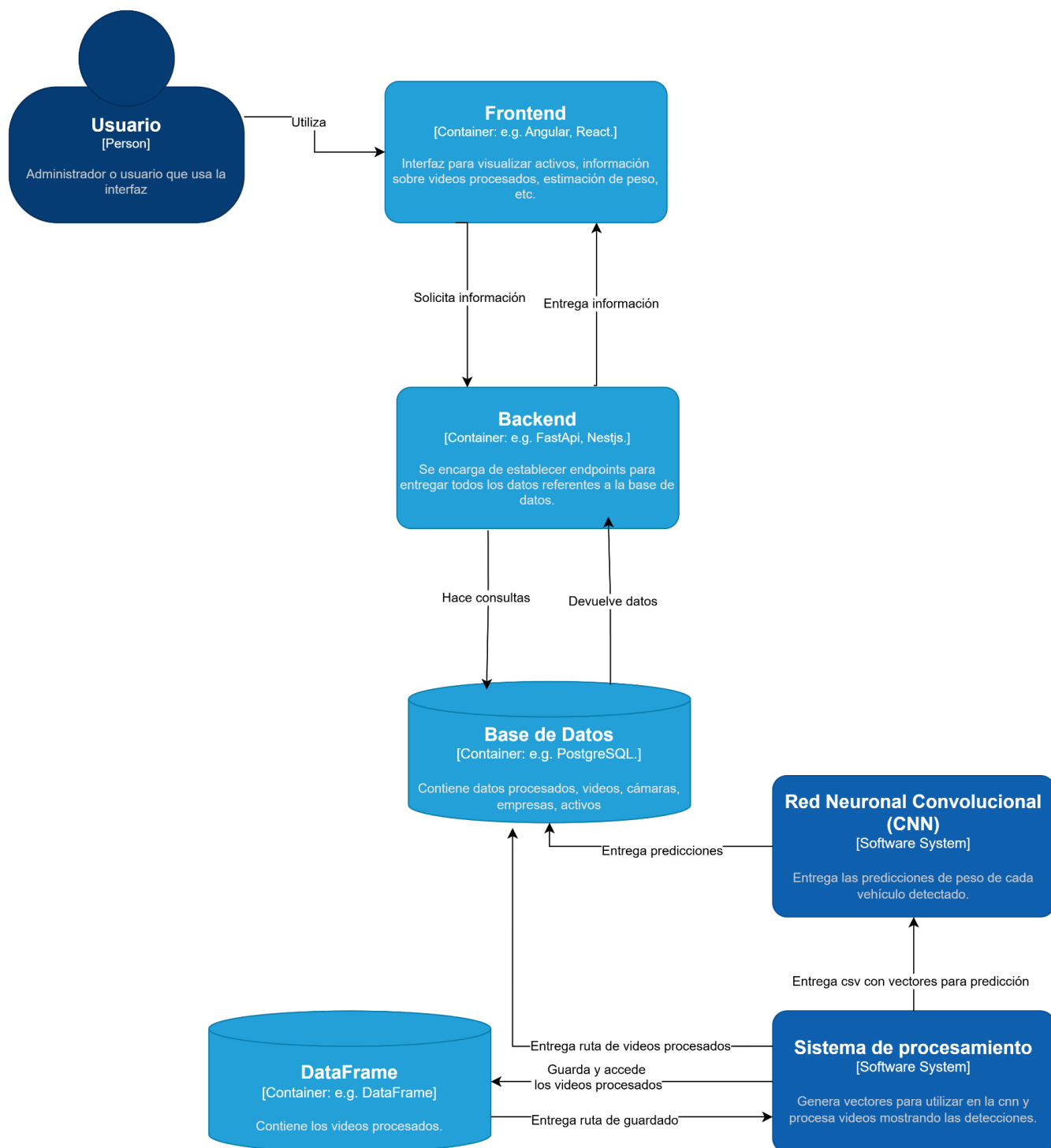


Figura 32: Diagrama de relaciones C4, capa 1.

Si revisamos más en profundidad el sistema de procesamiento podemos ver lo mostrado en la figura 33, donde podemos ver que la información del acelerómetro y videos de la cámara son enviados por agentes externos al sistema.

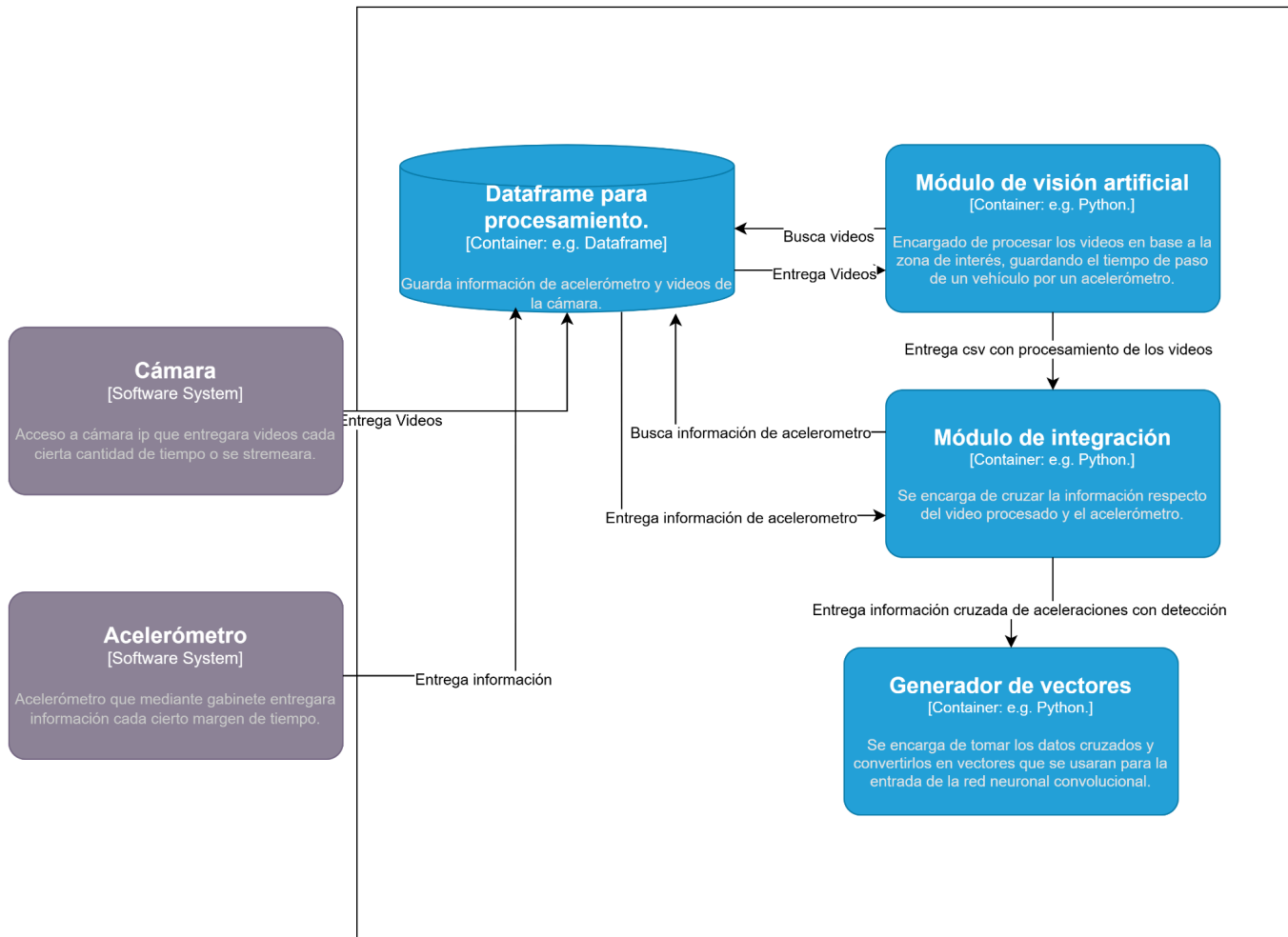


Figura 33: Parte de capa 2, mostrando el sistema de procesamiento.

La configuración inicial de un puente requiere una instalación física que incluye colocar una cámara en una posición elevada y con una vista clara del puente. Esto facilita la definición del área de detección para cada acelerómetro.

Los videos capturados por la cámara y los datos registrados por los acelerómetros son procesados por el módulo de procesamiento. Este módulo sincroniza la detección de vehículos con los datos de aceleración vertical, utilizando una ventana de tres segundos de medición asociada al paso del vehículo.

Finalmente, esta información se utiliza como entrada para una red neuronal convolucional, que se encarga de estimar el peso de los vehículos que pasan por el puente.

Para fines de este trabajo, se priorizo la implementación del sistema de procesamiento, y la aplicación web quedó como un mockup.

A continuación se utilizaran imágenes de un desarrollo pasado, para ilustrar cómo sería el funcionamiento de la aplicación, como se puede ver en la figura 34, primero se ingresa al activo, esté activo tendrá asociadas una o más cámaras.



Figura 34: Apartado de cámaras aplicación web.

Una vez se ingresa a la cámara se despliegan los videos pertenecientes a esta como se ve en la figura 35.

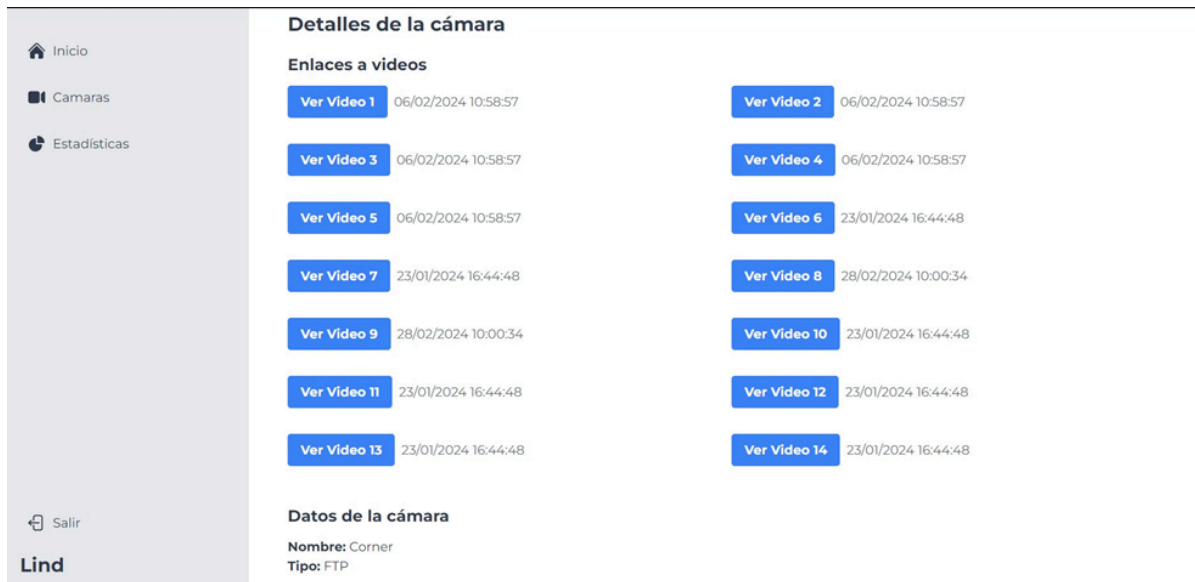


Figura 35: Lista de videos.

Cuando se ingresa al video se puede ver la vista previa del video procesado junto a información relevante, como se ve en las figuras 36 y 37.



Figura 36: Vista de video procesado

Vehículos Identificados				
ID del Vehículo	Tipo	Velocidad	Carril	Hora
7e3clac4-5bca-48d2-a93d-5a5ffcd5767f	car	46	0	23/01/2024 16:44:48
f1fffea5-c117-4cc1-ad60-91b9dd31b960	car	46	0	23/01/2024 16:44:50
d6a524f8-1932-4ba5-ab3a-71235c77f7c5	truck	46	1	23/01/2024 16:44:50
fa4216ae-7097-4176-9f78-aafb318f6c56	car	36	2	23/01/2024 16:44:50
fd26135a-762e-49dc-87ed-3913faa79c5b	car	49	0	23/01/2024 16:44:50
a406ab75-ebf9-4ea4-8820-3b4919360bbb	car	49	0	23/01/2024 16:44:51
26aa476d-ddcd-48df-b273-b5a823d2e481	car	57	0	23/01/2024 16:44:55
ad01e4fe-a3b2-490a-bcc1-5235c1cd9fd8	car	54	1	23/01/2024 16:44:56
6eb15ba2-fa6c-428d-bb29-b9887f63ae12	car	59	0	23/01/2024 16:45:01
814e74ba-859f-47f0-aa71-f008a7d02af2	car	46	1	23/01/2024 16:45:02
2c4cb47a-3787-41a6-b724-3ef2b6e54600	car	51	0	23/01/2024 16:45:04
43d6ae06-6e11-45a8-8449-defa451d33b1	car	46	1	23/01/2024 16:45:11
431c71b0-8e47-491a-aa47-c0df3b4627f5	car	57	0	23/01/2024 16:45:11
2ec7cdad-6c2c-4a12-9a2f-d6794a55c3b2	car	50	1	23/01/2024 16:45:13
d2c8419d-53b2-49f5-87d7-facef7a495ad	car	36	1	23/01/2024 16:45:20
d0ac3a5a-1a01-424c-b9c0-473042617462	car	47	0	23/01/2024 16:45:21

Figura 37: Lista con vehículos identificados.

Por otro lado, en la vista del administrador, está la posibilidad de configurar la cámara a la que se accede en ese puente, como se ve en la figura 38.

Inicio

Camaras

Estadísticas

Salir

Lind

Configuración de conexión

Dirección del servidor:

Puerto:

Nombre de usuario:

Contraseña:

Directorio Remoto:

Test de conexión

Figura 38: Configuración de cámara.

Este es el funcionamiento general del sistema, el flujo más específico de los componentes de los sistemas se listan a continuación:

4.1. Arquitectura de procesamiento de datos

Este sistema se divide en 3 módulos los cuales son:

- **Procesamiento de video con visión artificial:** Este módulo se encarga de la detección de vehículos, ingreso del ROI (región de interés o por sus siglas en inglés, Region of Interest) para guardar los vehículos que pasan por el acelerómetro y generar un video procesado y un archivo csv que contiene las detecciones.

I. Entrada: Archivo de video (mp3, avi, etc.) - archivo de configuración con el ROI.

II. Salida: Csv con detecciones de los vehículos, de la forma [TrackerID, Timestamp, Class].

Donde TrackerID corresponde al identificador del vehículo, Timestamp corresponde a la marca de tiempo en que se detectó el paso del vehículo por el acelerómetro y Class es la identificación del tipo de vehículo detectado

- **Integración video - acelerómetro:** Este módulo hace un cruce de información entre la salida del módulo antes descrito y un archivo que contiene la información cruda del acelerómetro, creando una tabla que asocia las detecciones de los vehículos con las aceleraciones verticales detectadas en el acelerómetro en el momento.

I. Entrada: Csv de detección de vehículos - csv del acelerómetro/s.

II. Salida: Csv con datos cruzados, de la forma [TrackerID, Class, Timestamp, Accelerations, accelerations_count, mean_accel, max_accel, min_accel, std_accel].

Donde TrackerID, Class y Timestamp corresponden a los mismos datos de la Salida de procesamiento de video, Accelerations corresponde a un vector de 100 aceleraciones tomados en 1 segundo por el acelerómetro, en base al tiempo de paso del vehículo, mean_accel, corresponde a la media del vector de aceleraciones, max_accel y min_accel a la máxima y mínima aceleración detectada en el vector y str_accel a la desviación estándar del vector de aceleraciones.

Durante el procesamiento se crean 3 filas para cada vehículo, que corresponde a una ventana de 3 segundos de aceleración.

- **Generación de vectores de aceleración:** Este módulo combina las filas de aceleraciones para cada vehículo y los transforma en un vector único, que se utiliza posteriormente para alimentar la red neuronal convolucional.

- I. Entrada: Csv con datos cruzados.
- II. Salida: Csv con vectores de aceleración.

En la figura, se puede ver de forma más específica cómo funciona el flujo de información, extendiendo un poco lo que se mostró anteriormente en el diagrama C4.

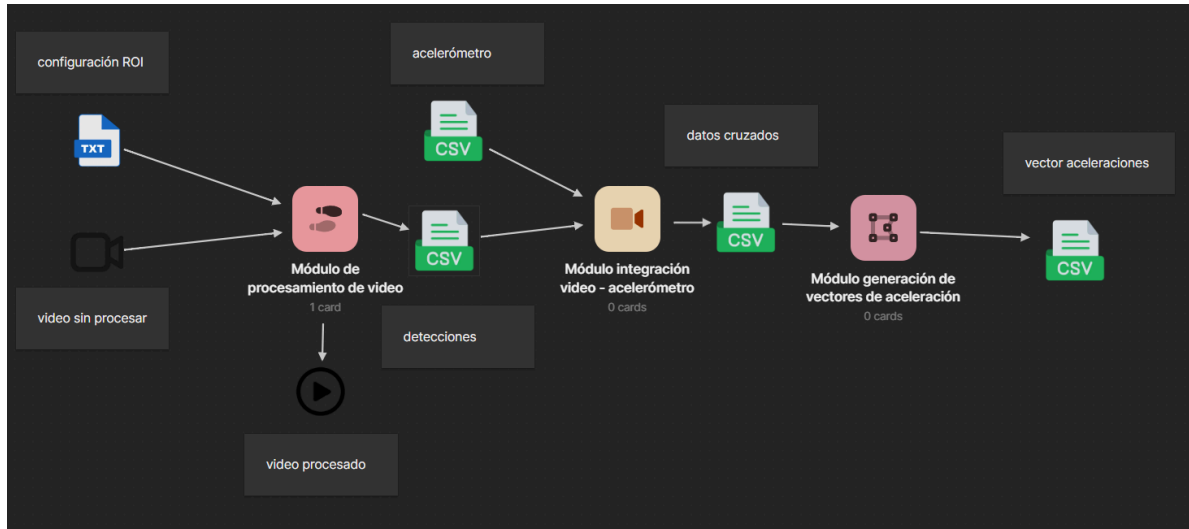


Figura 39: arquitectura de procesamiento de datos

4.2. Arquitectura CNN

La arquitectura propuesta tiene como objetivo estimar el peso de vehículos a partir de señales de aceleración vertical. Se eligió una CNN por su capacidad de procesar señales temporales y detectar patrones en los datos. La CNN recibe como entrada un vector de aceleraciones verticales captadas por un acelerómetro en una ventana de tiempo, en este caso 3 segundos.

Esta CNN está constituida por:

1. Primera capa convolucional: Convierte la entrada que corresponde al vector de aceleraciones en 20 canales de salida, utilizando un filtro o kernel de tamaño 100, esto significa que abarca 100 elementos consecutivos de la señal, esta capa ayuda a identificar características básicas de la señal, esto puede ser, tendencias locales, frecuencias dominantes o picos relacionados con las fuerzas que son inducidas por el peso de los vehículos.
2. ReLU 1: Es una función de activación que sirve para introducir no linealidad al modelo, con el fin de capturar relaciones más complejas en los datos.
3. Segunda capa convolucional: Opera sobre los 20 mapas de características generados previamente, produciendo 10 nuevos mapas mediante kernels de tamaño 50. Esta reducción permite extraer patrones de más alto nivel, combinando las características básicas detectadas

en la primera capa para formar representaciones más complejas del comportamiento dinámico del vehículo.

4. ReLu 2.
5. Primera capa completamente conectada: Reduce las características a 10 características esenciales, actuando como un cuello de botella que fuerza al modelo a retener sólo la información más relevante para la estimación del peso.
6. Segunda capa completamente conectada: Esta segunda capa se encarga de generar la salida utilizando las 10 características para estimar finalmente el peso en base a lo aprendido.

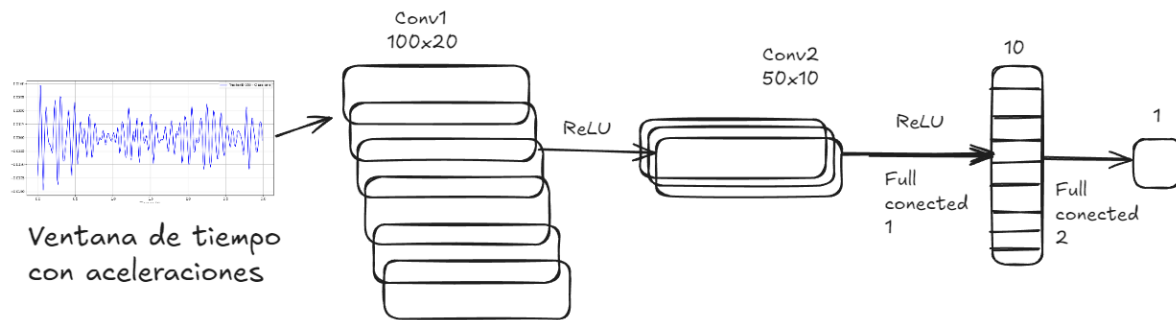


Figura 40: Modelo de red neuronal convolucional.

5. Implementación de módulos del sistema

En este capítulo se presenta el desarrollo e implementación de los componentes principales y de apoyo que conforman el sistema de pesaje dinámico vehicular.

5.1. Red neuronal convolucional (CNN)

La implementación de la CNN se desarrolló utilizando el framework PyTorch, una de las bibliotecas más populares para el aprendizaje profundo. El sistema se complementa con NumPy para el procesamiento de datos y Pandas para el almacenamiento y gestión.

El modelo se explicó con anterioridad en el capítulo 5.2.

La entrada, como se explicó previamente, corresponde a una ventana de 3 segundos donde se muestran las señales detectadas por el acelerómetro. Recibiendo una predicción de peso en kilogramos.

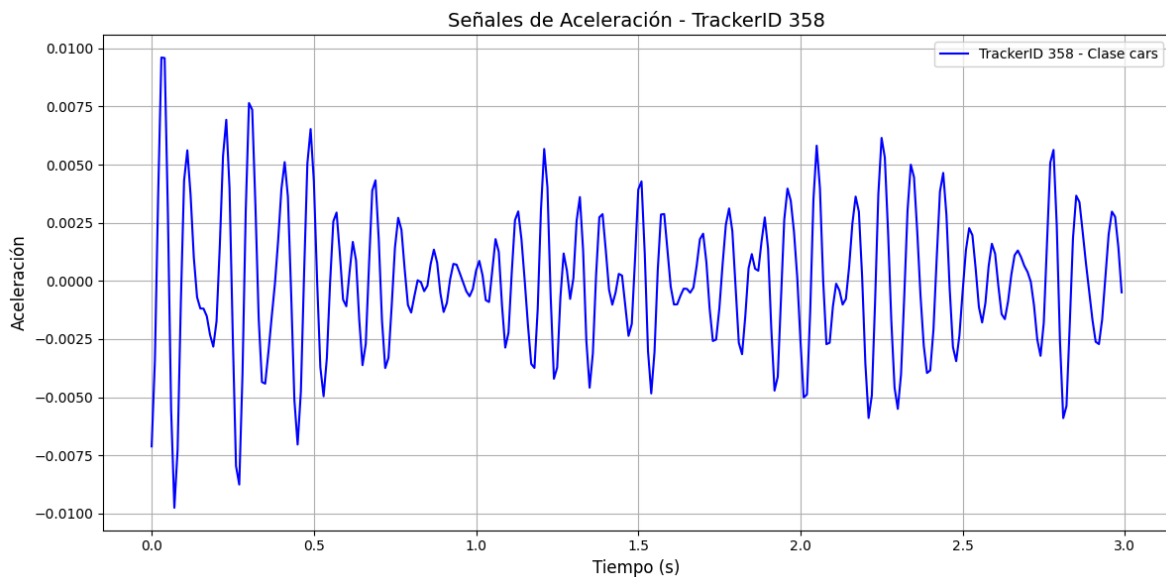


Figura 41: Ventana de tiempo de aceleraciones de un auto.

El modelo fue implementado considerando los siguientes parámetros de entrenamiento:

- Optimizador: Adam con tasa de aprendizaje de 0.001.
- Número de épocas: 1000.
- Tamaño de batch: 32.

Los datos fueron divididos de la siguiente forma:

- 80% para entrenamiento.
- 10% para validación.
- 10% para pruebas.

Las métricas con las que se evaluó el modelo consideran:

- Error Absoluto Medio (MAE).
- Raíz del Error Cuadrático Medio (RMSE).

5.2. Detección de vehículos

La implementación de este módulo de procesamiento se desarrolló utilizando YOLO (You Only Look once), en específico, una versión personalizada, entrenada específicamente para este proyecto.

El sistema se complementa con:

- OpenCV (cv2) para el procesamiento de imágenes y video.
- Ultralytics para la implementación de YOLO.
- Supervision para las herramientas de seguimiento y anotación visual.

Funcionamiento del módulo:

1. Configuración de parámetros: Se configura con tres parámetros:

- Input_video: Ruta del video a procesar.
- Txt_file: Ruta del archivo de donde se leerán los puntos de la zona de interés.
- Output_video: Ruta donde se guardará el vídeo procesado.

La zona de interés se define mediante cuatro puntos coordinados en un archivo de texto, por ejemplo:

455 257

557 259

523 485

418 480

Estos puntos definen el polígono que se utilizará para el procesamiento.



Figura 42: Zona de interés ejemplificada.

2. Detección y seguimiento: Utilizando YOLO, se clasifican los vehículos entre auto, moto, camión o bus, con un umbral de confianza configurable, y se establece el seguimiento utilizando ByteTrack, que asigna identificadores únicos a cada vehículo detectado y mantiene su trayectoria a través del procesamiento.
3. Análisis de zona de interés: Se define una zona poligonal específica para verificar la presencia de vehículos, si uno de los vehículos está dentro de la zona de interés, solo una vez por vehículo, se guardará un registro asociando el vehículo con su correspondiente marca temporal.
4. Registro: Cuando se procesa todo el video, se guarda la información de cada vehículo detectado incluyendo:
 - ID único del vehículo
 - Timestamp de detección
 - Clasificación del vehículo

Exportando los datos del vehículo para su posterior uso.

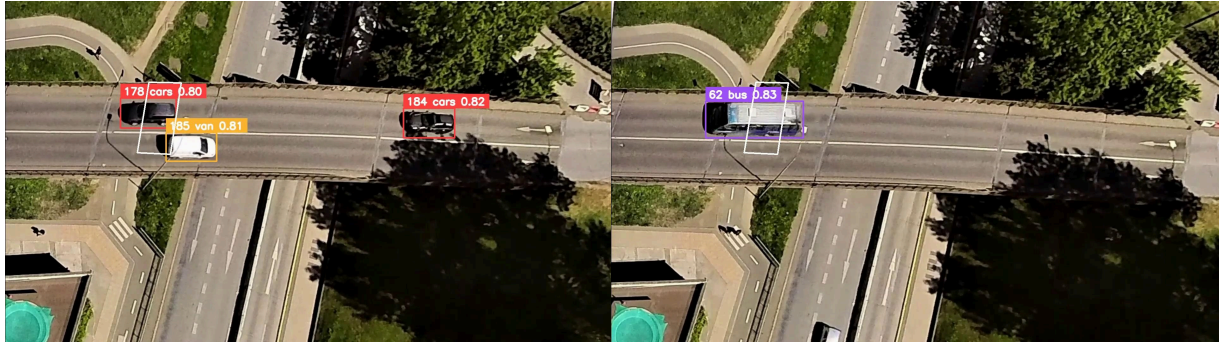


Figura 43: Procesamiento detección y seguimiento de vehículos con YOLO y ByteTrack.

Como se comentó anteriormente también se entrenó un modelo de YOLO con datos propios, esto se logró con ayuda de roboflow, que daba la facilidad de etiquetar los distintos objetos en una imagen y crear un set de datos para entrenamiento como se ve en la figura 44.

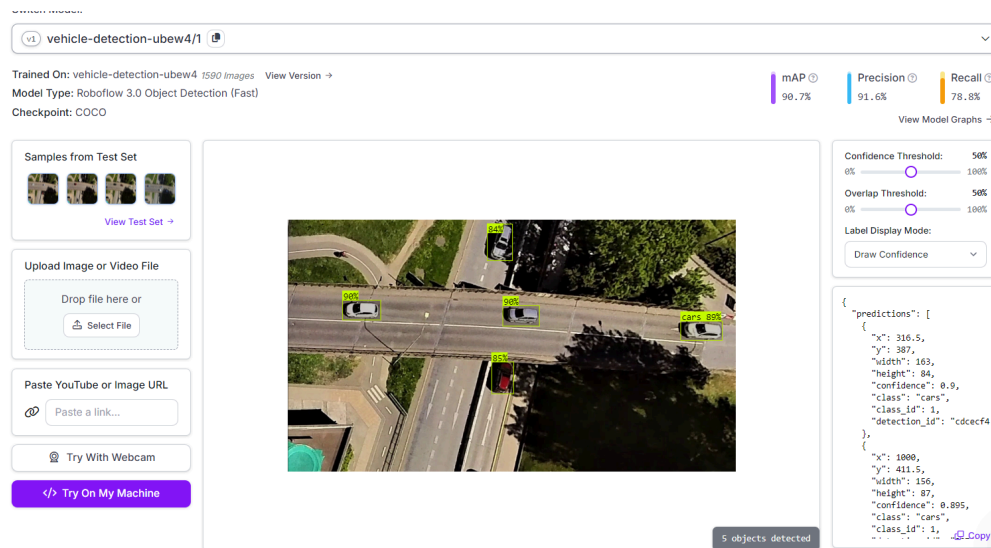


Figura 44: Interfaz de roboflow para la etiqueta y exportación del set de datos.

Luego se utilizó este set de datos en ultralytics HUB, donde finalmente se llevó a cabo el entrenamiento e importación del modelo personalizado, en la figura 45 se ven las métricas del entrenamiento hecho en ultralytics HUB.



Figura 45: Métricas de entrenamiento en ultralytics.

5.3. Selección de zona de interés

Este módulo fue diseñado como una herramienta de apoyo, para facilitar la selección precisa del área que se quiere trabajar para la detección de vehículos, mediante una interfaz visual.

Se desarrolló utilizando React.js, gestionando el dibujo en la imagen con un canvas, que permite al usuario interactuar de manera intuitiva con la selección de puntos clave.

Funcionamiento del módulo:

1. Selección de puntos: El usuario hace clic en la imagen para marcar hasta cuatro puntos que definan la zona de interés.
2. Visualización en tiempo real: Los puntos seleccionados se dibujan en el lienzo, formando un polígono cerrado una vez completados los cuatro puntos.
3. Coordenadas normalizadas: Las coordenadas se adaptan a la resolución original de la imagen, garantizando precisión en el procesamiento.
4. Gestión y eliminación de puntos: Los puntos seleccionados pueden visualizarse y eliminarse en caso de error o ajustes.
5. Multipropósito: Además de configurar zonas para detección de vehículos, también permite definir áreas de interés para otros módulos, como la detección de velocidad.

De esta forma se puede garantizar una selección rápida y precisa, minimizando los errores en esta fase. En la figura 46 se ve la interfaz principal para configurar la zona de interés, en el cual se puede

subir una imagen, en este caso del video grabado para poder seleccionar el área a evaluar. En la figura 47 se ve una imagen cargada con la zona de interés marcada y finalmente, en la figura 48 se pueden visualizar los puntos obtenidos que definen la zona de interés.

Zona de interés

Camera Configuration

Configuración

Subir Screenshot:

Seleccionar archivo Sin archivos seleccionados

Ancho:

0

Largo:

0

Guardar

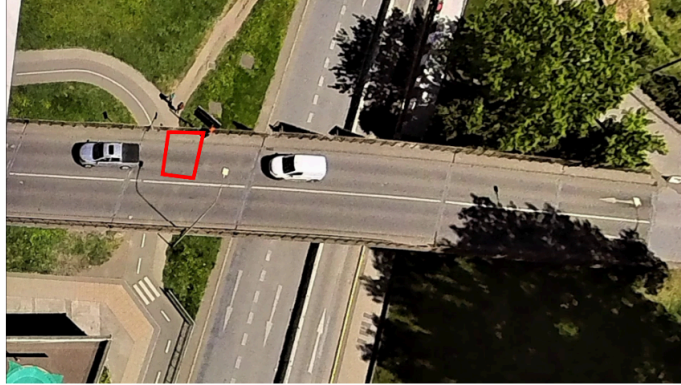
Vista previa de puntos de la zona de interés

Figura 46: Pantalla inicial de la configuración de zona de interés.

Configuración

Subir Screenshot:

Seleccionar archivo frame_0.jpg

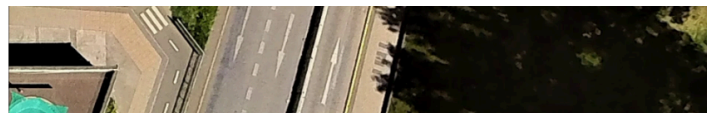


Ancho:

Largo:

Guardar

Figura 47: Imagen cargada en configuración de zona de interés.



Ancho:

Largo:

Guardar

Vista previa de puntos de la zona de interés

Punto 1: (456, 375) [Eliminar](#)

Punto 2: (554, 377) [Eliminar](#)

Punto 3: (526, 504) [Eliminar](#)

Punto 4: (439, 493) [Eliminar](#)

Figura 48: Puntos seleccionados en configuración de zona de interés.

6. Resultados

Luego de 2000 épocas para el entrenamiento, el modelo muestra la capacidad para identificar patrones generales sobre las diferencias de peso. Se debe considerar que los valores que se tratan como ‘pesos reales’ son en realidad estimaciones basadas en la clasificación de los vehículos y observaciones extraídas de la segunda recolección (sección 3.1.2), lo que impide evaluar completamente la precisión del modelo, pero sí permite analizar el comportamiento general frente a este rango de peso estimado. En la figura 49 se puede ver la predicción de pesos en contraste con su peso real (entendiéndose como el peso estimado según clasificación y/o modelo del vehículo), utilizando el conjunto de prueba.

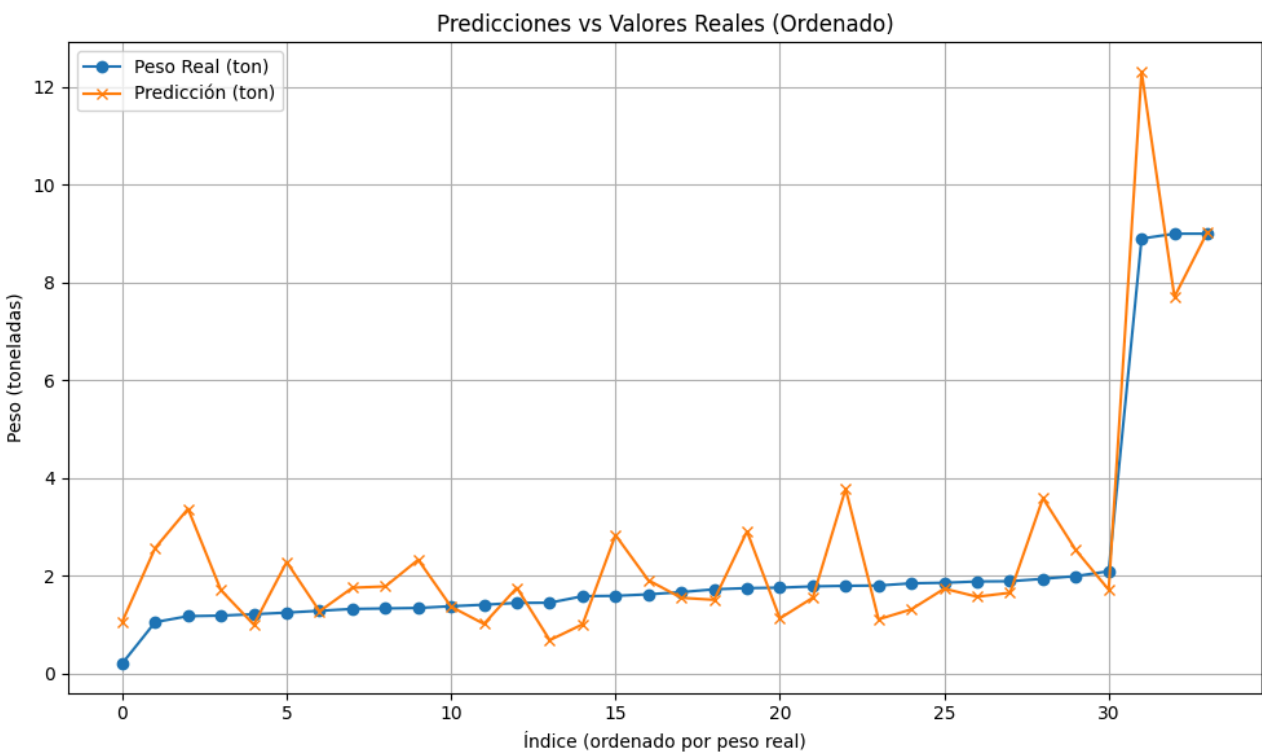


Figura 49: Predicciones de peso.

El RMSE (por sus siglas en inglés, Root Mean Square Error) y el MAE (por sus siglas en inglés, Mean Absolute Error) [35] son dos métricas para evaluar el error en modelos de regresión. El MAE mide el promedio de los errores absolutos entre las predicciones y los valores reales, por su parte, el RMSE penaliza los errores más grandes, lo que lo hace más sensible a valores atípicos, las fórmulas para calcular el MAE y el RMSE se pueden ver en las figuras 50 y 51.

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Figura 50: Fórmula MAE.

donde:

- n es el número de elementos.
- y_i es el valor real.
- x_i es el valor predicho.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}$$

Figura 51: Fórmula RMSE.

donde:

- n es el número de elementos.
- y_i es el valor real.
- x_i es el valor predicho.

En base a las evaluaciones de error mostradas en la tabla 1, se puede ver un castigo alto para los vehículos más ligeros, que tienden a un rango de entre 900 kilogramos a 2 toneladas, sin embargo, es un error bastante tolerable si se habla de vehículos pesados, hablando por ejemplo de 9 toneladas (considerando peso en vacío), ya que si se habla de vehículos más pesados, como camiones de 3 o 4 ejes, se hablaría de una magnitud que puede ir desde las 10 toneladas hasta las 30 o más. Para las aplicaciones prácticas los objetos de mayor interés son precisamente estos vehículos pesados, lo que da buenas señales para una implementación de este modelo en escenarios reales.

RMSE	MAE
Peso (kg)	Peso (kg)
1130	744

Tabla 1: Tabla de Error cuadrático medio y Error absoluto medio.

En la figura 52 se puede ver la evolución de la pérdida durante el entrenamiento y validación del modelo. La pérdida refleja el MSE [35] (por sus siglas en inglés, Mean Square Error), que ayuda a ver que tan bien se ajusta el modelo a los datos, en este caso se puede ver un inicio caótico, alrededor de la época 250 se estabilizan las pérdidas, pero estando muy alejado el entrenamiento de la validación, y llegando a la época 1500 se puede ver como empiezan a converger, lo que da señales del aprendizaje del modelo.

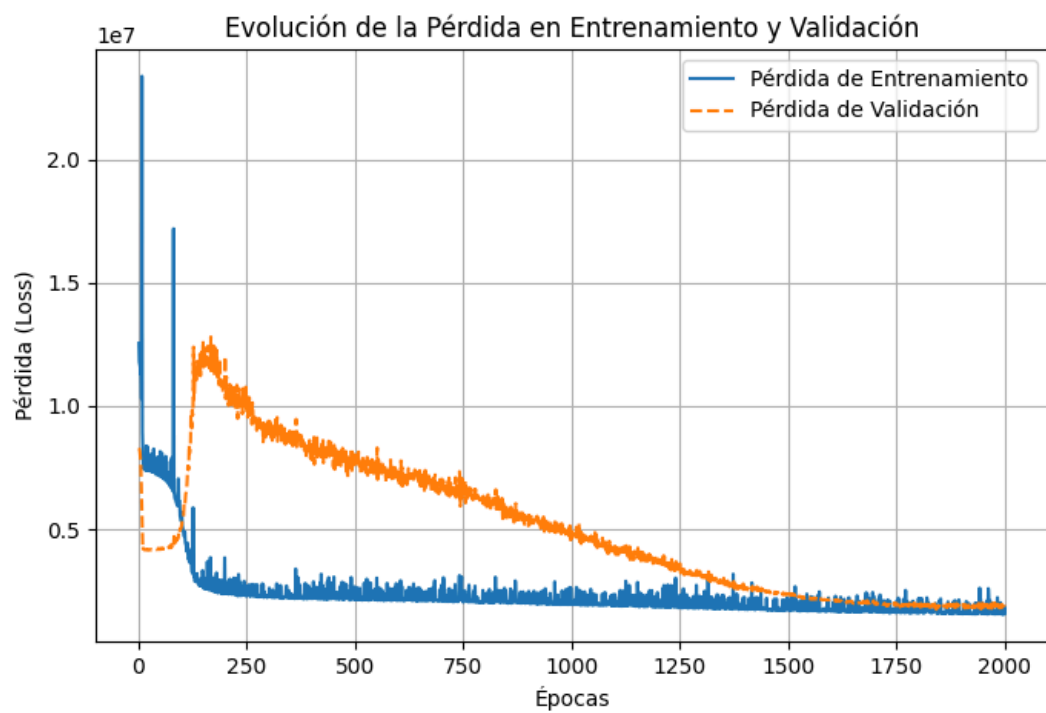


Figura 52: Evolución de la pérdida en entrenamiento y validación.

7. Siguientes pasos

Aún hay una serie de puntos importantes y de mejora por abordar en este modelo.

- Primero y principal, entrenar el modelo con pesos reales, para ver cómo se comporta la estimación. En la segunda obtención de datos se intentó estimar con más exactitud el peso, pero solo en base al modelo del vehículo y tomando el peso en vacío, por lo que los resultados pueden cambiar notablemente.
- La utilización de wavelets [23], que no son más que ondas pequeñas que ayudan a representar datos con variaciones o discontinuidades, esto implicaría un cambio en la entrada del modelo, pero puede ser un punto de mejora, ayudando a identificar patrones, filtrar ruido, etc.
- Prueba del modelo en un escenario real, para ver cómo se adapta a nuevos datos, principalmente en diferentes estaciones, ya que la humedad, viento, estado del puente, afectan la medición.
- Exploración de otras arquitecturas para el modelo, principalmente, redes recurrentes, ya que suelen capturar de mejor forma la naturaleza temporal de los datos, una opción puede ser LSTM.

8. Conclusión

En la presente memoria se logró la implementación de los sistemas propuestos, como lo son el sistema de visión artificial para identificar los vehículos, como también un sistema que permite estimar un peso en este caso utilizando una red neuronal convolucional, con ayuda de un acelerómetro y una cámara, por otro lado se utilizó una configuración tanto para la cámara como el acelerómetro, para poder obtener la información necesaria y finalmente se pudo modelar una arquitectura donde se puede hacer uso de este sistema de pesaje en movimiento no intrusivo, en el que se puede visualizar la información obtenida por los sistemas implementados.

Un tema que queda pendiente es la validación de los resultados de la red neuronal convolucional ya que se requiere de más recursos, la posibilidad de tener pesos reales, para hacer pruebas más acertadas y robustas.

El enfoque del sistema de pesaje en movimiento no intrusivo muestra mejor respuesta en vehículos pesados, ya que se vio que generan más información caracterizable, comparado con los más ligeros cuya información tiende a confundirse con la vibración natural del puente, esto es especialmente positivo, ya que son los vehículos pesados, los que más afectan la infraestructura del puente.

Este proyecto requiere una colaboración interdisciplinaria, ya que se necesitan en gran medida conocimientos estructurales, la capacidad de instrumentar y extraer datos y la creación del modelo para interpretar todos estos datos y lograr el sistema propuesto.

En general, hay una serie de pasos y pruebas con las que se puede seguir iterando, pero se sienta una base que muestra resultados, a pesar de la precariedad de los datos y se muestra una forma de aplicar inteligencia artificial en el ámbito civil.

9. Referencias

1. Wang, H., Nagayama, T., Kawakatsu, T., & Takasu, A. (2023). A Data-Driven Approach for Bridge Weigh-in-Motion from Impact Acceleration Responses at Bridge Joints. Structural Control and Health Monitoring, 2023, 2287978. <https://doi.org/10.1155/2023/2287978>
2. Lydon, M., Taylor, S. E., Robinson, D., Mufti, A., & Brien, E. J. O. (2016). Recent developments in bridge weigh in motion (B-WIM). Journal of Civil Structural Health Monitoring, 6. <https://doi.org/10.1007/s13349-015-0119-6>
3. Yu, Y., Cai, C., & Deng, L. (2016). State-of-the-art review on bridge weigh-in-motion technology. Advances in Structural Engineering. <https://doi.org/10.1177/1369433216655922>
4. Shi, S., Yarnold, M., & Mander, J. (2024). Evaluation of a load cell-based bridge weigh-in-motion system. OnlineFirst. <https://doi.org/10.1177/03611981241263336>
5. <https://docs.ultralytics.com/es> Recuperado el 10 de noviembre de 2024
6. <https://www.redecam.cl/preocupacion-por-el-estado-de-los-puentes-en-chile-especialmente-en-rutas-de-transito-de-camiones-pesados/> Recuperado el 10 de noviembre de 2024,
7. <https://eadic.com/blog/entrada/mantenimiento-de-puentes/> Recuperado el 10 de noviembre de 2024
8. <https://cooperativa.cl/noticias/pais/region-de-los-rios/tras-una-decada-de-verguenzas-afirman-que-ahora-si-el-puente-cau-cau/2024-01-21/105116.html> Recuperado el 10 de noviembre de 2024
9. <https://www.camaweigh.com/blog/post/what-is-the-difference-between-static-and-dynamic-weighing/?srsltid=AfmBOortJnkGq6V7ltrqqBR-rNgWcOSidKCDq8HeaPClp4bpeevPyNG> Recuperado el 10 de noviembre de 2024
10. https://www.youtube.com/watch?app=desktop&v=vrx2TQaF7_I&ab_channel=TxDOT Recuperado el 10 de noviembre de 2024
11. Wikipedia. (n.d.). Homografía (geometría). En Wikipedia. Recuperado el 10 de noviembre de 2024, de [https://es.wikipedia.org/wiki/Homografia_\(geometría\)](https://es.wikipedia.org/wiki/Homografia_(geometría))
12. Gillis, A. S., Craig, L., & Awati, R. (s.f.). What is a convolutional neural network (CNN)? TechTarget. Recuperado el 10 de noviembre de 2024, de <https://www.techtarget.com/searchenterpriseai/definicion/convolutional-neural-network>

13. Developers Breach. Convolutional Neural Network in Deep Learning. Recuperado el 10 de noviembre de 2024, de <https://developersbreach.com/convolution-neural-network-deep-learning/#1-what-is-cnn>
14. Byrd, E. (2022, April 7). What is a convolution? How to teach machines to see images. 8th Light Insights. Recuperado el 10 de noviembre de 2024, de <https://8thlight.com/insights/what-is-a-convolution-how-to-teach-machines-to-see-images>
15. C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, 2nd ed., Cham, Switzerland: Springer, 2023. [Online]. Available: <https://doi.org/10.1007/978-3-031-29642-0>.
16. Nagyfi, R. (2018, September 4). *The differences between artificial and biological neural networks*. Towards Data Science. <https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7>
17. CXAI: Explaining convolutional neural networks for medical imaging diagnostic. (2022, June). *Electronics*, 11(11), 1775. <https://doi.org/10.3390/electronics11111775>
18. Google. (n.d.). *Clasificación multiclase con redes neuronales*. Google Developers. <https://developers.google.com/machine-learning/crash-course/neural-networks/multi-class?hl=es-419>
19. Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, 201, 123-135. <https://doi.org/10.1016/j.procs.2022.01.135>
20. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788). IEEE. <https://doi.org/10.1109/CVPR.2016.91>
21. Sumit, Bisht, S., Joshi, S., & Rana, U. (2024). Comprehensive review of R-CNN and its variant architectures. *International Research Journal on Advanced Engineering Hub (IRJAEH)*, 2, 959-966. <https://doi.org/10.47392/IRJAEH.2024.0134>
22. Diario Financiero. (2024, 2 de abril). Transporte de carga: camiones livianos, medianos y pesados. https://www.df.cl/noticias/site/docs/20240401/20240401181903/suplemento_20240402_oko_k.pdf

23. Castro, L. R., & Castro, S. M. (s.f.). Wavelets y sus Aplicaciones. Universidad Nacional del Sur, Departamento de Matemática - Departamento de Computación, Bahía Blanca, Argentina.
https://sedici.unlp.edu.ar/bitstream/handle/10915/24289/Documento_completo.pdf?sequence=1&isAllowed=y#:~:text=Las%20wavelets%20
24. Data Science Process Management. (s.f.). OSEMN Framework. Data Science PM. de
<https://www.datascience-pm.com/osemn/>
25. Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z., & Li, H. (2019). The State of the Art of Data Science and Engineering in Structural Health Monitoring. Engineering.
<https://doi.org/10.1016/j.eng.2018.11.027>
26. Deng, Y., Zhao, Y., Ju, H., Yi, T. H., & Li, A. (2024). Abnormal data detection for structural health monitoring: State-of-the-art review. Developments in the Built Environment.
<https://doi.org/10.1016/j.dibe.2024.100337>
27. Wang, H., Nagayama, T., & Su, D. (2019). Estimation of dynamic tire force by measurement of vehicle body responses with numerical and experimental validation. Mechanical Systems and Signal Processing. <https://doi.org/10.1016/j.ymssp.2019.01.017>
28. Deng, L., & Cai, C. S. (2009). Identification of parameters of vehicles moving on bridges. Engineering Structures. <https://doi.org/10.1016/j.engstruct.2009.06.005>
29. Seguel Anativia, J. I. (2019). Bases metodológicas para implementación de sistemas de instrumentación y monitoreo de salud estructural en tiempo real de puentes en Chile (Memoria para optar al título de Ingeniero Civil). Universidad de Chile, Santiago de Chile. extraído de
https://repositorio.uchile.cl/bitstream/handle/2250/173750/cf-seguel_ja.pdf?sequence=1
30. SkyCiv. (s.f.). ¿Qué es un momento flector?. Recuperado el 4 de marzo de 2025, de
<https://skyciv.com/es/docs/tutorials/beam-tutorials/what-is-bending-moment/>
31. Vivanco, J. (2019, 5 de junio). El modelo C4 de documentación para la Arquitectura de Software. Medium.
<https://medium.com/@javiervivanco/el-modelo-c4-de-documentaci%C3%B3n-para-la-arquitectura-de-software-424704528390>
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions (arXiv:1409.4842). arXiv.
<https://doi.org/10.48550/arXiv.1409.4842>

33. J. Redmon, "Darknet: Open Source Neural Networks in C," 2013–2016. [Online]. Available: <http://pjreddie.com/darknet/>
34. C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.
35. Arrioja Landa Cosio, N. (2021, diciembre 20). *Métricas en regresión*. Medium. <https://medium.com/@nicolasarrioja/métricas-en-regresión-5e5d4259430b>

10. Anexos

Anexo 1: Módulo de detección de velocidad.

Este es un módulo que no fue utilizado para el enfoque final de estimación de peso, pero puede tener valor para futuras iteraciones o incluso un enfoque diferente.

El módulo de detección de velocidad se desarrolló utilizando las mismas tecnologías que en detección de vehículos, estos son, YOLO, ByteTrack y Supervisión, el que permite la gestión y procesamiento de videos, facilitando la visualización y almacenamiento de los resultados obtenidos, utilizando de base OpenCV.

Se utilizaron una combinación de técnicas y configuraciones específicas para calcular la velocidad de los vehículos de manera confiable:

- Homografía: Para transformar las coordenadas 2D de la imagen en coordenadas reales del mundo. Este mapeo es fundamental para convertir las distancias medidas en píxeles a distancias reales (metros) y obtener velocidades en unidades reales (kilómetros por hora). Esto es parte del dominio de la geometría proyectiva.
- Parámetros de la cámara: Se debe tener en cuenta la posición de la cámara y como se muestra la imagen, esto es, claridad con la distancia focal, el centro óptico, la posición y orientación de la cámara, lo importante es minimizar la distorsión al momento de grabar, de forma que la transformación de las coordenadas de píxeles a metros sea lo más exacta posible, de esta manera, aumenta en gran medida la precisión en la estimación de la velocidad de los vehículos. Los parámetros de la cámara conllevan otro dominio el cual no se considera en este informe.

Funcionamiento del módulo:

1. Configuración de Parámetros: se configura con 5 parámetros:
 - Input_video: Ruta del video a procesar
 - File_txt: Ruta del archivo de donde se leerán los puntos de la zona de interés.
 - Output_video: Ruta donde se guardará el vídeo procesado.
 - Width: Ancho calculado del cuadrilátero.
 - Height: Altura calculada del cuadrilátero.
2. Los vehículos detectados dentro de la zona de interés son rastreados continuamente utilizando ByteTrack.

3. A partir de las trayectorias obtenidas, se calcula la velocidad de cada vehículo transformando las distancias en píxeles a distancias reales mediante la homografía.
4. Los resultados se registran en un archivo, incluyendo:
 - ID único del vehículo.
 - Velocidad estimada.
 - Marca temporal asociada al cálculo.

En la figura 53 se puede ver una vista previa de cómo funciona la detección de velocidad de un vehículo.

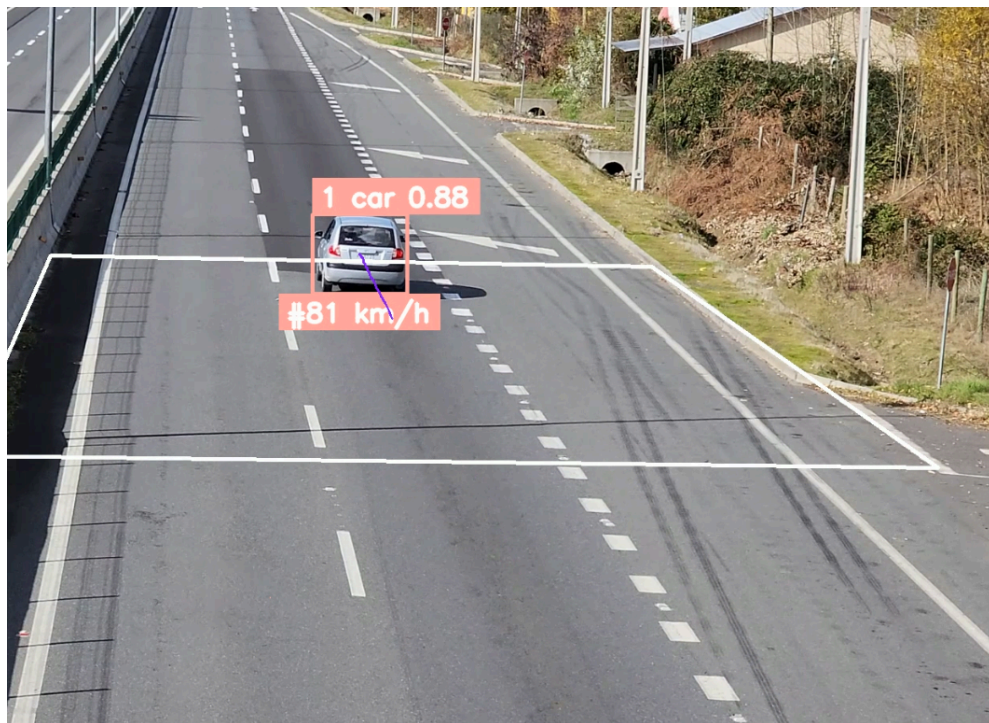


Figura 53: Detección de velocidad.

Anexo 2: Red neuronal.

Las redes neuronales artificiales [16] son técnicas populares de aprendizaje automático que simulan el mecanismo de aprendizaje en los organismos biológicos. El sistema nervioso humano contiene células, que se denominan neuronas. Las neuronas están conectadas entre sí mediante axones y dendritas, y las regiones de conexión entre axones y dendritas se denominan sinapsis. La fuerza de las conexiones sinápticas a menudo cambia en respuesta a estímulos externos. Este cambio es la forma en que se produce el aprendizaje en los organismos vivos [15]. En la figura 54 se puede observar la relación de una neurona artificial comparada con una neurona del sistema nervioso humano.

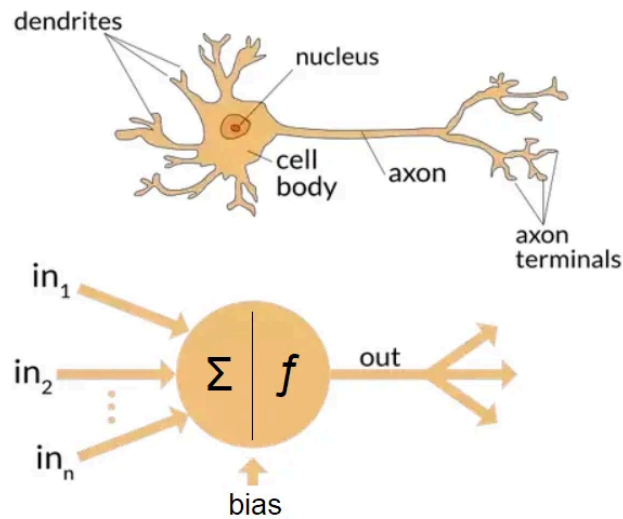


Figura 54: red neuronal biológica y artificial