UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

FACULTAD DE INGENIERÍA

Reconocimiento de la intención del peatón de cruzar una vía urbana mediante Inteligencia Artificial.

TESIS

Como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta
Ing. Armando Silva Veláquez

Dirigido por:
Dr. Andras Takacs

Querétaro, Qro. a 5 de septiembre de 2023

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

FACULTAD DE INGENIERÍA

MAESTRÍA EN CIENCIAS EN INTELIGENCIA ARTIFICIAL

Reconocimiento de la intención del peatón de cruzar una vía urbana mediante Inteligencia Artificial.

TESIS

Como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta
Armando Silva Velázquez

Dirigido por:
Dr. Andras Takacs

Dr. Andras Takacs
Presidente

Dr. Jesús Carlos Pedraza Ortega
Secretario

Dr. Juan Manuel Ramos Arreguín
Vocal

Dr. Gonzalo Macias Bobadilla
Suplente

Dra. Adriana Rojas Molina
Suplente

Centro Universitario, Querétaro, Qro.
5 de septiembre de 2023
México

A mis padres y familia por su apoyo incondicional.

Agradecimientos

A mi director de tesis por guiarme a lo largo de todo el programa y compartirme sus conocimientos, y experiencias profesionales y personales que me han servido en mi desarrollo profesional en el posgrado.

Sinodales por su dedicación y disposición de ayudar en cualquier momento.

A mis profesores por dedicación dar la mejor clase y estar dispuestos a resolver cualquier duda en cualquier momento.

A la Universidad Autónoma de Querétaro y la Facultad de Ingeniería por los apoyos económicos y por las instalaciones.

Al Consejo Nacional de Ciencia y Tecnología por apoyar con el financiamiento del programa de posgrado.

RESUMEN

Reconocer la intención de cruce de peatones es una de las tareas más importantes y ampliamente investigadas en esta era de autos inteligentes y conducción autónoma. El objetivo de reconocer las intenciones del peatón alrededor de los vehículos autónomos es realizar acciones evasivas en situaciones potencialmente peligrosas para evitar accidentes con peatones, ciclistas, animales, personas discapacitadas y otras personas que se acerquen al automóvil. El proceso de reconocimiento comienza con la recuperación de imágenes de peatones a partir de vídeos en tiempo real con aplicaciones de visión artificial. Luego, para categorizar la intención de cruce del peatón, estas imágenes se procesan para extraer características para el reconocimiento de patrones de comportamiento. Esta tesis estudia la combinación de técnicas de aprendizaje automático y profundo para extraer características de los peatones, como los ángulos internos de las rodillas generados por el paso natural del peatón, la orientación de la cabeza y también el giro, y la consideración de otros factores externos como el paso de peatones, semáforo y señal de alto. Analizar las características que aportan los algoritmos es parte fundamental para determinar la intención del peatón ya que juegan un papel importante otras muchas variables externas, además de las consideradas, como la climatología o estación del año y, sobre todo, la cultura del peatón en función del país donde reside. Las investigaciones sobre el tema y la experiencia adquirida durante el posgrado indican que la estructura del modelo para inferir rápidamente la intención del peatón debe ser secuencial y robusta. Para tener un algoritmo robusto, considere más variables y encuentre el equilibrio adecuado entre velocidad y rendimiento.

Palabras clave : Predicción, Peatón, Inteligencia artificial, Aprendizaje profundo, Aprendizaje maquina.

ABSTRACT

Recognizing the pedestrian crossing intention is among the most important and widely researched tasks in this age of smart cars and autonomous driving. The objective of recognizing the pedestrian's intentions around autonomous vehicles is to take evasive action in potentially dangerous situations to avoid accidents with pedestrians, cyclists, animals, disabled people, and others who come near the car. The recognition process starts with retrieving images of pedestrians from videos in real time with computer vision applications. Then to categorize the pedestrian's crossing intention, these images are processed to extract features for behaviour pattern recognition. This thesis studies the combination of deep and machine learning techniques to extract pedestrian characteristics, such as the internal knee angles generated by the natural gait of the pedestrian, the head orientation and turn also, and the consideration of other external factors such as pedestrian crossing, traffic light and stop sign. Analyzing the characteristics provided by the algorithms is an essential part of determining the pedestrian's intention since many other external variables, in addition to those considered, play an important factor, such as the weather or season of the year and, above all, the behavioral culture of the country of the pedestrian. The research on the subject and the gained experience during the postgraduate course indicate that the model structure to quickly infer the pedestrian's intention should be sequential and robust. To have a robust algorithm, consider more variables, and find the right balance of speed and performance.

Keywords: Prediction, Pedestrian, Artificial intelligence, Deep learning, Machine learning.

ÍNDICE

ÍNDICE DE FIGURAS

Lista de Acrónimos

ML Aprendizaje Automatico o Machine Learning

DL Aprendizaje Profundo o Deep Learning

WHO World Health Organization

AI Artificial intelligence

AV Autonomous Vehicles

CNN Convolutional Neural Network

FPS Frames Per Seconds

RF Ramdom Forest

SVM Support vector machine

KNN K-nearest neighbors

SAE Society of Automotive Engineers

ILP Integer Linear Programming

RNN Recurrent Neural Network

DNN Deep Neural Networks

VRU Vulnerable Road User

HDV Human Driven Vehicles

FAV Fully Automated Vehicles

C Crossing

NC No Crossing

MLP Multilayer Perceptron

JAAD Joint Attention in Autonomous Driving

GRU Gated Recurrent Unit

GB Gradient Boost

XGB Extreme Gradient Boost

RGB Red Blue Green

NN Neural Networks

LR Linear regression

ADASYN Adaptive synthetic sampling approach for imbalanced learning

PCA Principal Component Analysis

VP Verdaderos Positivos

VN Verdaderon Negativos

FP Falsos Positivos

FN Falsos Negativos

PR Precision vs Recall

ROC receiver operating characteristic

AUC Area Under Curve

DFL Distributional Focal Loss

ResNet Residual Neural Network

YOLO You Only Look Once

1. INTRODUCCIÓN

1.1. Introducción

La inteligencia artificial con los métodos ML por sus siglas en inglés y Aprendizaje Profundo o Deep Learning (DL) por sus siglas en inglés ha llegado hasta los automóviles de hoy en día para convertirlos en autos autónomos o vehículos inteligentes, esto con el fin de salvaguardar la seguridad y la integridad no solo de sus ocupantes sino también de los usuarios de la vía pública como lo son peatones de cualquier tipo, ciclistas o incluso animales.

Hoy en día se utiliza la detección de peatones mediante técnicas del área de DL para así extraer características tomadas de cada imagen (frame) de un vídeo en tiempo real, esta tarea de detección y extracción de características es un área bastante madura ya que se cuenta con muchos modelos que realizan estas tareas, como lo son YOLOVn (técnica de un paso) o ResNet (técnica de dos pasos), entre muchas otras. En la parte de ML, existen métodos que pueden tomar estas características y hacer una predicción de la intención. En base a lo investigado en el estado del arte, la determinación de las características mínimas necesarias para el reconocimiento de la intención de un peatón es un área que aún se sigue siendo investigada, ya que diferentes investigaciones utilizan diferentes características de una imagen o de un peatón como por ejemplo la orientación de la cabeza del peatón, uso de gestos del rostro o manos, orientación del torso o incluso del ambiente como las señales peatonales o vehiculares. Es por esto por lo que es la presente tesis se pretender usar algunas características ya usadas más la consideración de los ángulos internos generados por las rodillas de un peatón mediante técnicas ya conocidas de DL y ML para lograr un mejor desempeño para la velocidad y/o robustez y las métricas correspondientes a este ámbito.

1.2. Justificación

La Organización Mundial de la Salud [10] (World Health Organization (WHO) por sus siglas en ingles), reporta que en el mundo cada 1:41 minutos muere un peatón, 33 al día, 23,938 por mes y 257,007 por año que se ven involucrados en un accidente con un vehículo por varias razones, el exceso de velocidad, estar bajo efectos de alguna sustancia, distracciones por parte del conductor y también se debe a la mala infraestructura. Para México reporta por cada 100 mil personas 3.7 peatones mueren. Todas estas cifras son solo por accidentes terrestres.

WHO muestra que una ligera tendencia en aumento de las fatalidades a nivel mundial hasta el año 2016, y es por eso por lo que hoy día se ve o se suele hablar de carros inteligentes o autónomos, los cuales se conducen solos o tiene un grado de autonomía. Esto con el fin de reducir las fatalidades de los peatones. Y para lograr esto hay muchos estudios tratando de mejorar las técnicas para sensar, detectar, reconocer y trazar el comportamiento del peatón para que el carro inteligente tome decisiones por sí mismo y así poder evitar una colisión con el peatón.

A pesar de que en la actualidad, como ya se vio en los antecedentes, se están utilizando métodos de Inteligencia Artificial (AI por sus siglas en ingles) para poder lograr llegar a los Vehículos Autónomos (Autonomous Vehicles (AV) por sus siglas en inglés) los cuales tienen deferentes niveles de autonomía y pueden llegar a no requerir asistencia del conductor. Existen complicaciones en la vida real como se menciona en los trabajos [11, 12]. Por ahora existen muy pocos sistemas lo suficientemente robustos; es decir, que falle por una nueva entrada o la ausencia de información para que puedan ser implementados en un automóvil. Esto ocurre porque los nuevos sistemas que usan AI solo se enfocan en un solo atributo del peatón, y si llegara a fallar no hay otra manera de actuar frente a un accidente con un peatón.

La contribución al término del trabajo será un sistema que puede unir dos características extraídas del peatón y que no se ha encontrado muchos sistemas que puedan unir dichas características. Este sistema podrá brindará a la sociedad un mecanismo para AV que les permitirá ser más fiables y dar un paso más cerca de un AV con mayor autonomía.

1.3. Planteamiento del problema

Uno de los principales retos al usar vídeos como fuente, es la extracción de información de la imagen o imágenes a usar. Mientras la calidad del vídeo sea mayor, mayor y mejor serán las características que puedan ser extraídas, pero con un alto costo computacional. Mientras que, si se utiliza una calidad baja de vídeo, la velocidad y el costo computacional será mucho mejor pero la tarea de extracción de características será más difícil y menos precisa; así que se debe de seleccionar la calidad de vídeo adecuado para una tarea en específico.

En la búsqueda bibliográfica los trabajos [13, 14, 15, 16, 17, 18] mencionan o usan CNN, debido a que encontraron u obtuvieron resultados de precisión por arriba del 90 por ciento.

En el trabajo [18] se reporta una precisión de 94.4 por ciento usando cuadros de vídeos de una resolución de 1280x1024 usando el esqueleto de un peatón; aunque no menciona los cuadros por segundo (Frames Per Seconds (FPS) por sus siglas en inglés) presume de tener una velocidad de 0.25 ms por inferencia; aunque se basa en vídeos RGB, lo cual hace más pesado el procesamiento, y el trabajo [14] reporta una precisión de 91 y 92 por ciento para la orientación de la cabeza y la orientación del cuerpo para un peatón respectivamente. En estos y otros trabajos solo se enfocan en una característica, y evaluarla individualmente un área de oportunidad es combinar dos o más características para que el sistema sea más robusto; es decir, que sea menos propenso a fallas.

Algo también que queda pendiente para el mejoramiento de estos sistemas es la existencia de algunas controversias, ya que algunos trabajos revisados como [13] utiliza la dirección de la cabeza para el reconocimiento de la intención del peatón a diferencia del trabajo [18] argumenta que no es necesariamente útil en tareas específicas o para el mejor desempeño del sistema. Otra área de oportunidad para el trabajo [18] es la utilización de cuadro de vídeos de más baja resolución con la misma precesión para mejorar la velocidad y la carga computacional.

En el trabajo [19] presume de tener una precisión de 100 por ciento para detectar si el peatón está mirando al conductor por medio de los ángulos de la mirada entre ellos, pero la

limitante de este trabajo es que fue realizado sin técnicas de AI y en un ambiente controlado donde el auto con equipamiento esta inmóvil.

Otra área de oportunidad es que varios trabajos ya mencionados coinciden en el mejoramiento de la oclusión (la obstrucción del peatón por algún otro objeto) del peatón.

1.4. Hipótesis

Mediante la extracción y unificación del esqueleto virtual del peatón y la dirección de la cabeza desde vídeos por medio de métodos de AI se reconocerá la intención del peatón con mayor eficiencia y en menor tiempo.

1.5. Objetivos

1.5.1 Objetivo General

Desarrollar un sistema que permita reconocer si un peatón va a cruzar la calle por medio de técnicas y métodos de AI para reconocer la orientación de la cara y el movimiento del cuerpo usando su esqueleto con el fin de prevenir o disminuir accidentes viales que involucren peatones.

1.5.2 Objetivos Específicos

- Seleccionar y utilizar métodos para la extracción de información de vídeos.
- Desarrollar un sistema y aplicar métodos de AI que permitan reconocer la postura del cuerpo de peatón y la dirección de su cabeza.
- Proponer una estrategia para unir ambas características; dirección de la cabeza y postura del cuerpo.
- Realizar pruebas utilizando la estrategia propuesta, y aplicarle métricas.

1.6. Estructura de la tesis

En el presente proyecto de tesis se estudian diversos temas relacionados con inteligencia artificial, aprendizaje profundo y automático. La estructura de la tesis se plantea de la siguiente forma:

- Capitulo 2 Se abordan los antecedentes de las redes neuronales y las técnicas de aprendizaje automático utilizadas en el área de conducción autónoma llegando así al estado del arte.

- Capitulo 3 Nos concentraremos en los fundamentos que nos permiten la realización del proyecto de tesis, donde exponemos temas primordiales como: redes neuronales recurrentes, redes neuronales convolutivas, algoritmos de aprendizaje maquina como lo son bosque aleatorio (Ramdom Forest (RF) por sus siglas en ingles), máquina de soporte de vectores (Support vector machine (SVM) por sus siglas en ingles) y vecinos más cercanos (KNN por sus siglas en ingles).

- Capitulo 4 Se abordan los métodos y materiales utilizados para el desarrollo de esta, la metodología y algoritmos primordiales utilizados para la realización de pruebas.

- Capitulo 5 se mostrará el comportamiento obtenido de haber utilizado la característica de los ángulos interiores de la rodilla.

- En el capítulo 6 finalmente se verán las conclusiones del trabajo de tesis y las potenciales aplicaciones futuras.

2. ANTECEDENTES

Dado a la falta de pericia del conductor y los avances tecnológicos, se pretende tener AV que puedan evitar todo tipo de accidente, que para este trabajo se relaciona con los accidentes con peatones. Hoy en día los autos inteligentes llegan a un nivel de autonomía 3, estando a la mitad de camino para la autonomía total. Las investigaciones realizadas para llegar a esto se centran únicamente en una solo característica del peatón para reconocer si tiene la intención de cruzar una vía urbana y esto hace que el sistema no sea lo suficientemente robusto para enfrentar situaciones de la vida real y además de que no existen muchos modelos que pueda unir diferentes características del peatón y hacer un reconocimiento de la intención del peatón de cruzar la vía urbana de manera más fiable.

Desde los años 80's ya se hablaba de vehículos autónomos, principalmente estas aproximaciones eran hechas para proyectos de defensa nacional, como por ejemplo "Martin Materietta Denver Aerospace's Autonomus Land Vehicle project"[20]. El cual consistía en la información tomada desde un sistema de visión para localizar y evadir obstáculos con el propósito de que el vehículo tome decisiones tácticas. Este trabajo consistió en demostrar los avances de la inteligencia artificial, manejo de imágenes y arquitecturas avanzadas dentro del contexto militar. Aunque se consiguió terminar el proyecto aún quedaba por resolver la detección y modelamiento en intersecciones.

En el trabajo [21] habla sobre el incremento de vías vehiculares y el aumento del uso de vehículos, los problemas de tráfico, accidentes viales en otros, se empezó a tomar de base las investigaciones ya mencionadas para poder así detectar obstáculos y también el mejoramiento para ambientes más complejos. En el trabajo [22] habla sobre los primeros intentos donde solo se detectaba desde un auto, a otros vehículos estando en movimiento o estáticos. También se aplicaba a la predicción de trayectorias de los vehículos en los alrededores.

Los AV están en auge hoy en día en las ciudades urbanas donde existe una mayor concentración de carros y de personas que llamaremos peatones. Existe mucha interacción entre los peatones y los vehículos en lugares como un cruce peatonal, un cruce en zonas no permitidas para peatones, entre otras. Los accidentes sean fatales o leves son provocados por la mal toma de decisiones o la distracción de los peatones o de los conductores; estos factores suelen ser causados por factores ajenos a la interacción misma de peatones y conductores como los son la fatiga por diversos factores como la privación del sueño provocado por el estilo de vida en las zonas urbanas o por el uso de dispositivos como los celulares. Es por eso por lo que hoy en día existe mucho interés en la realización de AV capaces de tomar sus propias decisiones y acciones en base a los acontecimientos que pasan en sus alrededores y que son captados por sensores, cámaras, etc.

2.1. Niveles de Autonomía

Para poder decir que un auto es autónomo se debe de tener niveles que definan las características de un AV y poder partir de ahí, en la tabla 1 se muestra los niveles de autonomía de un carro que existen hoy en día propuesto por Sociedad de Ingenieros Automotrices (Society of Automotive Engineers (SAE) sus siglas en ingles). Los niveles van desde el cero al cinco, donde el nivel cero indica que no existe ningún sistema autónomo, pero cuenta con sensores visuales que ayudan al conductor, un ejemplo es la asistencia visual del punto muerto de los retrovisores, en el nivel uno se tiene un sistema crucero donde el conductor debe de tener las manos sobre el volante para controlar la trayectoria, para el nivel 2 el sistema toma el control total del vehículo mientras que el conductor está preparado para intervenir en caso de que se requiera, para el nivel 3 el conductor puede no prestar atención y el auto debe ser capaz de responder a situaciones de emergencia, para el nivel 4 no se requiere que el conductor este concentrado salvo en áreas especiales, Finalmente para el nivel 5 no se requiere intervención alguna del conductor.

El trabajo [1] propone que para poder alcanzar el nivel 1 el AV necesita proveer herramientas de asistencia, para hacer esto se necesita *detectar*. Para alcanzar el nivel 2 es necesario que el AV *reconozca* para así poder hacer predicciones a corto plazo. Para alcanzar

Tabla 2.1: Niveles de autonomía para autos [1].

Nivel(SEA)	Descripción	Requerimientos
0	Sin automatización. El sistema automatizado emite advertencias y puede intervenir momentáneamente, pero no tiene un control sostenido del vehículo	Sensar
1	Las manos en el volante. El conductor y el sistema automatizado comparten el control del vehículo. El conductor debe estar listo para retomar el control total cuando sea necesario	Detección
2	Manos fuera. El sistema automatizado toma el control total del vehículo (dirección y velocidad). El conductor debe estar preparado para intervenir de inmediato. El contacto ocasional entre la mano y el volante suele ser obligatorio.	Reconocimiento, Seguimiento
3	Ojos fuera El conductor puede desviar la atención de forma segura de las tareas de conducción, usar un teléfono o ver una película. El vehículo manejará situaciones que requieran una respuesta inmediata, como el frenado de emergencia. El conductor aún debe estar preparado para intervenir dentro de un tiempo limitado	Predicción comportamiento
4	Mente fuera. No se requiere la atención del conductor, excepto en áreas espaciales limitadas o circunstancias especiales en las que el vehículo debe poder cancelar o transferir el control al ser humano de manera segura.	Interacción peatonal, señalización
5	Automatización total. Ninguna intervención humana requerida	Robustez y fiabilidad extremas

el nivel 3 se requiere una *predicción* más robusta para así quitar tareas al conductor. Y para alcanzar el nivel 4 y 5 el AV debe ser capaz de entender las situaciones como lo haría un conductor, basados en el comportamiento del peatón y las interacciones existentes. Se muestra en la figura 2.1 los requerimientos para alcanzar dichos niveles.

2.2. Inteligencia artificial en el área de conducción autónoma

Con el avance de los años las nuevas tecnologías han crecido con mayor potencia y redujeron su tamaño. Esto abrió la puerta para las técnicas ya preexistentes de la AI para poder ser utilizadas; ya que algunos métodos requieren procesos más tardados o especializados. Derivado de esto se empezó a utilizar diversas técnicas para obtener mejores resultados en el área de la industria automotriz. Hablando de AV se están aplicando métodos de AI para

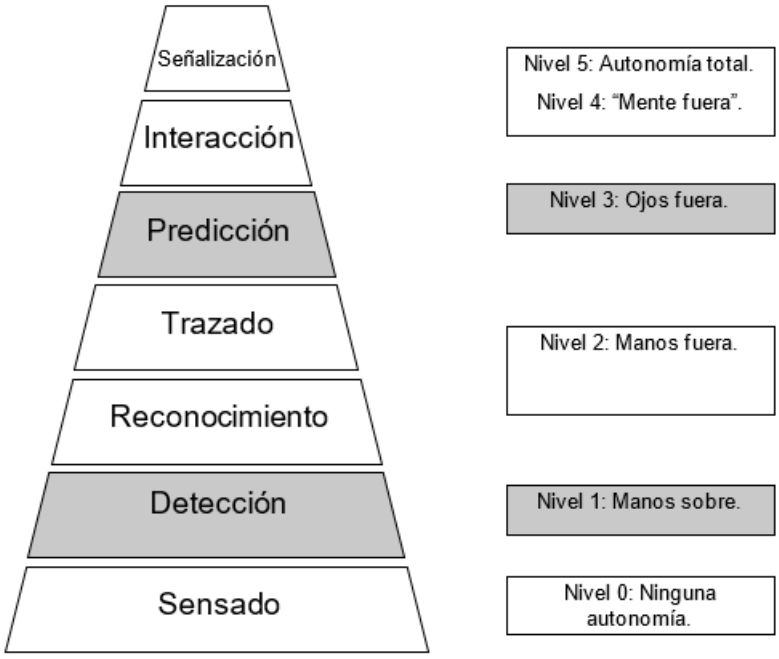


Figura 2.1: Niveles de autonomía para autos [1].

mejorar las áreas de *sensado, detección, reconocimiento, trazado, predicción e interacción.*

Este trabajo se concentrará en el área de reconocimiento. En el trabajo [1] menciona que el *reconocimiento* toma ciertos atributos específicos de un objeto (peatón) que ha sido detectado y sensado ya sea por sensores o entradas visuales. Existen muchos modelos de reconocimiento que se pueden apreciar en la figura 2.2.

En la actualidad existen una gran variedad de trabajos donde cada uno de ellos utiliza métodos de reconocimiento del área de la AI para poder reconocer o clasificar las características deseadas. En las recopilaciones de [1, 23] mencionan una gran variedad de técnicas y algunas que son comúnmente usadas son CNN, *Programación Lineal Entera* (Integer Linear Programming (ILP) por sus siglas en inglés), *Redes Neuronales Recurrentes* (Recurrent Neural Network (RNN) por sus siglas en inglés), *Redes Neuronales Profundas* (Deep Neural Networks (DNN) por sus siglas en inglés), *KNN*, *SVM* estos y más métodos son usados y también usan variantes o complementos con el propósito de clasificar y encontrar atributos específicos para el reconocimiento de las intenciones de los peatones.

Ahora bien, para poder aplicar y que funcione cualquiera de estos métodos de AI y

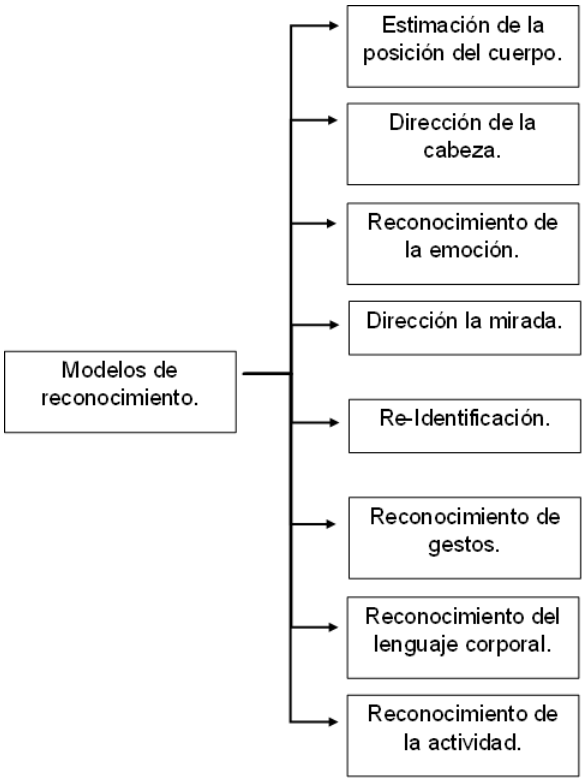


Figura 2.2: Modelos de Reconocimiento[1].

poder reconocer características se debe de tener ya sea una imagen o un vídeo de donde poder extraer la información y hacer todo el proceso necesario. Mientras mejor sea la calidad de la imagen mejor será la precisión del reconocimiento y para esto existen también técnicas de AI para mejorar o modificar ya sean las imágenes o vídeos, con el fin de facilitar el trabajo de estudio. En el trabajo [12] menciona que existen varios retos al momento de clasificar debido a muchas circunstancias en el mundo real como los son:

- *Imágenes de baja resolución* debido a la distancia o movimientos súbitos del peatón.
- *Diferentes escenas* esto implica diferentes ángulos, iluminación entre otros.
- *Oclusión* lo cual se refiere a que los objetos en cuestión sean tapados por otros creando ambigüedad.

De igual manera en el trabajo [11] comenta que es aún más complicado a la hora de reconocer los comportamientos de peatones individualmente dentro de una multitud, ya

que, a pesar de los problemas ya mencionados, cada peatón de este *rebaño* tiene su propia dirección, dependiendo de su objetivo. Cabe recordar que un vídeo es una serie de imágenes consecutivas. El trabajo [24] propone los pasos para el procesamiento de imágenes, los cuales son: *Adquisición de la imagen, mejora de imagen, procesamiento morfológico, restauración de la imagen, procesamiento de imágenes a color, compresión de la imagen y segmentación.*

2.3. Comportamiento al caminar

Para Predecir la intención de los peatones, es fundamental conocer y establecer las principales características que un peatón muestra de forma no verbal. Los trabajos [2, 25] investigan el andar de los peatones y miden los ángulos de algunas articulaciones de la parte inferior del cuerpo, como las rodillas y se puede apreciar en la figura 2.3. Otros artículos como [13, 14, 15, 26, 27, 28] mencionan o usan el movimiento/orientación de la mirada, la cabeza, los ojos, los brazos y las piernas. La referencia [29] argumenta que, si un peatón mira al conductor en un automóvil, que se aproxima, lo más probable es que la intención del peatón sea no cruzar. Por el contrario, si el peatón no está mirando al vehículo la probabilidad de cruzar es mayor y más aún, la acción se está desarrollando sobre el paso de peatones. Y el artículo [30] nos dice que es más probable que ocurra la acción de cruzar si un peatón está caminando hacia el cruce de peatones que otra que está parada en la acera.

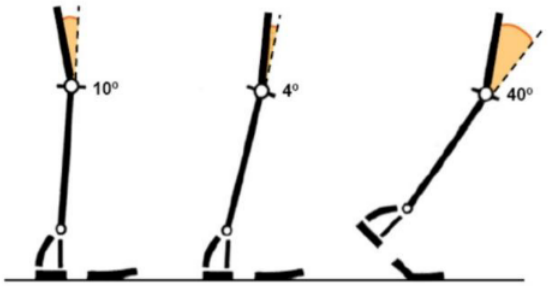


Figura 2.3: Ángulo de la rodilla al caminar de un peatón [2].

Además, casi todas las búsquedas de artículos se centran en los peatones que cruzan el cruce de peatones y no consideran todas las variables que pueden influir en el comportamiento de cada peatón [31], como el entorno y las características individuales y situaciones

personales, pero Rasouli et al. [31] tiene en cuenta algunas características para la intención de cruzar:

- Tipo: toma la edad de los peatones y los clasifica en tres clases, adulto, niño y anciano. Estas clases obtienen diferentes formas de cruzar.
- Rasgo: Se trata del carácter de cada peatón, agresivo, conservador y promedio. El rasgo afecta el comportamiento de cada uno, mientras que un Vulnerable Road User (VRU) o peatón agresivo camina más rápido y los VRU promedio conservadores caminan más lento.
- Ley de obediencia: Un VRU promedio es que uno puede obedecer o romper dependiendo del entorno. Una persona media es aquella que se encuentra por debajo de un umbral predefinido, es decir, que se encuentra más cerca de una zona designada como paso de peatones o similar para peatones.
- Aceptación del espacio: Es el espacio que acepta un peatón para cruzar la calle imprudentemente. Y está relacionado con los VRU agresivos.
- Patrón de cruce: se trata de dos categorías, una etapa en la que VRU espera para cruzar cuando no se acerca ningún vehículo o no hay ningún vehículo en la carretera y el espacio rodante donde VRU intenta el primer carril libre disponible y así sucesivamente hasta llegar en la acera
- Ruido perceptual: La capacidad de cada VRU para escanear su entorno afecta la aceptación del espacio para cruzar la calle. Y este ruido perceptible depende de la edad del peatón y de la incapacidad fiscal (si la hay).

Otra consideración importante es que con la llegada de la era de los teléfonos inteligentes, nuestros estilos de vida se han modificado. Hablando sobre el tema de esta encuesta, la influencia de los teléfonos inteligentes ha cambiado el comportamiento de los peatones, y esto hace que los artículos e investigaciones que han estado trabajando en características como la orientación de la cabeza necesiten cambiar nuevos enfoques.

En [32] comenta que los peatones, que usan teléfonos inteligentes en las intersecciones, están menos preocupados por su entorno, y aquellos que caminan y usan teléfonos inteligentes se denominan Çaminantes distraídosz "Zombis con teléfonos inteligentes", respectivamente. Y también, algunos los estudios han encontrado que el 50 por ciento están grabando mientras caminan y el 69.5 por ciento de los accidentes son peatones hablando por teléfono. Referencia [33] investigó y encontró que el campo de visión de un peatón usando un teléfono inteligente es el 5 por ciento del total de los VRU normales y las hembras adoptan un comportamiento de cruce más peligroso, especialmente cuando están distraídas.

En [34] se llevó a cabo una encuesta de texto en la que los participantes se dividieron en dos grupos, en uno de ellos se introdujo la información sobre vehículos impulsados por humanos (Human Driven Vehicles (HDV)) y en el otro sobre vehículos totalmente automatizados (Fully Automated Vehicles (FAV)). Encontraron que los peatones que son conscientes de que un automóvil es un FAV, se sienten más seguros para cruzar en acciones de riesgo donde el FAV no se detuvo. Así, se modifica el comportamiento de un peatón original.

2.4. Estado del arte

2.4.1 Detección de comportamiento de peatones

Para la detección del comportamiento de los peatones, existen muchos enfoques diferentes, y la forma en que se consideran algunas partes o movimientos para los peatones podría arrojar mejores resultados sobre la intención del peatón. Cada trabajo tomó diferentes partes o movimientos de VRU que consideraron relevantes para las intenciones de los peatones. A continuación, se presentará.

La referencia [35] usó puntos clave de imágenes recortadas y el enfoque de arriba hacia abajo que primero detecta un peatón mediante un cuadro delimitador y luego asigna los puntos clave.

El esqueleto de peatón es utilizado actualmente por muchos artículos como [13, 27, 28] porque se necesita menos dimensionalidad. La figura 2.4 se puede ver el esqueleto de una persona.

Algunos trabajos han utilizado ángulos tomados de esqueletos para predecir la de-

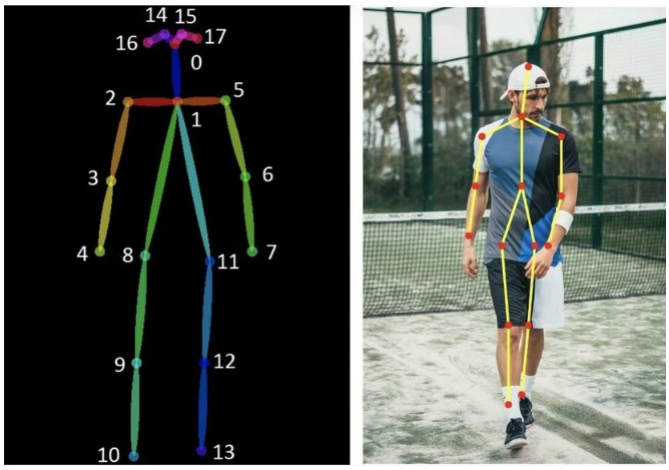


Figura 2.4: Representación de un esqueleto de una persona por OpenPose [3].

tección de la intención de comportamiento de cruce.

El uso del género está relacionado con el comportamiento y para el reconocimiento de género, trabajos como [36] y [37] utilizaron imágenes recortadas de la cabeza para la detección. Los codos o las muñecas son útiles para comportamientos de atención específicos de los peatones porque se utilizan para la comunicación con los conductores como se menciona en [18]. Además, la referencia [36] usó gestos con las manos, y en la referencia [18] se prefieren el torso y los hombros para modelar la orientación del peatón. Por otro lado, [18, 26] utilizó puntos clave de rodilla y tobillo para detectar si el VRU está caminando o de pie y si el proceso es dinámico. Además, [13, 38] utilizó la dirección, la orientación y el estado de movimiento de mirar/mirar.

2.4.2 Predicción de la intención del peatón

Para reconocer la intención de un peatón se necesita alguna información extraída de algún vídeo, aquí es donde la visión por computadora es importante debido al procesamiento de vídeo/imagen, que proporciona un mejor rendimiento para el reconocimiento de la intención. Por otro lado, el uso de vídeo conlleva algunas dificultades como el costo computacional debido a su alto dimensional. En [24] se mencionan los pasos del procesamiento de imágenes. Para obtener la Intención de Cruce / No Cruce (Crossing (C)/No Crossing (NC)), es necesario:

1. Extraer características.

2. Clasificar según características.

La referencia [39] trabajó en la predicción de la trayectoria del peatón basada en la biomecánica de la marcha. Usaron solo una variable biomecánica, 10 cámaras Flex3 para monitorear los experimentos y las leyes de Newton. Sus ventanas de predicción eran de alrededor de 1 segundo. Pidieron a cinco sujetos adultos de entre 21 y 52 años que realizaran los experimentos.

[40] propuso un novedoso generador de trayectorias multimodales basado en el movimiento natural de los peatones. Su modelo utiliza una red de puntuación y una red de regresión para extraer características de los distintos patrones de movimiento y refinar los distintos patrones de movimiento punto por punto con dos capas de perceptrón multicapa (Multilayer Perceptron (MLP) por sus siglas en ingles), respectivamente. Utilizan cuatro métricas (Error de desplazamiento (ADE), Error de desplazamiento final (FDE), Distancia promedio por pares (APD) y Distancia final por pares (FPD)) para evaluar la precisión de las predicciones de trayectoria.

2.4.3 Extracción de características

Para Extraer características, los artículos [13, 14, 15, 18, 28, 41] utilizan algunas características como la orientación de la cabeza y la esqueletización de la pose del cuerpo. Mientras que en [42, 43, 44] argumentan que la orientación de la cabeza y el movimiento de las piernas son las principales características que indican la intención de los peatones. Además, el artículo [42] afirma que el contexto de los peatones es útil y puede mejorar el nivel de reconocimiento de la intención del peatón. El uso de esqueletización brinda la ventaja de hablar sobre el tiempo de procesamiento, en [45] argumenta que el uso de un área de interés de recorte y esqueletización afecta positivamente el rendimiento "17 articulaciones de esqueleto en comparación con 2048 ResNet Feature Vector".

Con el uso del esqueleto de las piernas, el artículo [46] calcula algunos ángulos para diferentes articulaciones. En [2] establece ciertos ángulos para la marcha normal de un peatón. Estos ángulos pasan por 40° y 60° de las rodillas. Además, en [4] propuso un método

de predicción de intención de cruce de peatones que considera usar la fusión de múltiples características, estas características son características de un peatón, como los ángulos de las partes inferiores del peatón usando su pose, la distancia relativa entre El AI y el peatón, el entorno y la información del AI como la velocidad y son extraídos por una cámara monocular. Un ejemplo de los ángulos usados por [4] se puede ver en la figura 2.5.

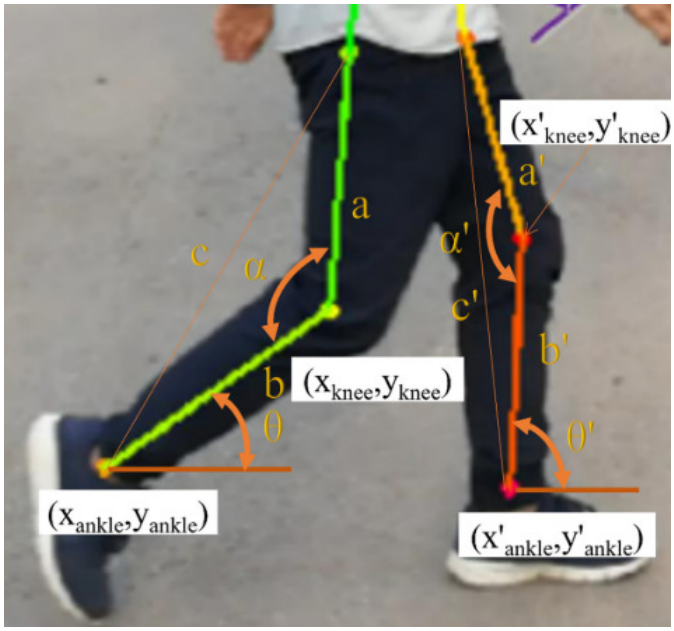


Figura 2.5: Ángulos de centrado utilizados por [4].

Ahora, hay muchas maneras de obtener estas características. Actualmente existe una gran variedad de trabajos donde cada uno de ellos utiliza métodos de reconocimiento del área de AI para poder reconocer o clasificar las características deseadas. Los artículos [1, 23] mencionan las dos técnicas más comunes de AI que son CNN y RNN. Ambas redes se pueden aplicar para extraer características, pero existen algunas diferencias que hacen que se utilicen para aplicaciones similares y no para la misma tarea. [47], utilizó una pose 2D y una combinación de tipos de CNN para extraer las características de los peatones en el conjunto de datos Joint Attention in Autonomous Driving (JAAD).

La referencia [48] usó CNN (VGG19) y RNN (Gated Recurrent Unit (GRU)) para extraer características para un peatón y un AV y usó solo CNN en un reloj para obtener más información sobre el peatón a fin de predecir la intención del peatón.

2.4.4 Clasificación de las intenciones

Ahora, para el paso 2. Basado en algunas características del peatón, agregado al contexto del peatón. Hay muchos métodos de ML que pueden usarse como clasificador y determinar la intención del peatón de cruzar o no cruzar.

En [49] se mencionó el uso de SVM como clasificador, la referencia [37] usó un módulo de atención facial como clasificador, SVM fue usado como clasificador del modelo de decisión de cruce por [50]. SVM, RF, Gradient Boost (Gradient Boost (GB)) y Extreme Gradient Boost (Extreme Gradient Boost (XGB)) fueron usados por [27], y más trabajos hablan sobre el uso de algunos clasificadores similares SVM, RF, GB y XGB, KNN. De la misma forma, In [4] y [47] solo usaron un RF como clasificador para practicar la intención de un peatón en base a ángulos, alrededores e información de AV.

Sabiendo que no es posible realizar pruebas con sujetos reales debido a la naturaleza del tema, todos los trabajos utilizan algún tipo de base de datos para poder generar resultados. Una de las bases de datos más utilizadas en la Atención Conjunta en Conducción Autónoma JAAD [51, 52], que contiene 346 videoclips HD de 5-10 segundos de duración grabados con encendido -cámara de a bordo a 30 FPS.

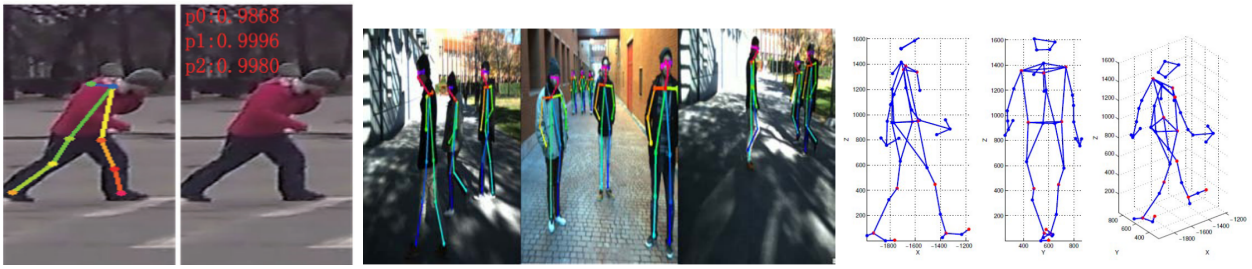
2.4.5 Modelos de predicciones

La tabla 2.2 muestra modelos Tradicionales, que no utilizan métodos de DL y modelos AI para predecir algunas características de los peatones por trabajos actuales. Algunos trabajos como [39] no utilizan las métricas mencionadas en la sección 6, sin embargo, el método de predicción propuesto para este trabajo logra un error promedio entre 100 y 200 mm para un tiempo de 0 - 0.5 s. De la misma manera, el trabajo [19] informó una muestra de mirada media de 0.996 para el conductor y 0.997 para la mirada del peatón, umbral de 4° y detección de contacto visual.

2.4.6 Resumen

Teniendo las herramientas, métodos y una buena fuente de donde sacar la información, es posible reconocer si el peatón tiene intención de cruzar o no la calle o una avenida.

Existen varias maneras para determinar esto, varios trabajos [13, 14, 15, 28] se centran en la dirección del cuerpo completo, la dirección de la cabeza, la dirección de los ojos, otros se basan en la partición del cuerpo sea por partes o la partición de puntos o un esqueleto que de información derivado de ciertos ángulos o separación entre puntos claves del esqueleto. En la figura 2.6 se muestra los modelos de esqueletos que se obtuvieron en varios trabajos diferentes. En el trabajo [13] estudia la dirección de los ojos y el esqueleto del peatón para determinar el comportamiento. Se concluye que es más difícil trabajar y obtener una precisión por medio del esqueleto a largas distancias. Para el trabajo [14] se enfocó sobre la dirección de la cabeza y la orientación del cuerpo completo usando CNN, obteniendo una precisión de 0.91 y 0.92 para la posición de la cabeza y la orientación del cuerpo respectivamente, comenta que es posible mejorar la precisión aumentando la cantidad de clasificaciones para la orientación. En el trabajo [15] usa la posición relativa para detectar al peatón y es posible mejorar su precisión usando la posición absoluta. En el trabajo [28] trabaja con el esqueleto, pero no utiliza métodos de AI y concluyó que es posible obtener mejores resultados usando CNN. En la figura 2.7 se muestra la clasificación de los peatones que están mirando hacia la cámara o carro y los que no lo hacen.



(a) Intención de cruzar[13]. (b) Detección de postura[15]. (c) Modelo del esqueleto[28].

Figura 2.6: Representación del modelo esquelético.

Es posible reconocer las intenciones del peatón mediante cinco principales características que son propuestas en el trabajo [62] las cuales son: *Movimientos de la cabeza, movimiento de las piernas, dinámica del caminar, características del peatón y trafico.* Un dato interesante es que los conductores tienden más a observar y durante más tiempo en la parte superior del peatón para intentar predecir sus posibles movimientos.

(a) Peatón mirando[14].

(b) Peatón no mirando[14].

Figura 2.7: Orientación de la cabeza.

En el trabajo [16] menciona que partiendo del esqueleto de un peatón es posible conocer sus intenciones mediante la pendiente, distancias o ángulos entre los puntos claves o articulaciones, como los que podría ser la columna, rodillas, codos etc.

De manera análoga es posible mediante ángulos reconocer si el peatón está prestando atención al cruzar debido que un peatón la mayor parte del tiempo trata de establecer contacto visual con el conductor. En el trabajo [19] calculan los ángulos entre el peatón y el conductor y establecen a que ciertos ángulos hay contacto visual, este trabajo se centró con un solo peatón y un carro inmóvil. Nathanael [63] comenta que es posible solo utilizar la dirección de los ojos y por ende la cabeza junto con la posición o postura del peatón para reconocer las intenciones de los peatones.

Se encontró con el trabajo [18] que presume de tener la mayor precisión en el estado del arte con un 94.4 por ciento para el reconocimiento de la intención del peatón, basándose solamente en el esqueleto del cuerpo, utilizando métodos como CNN. Y también menciona que es posible mejorar sus resultados mediante la unión de otras características.

Después de la revisión del estado del arte que para el trabajo [18] usa una secuencia de vídeos con una resolución de 1280x1024, lo cual representa un costo computacional más alto, y sumado a ellos utilizan información con un formato Red Blue Green (RGB) por lo cual el costo es aún mayor ya que involucra 3 matrices de 1280x1024, por lo cual es una limitante por el hecho de que aún no todos los automóviles cuenta con cámaras de alta resolución y los

que los tienen son de un costo poco accesible a la población en general. Otra limitante de este trabajo es que sólo se enfoca en una sola característica (esqueleto del peatón), lo cual para su servidor valdría la pena el usar otro atributo para el reconocimiento del comportamiento del peatón.

También para el trabajo [14] sus limitaciones el uso vídeos de baja resolución y que utilizan la orientación del cuerpo, esto no da una referencia clara si el peatón cruzará, a diferencia del uso del esqueleto del peatón ya que este da información de ciertas posiciones en una secuencia sobre cinética del peatón. En la tabla 2.3 se muestra un par de trabajos con sus aportaciones, precisión y limitaciones.

Tabla 2.2: Resultados de algunos trabajos para sus propios enfoques. No disponible (N/A).

	Autor	Año	Modelo	Pre.	Re.	F1	Acc.	AUC	Alcance
Métodos tradicionales	[39]	2022	Predicción a corto plazo basada en el peatón en las leyes de Newton.	N/A	N/A	N/A	N/A	N/A	Predicción de la trayectoria humana.
	[50]	2021	Peatón Híbrido Multimodal (MHP), SVM.	0.79	0.73	0.75	0.88	N/A	Paso de peatones/No paso de peatones.
	[19]	2021	distancia pitagórica, ángulos, contacto visual de peatones.	N/A	N/A	N/A	N/A	N/A	Eye contact detection.
	[12]	2020	Hierarchical Feature Embedding (HFE) framework.	0.87	0.85	0.86	0.92	N/A	Reconocimiento de atributos.
	[28]	2019	Gaussian Process Dynamical Models (B-GPDMs).	0.95	0.99	0.97	0.95	N/A	Predicción de trayectorias peatonales, poses e intenciones.
Métodos IA	[53]	2022	Kernelized convolutional transformer network (KCTN) with multihead attention (MHA) mechanism.	N/A	N/A	N/A	0.99	N/A	Predicción de la intención del conductor.
	[54]	2022	YOLOv3, LSTM.	N/A	N/A	N/A	0.97	N/A	Reconocimiento de comandos de tráfico.
	[55]	2021	Long short-term memory network with attention mechanism (AT-LSTM).	0.95	0.98	N/A	0.96	N/A	Intención de paso de peatones.
	[36]	2021	LSTM network, dense network.	N/A	N/A	0.90	0.91	N/A	Pedestrian's Intention Recognition.
	[56]	2021	GRU, multilayer perceptron (MLP), VGG16.	0.96	N/A	0.90	0.87	0.92	Predicción de acciones e intenciones para pasos de peatones.
	[44]	2021	VGG19, GRU.	0.51	0.81	0.63	0.83	N/A	Predicción de la intención de cruce de peatones.
	[45]	2021	ResNet-50 and two parallel single-layered convolutional task head.	0.71	0.79	N/A	N/A	N/A	Pedestrian intention prediction.
	[41]	2021	U-GRU, TrouSPI-Net.	0.73	0.89	0.80	0.88	0.88	Predicción de las acciones de los peatones.
	[57]	2021	YinYang-Net, ResNet50.	N/A	N/A	N/A	0.93	N/A	Reconocimiento de Género.
	[27]	2021	SVM, RF, GBM, and XGBoost.	0.80	0.75	0.77	0.92	0.84	Predicción de intención de cruce de peatones.
	[35]	2021	High Resolution Net (HR-NET), VGG16.	N/A	N/A	0.84	0.79	N/A	Predicción de intención de cruce de peatones.
	[58]	2020	Feature pyramid attention model (FPAM) based on ResNet-50, Multi label focal loss (MLFL).	0.87	0.86	0.86	0.79	N/A	Reconocimiento de atributos de peatones.
	[38]	2020	Pretrained CNN models from ResNet, ResNeXt, ResNet34. RNNs modelos Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).	0.83	1.0	N/A	0.70	N/A	Predicción de cruce de peatones.
	[18]	2020	Cascaded Pyramid Network (CPN), SPI-Net (Skeleton-based Pedestrian Intention network).	N/A	N/A	N/A	0.94	N/A	Predicción de las intenciones de los peatones.
	[59]	2020	CNN–LSTM architecture, VGG16, LSTM.	N/A	N/A	N/A	0.93	N/A	Reconocimiento de la Acción Humana.
	[60]	2019	Mask R-CNN, ResNet-50.	N/A	N/A	N/A	0.89	N/A	Reidentificación de persona.
	[13]	2019	Resnet (Resnet18, Resnet34, Resnet50, Resnet18, Resnet101).	N/A	N/A	N/A	0.92	N/A	Reconocimiento de intención de paso de peatones.
	[61]	2019	Dense41-YOLOV3-TINY.	N/A	N/A	N/A	0.80	N/A	Detección de peatones y reconocimiento de comportamiento.

Tabla 2.3: Comparaciones de trabajos.

Trabajos	Aportación	Precisión %	Limitaciones
Predicting Intentions of Pedestrians from 2D Skeletal Pose Sequences with a Representation-Focused Multi-Branch Deep Learning Network [18].	Reconocimiento de la intención de cruzar o no, mediante la extracción del esqueleto y el uso de capas profundas 2D-CNN.	94.4	Uso de videos de alta resolución y uso de una sola característica del peatón.
Appearance based pedestrians' head pose and body orientation estimation using deep learning [14].	CNN para el reconocimiento de la dirección de la cabeza y la orientación del cuerpo.	91 (cabeza) y 92 (cuerpo).	Vídeos de baja calidad y la orientación no de información convincente de la intención del peatón.
Vision-based recognition of pedestrian crossing intention in an urban environment[13].	Clasificador "mirando o no (looking/no looking)"por medio de 2D-CNN. Clasificador, çruzar o no (crossing/ no crossing) "por medio de 3D-CNN.	93.0 - (looking/no looking). 92.6 - (C/ NC).	Uso de vídeos de alta resolución.
Pedestrian Path, Pose and Intention Prediction through Gaussian Process Dynamical Models and Pedestrian Activity Recognition [28].	Reconocimiento de la actividad del peatón a partir del movimiento, la posición y articulaciones claves del peatón.	95.13	Desplazamientos articulares, los cuales en diferentes situaciones puede ser bueno o no en la clasificación. Se puede mejorar la precisión usando CNN. Pruebas en carros estáticos.
Design of A Prediction based Pedestrian Tracking System by UAV [16].	Sistema UAV, el cual monitorea la seguridad pública. Consiste en el trazado y el reconocimiento de la pose de la persona.	89.0	Distancia de alcance de 3.4m.
Towards the detection of driver–pedestrian eye contact [19].	Modelo para la detección del cruce de miradas entre el peatón y conductor mediante el cálculo ángulos.	100	Ambiente controlado (laboratorio), carro estático. Técnicas IA no usadas.

3. FUNDAMENTACIÓN TEÓRICA

En la presente sección se discutirán los fundamentos teóricos sobre los cuales se basa la metodología propuesta.

Para realizar el trabajo se necesita una base de datos que nos ayude simular y probar el sistema en un ambiente urbano, sin poner en riegos la integridad de un ser humano o animal. Esta base de datos consta de vídeos de corta duración, y en cada vídeo seleccionado se le hará un procesamiento de imagen a una serie de imágenes del mismo vídeo en una cierta secuencia. Estas imágenes consecutivas serán con las cuales se entrene y pruebe la CNN para conseguir la dirección de la cabeza y el esqueleto virtual del peatón. Después se buscará la forma de unir estas dos características para hacer una clasificación (SVM, KNN y RF) de la intención del peatón.

3.1. Base de datos

Una base de datos es un conjunto o colecciones de cosas, y para el uso de este proyecto serán base de datos de vídeos sobre peatones en zonas urbanas. En la actualidad existen muchas bases de datos, como por ejemplo [64] *Datasets for Computer Vision and Image Processing on CVonline* donde contiene una colección de diferentes bases de datos públicas y también existe GOOGLE *Dataset Search*. La base de datos elegida para esta investigación es conocida como JAAD dataset [51] la cual es una base de datos publica la cual consta de 346 vídeos cortos de 5 a 10 segundos a 30 FPS, 2793 cajas delimitadores (Bounding box). Esta base de datos también contiene las anotaciones para los peatones, el vehículo, el clima y entre otros más. La razón por la selección de esta base de datos fue por la gran variedad de ambientes, la duración de los vídeos y después de una investigación de las bases de datos utilizadas para este tipo de tareas que utiliza solo imágenes RGB fue la más utiliza, en la tabla

3.1 se observa las bases de datos investigadas y los trabajos que las utilizaron.

3.2. Procesamiento de vídeos

El libro [81] explica que un vídeo consiste en un conjunto de imágenes que se le conocen como **frames** las cuales son mostradas a la persona a una determinada velocidad (**frame rate**) la cual se mide en FPS. Se ha determinado que entre 25 a 30 FPS se percibe un movimiento suave y continuo. Una desventaja de un vídeo es que se cuenta con mucha información, lo que incrementa el tamaño del archivo y por consecuente el procesamiento del vídeo. " Un vídeo de un minuto consiste en 30 FPS y cada uno de ellos es de 640 por 480 píxeles y usando 24-bits de color y esto tomo más de 1582MB". Los vídeos a color utilizan señales *RGB* que corresponde a los colores principales los cuales son rojo, verde y azul cada uno de estos colores a nivel de programación es manejado como una matriz lo cual lo hace pesado computacionalmente. " Cada una de estas matrices o longitudes de onda está compuesta de píxeles y cada píxel trabaja con 8 bits y esto nos da 256 valores diferentes por consecuente al tener tres canales se tiene 256^3 lo cual representa 16,777,216 colores por pixel"[82].

Según el libro [83] el análisis de videos incluye la *detección de límites de tiro*; es decir, segmentos del video, y la *extracción de frames clave* del contenido.

Para la *detección de límites de tiro*, explica que no es eficiente procesar todo el video a la vez, por lo cual es mejor que se descomponga el video en secciones más pequeñas y procesarlas a cada una por separado. Existen técnicas de *detección de límites de tiro automático* y que pueden ser clasificados en cinco categorías, *basado en píxeles, basado en estadísticas, basado en transformación, basado en características y basado en histograma.* La *extracción de fotogramas(frames) clave* se realiza después de la detección de límites de tiro, los cuales serán los fotogramas claves.

Conscientes de que en los libros [81] [83] expresan que los videos, al ser una secuencia de imágenes o frames, es posible aplicar técnicas de procesamiento de imágenes.

Tabla 3.1: Base de datos investigadas.

.

Dataset	Works	Details	Download
VOC2012	[65]	20 types of classes. The train and validate data have 11,530 images containing 27,450 ROI-labeled objects and 6,929 segmentations.	http://host.robots.ox.ac.uk/ pascal/VOC/voc2012/
CMU-Perceptual-Computing Lab	[66]	2D real-time multi-person joints detection. 3D real-time single-person joints detection. Adjust toolbox. Single-person tracking	https://github.com/CMU-Perceptual-Computing-Lab/openpose
MARS	[67]	Reidentification dataset, expansion of Market-1501 dataset.Contains 1,261 different VRUs, and used at least 2 cameras.	https://paperswith code.com/dataset/mars
JAAD	[51, 52]	346 HD videos 5-10 seconds. It is used a camera at 30 FPS on the board. Variety of labels.	https://data.nvision2 .eecs.yorku.ca/JAAD _dataset/
Market-1501	[68]	1501 identities taken by six different cameras, and 32,668 pedestrians image bounding-boxes. 750 for training and 751 for testing.	https://paperswith code.com/dataset/ market-1501
DukeMTMC-reID	[69]	HD videos were taken by 8 different cameras. 16,522 training images of 702 identities, 2,228 inquiry images of the other 702 identities, and 17,661 gallery images.	https://paperswith code.com/dataset/ dukemtmc-reid
VIPER	[70]	Contains 632 VRUs and two outdoor cameras. Each image has been resized to be 128×48 pixels. Pose the angle of each person.	https://paperswith code.com/dataset/viper
USC	[71]	Video Temporal Analysis,Face Recognition and Unsupervised 3D Geometry Learning	https://sites.usc.edu /iris-cvlab/
KITTI	[72]	A vehicle equipped with HD color and grayscale video cameras.	http://www.cvlibs.-net/datasets/kitti/
PIE	[73]	6 hrs of HD video is taken with a camera (30 FPS) on-board and split into 10-minute chunks. Many annotations.	https://data.nvision2 .eecs.yorku.ca/ PIE_dataset/
MPII	[74]	The data set contains 25,000 images which contain over 40,000 VRUs with key points labels.	http://human-pose.mpi-inf.mpg.de/
PETA	[75]	Recognizing pedestrian attributes, gender, and clothing style, at a far distance. 19,000 pedestrian images with 65 attributes.	https://paperswith code.com/dataset/peta
KTH	[76]	Made by six movements: walk, jog, run, box,handwave, and hand clap.	https://paperswith code.com/dataset/kth
OPV2V	[77]	70 scenes, 11,464 frames, and 232,913 annotated vehicle bounding boxes.	https://paperswith code.com/dataset/opv2v
CODD	[78]	LIDAR information from multiple vehicles,3D object detection, cooperative object tracking. Annotations.	https://paperswith code.com/dataset/codd
ETH	[79]	Contains 1,804 images in three video clips extracted from a stereo rig mounted on a car (13-14 FPS).	https://paperswith code.com/dataset/eth
UCY	[80]	Real pedestrian trajectories composed of Zara01, Zara02, and UCY.	https://paperswith code.com/dataset/ucy

3.3. Extracción de atributos del peatón mediante métodos de AI

La extracción de atributos o características del peatón es una tarea indispensable para el reconocimiento del comportamiento del peatón, ya que con ellas se puede reconocer si el peatón está tendiendo a cruzar o no, y para ello existen muchas maneras de extraer dichas características. Durante la búsqueda de los antecedentes se encontró que la gran mayoría de los trabajos utiliza las CNN, y en específico el trabajo [17] hace una tabla 3.2 comparativa, donde se muestra que las razones del porque son más usados las CNN's ya que son más fáciles de entrenar, la carga computacional es menor a comparación de otras redes. Ahora para a los trabajos [14, 18] reportan una precisión de 94.4 y 91-92 por ciento en el reconocimiento de sus atributos respectivamente. En el trabajo [18] se basa solamente en el esqueleto (94.4 por ciento) y para el trabajo [14] utiliza la dirección de la cabeza y del cuerpo del peatón (91 y 92 por ciento respectivamente). Basado en los trabajos revisados se considera para este trabajo, la extracción y reconocimiento del esqueleto del peatón y la dirección de la cabeza para reconocer la intención del peatón.

Tabla 3.2: Comparaciones de métodos [17]

.

	Métodos basado en CNN	Métodos basado en RNN	Métodos NO basados en Neural Networks (NN)
Ventajas	-Bueno para aprender características espaciales, geométricas y de forma. -Carga computacional reducida en comparación con las redes neuronales normales - Invariancia de traducción.	-Bueno para procesar series de datos. -Bueno para aprender características y dependencias temporales. -Asegurar la coherencia temporal.	-Diseñar un modelo de trabajo es más sencillo en comparación con las redes neuronales. -La mayoría no dependen de los datos de entrenamiento.
Desventajas	-No puede determinar las dependencias temporales en los datos de secuencia sin mecanismos adicionales.	-Generalmente más difícil de entrenar.	No modelan las dependencias de secuencia con tanta eficiencia.

3.3.1 Redes Neuronales Convolucionales (CNN)

La investigación [84] comenta que las redes neuronales convolucionales (CNNs) han sido actualmente usadas para la clasificación y trazado de problemas, ya que ha demostrado que un comportamiento superior en tareas de visión y puede generar características robustas y genéricas. El trabajo [17] menciona que algunas ventajas de CNNs sobre los métodos más convencionales de clasificación es debido a las capas convolucionales ya que estas pueden tener varios filtros, características y este método puede aprender mediante un entrenamiento previo. Y estas capas y características son modificables para así poder mejorar los resultados de dependiendo de los requerimientos del problema en cuestión.

El trabajo [14] argumenta que para poder utilizar CNNs es necesario tener las imágenes o la secuencia de ellas (vídeo) *normalizadas*, esto para que todas estén en un mismo formato. Después del conjunto de datos se separa en subconjuntos que llamaremos datos de entrenamiento y validación (prueba). Las imágenes son marcadas con sus características y así tenemos listo nuestros datos para el entrenamiento supervisado. Estos subconjuntos ahora están listos para pasar por el método seleccionado de CNN. Esto se hará para la base de datos para el reconocimiento de la orientación de la cabeza y del esqueleto de manera separada.

En la figura 3.1 se muestra el diagrama de flujo donde se conceptualiza el procedimiento para obtener la dirección de la cara o vista y el esqueleto del peatón mediante CNN con imágenes procesadas previamente con el fin de obtener una salida donde se tenga información de la orientación de la cabeza y el esqueleto del cuerpo con puntos claves, con el fin de pasar la salida ya mencionada por clasificadores de la AI. A continuación, se muestran los modelos que se utilizaron para la extracción de características del peatón y su entorno.

3.3.2 YOLOV8

Los modelos de You Only Look Once (YOLO)vn son del tipo conocidos como método de una etapa la cual priorizan la velocidad de inferencia. El funcionamiento de este modelo se cometa en el trabajo [85] el cual se basa en analizar una imagen completa para entender el contexto de la imagen, pero al mismo tiempo divide la imagen en una red o secciones cuadradas SxS y cada sección es clasificada y da un nivel de confiabilidad a cada

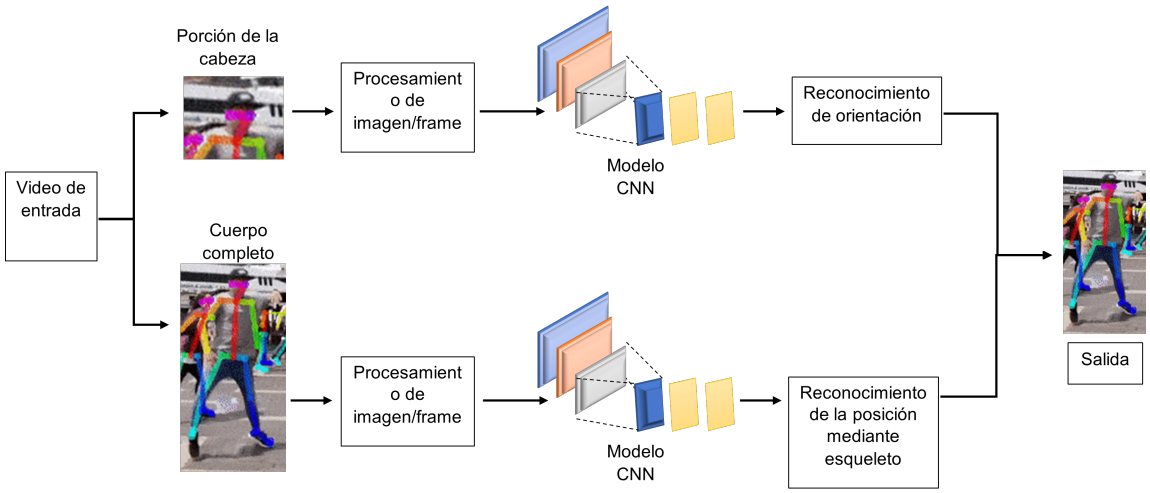


Figura 3.1: Modelo CNN (Imágenes obtenidas de [5]).

sección, la sección con mayor valor serán las detecciones finales que el modelo YOLOvn dará, es por esto que al realizar todo en una solo toma hace que sea más rápido.

El modelo YOLOv8 [6] es de las últimas versiones del año 2023, la cual tiene la capacidad de hacer detecciones, estimación de poses, y segmentaciones en tiempo real, ofreciendo altos rendimiento en velocidad y precisión. También resalta su fácil implementación con otras versiones anteriores. En la figura 3.2 se observa la comparaciones de YOLOv8 con sus predecesoras versiones

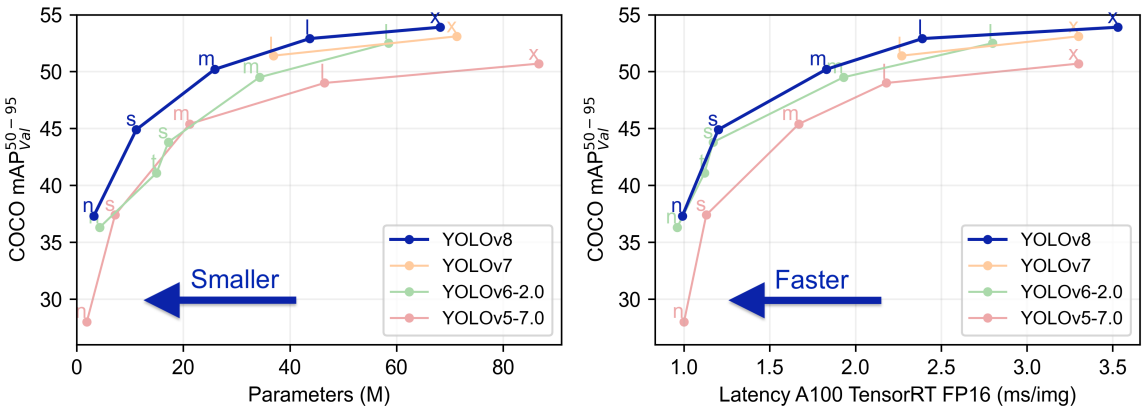


Figura 3.2: Comparaciones de versiones del modelo YOLO [6].

3.3.3 ResNet50

Los modelos de Residual Neural Network (ResNet)n se puede considerar también se un solo paso, pero la diferencia con YOLO es que contiene módulos residuales que alimentan al siguiente modulo y a la salida para así poder tener una red más profunda y no tener problemas del desvanecimiento de gradiente como se comenta en el trabajo [86], lo cual ocurre en redes profundas sin estas conexiones residuales. Se selecciono este modelo,ResNet50 por que tarda menos tiempo que una ResNet101 pero es más lenta que una YOLO pero en teoría más precisa. En el trabajo [87] hace una comparación del tiempo y la exactitud de los modelos de ResNet, donde el modelo Resnet50 se encuentra a la mitad de los demás modelos con una exactitud de 0.86 y una velocidad 5580 segundos de entrenamiento. Con esto buscamos el modelos con un rendimiento bueno sin sacrificar mucho el tiempo. Tambien fue seleccionado ya que se encontró este modelo pre-entrenado con la base de datos publica COCO [88] con el fin de ahorrar tiempo para la generación del nuestro sistema en cuestión.

3.4. Métodos de AI para la clasificación de la intención del peatón

Habiendo obtenido las características del esqueleto y la dirección de la cabeza, el siguiente paso es clasificar de acuerdo con las características ya mencionadas y para esto existen varios métodos de la AI del apartado de aprendizaje supervisado, el cual consiste en proporcionales datos de entrenamiento los cuales pasaron por la CNN. Después se le hacen llegar nuevos datos y el método deberá de poder clasificarlos. Aunque existen varios métodos de clasificación como *KNN, RF, Regresión lineal (Linear regression (LR) por sus siglas en inglés)*, pero el más mencionado en las referencias utilizadas en este trabajo es la *SVM*.

3.4.1 Máquina de Soporte de Vectores (SVM)

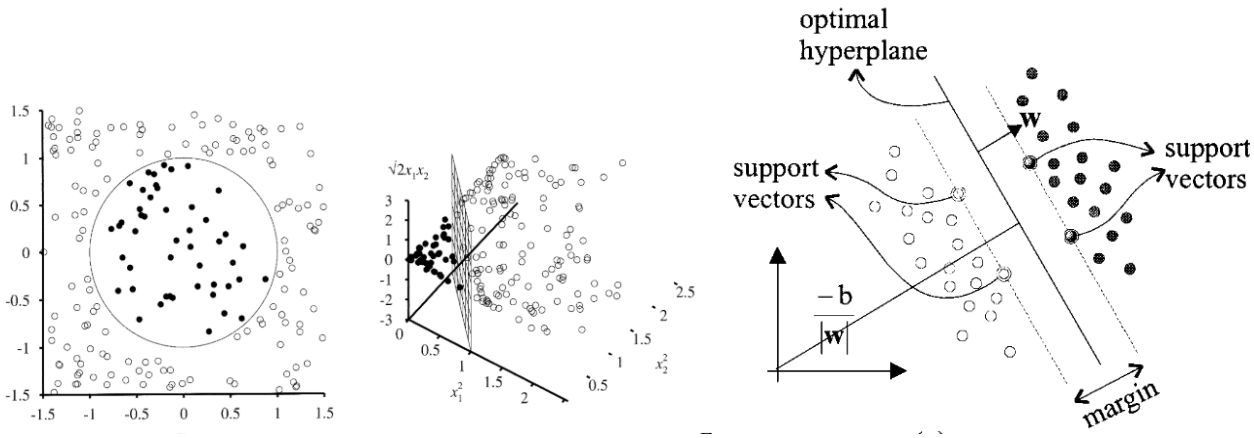
La investigación [89] de una noción sobre lo que es la Máquina de Soporte de Vectores y que fue concebida en 1992 e introducida por Boser, Guyon y Vapnik. Este método está relacionado al campo del aprendizaje supervisado y es usado para la clasificación y regresión. Este método utiliza la teoría ML para maximizar la precisión mientras evita el sobre ajuste de los datos. SVM es ampliamente utilizado en el mapeo de píxeles ya que da una precisión

comparable con los nuevos métodos como, por ejemplo, las NN. SVM se desempeña mejor al no sobre generalizar, no como los métodos de NN.

Los trabajos [89, 90] mencionan que la teoría del aprendizaje estadístico permite obtener conocimiento, hacer predicciones y tomar decisiones de un conjunto de datos. Esto permite escoger hiper-planos que son utilizados por SVM.

El trabajo [89] explica como SVM realiza una clasificación o una regresión. La cual es realización de un hiperplano, es decir dado un plano, sea 2D, 3D etc. El SVM agrega otro plano para así poder separar los datos que aparentemente no puede ser separados por una línea recta. Lo que hace posible la creación de hiperplano es por el uso de *Kernels* y existen diferentes tipos de kernels que dan diferentes propiedades de acuerdo al tipo de kernel a usar. Existen muchos hiper-planos para separar la información, pero hay solo uno que separa de manera más uniforme y de mejor manera los datos, para conocer cuál es el mejor hiperplano se usa el *margen máximo*. El margen máximo nos asegura que incluso, si es cometido un pequeño error en los límites, no se tendrá una mala clasificación, y por otra parte evita los mínimos locales y una mejor clasificación. Ambos trabajos [89, 90] dicen que el margen máximo es el que se encuentra más alejado de las clases a separar y está a la mitad exacta de los datos de cada clase más cercanos. En la figura 3.3a se muestra un conjunto de datos aparentemente no separables, pero al usar SVM se convierte de una dimensión 2D a 3D y ahora es posible clasificar por medio de un hiperplano. En la figura 3.3b es un hiperplano para clasificar un conjunto de datos en un plano 2D que es fácilmente separable en este caso.

Ventajas y desventajas En los trabajos [89, 91] *m*encionan las ventajas y desventajas. Las ventajas: Es de más fácil entrenamiento, no se estanca en un mínimo local a diferencia de las NN, la complejidad y el error puede ser modificable, es eficaz en espacios de gran dimensión, eficiente en la memoria, se pueden especificar diferentes funciones del núcleo para la función de decisión. *Las desventajas*: Selección del kernel ideal y no proporcionan directamente estimaciones de probabilidad.



(a) Separación para 3D[89].

(b) Separación para 2D[90].

Figura 3.3: Hiper-planos.

3.4.2 Hiperparametros

Como se meconio anteriormente, el kernel hace que el comportamiento de este modelo cambien a nuestras necesidades, a continuación se muestran los kernels más comunes[91, 92]:

- *Lineal*:

$$K(x1, x2) = x_1^T x_2 \tag{3.1}$$

  Aprendizaje de dos clases.

- *Polinómica*

$$K(x1, x2) = (x_1^T x_2 + 1)^\rho \tag{3.2}$$

  ρ representa el orden del polinomio.

- *Sigmoide*

$$K(x1, x2) = tanh(\beta_0 x_1^T x_2 + \beta_1) \tag{3.3}$$

  β_0 y β_1 son coeficientes arbitrarios.

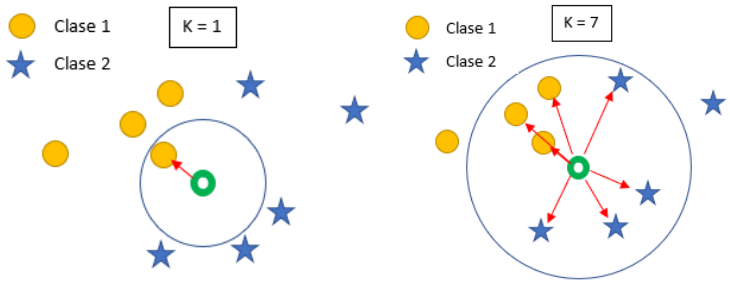
- *Función de base radial (RBF) o gaussiana*

$$K(x1, x2) = exp\left(-\frac{||x_1^T x_2||2}{2\sigma^2}\right) \tag{3.4}$$

σ representa la anchura del kernel.

3.4.3 Vecinos más cercanos (KNN)

En los trabajos [93, 94] KNN, es uno de los más antiguos, pero sigue siendo unos de los métodos de clasificación y regresión con un alto valor de precisión comparado con nuevos métodos como el SVM. Este método clasifica datos de prueba basándose en datos de entrenamiento previamente clasificados. KNN normalmente utiliza la distancia euclidiana como métrica y una *K* como la cantidad de vecinos a usar para una comparación. Dependiente de la cantidad de vecinos establecidos en K, tomará esa cantidad de datos con las menores distancias euclidianas con relación al nuevo dato de entrada. Se hace una comparación de la cantidad de cada clase y el que tenga la mayor presencia será la clasificación que tendrá el nuevo dato. En la figura 7 se muestran dos ejemplos usando diferentes valores de K, donde en la figura 3.4a se establece la cantidad de vecinos K = 1 y el resultado será que el nuevo dato será clasificado como clase 1 y para la figura 3.4b se utiliza un K = 7, donde la clasificación para el nuevo dato será de clase 2.



(a) KNN con K = 1. (b) KNN con K = 7.

Figura 3.4: Clasificación KNN.

Aunque este método es fácil de entender, implementar y entrenar. El sitio [91] y el trabajo [93] mencionan que una de las principales desventajas de este método es la elección

del valor de K, ya que; "La elección óptima del valor depende en gran medida de los datos: en general, un mayor K suprime los efectos del ruido, pero hace que los límites de clasificación sean menos distintos"[91]. Otra desventaja importante que es muy mencionada es el costo computacional, ya que guarda (almacena todos los todos ya clasificados) y utiliza los datos ya clasificados.

3.4.4 Hiperparametros

Los hiper parametros que presenta este modelo en base a la librería **scikit-learn** son los siguientes [91]:

- N° vecinos: tipo entero, default=5.

  Esta es la cantidad de vecinos a considerar.

- Pesos: ‘uniform’, ‘distance’,default=’uniform’.

  ‘uniform’ : La influencia de todos los vecinos es igual.

  ‘distance’ : Los vecinos más cercanos tendran un mayor pesos o importancia.

- Algoritmo : Es el algoritmo para calcular los vecinos más cercanos. ‘ball_tree’, ’kd_tree’, ’brute’, y ’auto’. ’auto’ trata de seleccionar el mejor de los anteriores algoritmos.

- Tamaño de hoja: tipo entero, default=30.

  Se utiliza este parámetro cuando el algoritmo seleccionado sea ’ball tree’ o ’kd tree’ y esto afecta la velocidad y la memoria requerida para el árbol.

- P: entero, default=2. Parámetro de potencia para la métrica de Minkowski. p = 1, esto es equivalente a usar la distancia manhattan (l1) y la distancia euclidean (l2) para p = 2. Para p arbitraria, se usa la distancia minkowski (l_p)

- Metrica: cadena, default=’minkowski’. El valor predeterminado es "minkowski", que da como resultado la distancia euclidiana estándar cuando p = 2. Otras tipo de distancias son ’cityblock’, ‘cosine’, ‘euclidean’, ‘haversine’, ‘l1’, ‘l2’, ‘manhattan’, ‘nan euclidean’.

- N° trabajos: entero, default=None. El número de trabajos paralelos a ejecutar para la búsqueda de vecinos.

3.4.5 Bosque Aleatorio

En los trabajos [95, 96, 97] hablan sobre el RF siendo este el más utilizado ampliamente como un método de clasificación ya que éste ofrece un gran grado de precisión, el manejo de grandes cantidades de datos, el manejo de la correlación entre variables, mientras menor sea la correlación entre los árboles menor será el sobre ajuste del RF, lo cual pocos métodos pueden ofrecer. El RF es una colección de un número finito de árboles de decisión, los cuales los nodos y las características son puestas aleatoriamente para reducir la correlación. Cada árbol de decisión tiene como resultado una clasificación y la clasificación con mayores ocurrencias será la mejor clasificación posible. Las desventajas de este método son: mientras más árboles el costo aumentara. En la figura 3.5 se muestra el proceso de RF, donde se tiene nodos para ir separando por características, al final de cada árbol se concluyó con una clasificación, *Reptil, Reptil y Mamífero*, de las cuales Reptiles apareció dos veces, esa será la clasificación final para el dato de entrada.

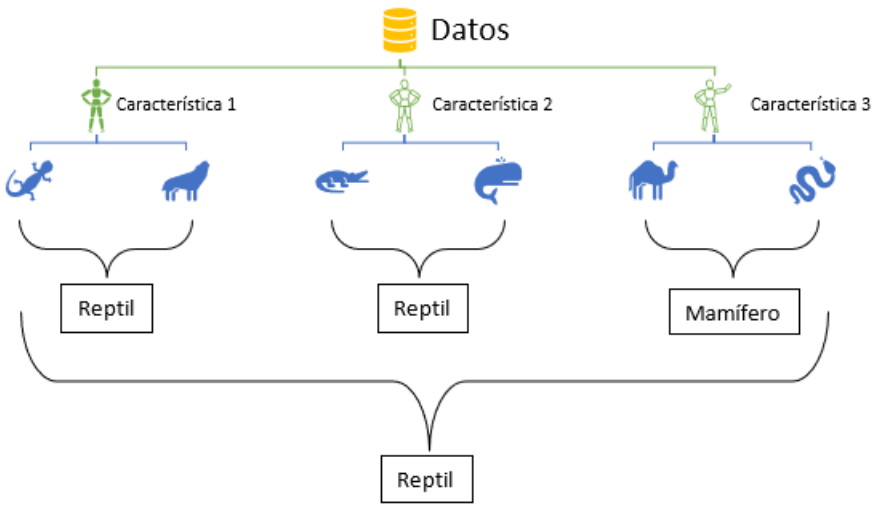


Figura 3.5: Modelo Bosque Aleatorio.

3.4.6 Hiperparametros

Los hiper parametros que presenta este modelo en base a la librería **scikit-learn** son los siguientes [91]:

- N° estimadores: entero, default=100. Es el numero de arboles en el bosque.
- Criterio: “gini”, “entropy”, “log_loss”, default=”gini”. La función para medir la calidad de una división.
- Profundidad maxima: entero, default=None.
- La profundidad máxima del árbol. Si es Ninguno, los nodos se expanden hasta que todas las hojas sean puras o hasta que todas las hojas contengan menos de min_samples_split samples.
- min_samples_split: entero o flotante, default=2. El número mínimo de muestras requeridas para dividir un nodo interno.
- min_samples_leaf:entero o flotante, default=1. El número mínimo de muestras requeridas para estar en un nodo de hoja
- min_weight_fraction_leaf:flotante, default=0.0. La fracción ponderada mínima de la suma total de pesos (de todas las muestras de entrada) requerida para estar en un nodo hoja. Las muestras tienen el mismo peso cuando no se proporciona sample_weight.
- max_features“sqrt”, “log2”, None: entero o flotante, default=”sqrt”. La cantidad de características a considerar al buscar la mejor división:

3.5. Procesamiento de la información para modelos de ML

3.5.1 Imputación por moda

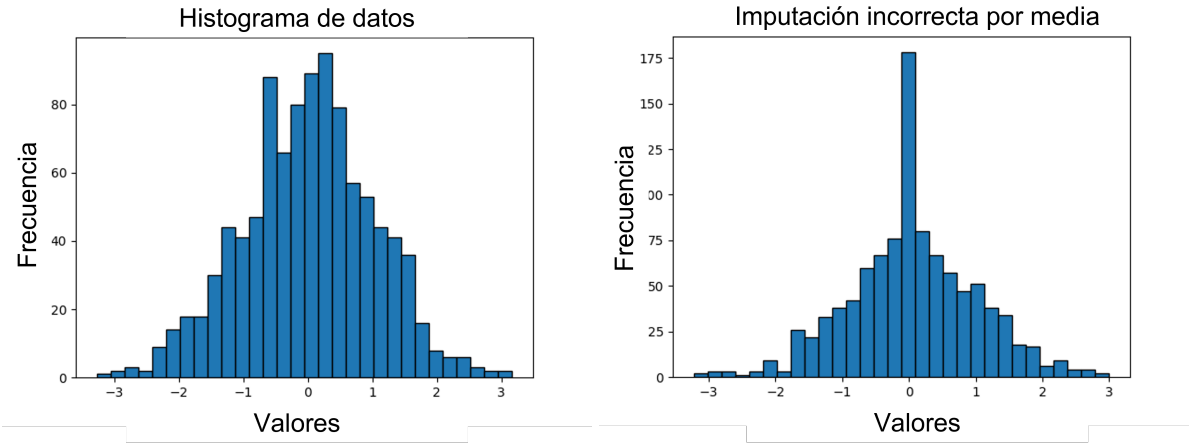
Los métodos de imputación es un conjunto de métodos que rellenan los valores faltantes de una base de datos. Dependiendo de la naturaleza de cada base de datos existe una

técnica de imputación que de mejores aproximaciones para no afectar la distribución de los datos.

Con el método de imputación por moda se busca tomar una cierta cantidad de vecinos (3,5 o 10) que se encuentren más cercanos desde arriba y hacia abajo del valor faltante y así poder calcular la moda para rellenar los datos faltantes. La desventaja es la reducción de su varianza y por ende la distribución se ve modificada. también es importante no hacer esta imputación con valores que han sido también imputados [98].

3.5.2 Distribución de los datos

Como se mencionó anteriormente, existen varias maneras de llenar los valores faltantes lo cual es bueno ya que no se pierde la instancia completa y se evita que afecte de forma negativa a cualquier modelo de ML. Sin embargo hay que tener cuidado en no modificar el comportamiento de los datos, esto se puede lograr mediante la inspección visual de la distribución de los datos como se comente en [98]. Lo que se busca es que los histogramas de cada atributo no cambien en forma al hacer algún tipo de imputación de datos. En la figura 3.6 se puede observar una mala selección de imputación de datos por moda(fig. 3.6b), donde claramente la distribución de los datos se ve afectada.



(a) Histograma Original. (b) Histograma imputado.

Figura 3.6: Mala imputación de media.

3.5.3 Normalizaciones

La normalización de forma simplificada es un método que nos permite tener la misma relación entre cada uno de los valores de cualquier conjunto de datos. Esto para que cualquier método de ML tenga un comportamiento más estable, ya que la distancia entre cada dato es la misma. A continuación, se muestran los dos tipos de normalizaciones utilizadas en este proyecto.

- Normalización MaxMin

  Este método de normalización es muy común. Convierte el valor más grande en uno y el valor más chico en cero, de esta forma están modificados nuestro punto de referencia para que la distancias en valores sea las homogéneo. Sin embargo, como lo comentan los sitios [7]. [99] este tipo de normalización NO es bueno trabajando con valores atípicos. La fórmula se aprecia en la ecuación 3.5. En la figura 3.7 se puede apreciar el problema con los datos atípicos.

$$x^{'} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.5}$$

- Normalización StandardScale o Z-score

  Es una técnica que evita el problema de los datos atípicos como lo refiere [7] donde a cada dato se le resta la media de la variable y se le divide por la desviación típica. Pero el sitio [99] comenta que sirve para algunos valores atípicos, pero no tan extremos. La ecuación se muestra en la formula 3.6. Cabe mencionar que la fórmula es igual a la de estandarización.

$$x^{'} = \frac{x - \mu}{\sigma} \tag{3.6}$$

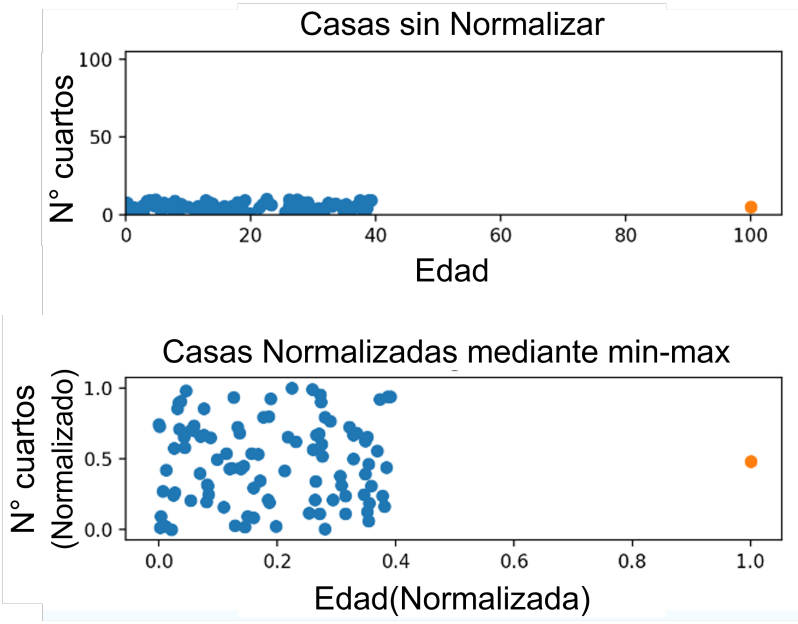


Figura 3.7: Mala normalización con datos atípicos [7].

3.5.4 PCA

El análisis de componentes principales, conocido como Principal Component Analysis (PCA) por sus siglas en inglés, es un método para reducir la dimensionalidad de una base de dato y que minimiza la perdida de información. Además, que al final de aplicar este método, los atributos son independientes de los demás [98, 100]. Los pasos para aplicar PCA como se menciona en [98, 100] son:

1. Eliminar el valor de decisión.
2. Estandarizar los datos de entrada.
3. Restar a cada valor la media de su propio atributo.
4. Calcular la matriz de covarianza.
5. Calcular eigenvalores y eigenvectores.
6. Ordenar los eigenvalores de mayor a menor.

7. Seleccionar los componentes principales que sumen la mayor información deseada.

8. Obtener la nueva base de datos con los atributos seleccionados.

Este método tiene como desventaja ser muy sensible a los *outliers*."Por esta razón, surgieron variantes de PCA para minimizar esta debilidad. Entre otros se encuentran: RandomizedPCA, SparcePCA y KernelPCA" [100].

3.5.5 Matriz de Correlación de Pearson

La matriz de confusión de Pearson, también conocida como matriz de correlación, es una matriz cuadrada que muestra las correlaciones entre las variables de un conjunto de datos. Es una medida de la relación lineal entre dos variables y su coeficiente de correlación se conoce como coeficiente de correlación de Pearson.

El coeficiente de correlación de Pearson puede variar entre -1 y 1, donde un valor de -1 indica una correlación negativa perfecta, 0 indica que no hay correlación y 1 indica una correlación positiva perfecta. Esta matriz ayuda a comprender las asociaciones y patrones de correlación entre las variables en un conjunto de datos.

Los que se busca en este trabajo al usar la correlación de Pearson y conocer cuáles de las características esta más relacionada con el atributo de decisión el cual es el estado del peatón (Cross True), que se puede apreciar en la tabla 4.1. A continuación se presenta una matriz de correlación de Pearson:

$$\text{Matriz de correlación de Pearson} : \begin{bmatrix} 1 & r_{12} & \dots & r_{1n} \\ r_{21} & 1 & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \dots & 1 \end{bmatrix}$$

3.6. Métodos de evaluación en el comportamiento de modelos DL y ML

3.6.1 K-fold

El método K-fold Cross Validación es utilizado para conocer el desempeño de predicción de algún modelo y tener una métrica de comparación contra otros modelos. Este método trabaja bien incluso para pocos datos, ya que los datos son divididos en subconjuntos dependiendo del valor de la variable K.

El método de K-fold es descrito en los siguientes pasos y cabe recalcar que estos pasos solo son una prueba, lo cual no es suficiente por lo cual se deben realzar el procedimiento varias veces [8, 101]. En la figura 3.8 se aprecia el procedimiento.

1. Revolver el conjunto de datos aleatoriamente.
2. Separar los datos en K conjuntos, donde k debe ser menor o igual a la cantidad de datos.
3. Selecciona un conjunto de los k conjuntos, el cual será el conjunto de datos para pruebas, mientras que los conjuntos restantes son utilizados para el entrenamiento.
4. Entrenar el modelo con los conjuntos de entrenamiento.
5. Probar el modelo con el conjunto de entrenamiento.
6. Guardar las métricas de evaluación.
7. Eliminar el modelo.
8. Seleccionar otro conjunto como de prueba y hacer los demás conjuntos como los de entrenamiento.
9. Repetir el procedimiento K veces necesarias del punto 2 al 8.

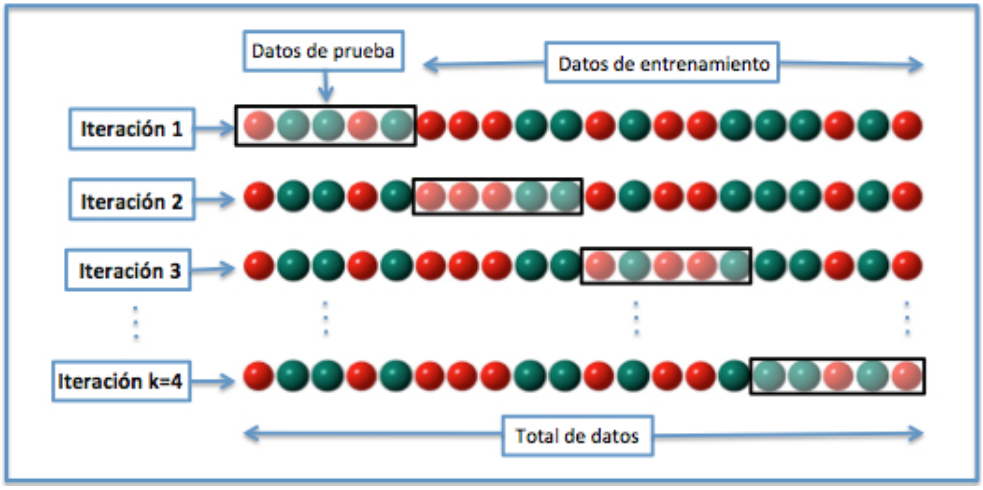


Figura 3.8: K-fold [8].

3.6.2 Matriz de Confusión

Las matrices de confusión son una herramienta utilizada en la evaluación de modelos de clasificación. Es una tabla que muestra la comparación entre las etiquetas reales de un conjunto de datos y las etiquetas predichas por un modelo.

La matriz de confusión tiene dimensiones NxN, donde N es el número de clases o categorías en el problema de clasificación. Los elementos de la matriz representan la cantidad de instancias que se clasificaron correcta o incorrectamente en cada clase.

En una matriz de confusión típica, los elementos en la diagonal principal representan las predicciones correctas, mientras que los elementos fuera de la diagonal principal representan las predicciones incorrectas.

Las matrices de confusión son útiles para evaluar el rendimiento del modelo de clasificación en términos de precisión, recall (sensibilidad), especificidad y otras métricas. También permiten identificar patrones de errores, como falsos positivos, falsos negativos y confusiones entre clases similares.

Para el objetivo de este proyecto la clasificación binaria con las clases Cruzando y No cruzando, una matriz de confusión típica tendría el siguiente formato:

- Verdaderos Positivos (VP): Indica la cantidad de instancias de la clase Cruzando que se predijeron correctamente como Cruzando.

		Predicciones		
		No Cruzando	Cruzando	Total
Real	No Cruzando	VN	FP	$VN + FP$
	Cruzando	FN	VP	$FN + VP$
	Total	$VN + FN$	$FP + VP$	N

- Verdaderon Negativos (VN): Indica la cantidad de instancias de la clase No cruzando que se predijeron correctamente como No cruzando.
- Falsos Positivos (FP): Indica la cantidad de instancias de la clase No cruzando que se predijeron incorrectamente como Cruzando (falsos positivos).
- Falsos Negativos (FN): Indica la cantidad de instancias de la clase Cruzando que se predijeron incorrectamente como No cruzando (falsos negativos).

En resumen, una matriz de confusión en un problema de clasificación binaria con las clases Cruzando y No cruzando muestra la cantidad de aciertos y errores en las predicciones del modelo para cada clase. Es una herramienta útil para evaluar el rendimiento y la precisión del modelo en términos de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

3.6.3 Métricas de evaluación

En el área de ML y de DL existen cuatro métricas básicas que surgen a partir de la matriz de confusión, estas son la exactitud, precisión, recall, y f1-score.

- Exactitud: "Es la proporción de muestras clasificadas correctamente entre las muestras positivas clasificadas" [27]. Se puede ver en la fórmula 3.7.

$$Exactitud = \frac{\text{VP} + \text{VN}}{\text{AV} + \text{VN} + \text{FP} + \text{FN}} \tag{3.7}$$

- Precisión:"Es la proporción de muestras clasificadas correctamente entre todas las muestras" [27]. Se puede ver en la fórmula 3.8.

$$Precisin = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{3.8}$$

- Recall: ”También conocida como sensibilidad, es la proporción de muestras clasificadas correctamente entre las muestras positivas reales” [27]. Se puede ver en la fórmula 3.9.

$$Recall = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{3.9}$$

- F1-Score: ”Es un promedio ponderado de la precisión y recall” [27]. Se puede ver en la fórmula 3.10.

$$F1 = \frac{2 * Precisin * Recall}{Precisin + Recall} = \frac{2 * \text{VP}}{2 * \text{VP} + \text{FP} + \text{FN}} \tag{3.10}$$

3.6.4 Gráficas de comportamiento de modelos ML

Existen un par de gráficas para la clasificación binaria que son utilizadas para conocer el rendimiento de cualquier modelo de ML para tareas de clasificación binaria, receiver operating characteristic (ROC) y PR.

- La gráfica de ROC contiene en el eje x ratio de falsos positivos y el eje y ratio de falsos negativos, y cada eje va desde cero hasta uno. ” La tasa de falsos positivos se calcula como el número de positivos verdaderos divididos entre el número de positivos verdaderos y de falsos negativos” [102], esto también es conocido como sensibilidad o recall.

  Esta gráfica es muy útil para saber la capacidad de los modelos de diferenciar entre las clases. Pero suele utilizarse para conjuntos de datos que tengas sus *clases balanceas*, ya que es sensible al desequilibrio de las clases. En la figura 3.9 se observan tres ejemplos de diferentes comportamientos, la gráfica de color rojo muestra un perfecto desempeño, la gráfica de color azul se considera un buen desempeño, pero la gráfica de color verde

un el peor comportamiento, ya que el modelo no sabe distinguir entre las clases y sus predicciones seria aleatorias

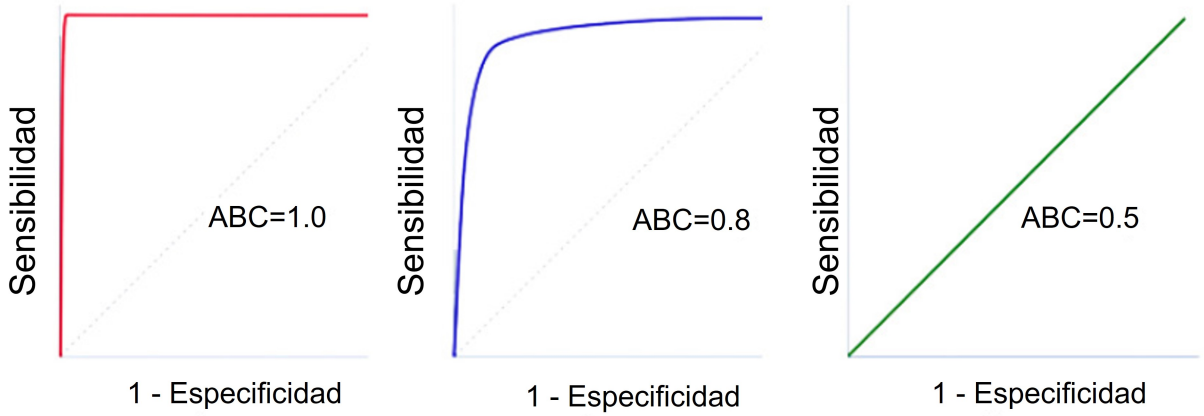


Figura 3.9: Comparación de diferentes gráficas ROC.

En resumen, valores pequeños en el eje X indican pocos falsos positivos y muchos verdaderos negativos. Valores grandes en el eje Y indican elevados verdaderos positivos y pocos falsos negativos.

- La curva PR también es utilizada para conocer el desempeño de cualquier modelo de ML y a diferencia de la cuerva ROC esta es utilizad para conjuntos de datos *des balanceados* y "La clave del uso de la curva de precisión-sensibilidad es que no tiene en cuenta los falsos negativos. La curva de precisión-sensibilidad solo se preocupa de la clase positiva, es decir, de la clase minoritaria"[102]. En la figura 3.10 se observan diferentes cuervas de PR. La curva en color morado corresponde a un excelente desempeño, la curva de color verde es un desempeño relativamente bueno y la cuerva de color azul tiene el peor comportamiento. Lo que se busca en esta gráfica es que modelos es que la precisión se mantenga lo más alta posible mientras el valor de recall va aumentado y que sea también lo más alto posible. En otras palabras, se busca que el porcentaje de predicciones positivas correctas contra todas las predicciones positivas sea siempre cercano a uno y que al mismo tiempo el porcentaje de predicciones positivas contra los estados reales positivos sean lo más cercano a uno.

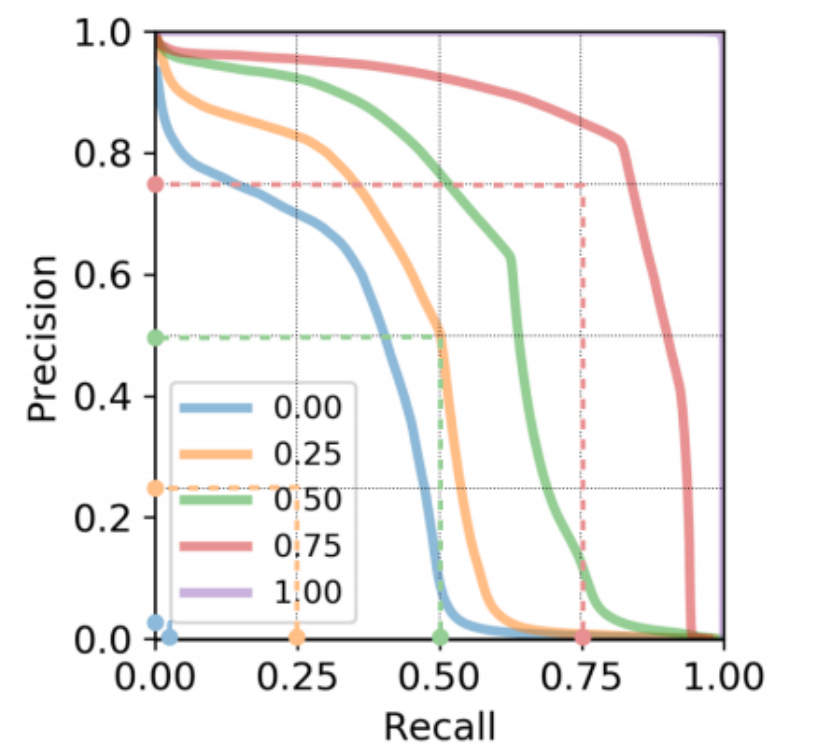


Figura 3.10: Comparación de diferentes gráficas PR [9] .

3.7. Unificación de características

Para poder utilizar las características deseadas como la orientación de la cabeza, la propuesta de utilizar los ángulos de las rodillas y entre otras más, se utilizó un formato CSV o Excel para así poder manejar y entrenar a los modelos KNN, SVM, y RF de mejor manera. En la tabla 3.3 se muestra la estructura que tendrá la base de datos para el entrenamiento de los modelos ML y la generación de la misma.

Tabla 3.3: Ejemplificación de unión de características

Atributo 1	Atributo 2	...	Atributo n	Atributo decisión
a	b		c	d

4. MATERIALES Y MÉTODOS

En esta sección se verá los pasos realizados para cumplir el objetivo general y los objetivos específicos de esta tesis; también se vera la base de datos que se utilizó, así como las métricas a considerar y las herramientas empleadas.

4.1. Metodología

La metodología empleada consta de tres secciones, 1.- Generación de base de datos la cual consiste en hacer detecciones de semáforos, señales de alto vehicular, pasos peatonales, y el peatón más cercano para poder obtener los ángulos de sus rodillas con el fin de generar una base de datos de tipo CSV para entrenar los modelos ML. 2.- Entrenamiento de modelos ML, este paso utilizara la base de datos creada en el paso 1, esta base de datos será procesada de tal modo que se tendrán varias diferentes versiones de la base de datos original, se seleccionaran las mejores bases de datos para los modelos de clasificación de ML y finalmente seleccionar el mejor modelo ML. 3.- Sistema Final, este último paso es la combinación de los dos anteriores, el cual solo detectara las variables ya mencionadas y serán mandadas a los modelos de ML seleccionados para la clasificación del estado del peatón imagen por imagen. En la figura 4.1 se maestra el diagrama de flujo general explicado anteriormente.

4.1.1 *Generación de base de datos*

Para la generación de la base de datos se tuvieron que realizar algunos puntos para lograrlo. Se utilizó un modelo pre-entrenado (ResNet50) para la detección de algunas variables. Se entrenó el modelo de YOLOV8 para la detección de dos variables más que no se pudieron encontrar en alguna base de datos pública. Una vez obteniendo las variables y de haber detectado al peatón más cercano, este último es recortado de la imagen original y por medio de la librería Mediapipe se le realiza una esqueletización y se calcula los ángulos inter-

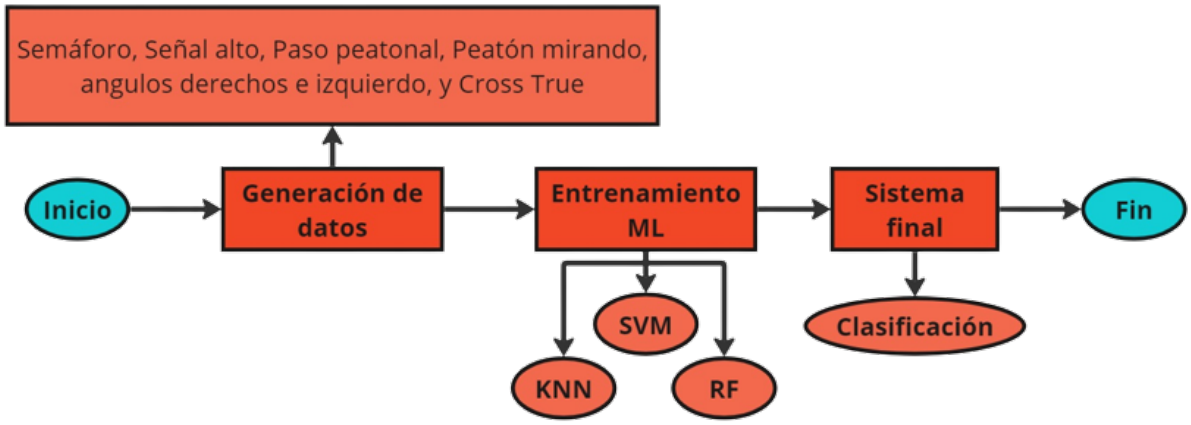


Figura 4.1: Diagrama general de metodología.

nos de las rodillas. Estos dos ángulos será parte también de la base de datos final en este paso. En la figura 4.2 se observa el diagrama de flujo con la secuencia para la generación de la base de datos, se tiene como entrada un vídeo, del cual se va analizando imagen por imagen hasta acabar. La detección es realizada a cada imagen para obtener las variables del entorno y la presencia del peatón, en dado caso que no detecte o no pueda detectar el modelo al peatón, se asigna el valor de -9999 para poder así guardas las demás variables con una instancia en la base de datos. Una vez pasada esta etapa y se aúlla podido detectar al peatón, las coordenadas de esta detección son utilizadas para encontrar en la base de datos JAAD el estado real del peatón (Cruzando o No Cruzando). En seguida se recorta el peatón de la imagen original y se obtiene su esqueleto para calcular los ángulos internos de las rodillas por medio de la ley de cosenos, siendo la rodilla el punto medio entre la cadera y el talón. A continuación, se muestra algo de información del entrenamiento realizado para los modelos de detección en forma de lista.

1. Entrenamiento de modelos de Detección

   a) ResNet50

   ResNet50 es un modelo pre-entrenado con la base de datos publica COCO, las clases que este modelo detectara son personas, semáforos, y señales de alto.

   b) YOLOV8

Debido a que las características pasas peatonales y lo orientación de la cabeza (Mirando, No Mirando) no se encontraron juntas en ninguna base de datos publica, fue necesario generar una base de datos propia a partir de la base de datos publica JAAD. La base de datos publica generada fue nombrada Crosswalk&Pedestrian Looking [103] y puede ser encontrada en la página de Robloflow.

c) Crosswalk&Pedestrian_Looking

Esta base de datos fue creada por la necesidad de detectar los pasos peatonales y la orientación de la cabeza. La obtención de estas imágenes fue tomada de la base de datos JAAD y para evitar un sobre entrenamiento para cualquier modelo de detección, se tomaron los primeros 276 vídeos (80 %) del total de 346. De cada uno de estos vídeos se tomó una imagen cada segundo o cada 30 FPS, dando una cantidad de 2237 imágenes. Esta base cuenta con las siguientes características:

- 2237 imágenes.
- 4680 anotaciones.
- Tamaño de imagen 1920x1080.
- Balance de clases 2,270, 1,838, y 572 para No Mirando, Paso peatonal, y Mirando.

Al final de la generación de esta base de datos se obtendrá una base de dato del tipo CSV con los atributos Instancia, Semáforo, Señal alto, Paso peatonal, Peatón Mirando, Ángulo Rodilla Derecho, Ángulo Rodilla Izquierda, y Cross True como se muestra en la tabla 4.1.

Tabla 4.1: Muestra de la Base de datos a crear.

Instancia	Semáforo	Señal alto	Paso peatonal	Peatón Mirando	Ángulo Rodilla Der	Ángulo Rodilla Izq	Cross True

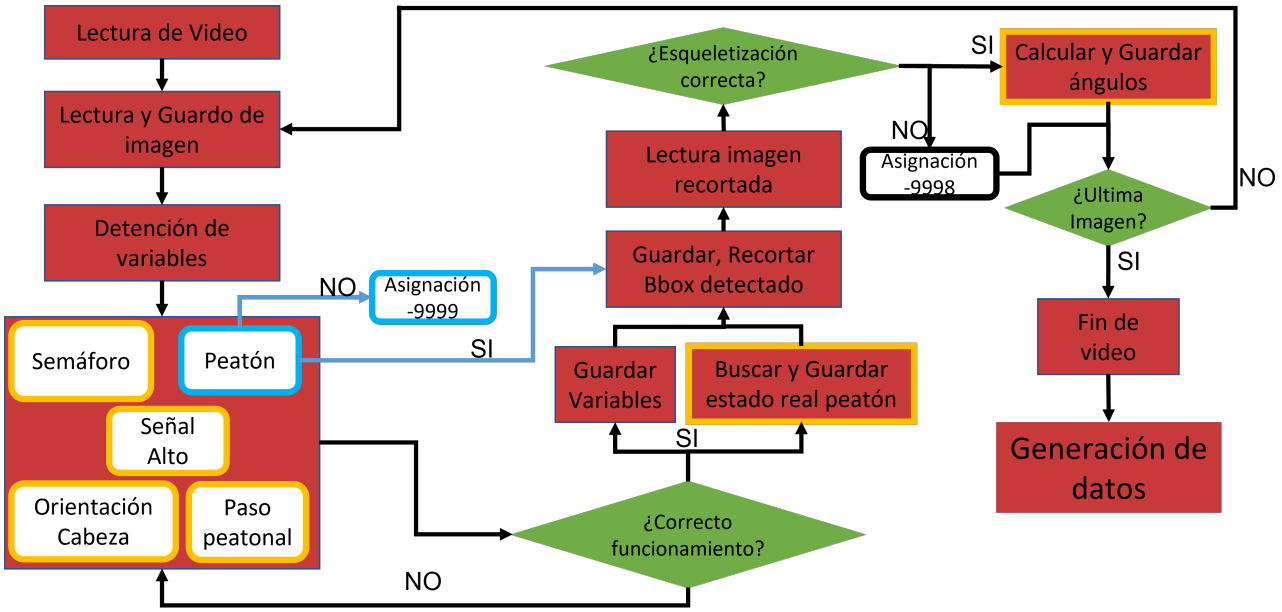


Figura 4.2: Generación de la base de datos.

4.1.2 *Entrenamiento de modelos ML*

El segundo paso para esta metodología es el entrenamiento de los modelos de aprendizaje máquina, donde en base a la investigación donde los más utilizados para la clasificación de la intención del peatón son: KNN, SVM, y RF. Los hiperparametros seleccionados para estos modelos fueron los siguientes:

- SVM: Kernel = ‘rbf’, gamma = ’auto’, probability = True. Se seleccionó el kernel rbf por este trabaja en infinitas dimensiones y la función gamma fue para encuentre la mejor separación de los datos, y el parámetro de probabilidad para poder generar gráficas de PR.

- KNN: n_neighbors = 5, metric = ’minkowski’. Estos parámetros fueron utilizadas ya que el valor de 5 vecino es comúnmente usada y la métrica de distancia, fue la que mejores resultados dieron en la materia de ML.

- RF: Criterion = ”gini”, splitter = ”best”, max_Depth = None, min_samples_Split = 2. Estos parametros son los de default y se dejaron así ya que se tiene una gran cantidad de

combinaciones para explorar, lo cual pudo afectar el tiempo de termino de este trabajo en tiempo y forma.

Cualquier modelo de ML es tan bueno como la base de datos con la que fue entrenada. Por esta razón la base de datos generada en el paso anterior, se le aplicará una preparación y un análisis de los datos para así tener no solo variantes de la base de datos, sino también conocer que atributos realmente son necesarios para los modelos para realizar la clasificación. En la figura 4.3 se presenta el diagrama de flujo para el procedimiento de preparación, análisis, entrenamiento y evaluación de los modelos, KNN,SVM,y RF.

Con lo mencionado, se realizaron varias bases de datos con variaciones a partir de la base de datos original 5.1 para ver el comportamiento que los modelos ML pudieran tener. Es por eso por lo que se generarán las siguientes variantes:

- **Imputada**

  Esta variante se eliminarán varias instancias y se le imputaran algunas instancias que presentaron datos faltantes en el atributo de decisión, lo cual se explica en la sección 5. El método de imputación utilizado fue por moda. Y las instancias eliminadas fueron las que presentaron valores menores a ceros para los atributos de los ángulos de las rodillas. Al final esta base de datos consto de los 6 atributos seleccionados Semáforo, Señal de alto vehicular, Pasos Peatonales, Orientación de la cabeza, ángulo derecho y ángulo izquierdo de las rodillas.

- **Imputada Reducida**

  Esta variante surgió de la base de datos imputada y la aplicación del método de reducción de dimensionalidad, PCA, lo cual nos da una base de datos con solo dos atributos, los ángulos de las rodillas.

- **Imputada maxmin**

  Esta variante surgió de la base de datos imputada y la aplicación del método de normalización maxmin para los atributos, ángulos de las rodillas ya que los demás atributos

son del tipo booleano. Siendo así que se tendrán los 6 atributos totales con los ángulos de las rodillas normalizados.

- **Imputada maxmin Reducido**

  Esta variante surge de la base de datos imputada maxmin a la cual se le aplicó el método de reducción de dimensionalidad PCA. Esto no deja una base de datos con 4 atributos de los 6 originales, estos atributos son Semáforos, Pasos peatonales, Orientación de la cabeza, Ángulo derecho de la rodilla normalizada.

- **Sintéticos**

  Esta variante surge de la base de datos imputada a la cual se le aplico el método de generación de datos sintéticos, Adaptive synthetic sampling approach for imbalanced learning (ADASYN) con el fin de balancear las clases cruzando y no cruzando. Esta base mantiene los 6 atributos originales.

- **Sintéticos Reducidos**

  Esta base de datos surgió de la base de datos sintéticos y se aplicó el método PCA lo cual arrojo una base de datos con solo 2 atributos, ángulos de las rodillas.

- **Sintéticos maxmin**

  Esta variante surgió de la base de datos sintética y la aplicación del método de normalización maxmin para los atributos, ángulos de las rodillas ya que los demás atributos son del tipo booleano. Siendo así que se tendrán los 6 atributos totales con los ángulos de las rodillas normalizados.

- **Sintéticos maxmin Reducido**

  Esta variante surge de la base de datos sintética maxmin a la cual se le aplicó el método de reducción de dimensionalidad PCA. Esto no deja una base de datos con 4 atributos de los 6 originales, estos atributos son Semáforos, Pasos peatonales, Orientación de la cabeza, Ángulo derecho de la rodilla normalizada.

- **Imputados StandarScale Reducido**

  Esta base de datos surgió de la base de datos imputada, y a la cual se le aplico otro método de normalización llamando StandardScale y también se le aplico el método PCA. Obteniendo al final una base de datos con 4 atributos y todos de ellos normalizados Semáforos, Pasos peatonales, Orientación de la cabeza, Ángulo izquierdo.

Teniendo 9 base de datos y 3 modelos a entrenar, el total de modelos entrenados fue de 27 modelos a analizar y seleccionar. Una vez seleccionados los modelos se guardan para su implementación en el sistema final.

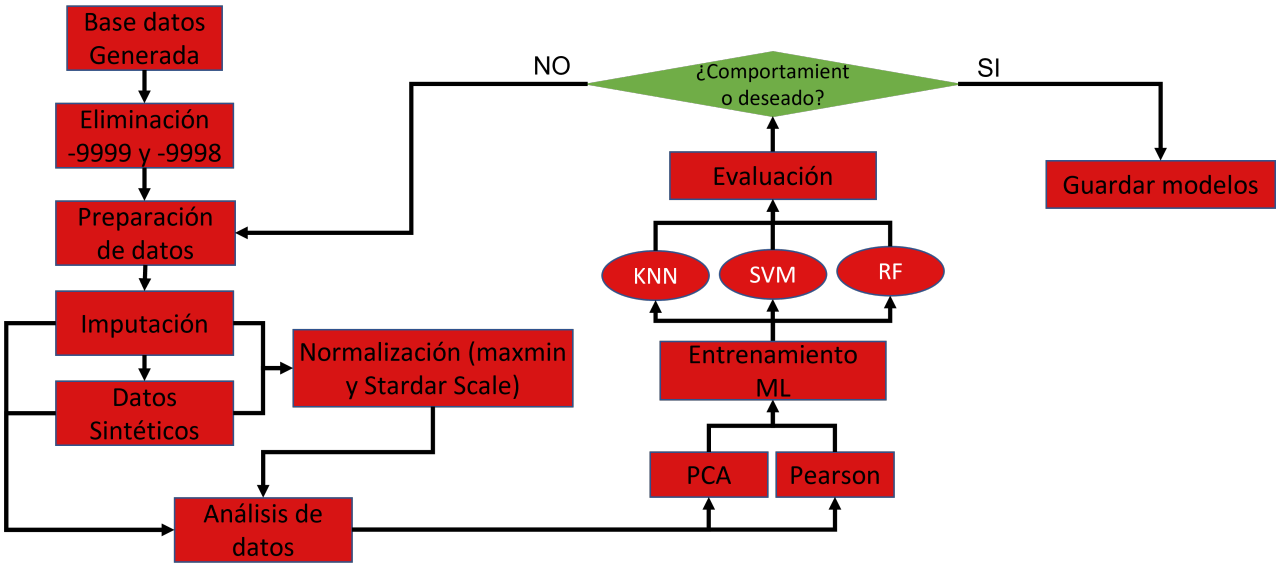


Figura 4.3: Entrenamiento de modelos de ML.

4.1.3 *Sistema final*

Como último paso se tiene el sistema final, el cual consiste en la unión de punto 1 y los modelos entrenados del punto 2. Como se puede apreciar en la figura 4.4, este sistema en vez de detectar y generar una base de datos, este guarda y manda las variables detectadas y calculadas a los modelos de ML para la clasificación del estado del peatón. En caso de que sistema no pueda detectar, se enviara un mensaje de peatón no detectado y se seguirá con la siguiente imagen en caso de que se haya detectado al peatón, pero no se logró generar el esqueleto de este, se mandará un mensaje de no se pudo renderizar el esqueleto.

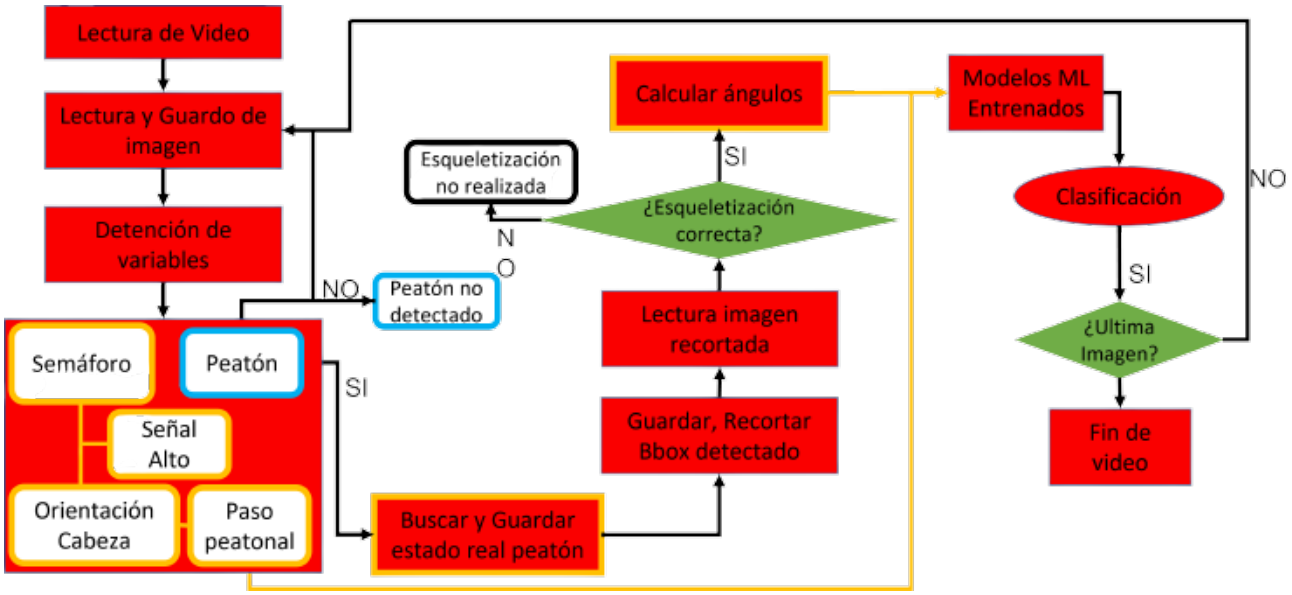


Figura 4.4: Sistema final.

4.2. Recursos

Dado a que los recursos requeridos para este trabajo son el uso de algún hardware con algún tipo de tarjeta gráfica o GPU, se opto por trabajar por medio de la herramienta conocida como Google Colab Pro. Del cual se trabajó con las siguientes características:

- Procesador: TU104
- Disco : 167 GB
- RAM sistema: 12.7 GB
- GPU: Tesla T4

4.3. Software

Los softwares utilizados en este trabajo se en listan a continuación con el propósito de dar toda las características necesarias para que este trabajo se reproducible en caso de usarlo.

- Python 3.10.12
- pycocotools 2.0.6
- pixellib 0.7.1
- mediapipe 0.10.1
- OpenCV 4.7.0
- Ultralytics YOLOv8.0.20
- torch-2.0.1+cu118

5. RESULTADOS Y DISCUSIÓN

5.1. Modelos de detección

Los modelos de detección se encargan de localizar y clasificar por medio de la caja delimitadora en toda la imagen. Para la tarea de detectar las variables deseadas, *semáforos, paso peatonal, señal de alto, orientación de la cabeza, y el peatón* se ha implementado los modelos de Resnet50 y YOLOV8. Donde el modelo ResNet50 es el encargado de detectar a un peatón, señales de tráfico(semáforos y señales de alto vehicular), y mientras que YOLOV8 se encarga de la detección de los pasos peatonales y la orientación de la cabeza del peatón.

Estos modelos son entrenados no solo para poder generar una base de datos para los entrenamientos de los modelos de ML , sino también son utilizados para el sistema final con el objetivo ya no de entrenar a los modelos de ML si no de alimentar a los modelos ya entrenados.

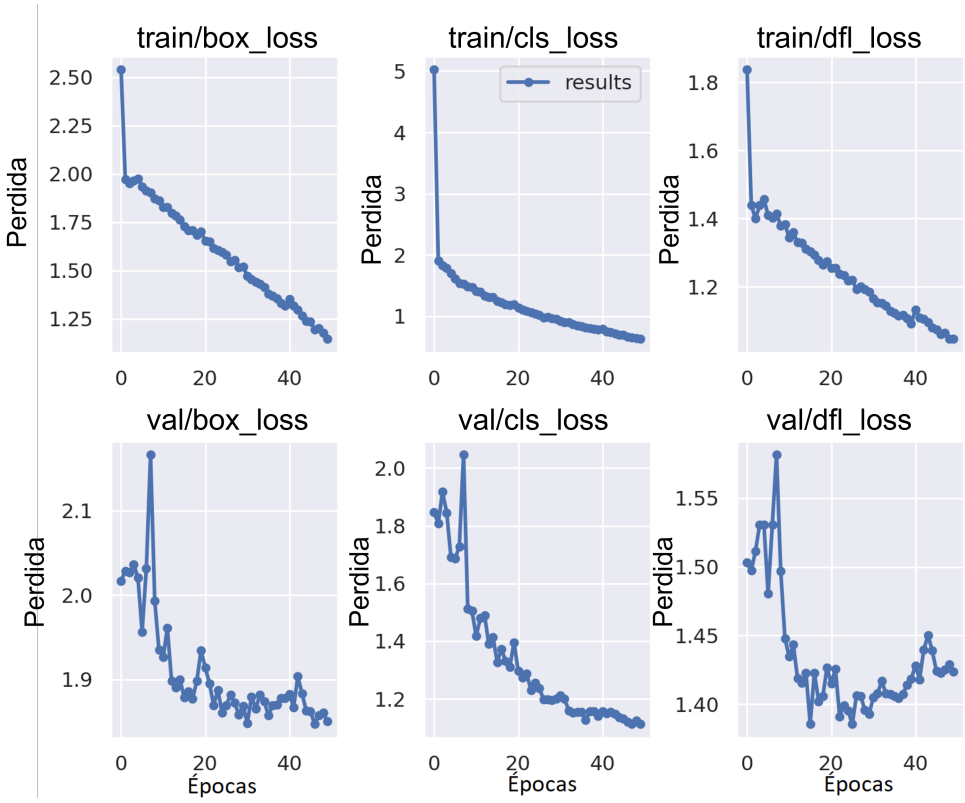
5.1.1 YOLO V8

Este modelo YOLOV8 fue entrenado con el objetivo de detectar los pasos peatonales y la orientación de la cabeza del peatón (mirando o no mirando). Para que este modelo pudiera cumplir su tarea se debido crear una base de datos con las variables deseadas (pasos peatonales y la orientación de la cabeza), debido a que no se encontró ninguna base de datos publica con estas características. Esta base de datos publica puede encontrarse en la plataforma de robooflow como Crosswalk&Pedestrian_Looking [103].

Para llevar a cabo el entrenamiento se tomó y utilizo la notebook de roboflow para el modelo de detección YOLOv8 [6]. Este entrenamiento se llevó a cabo con los parámetros pre-establecidos, el único parámetro que se vario fue la cantidad de épocas, 30 y 50 épocas fueron probadas. A continuación, se muestran los resultados de los dos entrenamientos.

- Curvas de entrenamiento y validación.

  En la figura 5.1 las gráficas de entrenamiento y validación para el valor de la pérdida para las cajas delimitadoras o Bounding Box, la perdida de las clases, y la perdida de la función de perdida pérdida focal distributiva (Distributional Focal Loss (DFL) por sus siglas en ingles). Para el comportamiento de las cajas delimitadoras nos dice como va bajando el valor de la función de perdida, siendo así para el entrenamiento baja de manera desea, sin embargo, la de validación no baja tanto lo cual podría ser un indicativo de un sobre entrenamiento. Este comportamiento se repite para DFL. Por otra parte, el comportamiento es buenos para la gráfica que contiene la pérdida con respecto a las clases, C Y NC, ya que siempre se encentre entre los valores 1 y 2.

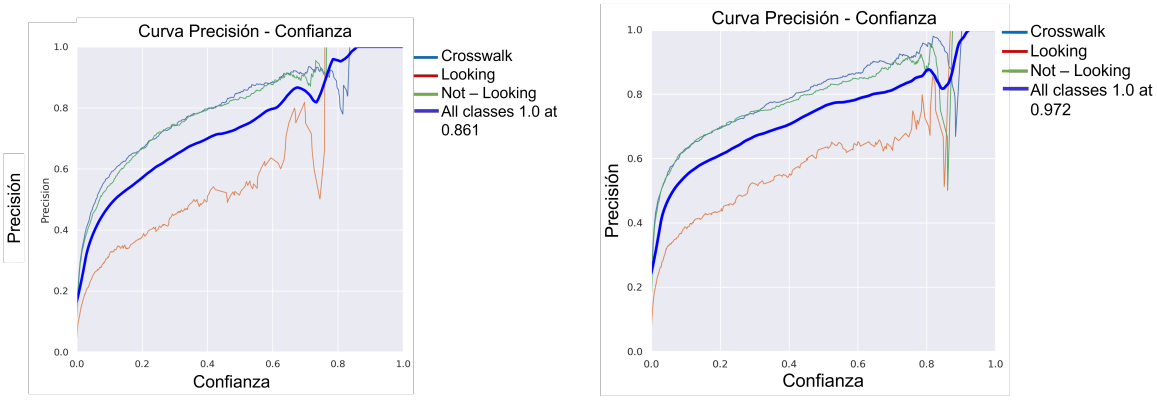


(a) 50 épocas.

Figura 5.1: Curvas de entrenamiento y validación.

- Curvas de precisión

  En la figura 5.2 se presenta las gráficas de precisión para los dos entrenamientos la figura 5.2a para el entrenamiento con 30 épocas y la figura 5.2b con 50 épocas. Estas gráficas muestran el valor que toma la precisión del modelo al ir incrementando el valor de confiabilidad. La gráfica 5.2b se observa que obtiene una mayor precisión y sube de manera constante.



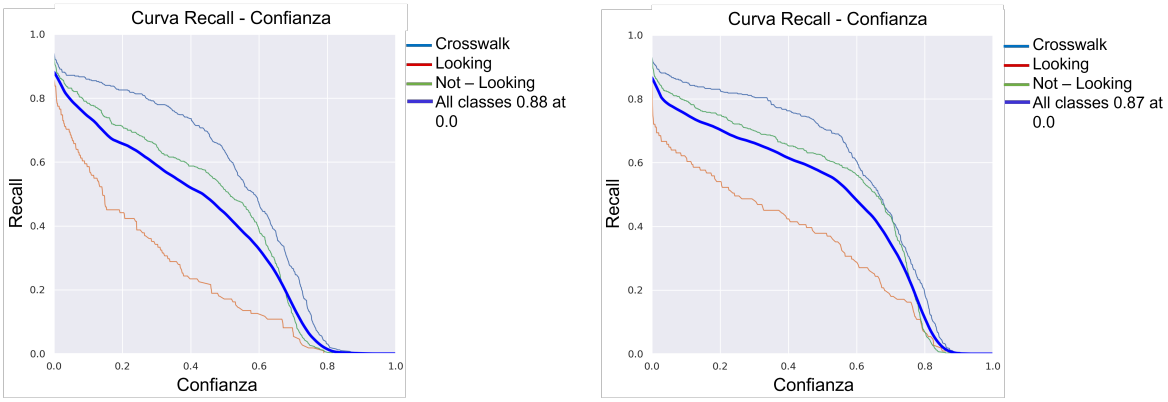
(a) 30 épocas.

(b) 50 épocas.

Figura 5.2: Curvas de precisión.

- Curvas de Recall o Sensitividad

  En la figura 5.3 se presenta las gráficas para la métrica Recall. Se puede observar una mejora en la métrica para el entrenamiento de 50 épocas (fig. 5.3b), ya que no decae tan rápido el valor del Recall. Esto nos dice que con el mismo valor de confiabilidad se puede obtener mayor valor en la métrica recall.

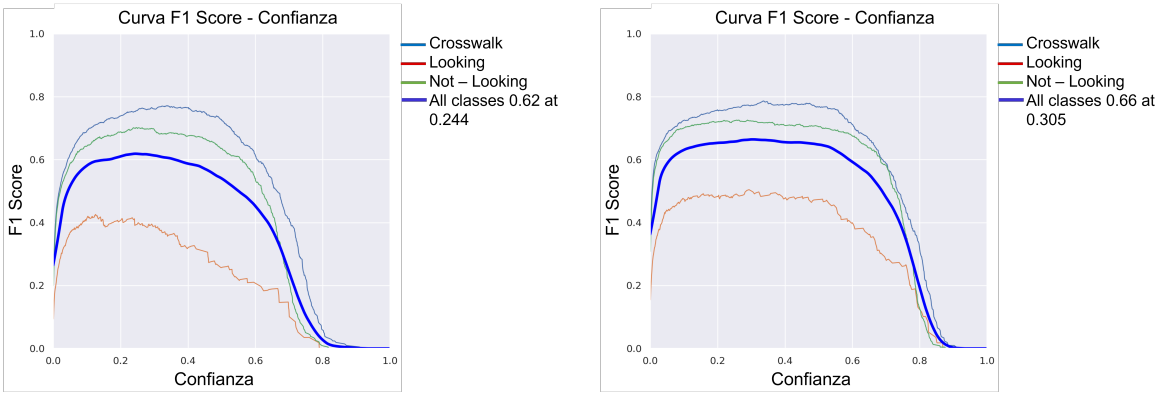


(a) 30 épocas. (b) 50 épocas.

Figura 5.3: Curvas de Recall o Sensitividad.

- Curvas de F1-Score

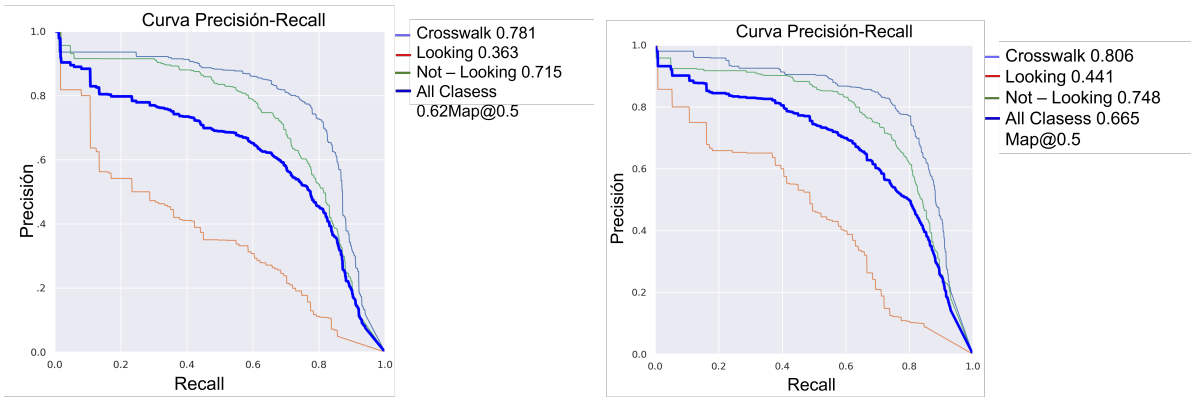
  En la figura 5.4 se presenta las gráficas para la métrica F1-Score o la media harmónica. Al igual que la gráfica de Recall (5.3), se puede apreciar que el entrenamiento de 50 épocas (fig. 5.4b) tiene un comportamiento más adecuado ya que se presenta una meseta y a una altura mayor que en la gráfica 5.4a.



(a) 30 épocas. (b) 50 épocas.

Figura 5.4: Curvas de F1-Score.

- Curvas de Precisión-Recall (PR) En la figura 5.5 se muestran las dos métricas precisión y Recall. Este comportamiento nos dice más claramente el comportamiento del modelo, por lo cual vemos que la figura 5.5b es mejor ya que las áreas bajo la curva tienen un valor mayor.



(a) 30 épocas. (b) 50 épocas.

Figura 5.5: Curvas de Precisión-Recall.

En base a las gráficas se seleccionó el modelo de YOLOV8 con 50 épocas de entrenamiento para la detección de pasos peatonales y la orientación de la cabeza del peatón, para los pasos de generación de base de dato y para el sistema final.

5.2. Generación de la base de datos

Una vez que los modelos de detección han sido seleccionado y entrenado, se implementaron para llevar a cabo el diagrama de flujo 4.2. Siendo así que al final de este proceso se obtuvo una base de dato del tipo Excel (.CSV). Esta base de datos consta de 670,927 instancias, es de tipo booleano, y consta de 7 atributos los cuales son: Semáforo, Señal de alto, Paso peatonal, Ángulo de la rodilla derecha e izquierda y como ultimo y séptimo atributo el atributo de decisión "*Cross True*". La base de datos se puede apreciar en la Tabla 5.1 donde el tipo de dato para los atributos Semáforo, Señal de alto, Paso peatonal, Peatón mirando, y Cross True son del tipo booleano, mientras que los atributos de los ángulos derecho e izquierdo son continuos.

Tabla 5.1: Base de datos original.

Instancia	Semáforo	Señal alto	Paso peatonal	Peatón Mirando	Ángulo Rodilla Der	Ángulo Rodilla Izq	Cross True
0	0	0	0	0	136.45	128.33	not-crossing
1	0	0	0	0	175.17	179.07	not-crossing
2	0	0	0	0	176.15	158.88	not-crossing
3	0	0	0	0	168.56	162.41	not-crossing
4	0	0	0	0	168.74	169.90	not-crossing
5	0	0	0	0	172.20	177.88	not-crossing
6	0	0	0	0	164.15	146.01	not-crossing
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
4882	0	0	0	0	-99999	-99999	not-crossing
4883	0	0	0	0	163.84	176.53	-99999
4884	0	0	0	0	132.70	112.99	not-crossing
4885	0	0	0	0	-99998	-99998	not-crossing
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
67086	0	0	0	0	170.34	163.99	crossing
67087	0	0	0	0	176.93	162.04	crossing
67088	0	0	0	1	171.90	122.92	crossing
67089	0	0	0	0	-99998	-99998	crossing
67090	0	0	0	0	-99998	-99998	crossing
67091	0	0	0	0	163.12	132.47	crossing

5.3. Modelos de aprendizaje maquina

Como se vio en la sección 3.4, los modelos seleccionados para realizar la clasificación y el reconocimiento de la intención del peatón son los modelos KNN, SVM, y RF. Pero antes de poder implementar y experimentar con estos modelos, es necesario recodar que cualquier modelo de inteligencia artificial es tan bueno como los datos con los que fueron entrenados. Por lo cual la calidad de los datos es primordial. A continuación, se presentan estos pasos realizados para mejorar la base de datos obtenida.

5.3.1 Preparación de los datos

1. Base de datos original

   Una vez que se cuenta con la base de datos 5.1. Se procedió a conocer sus características Esta base de datos contiene los atributos detectados, Semáforo, Señal de alto, Paso peatonal, y Peatón mirando. También contiene las características de los ángulos calculados en base a la detección del peatón y la generación de su esqueleto. Y como atributo de decisión o clase se encuentra como la última columna. Se recalcaron en color naranja los valores de los ángulos cuando no se pudo detectar el peatón(-99999) o cuando el sistema no pudo generar el esqueleto (-99998). También se recalcó de color rojo los valores en el atributo de decisión cuando hubo un problema de captura y se el valor como -99999.

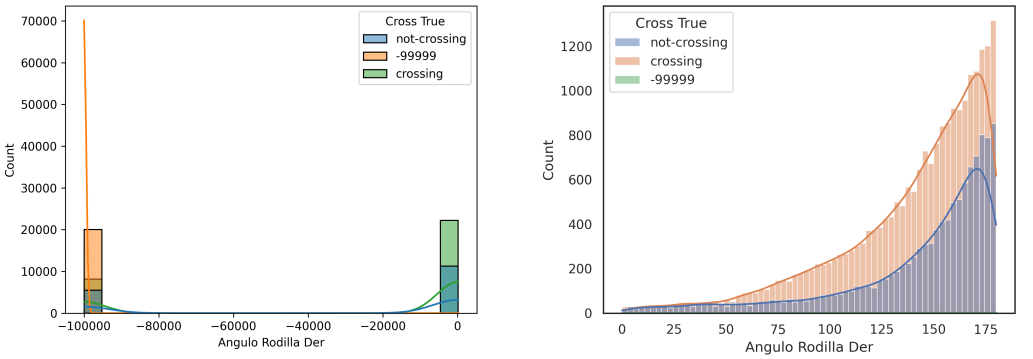
   Una vez identificado estos valores, la preparación de la base de datos tiene una serie de puntos que se deben de realizar para conocer el comportamiento de los datos y decidir qué tipo de técnicas son mejores para preparar la base de datos. Estos pasos son:

   - Conocer si hay datos faltantes. Para saber si existen valores faltantes se utilizó el método .unique() de la librería pandas para cada uno de los tributos. Para todos los atributos excepto los ángulos, los valores serán solo cero o 1. Se determino de esta manera que no contaba con valores faltantes a parte de los ya conocidos (-9999 y -9998).
   - Conocer el Balance de Clases. Se utilizo una función propia para conocer el balance de clases. Siendo así, la base de datos contiene un imbalance de clases marcado, not-crossing: 16,713 (24.91 %), -99999: 20,032 (29.86 %), y crossing: 30,347 (45.23 %).
   - La distribución de cada uno de los atributos.

     A continuación, se presenta las distribuciones de cada atributo en función de la clase de decisión para la base de dato original y junto a una base de datos donde se le ha eliminado las instancias cuyos atributos de los ángulos sea menor a ce-

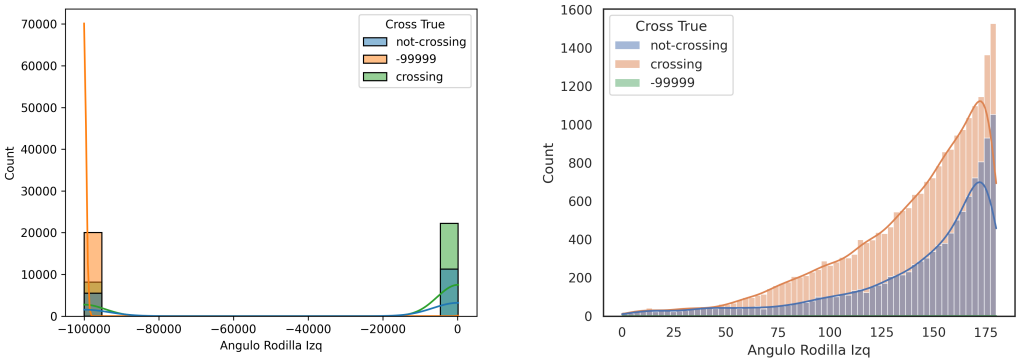
ro, ya que estos valores no dejan apreciar el comportamiento de los atributos en cuestión.

En las figuras 5.6 y 5.7 se puede observar de mejor manera el comportamiento de los ángulos de cada rodilla. Esta eliminación de instancias deja a la base de dato con una cantidad de 33,474 instancias.



(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.6: Distribución del atributo ángulo derecho.

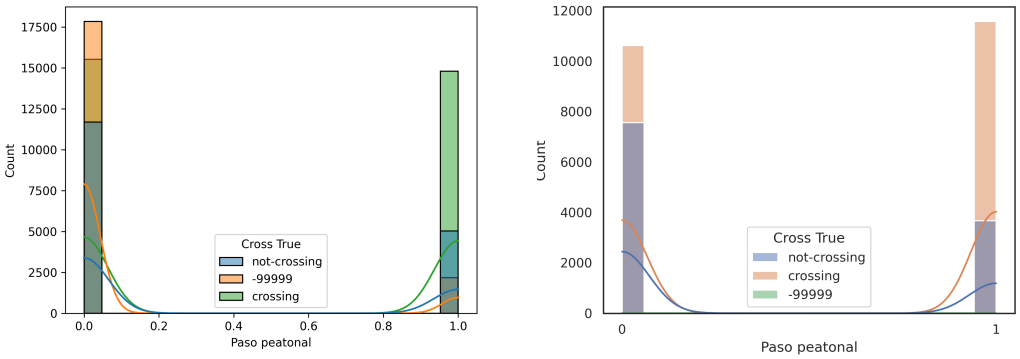


(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.7: Distribución del atributo ángulo izquierdo.

En la figura 5.8 se tiene la distribución para el atributo de pasos peatonales, donde al eliminar las instancias no deseadas, se puede apreciar que, en la presencia de un paso peatonal, los peatones tienden a cruzar más. Por otra parte, cuando

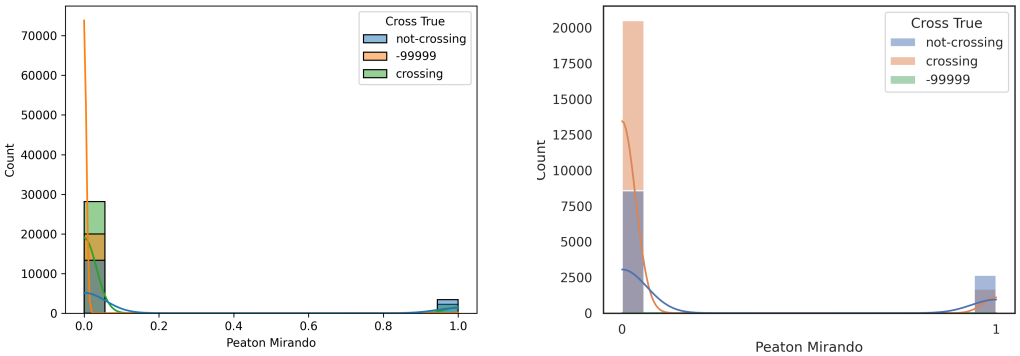
no existe un paso peatonal casi la misma cantidad de peatones tiende a cruzar a comparación de los que no cruzan.



(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.8: Distribuciones del atributo paso peatonales.

En la figura 5.9 se tiene la distribución para el atributo de orientación de la cabeza del peatón, el cual nos indica si el peatón está mirando o no. Se puede apreciar que la mayoría de los peatones que no miran hacia el vehículo, tienen a cruzar. Por otro lado, cuando los peatones miran hacia el vehículo, un poco más de los peatones prefieren no cruzar.

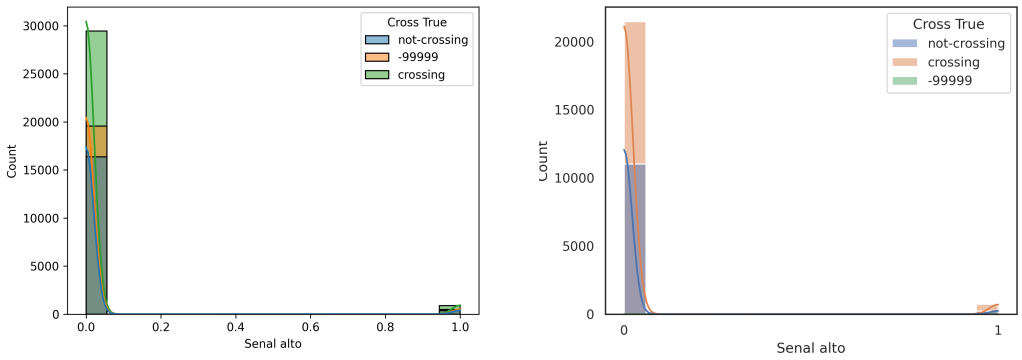


(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.9: Distribuciones del atributo orientación de cabeza.

En la figura 5.10 se muestra la distribución del atributo señal de alto vehicular. Y

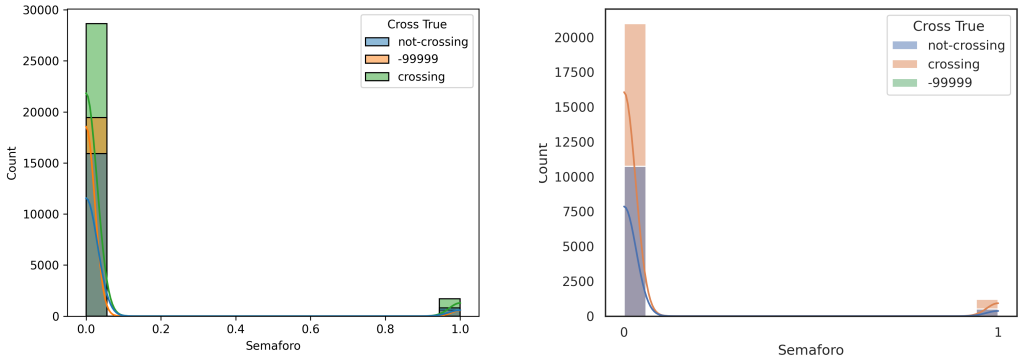
se puede observar que al no estar presente un semáforo la cantidad de peatones que cruza es mayor. En la presencia de algún semáforo, se alcanza a apreciar que la mayoría de los peatones cruza con la presencia de este atributo.



(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.10: Distribuciones del atributo señal de alto vehicular.

En la figura 5.11 tenemos la distribución del atributo semáforo. El cual se puede observar un comportamiento muy similar a la distribución del alto vehicular.



(a) Base de datos original. (b) Base de datos con instancias eliminadas.

Figura 5.11: Distribuciones del atributo semáforo.

2. Imputación de datos

   Como se observó en las gráficas de distribución de los datos, persiste la clases -99999 debido a esto se volvió a revisar el balance de clases. El balance de clases quedo de la siguiente manera: 16,713 (24.91 %) instancias para la clase not-crossing, 22,201 (66.32 %) instancias para la clase crossing y 31 (0.09 %) instancias para la clase -99999. Las 31 instancias que no se eliminaron pudo ser por un error de captura en código. Para resolver esto se decidió a aplicar una técnica de imputación por moda. Al final de esta imputación el balance de clases quedo de la siguiente manera: 11,253 (33.62 %) instancias para la clase not-crossing, 22,221 (66.38 %) instancias para la clase crossing. En la figura 5.12 se puede ver el balance de clases después de la imputación de datos, donde el valor cero y uno corresponde a no-cruzando y cruzando respectivamente.

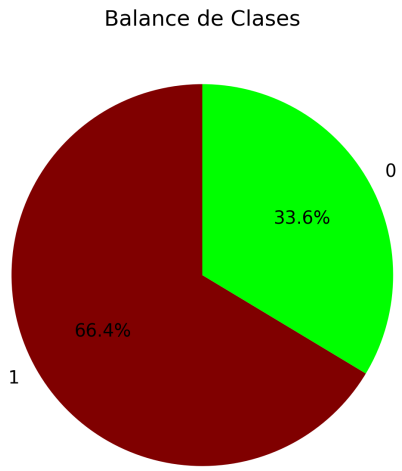
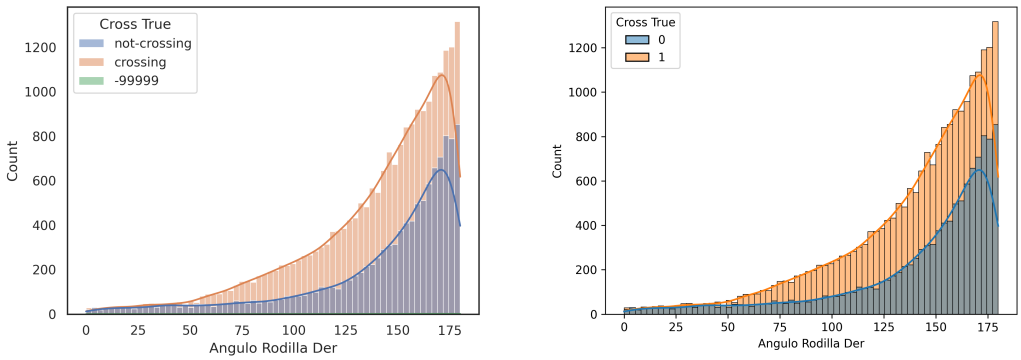


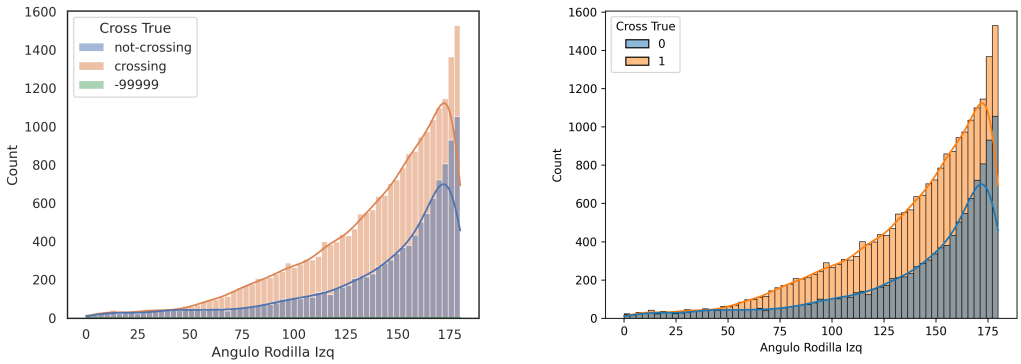
Figura 5.12: Balance de clases de la base de datos imputada.

A continuación, se muestra la comparación de la base de datos original y la base de datos aplica la imputación para verificar que el comportamiento de los datos no sea afectado.



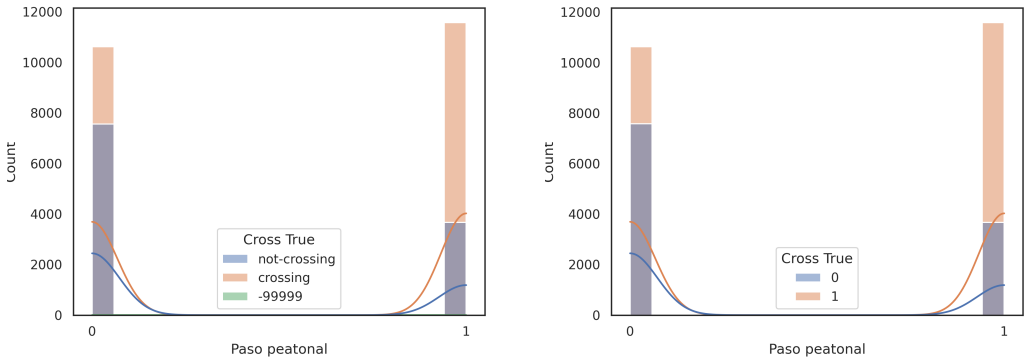
(a) Base de datos original. (b) Base de datos imputadas.

Figura 5.13: Distribución del atributo ángulo derecho.



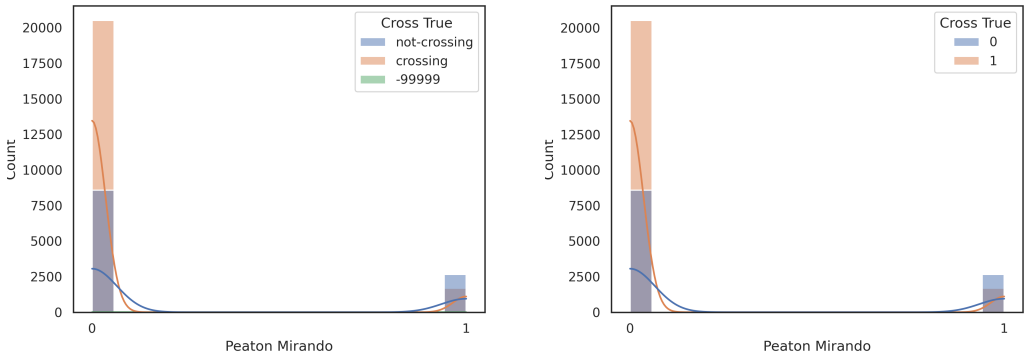
(a) Base de datos original. (b) Base de datos imputadas.

Figura 5.14: Distribución del atributo ángulo izquierdo.



(a) Base de datos original. (b) Base de datos imputada.

Figura 5.15: Distribuciones del atributo paso peatonales.

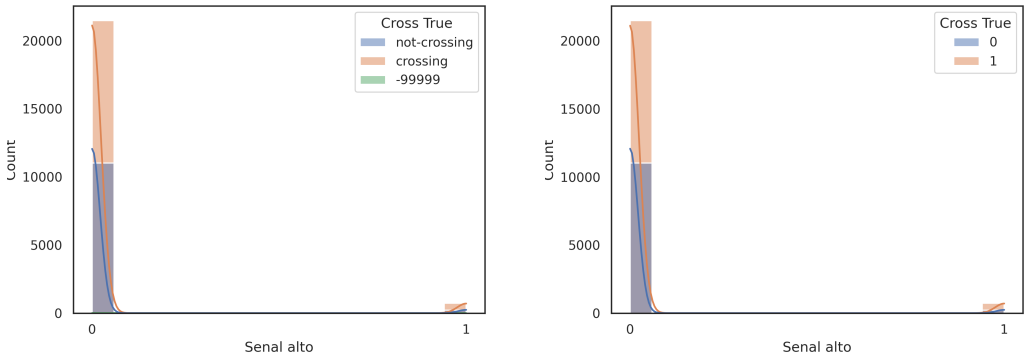


(a) Base de datos original. (b) Base de datos imputada.

Figura 5.16: Distribuciones del atributo orientación de cabeza.

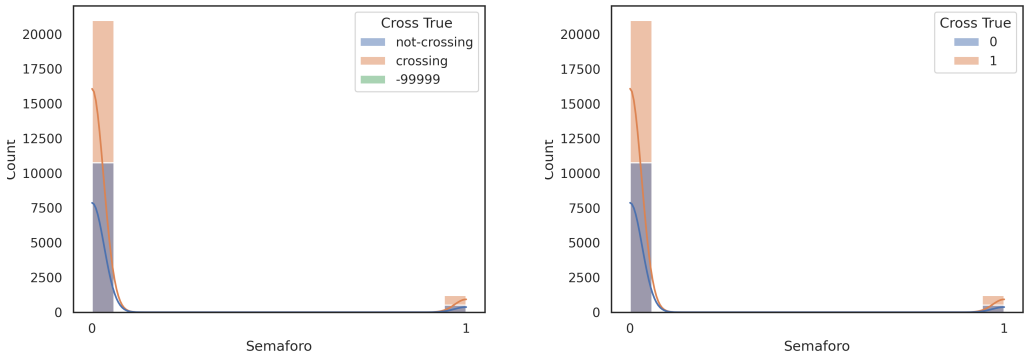
3. Datos sintéticos

Una vez que la imputación de datos fue realizada, se procedió a crear una variación de la base de datos imputada. Esta nueva versión se le aplicara el método de ADASYN esperando así que los modelos de ML tengan mejores resultados. Siendo así, el balance de clases que da la siguiente forma. Para la clase no-cruzando o cero tiene una cantidad de 22,598(50.42 %) y para la clase cruzando o uno 1.0: 22,221(49.58 %). En la figura 5.19 se observa que se logró el balance satisfactoriamente, sin embargo, se presentaran a continuación las distribuciones de cada atributo como se ha estado presentado antes para verificar que el comportamiento de los datos no ha sido afectado.



(a) Base de datos original. (b) Base de datos imputada.

Figura 5.17: Distribuciones del atributo señal de alto vehicular.



(a) Base de datos original. (b) Base de datos imputada.

Figura 5.18: Distribuciones del atributo semáforo.

4. Normalización de los datos

   Se realizo otra variación para las bases de datos ya mencionadas, imputada y sintética, esta variación es la aplicación de la normalización maxmin, y StardarScale para ver que comportamiento pudiera surgir con tan solo hacer una normalización. Y además en la parte de análisis de los datos, la normalización debe hacer se antes y no después al aplicar las técnicas PCA y la matriz de Pearson.

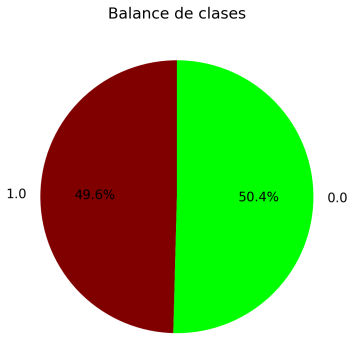
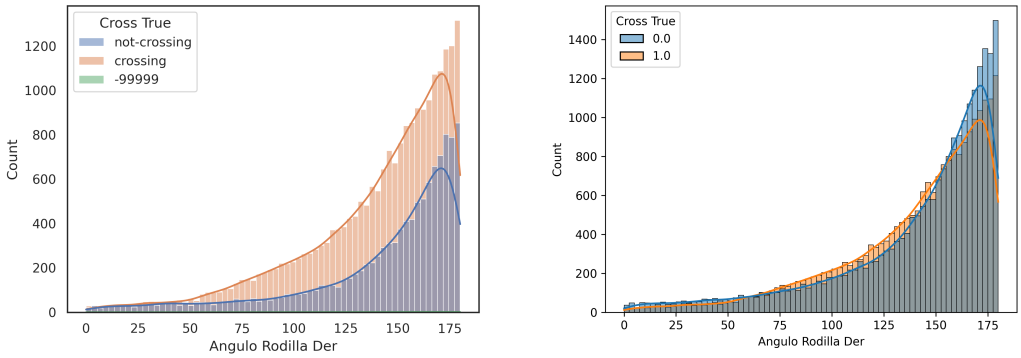


Figura 5.19: Balance de clases de la base de datos sintéticos.



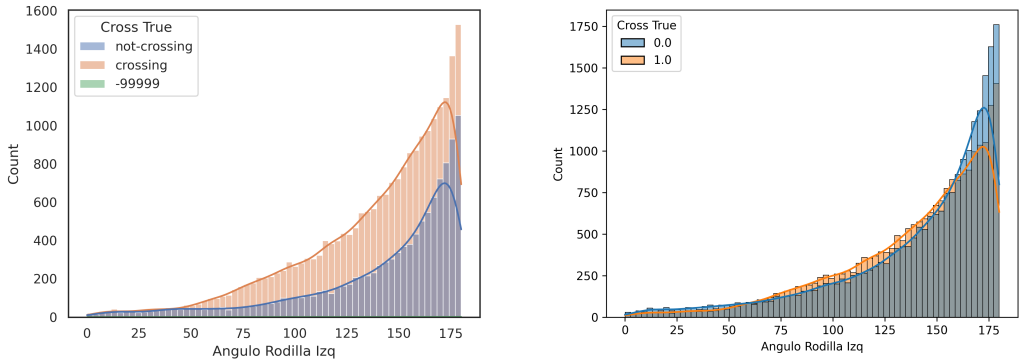
(a) Base de datos original.

(b) Base de datos sintética.

Figura 5.20: Distribución del atributo ángulo derecho.

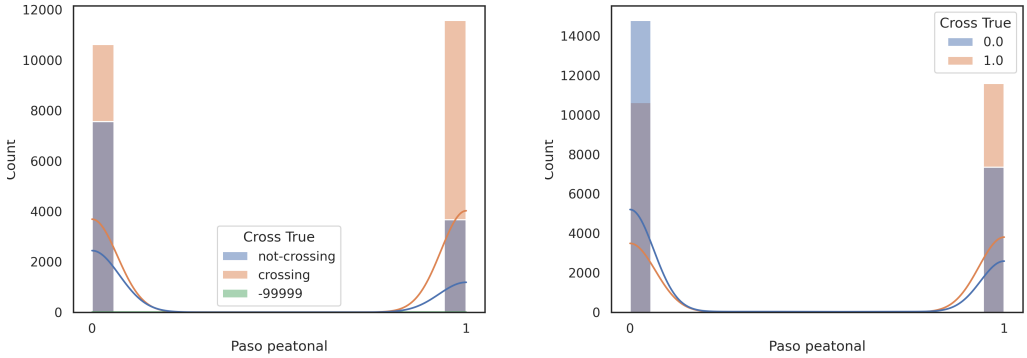
5.3.2 Análisis de los datos

El análisis de datos consiste en una variedad de técnicas a seleccionar para así poder discernir que características de la base de datos son las más importantes y así poder ahorrar tiempo y costo computacional. Para esta tesis se utilizaron las técnicas: Matriz de confusión de Pearson y la técnica de PCA. Con estas dos técnicas se establecieron que atributos realmente son necesarios para por hacer la clasificación o reconocimiento de la intención del peatón de cruza. A continuación, se mostrarán los resultados al aplicar la matriz de Pearson para las bases de datos imputada, sintética y la normalización maxmin a cada una de ellas, teniendo así 4 base de datos para, y para la técnica de PCA se le suma además de las anteriores,



(a) Base de datos original. (b) Base de datos sintéticos.

Figura 5.21: Distribución del atributo ángulo izquierdo.



(a) Base de datos original. (b) Base de datos Sintética.

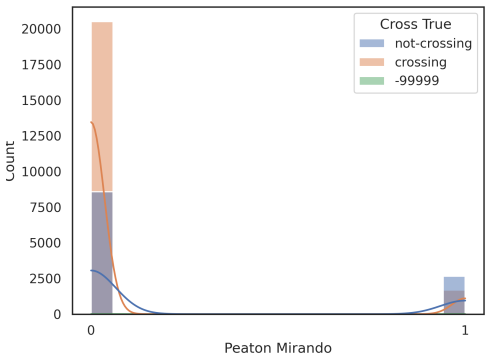
Figura 5.22: Distribuciones del atributo paso peatonales.

la normalización StandatScale, teniendo así 6 variaciones de la base de datos.

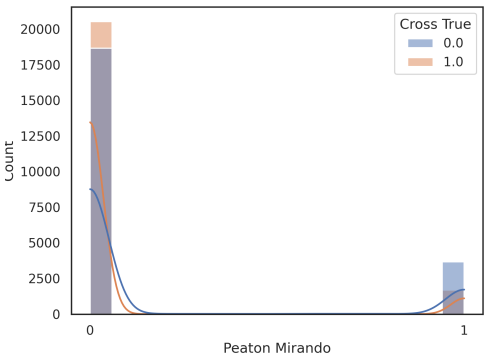
- Matriz de Confusión de Pearson

  Para este análisis se utilizaron la base de datos imputada, se utilizaron la base de datos imputada, sintética(balanceada) y sus normalizaciones maxmin.

  En la tabla 5.2 se puede apreciar estos atributos con mayor correlación con el atributo de decisión.

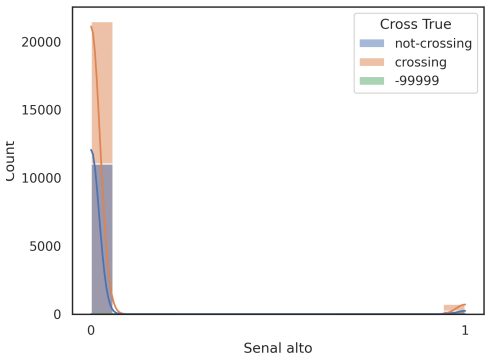


(a) Base de datos original.

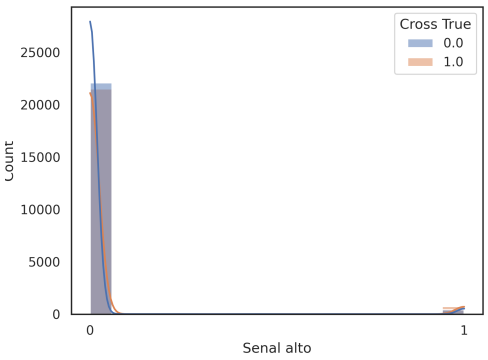


(b) Base de datos sintética.

Figura 5.23: Distribuciones del atributo orientación de cabeza.

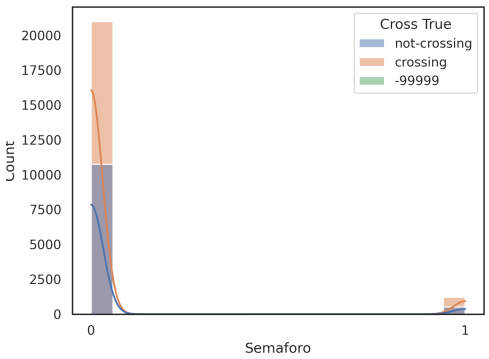


(a) Base de datos original.

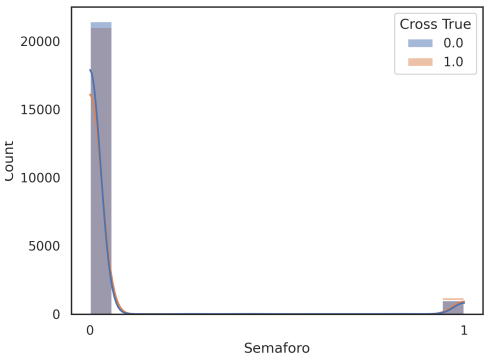


(b) Base de datos sintética.

Figura 5.24: Distribuciones del atributo señal de alto vehicular.



(a) Base de datos original.



(b) Base de datos sintética.

Figura 5.25: Distribuciones del atributo semáforo.

Tabla 5.2: Atributos más correlacionado al atributo de decisión.

Imputados		Imputados maxmin		Sintéticos		Sintéticos maxmin	
Paso peatonal	0.185	Paso peatonal	0.185	Paso peatonal	0.189	Paso peatonal	0.189
Peatón mirando	-0.226	Peatón mirando	-0.226	Peatón mirando	-0.141	Peatón mirando	-0.141

- Análisis de componentes principales (PCA)

  Para este paso ya se contaba con 6 bases de datos, la base de datos imputada, imputada normalizada (maxmin), imputada (StandarScale), sintética , sintética normalizada (maxmin), sintética (StandarScale). Estas 6 variantes de bases de datos fueron analizadas con esta técnica y el resultado se puede apreciar en la Tabla 5.3, donde se puede observar el porcentaje de información cada atributo para cada variante de base de datos, la base de datos imputada (imp.), imputada normalizada con el método maxmin (imp.maxmin), sintética (sint.), sintética normalizada maxmin (imp. maxmin),importada con la normalización StandardScale, y sintética con StandardScale.

Tabla 5.3: Porcentaje de información de cada atributo para las bases de datos.

Imp.		Imp. maxmin		Sint.		Sint. maxmin		Imp. StandarS.		Sint. StandarS.	
Ángulo derecho	0.81	Paso peatonal	0.47	Ángulo derecho	0.81	Paso peatonal	0.48	Ángulo izquierdo	0.27	Ángulo derecho	0.27
Ángulo izquierdo	0.19	Peatón mirando	0.22	Ángulo izquierdo	0.19	Peatón mirando	0.21	Paso peatonal	0.18	Semáforo	0.17
Paso peatonal	0.00	Ángulo derecho	0.13	Paso peatonal	0.00	Ángulo derecho	0.14	Peatón mirando	0.17	Señal-alto	0.17
Peatón mirando	0.00	Semáforo	0.09	Peatón mirando	0.00	Semáforo	0.09	Semáforo	0.16	Peatón mirando	0.16
Semáforo	0.00	Señal-alto	0.05	Semáforo	0.00	Señal-alto	0.05	Señal-alto	0.16	Señal-alto	0.16
Señal-alto	0.00	Ángulo izquierdo	0.03	Señal-alto	0.00	Ángulo izquierdo	0.03	Ángulo izquierdo	0.06	Ángulo derecho	0.06

Como resultado de este análisis de los datos por medio de la técnica de PCA y la matriz de correlación de Pearson, se obtuvo al final 9 variaciones de la base de datos original para poder así ver el comportamiento de los modelos de ML. La base de datos que se utilizaran para entrenar y validar los modelos de ML se en listan a continuación:

- **Imputada**

  Esta base de datos contiene todos los atributos (6) y el atributo de decisión *Cross True*.

- **Imputada Reducida**

  Esta base de datos contiene 2 atributos de los 6 originales (Ángulos de las rodillas) y el atributo de decisión *Cross True*.

- **Imputada maxmin**

  Esta base de datos contiene todos los atributos (6) y el atributo de decisión *Cross True*. A los atributos de las rollas se les fue aplicado una normalización maxmin.

- **Imputada maxmin Reducida**

  Esta base de datos contiene 4 atributos de los 6 originales (Paso peatonal, Peatón mirando, ángulo derecho, y semáforo) y el atributo de decisión *Cross True*. A los atributos de las rollas se les fue aplicado una normalización maxmin.

- **Sintéticos**

  Esta base de datos contiene todos los atributos (6) y el atributo de decisión *Cross True*. Y también se le aplico un balance de clases con el método ADASYN.

- **Sintéticos Reducidos**

  Esta base de datos contiene 2 atributos de los 6 originales (Ángulos de las rodillas) y el atributo de decisión *Cross True*. Y también se le aplico un balance de clases con el método ADASYN.

- **Sintéticos maxmin**

Esta base de datos contiene todos los atributos (6) y el atributo de decisión *Cross True*. A los atributos de las rollas se les fue aplicado una normalización maxmin. Y también se le aplico un balance de clases con el método ADASYN.

- **Sintéticos maxmin Reducida**

  Esta base de datos contiene 4 atributos de los 6 originales (Paso peatonal, Peatón mirando, ángulo derecho, y semáforo) y el atributo de decisión *Cross True*. A los atributos de las rollas se les fue aplicado una normalización maxmin. Y también se le aplico un balance de clases con el método ADASYN.

- **Imputados StandarScale Reducido**

  Esta base de datos contiene 4 atributos de los 6 originales (ángulo izquierdo, Paso peatonal, Peatón mirando, y semáforo) y el atributo de decisión *Cross True*. A los atributos de las rollas se les fue aplicado una normalización maxmin.

5.3.3 Entrenamiento y pruebas

Una vez que se contó con estas 9 bases de datos, se procedió a entrenar y probar los modelos de ML (KNN, SVM, y RF). Para realizar este proceso se utilizó la técnica de K-fold con un valor para K=5, esto debido a que las particiones más comunes son 5 y 10, siendo este último mucho más tardado en entrenar, la cual fue la razón del valor de 5. Los parámetros utilizados para cada modelo se muestran a continuación al igual que se presenta la figura 5.26 se muestran los resultados promedios de las métricas de los 27 modelos mediante el método K-fold.

- KNN : N° vecinos = 5, distancia = minkowski
- SVM : Kernel = ‘rbf’, gamma = ’auto’, Probabilidad = True
- RF : Criterio = ”gini”, Separador = ”best”, máxima profundidad = None, separaciones mínimas de pruebas = 2

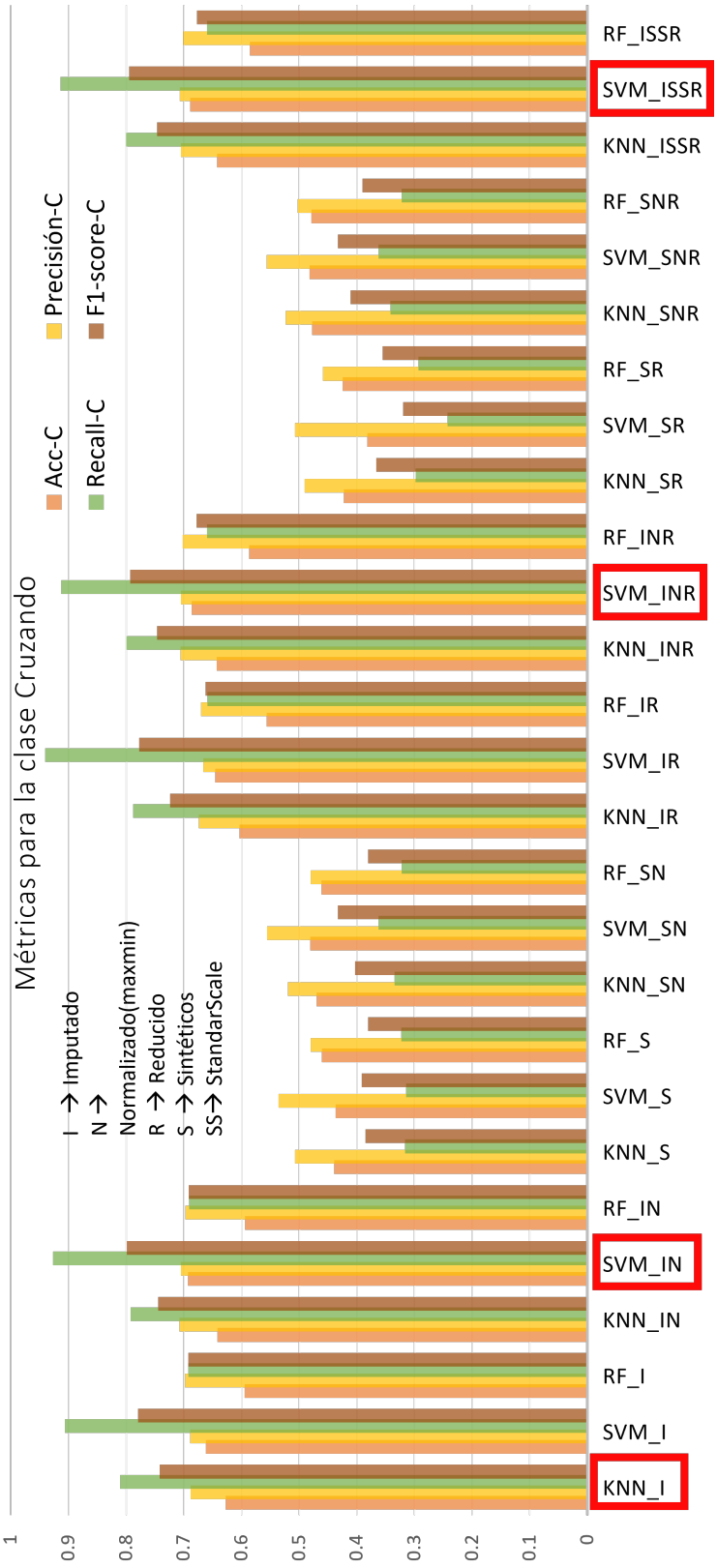


Figura 5.26: Gráfica de barras de los 27 modelos entrenado y probados con diferentes versiones de la base de datos original.

A continuación, se muestra una seria de tablas que muestran los resultados de las métricas obtenidas en el entrenamiento y prueba de los 4 modelos seleccionados al utilizar el método de k-fold. Cada tabla corresponde a un tipo de base de datos y a además se hizo un enfoque hacia las dos clases para ver como el comportamiento hacia cada uno de ellos. El promedio de los resultados se resalta en color rojo.

En las tablas 5.4, 5.5, 5.6, y 5.7 se tiene los resultados de los modelos SVM entrenados para las bases de datos imputada, imputada normalizada, imputada normalizada(maxmin) reducida, imputada normalizada(StandardScale) reducida respectivamente . Los resultados con el enfoque de hacia la clase cruzando presenta buenos valores para el F1-score, Recall y Precisión. Pero los resultados hacia el enfoque de NO cruzando no son muy buenos, lo cual está relacionado a las matrices de confusión y se corrobora que la clasificación de no cruzando tiene que ser mejorada para que el sistema en general mejore también.

Analizando los resultados de cada tabla se puede observar que todos tienen valores muy similares en sus métricas, pero se recalca en color verde los valores de la Exactitud, y F1-score en la tabla 5.5, para la métrica Recall en la tabla 5.6 ,y para la métrica Precisión en la tabla 5.7 como los mejores en comparación de las además tablas. En la tabla 5.8 se muestran los promedios de los resultados para la clase cruzando con sus rangos y para cada base de datos utilizada en el entrenamiento de los modelos de SVM.

Tabla 5.4: Resultados K-fold para la base de datos Imputados.

Cruzando					NO Cruzando				
K-fold	Exac.	Pre.	Recall	F1-Score	K-fold	Exac.	Pre.	Recall	F1-Score
1	0.686	0.686	0.882	0.771	1	0.686	0.492	0.220	0.304
2	0.695	0.735	0.885	0.803	2	0.695	0.472	0.244	0.326
3	0.697	0.753	0.866	0.805	3	0.697	0.417	0.253	0.315
4	0.584	0.578	0.970	0.724	4	0.584	0.683	0.084	0.149
5	0.678	0.693	0.928	0.794	5	0.678	0.543	0.173	0.262
Prom.	0.668	0.689	0.906	0.779	Prom.	0.668	0.521	0.195	0.271

Tabla 5.5: Resultados K-fold para la base de datos Imputados Normalizada (maxmin).

Cruzando					NO Cruzando				
K-fold	Exac.	Pre.	Recall	F1-Score	K-fold	Exac.	Pre.	Recall	F1-Score
1	0.664	0.682	0.916	0.782	1	0.664	0.525	0.178	0.266
2	0.711	0.733	0.929	0.819	2	0.712	0.537	0.196	0.288
3	0.765	0.785	0.931	0.852	3	0.765	0.644	0.180	0.437
4	0.612	0.600	0.954	0.737	4	0.616	0.751	0.180	0.290
5	0.706	0.725	0.904	0.804	5	0.706	0.613	0.307	0.409
Prom.	0.691	0.705	0.927	0.799	Prom.	0.693	0.614	0.208	0.338

Tabla 5.6: Resultados K-fold para la base de datos Imputados Normalizada (maxmin) Reducida (S,PP,OC,AD).

Cruzando					NO Cruzando				
K-fold	Exac.	Pre.	Recall	F1-Score	K-fold	Exac.	Pre.	Recall	F1-Score
1	0.647	0.679	0.879	0.766	1	0.647	0.460	0.199	0.277
2	0.714	0.735	0.928	0.820	2	0.714	0.544	0.205	0.298
3	0.750	0.785	0.901	0.839	3	0.750	0.575	0.351	0.436
4	0.616	0.600	0.952	0.737	4	0.616	0.745	0.181	0.291
5	0.705	0.725	0.901	0.804	5	0.705	0.609	0.309	0.410
Prom.	0.686	0.705	0.912	0.807	Prom.	0.686	0.587	0.249	0.324

Tabla 5.7: Resultados K-fold para la base de datos Imputados Normalizada (StandardScale) Reducida (S,PP,OC,AD).

Cruzando					NO Cruzando				
K-fold	Exac.	Pre.	Recall	F1-Score	K-fold	Exac.	Pre.	Recall	F1-Score
1	0.658	0.682	0.899	0.776	1	0.658	0.498	0.193	0.279
2	0.711	0.739	0.910	0.816	2	0.710	0.526	0.237	0.327
3	0.755	0.787	0.908	0.843	3	0.755	0.593	0.352	0.442
4	0.613	0.598	0.955	0.735	4	0.613	0.744	0.170	0.276
5	0.708	0.728	0.899	0.804	5	0.708	0.613	0.322	0.422
Prom.	0.689	0.707	0.914	0.795	Prom.	0.689	0.595	0.255	0.349

En la tabla 5.8 se muestran los resultados promedio para cada tipo de base con su rango de operación. En la tabla 5.9 se hace una comparación de los resultados obtenidos contra los trabajos que utilizaron la misma base de datos publica JAAD. El resultado(promedios de las métricas) utilizado para la comparación fue tomada de la tabla 5.5, ya que fue la que obtuvo dos métricas como mejores resultados. Y como se puede apreciar este trabajo se des-

Tabla 5.8: Métricas promedio con sus rangos de operación.

Base de datos	Exactitud	Presición	Recall	F1-score
Imputada	0.668 ±0.056	0.689 ±0.087	0.906 ±0.044	0.779 ±0.040
Imputada maxmin	0.691 ±0.076	0.705 ±0.092	0.927 ±0.025	0.799 ±0.057
Imputada maxmin reducida	0.686 ±0.067	0.705 ±0.092	0.912 ±0.036	0.807 ±0.051
Imputada StandardScale reducida	0.689 ±0.071	0.707 ±0.094	0.914 ±0.028	0.795 ±0.054

taca en la métrica Recall y F1-score las demás métricas se encuentra entre el estado del arte con los trabajos que utilizan la misma base de datos.

Tabla 5.9: Comparación de trabajo que usan la misma base de da JAAD.

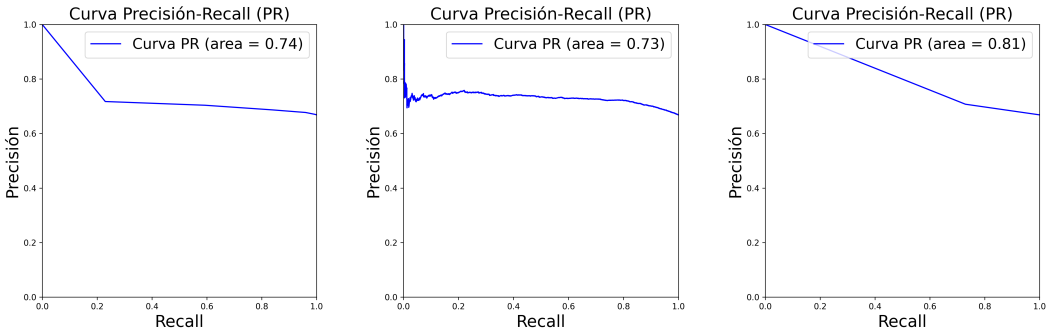
Trabajo	Exactitud	Presición	Recall	F1-score
J.Gesnouin, 2020,[18].	**0.944**	-	-	-
Z.Fang, 2018,[47].	0.88	-	-	-
J.Gesnouin,2021,[41].	0.85	0.56	0.57	0.55
D. Yang, 2022,[44].	0.83	0.51	0.81	0.63
Y. Yao,2021,[56].	0.82	-	-	**0.88**
J.A. Abbasi, 2022,[48].	0.7676	**0.879**	0.7172	0.7899
Lorenzo,2020,[38].	0.6882	0.7420	0.7703	0.7559
Nosotros	0.691 ±0.076	0.705 ±0.092	**0.927** ±0.025	0.799 ±0.057

A continuación, se presenta la tabla 5.10 la cual contiene los resultados de las matrices de confusión de cada modelo entrenado con las diferentes variables de la base de datos. Esta tabla contiene los VP, VN), FP, y FN. Lo que se busca en esta tabla es que los VP, y VN sean los más altos posibles, ya que estos indican que el modelo está clasificando de mejor manera a los peatones que está cruzando(VP), y los que no están cruzando (VN). Por otro lado, se busque que los FP, y los FN sean los más bajos posibles, ya que estos representan una mala clasificación. Es decir que mientras más alto sean los FP esto indica que el modelo está clasificando a peatones que no están cruzando como si estuvieran cruzando. Y la peor situación es que se presente altos valores de FN, ya que el modelo está clasificando a los peatones como no cruzando cuando en realidad el peatón es cruzando, y esto es de alto riesgo. En dicha tabla se resaltan en negrillas los valores más altos para los VP, y VN para cada modelo entrenado con su variante de base de datos. También se resalta para los FP, y FN los valores más pequeños ya que son los valores se requieren. En base a estos resultados se resaltan en verde los modelos que tuvieron altos valores de VP y los más bajos en FN. Esto debido a que se pretende reducir los accidentes a peatones, sin embargo, esto hace seleccionar a estos 4 modelos (SVM) tienda a generar más FP haciendo a los modelos "Preventivos".

Tabla 5.10: Resultados de las matrices de confusión de los modelos KNN, SVM, y RF.

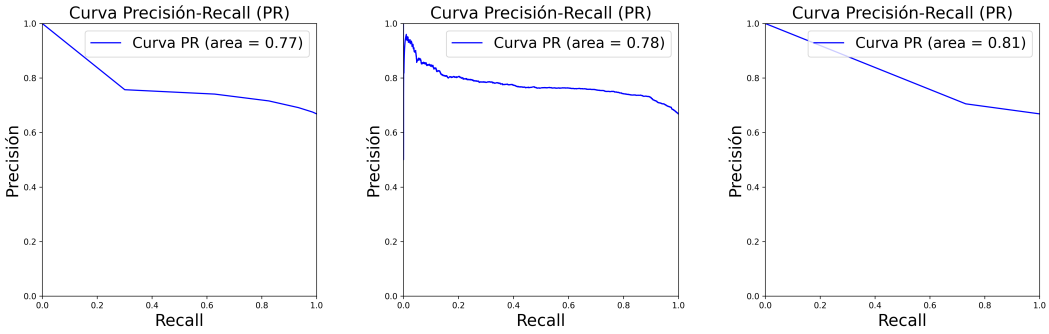
Base datos	Modelo	Verdaderos Positivos	Verdaderos Negativos	Falsos Positivos	Falsos Negativos
Imputada	KNN	3749	510	1710	726
	SVM	**4135**	383	1837	**322**
	RF	3237	**860**	**1360**	1238
Imputada maxmin	KNN	3695	754	1466	780
	SVM	**4045**	682	1538	**430**
	RF	3270	**853**	**1367**	1205
Sintética	KNN	2873	1921	1292	2878
	SVM	2684	**2138**	**1075**	3067
	RF	**3195**	1685	1528	**2556**
Sintética maxmin	KNN	**3089**	1956	1257	2662
	SVM	3805	**2173**	**1040**	**1946**
	RF	3208	1695	1518	2543
Imputada reducida	KNN	1912	1331	889	2563
	SVM	1483	**1563**	**657**	2992
	RF	**2147**	1163	1057	**2328**
Imputada maxmin reducida	KNN	3793	719	1501	682
	SVM	**4036**	685	1535	**439**
	RF	3192	**965**	**1255**	1283
Sintética reducida	KNN	2438	1956	1257	3313
	SVM	1916	**2312**	**901**	3835
	RF	**2749**	1695	1518	**3002**
Sintética maxmin reducida	KNN	3088	1915	1298	2663
	SVM	**3797**	**2187**	**1026**	**1954**
	RF	2990	1793	1420	2761
Imputada maxmin reducida	KNN	3794	710	1510	681
	SVM	**4023**	715	1505	**452**
	RF	3184	**957**	**1263**	1291

A continuación, se presentan las gráficas PR de cada uno de los 27 modelos. Analizando todas las gráficas 5.27, 5.35, 5.29, 5.35, 5.33, 5.35, 5.33, 5.35, y5.35 se puede ver un patrón donde todos los modelos entrenados con cualquier base de datos del tipo KNN y RF tienden a tener una pendiente continua y luego se estabilizan. Pero los modelos SVM tienen un comportamiento diferente, bajan drásticamente pero muy poco, y se mantienen por un tramo más largo. Cabe recalcar que la escala en este caso es engañosa y la gráfica 5.29b se aprecia mejor su resultado. En conclusión, para esta parte los modelos SVM tienen buenos resultados para estas gráficas, pero no son los mejores, pero su gran ventaja de ellos es que tienen un comportamiento más estable, por lo cual es preferible seleccionar los modelos de SVM resaltados en color verde por su estabilidad y los resultados de las matrices de confusión. Por otra parte, los modelos de KNN y RF no son elegibles a pesar de tener mejores resultados para las gráficas de Precisión vs Recall ya que sus matrices de confusión arrogaron malos resultados.



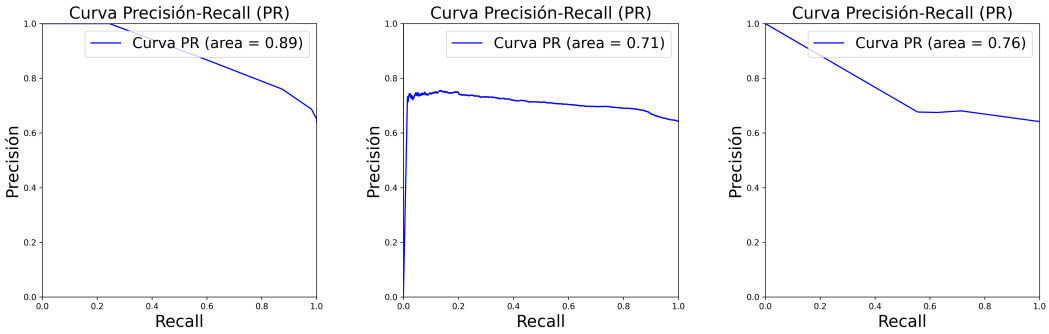
(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

Figura 5.27: Gráficas PR con modelos ML entrenados con la base de datos imputada.



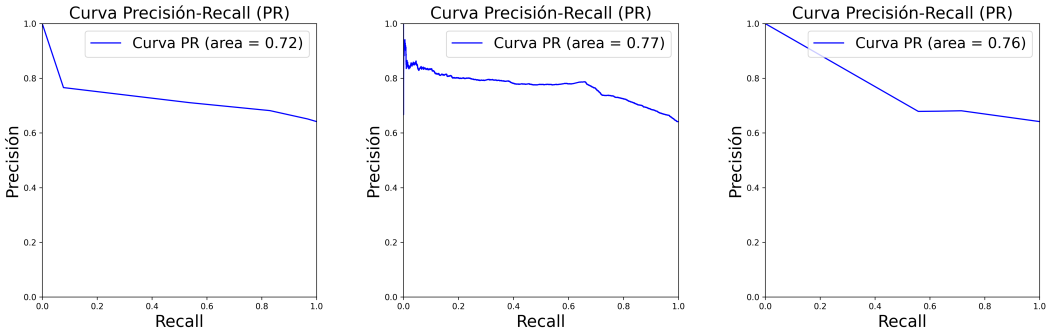
(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

Figura 5.28: Gráficas PR con modelos ML entrenados con la base de datos imputada y normalizada (maxmin).



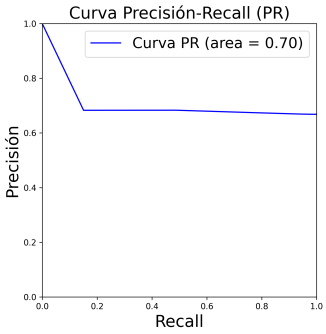
(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

Figura 5.29: Gráficas PR para los tres modelos ML entrenados con la base de datos sintéticos.

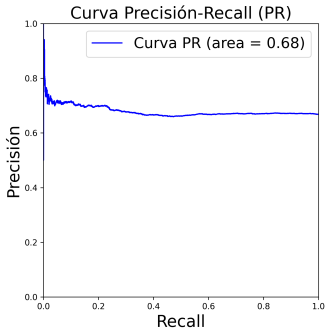


(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

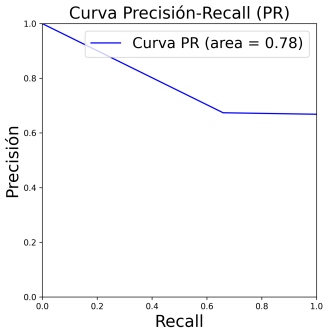
Figura 5.30: Gráficas PR para los tres modelos ML entrenados con la base de datos sintética y normalizada (maxmin).



(a) Modelo KNN(K=5).

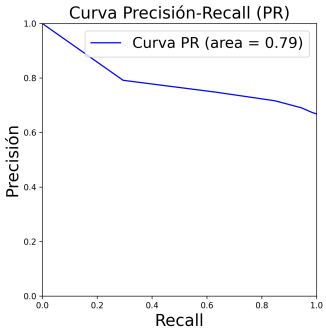


(b) Modelo SVM.

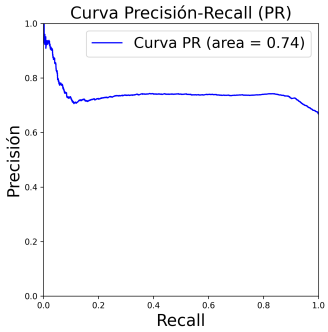


(c) Modelo RF.

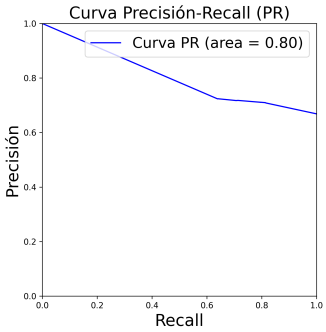
Figura 5.31: Gráficas PR para los tres modelos ML entrenados con la base de datos imputada reducida (ángulos derecho e izquierdo).



(a) Modelo KNN(K=5).

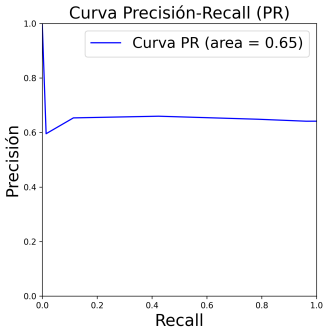


(b) Modelo SVM.

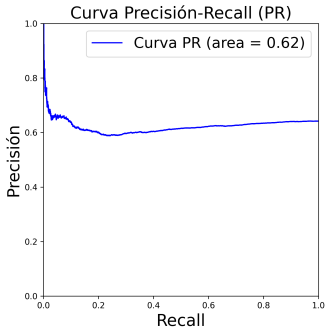


(c) Modelo RF.

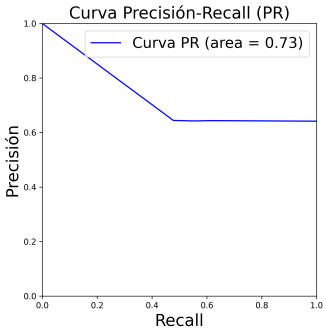
Figura 5.32: Gráficas PR para los tres modelos ML entrenados con la base de datos imputada, normalizada (maxmin), y reducida (Semáforo, P. P., O.C., Ang. Der.).



(a) Modelo KNN(K=5).

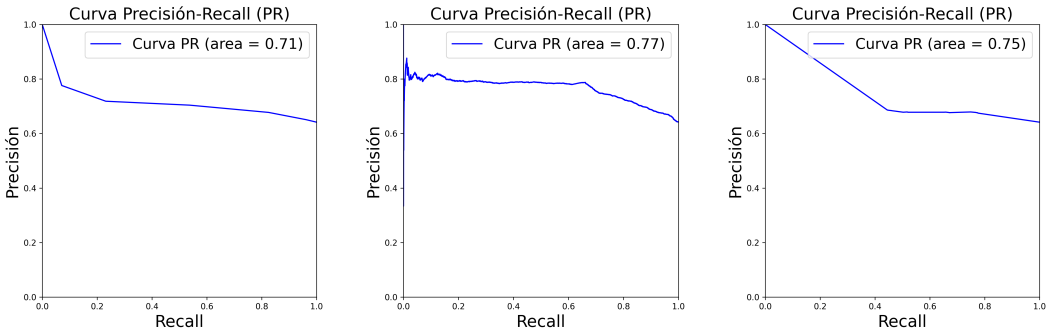


(b) Modelo SVM.



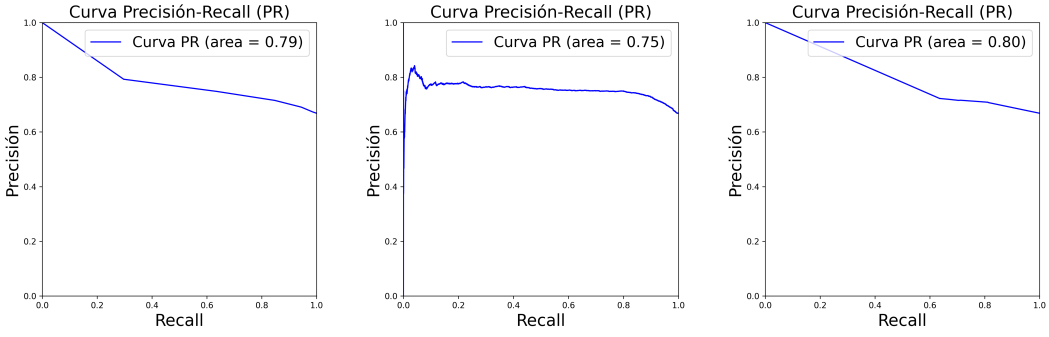
(c) Modelo RF.

Figura 5.33: Gráficas PR para los tres modelos ML entrenados con la base de datos sintético reducido (ángulos derecho e izquierdo).



(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

Figura 5.34: Gráficas PR para los tres modelos ML entrenados con la base de datos sintético, normalizado (maxmin), y reducida (Semáforo, P.P., O.C., Ang. Der.).



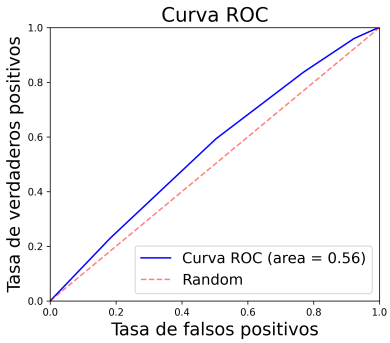
(a) Modelo KNN(K=5). (b) Modelo SVM. (c) Modelo RF.

Figura 5.35: Gráficas PR con modelos ML entrenados con base de datos imputada, normalizada (StandardScale), y reducida (Semáforo, P.P., O.C., Ang. Der.).

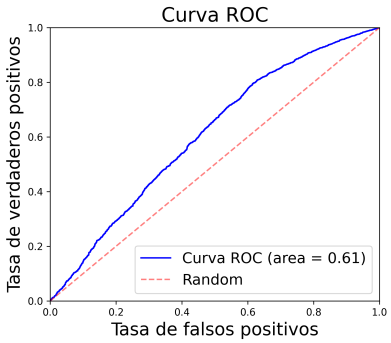
Ahora se presentarán las gráficas ROC. Como se puede apreciar en las figuras 5.36, 5.37, 5.38, 5.39, 5.41, 5.43, y 5.44, el modelo que mejor se desempeñó en cada una de las bases de datos fue **SVM**, lo cual nos esta reafirmado lo obtenido en las matrices de confusión. Sin embargo, para los modelos 5.38b, 5.39b, y 5.43b no fueron seleccionados ya que sus matrices de confusión resultaron con una gran cantidad de falsos negativos. Por otro lado, se puede observar un patrón en las figuras 5.40, 5.42 donde la característica en común es que ambas usan los atributos ángulo derecho e izquierdo, para la base de datos imputada y sintética, y se puede ver que cualquiera de modelos no puede diferenciar de la clase cruzando y no cruzando. Por lo cual nos dice que solo estos dos atributos no bastan para reconocer la intención de cruzar por parte del peatón.

Una vez habiendo analizado las matrices de confusión, las gráficas de PR, y las gráficas ROC se seleccionaron 4 modelos de las 9 iniciales. A continuación, se presenta un listado de la base de datos de los cuales fueron entrenados los modelos de SVM seleccionados. Estos coinciden con los resultados de K-fold aplicado en la tabla 5.8.

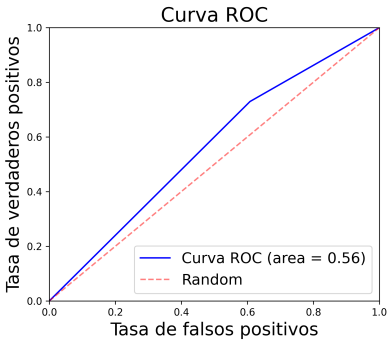
- Base de dato imputada.
- Base de dato imputada normalizada(maxmin).
- Base de dato imputada normalizada(maxmin) y reducida con los atributos Semáforo, Paso peatonal, Orientación de la cabeza, y Ángulo derecho.
- Base de dato imputada normalizada(StandarScale) y reducida con los atributos Semáforo, Paso peatonal, Orientación de la cabeza, y Ángulo Izquierdo.



(a) Modelo KNN(K=5).

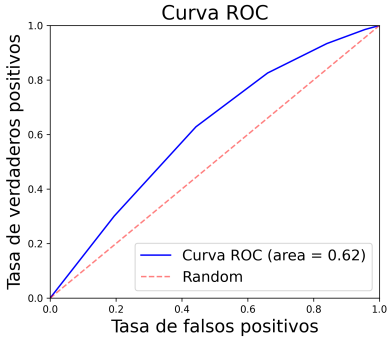


(b) Modelo SVM.

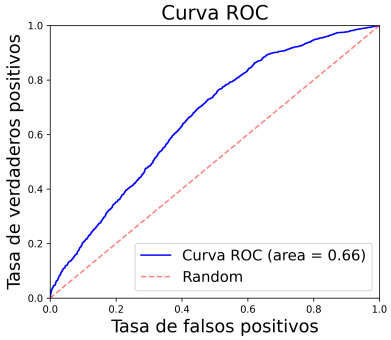


(c) Modelo RF.

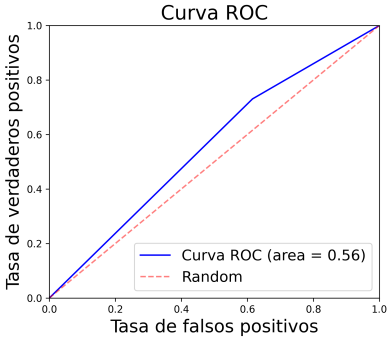
Figura 5.36: Gráficas ROC para los modelos ML entrenados con la base de datos imputada.



(a) Modelo KNN(K=5).

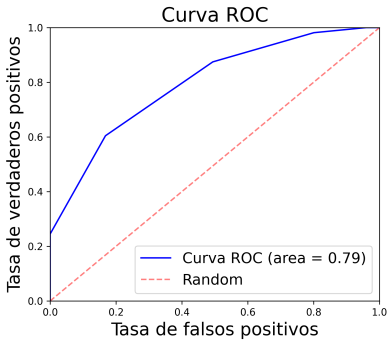


(b) Modelo SVM.

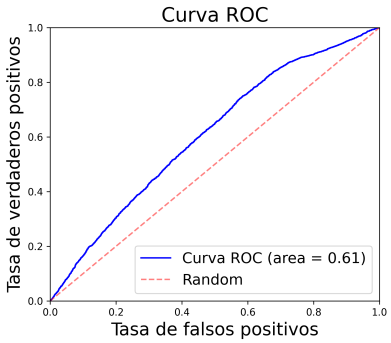


(c) Modelo RF.

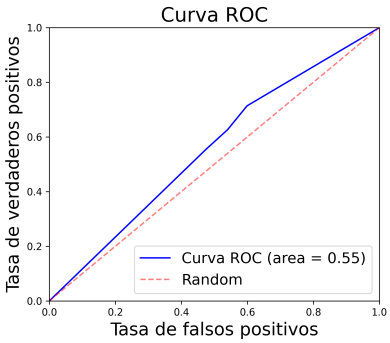
Figura 5.37: Gráficas ROC para los modelos ML entrenados con la base de datos imputada y normalizada(maxmin).



(a) Modelo KNN(K=5).

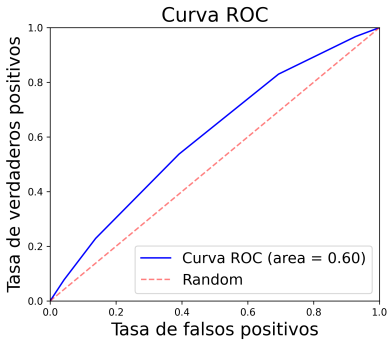


(b) Modelo SVM.

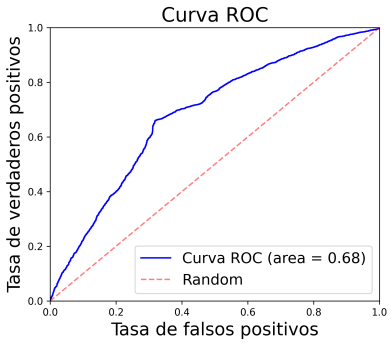


(c) Modelo RF.

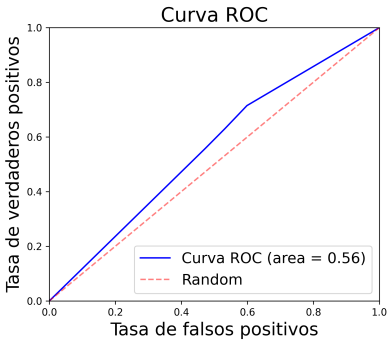
Figura 5.38: Gráficas ROC para los modelos ML entrenados con la base de datos sintéticos.



(a) Modelo KNN(K=5).

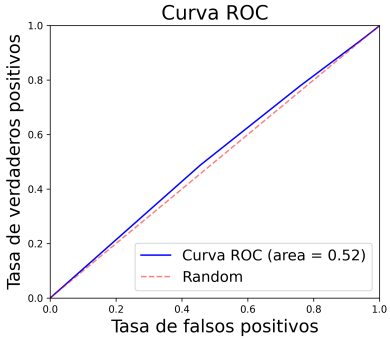


(b) Modelo SVM.

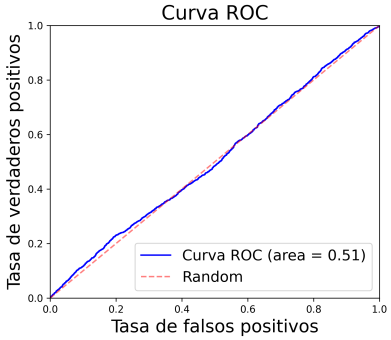


(c) Modelo RF.

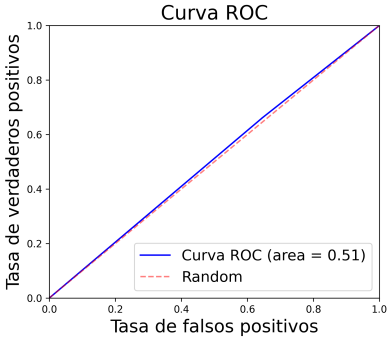
Figura 5.39: Gráficas ROC para los modelos ML entrenados con la base de datos sintética y normalizada(maxmin).



(a) Modelo KNN(K=5).

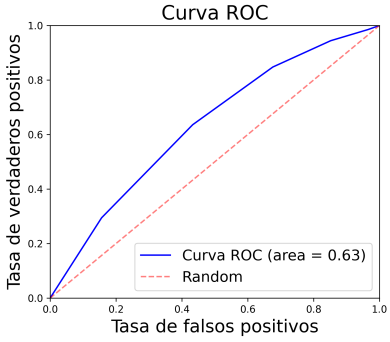


(b) Modelo SVM.

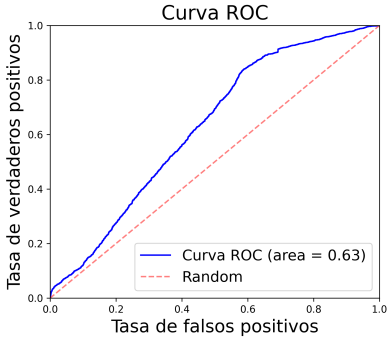


(c) Modelo RF.

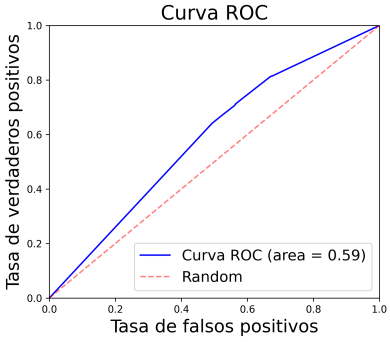
Figura 5.40: Gráficas ROC para los modelos ML entrenados con la base de datos imputada reducida (ángulos derecho e izquierdo).



(a) Modelo KNN(K=5).

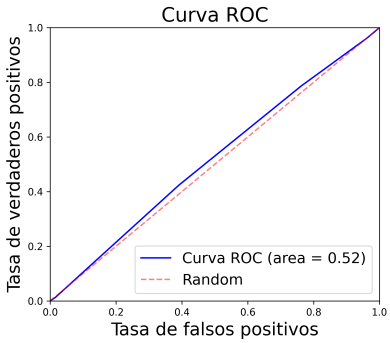


(b) Modelo SVM.

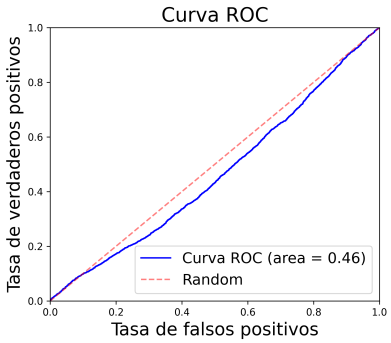


(c) Modelo RF.

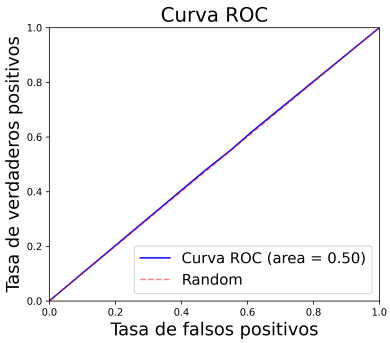
Figura 5.41: Gráficas ROC para los modelos ML entrenados con la base de datos imputada, normalizada(maxmin), y reducida (Semáforo, P.P., O.C., Ang. Der.).



(a) Modelo KNN(K=5).

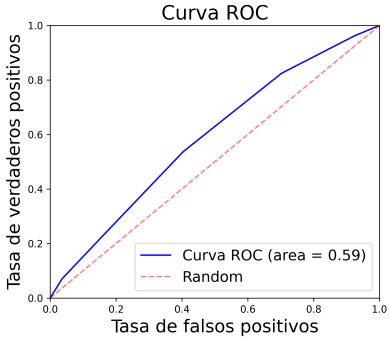


(b) Modelo SVM.

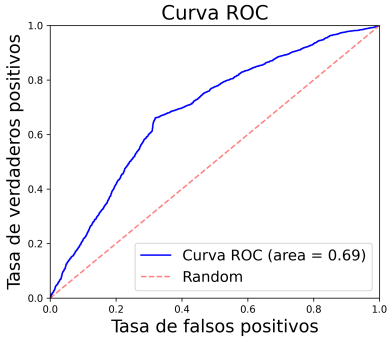


(c) Modelo RF.

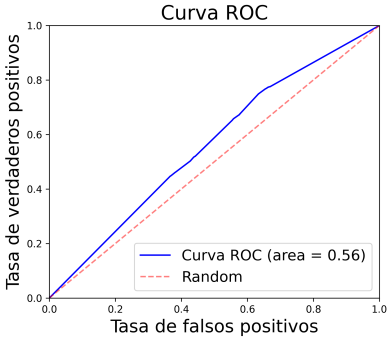
Figura 5.42: Gráficas ROC para los modelos ML entrenados con la base de datos sintéticos reducidos (ángulos derecho e izquierdo).



(a) Modelo KNN(K=5).

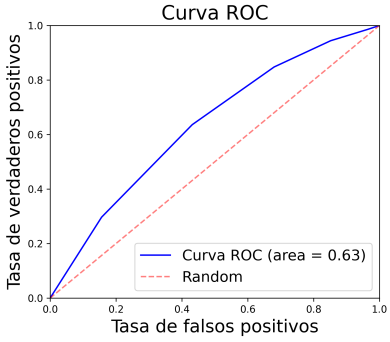


(b) Modelo SVM.

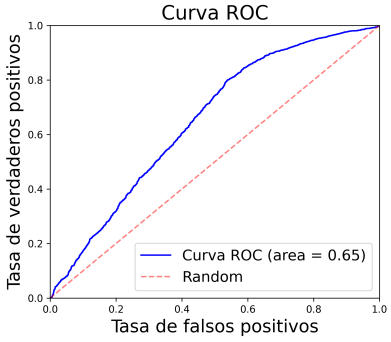


(c) Modelo RF.

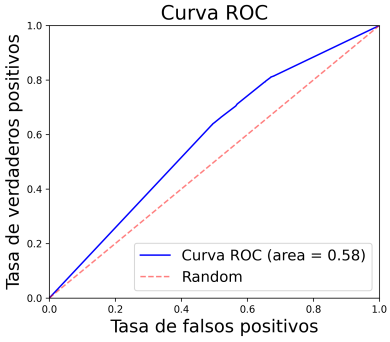
Figura 5.43: Gráficas ROC para los modelos ML entrenados con la base de datos sintética, normalizada(maxmin), y reducida (Semáforo, P.P., O.C., Ang. Der.).



(a) Modelo KNN(K=5).



(b) Modelo SVM.



(c) Modelo RF.

Figura 5.44: Gráficas ROC para los modelos ML entrenados con la base de datos imputada, StandardScale, y reducida (Semáforo, P.P., O.C., Ang. Der.).

5.4. Sistema final

El sistema final como ya se ha mencionado, es la unión de la arquitectura del sistema que generó la base de datos para entrenar a los modelos de ML y los modelos entrenados. Dado que los resultados en el entrenamiento y prueba de los modelos de ML fueron muy similares, se probó el sistema final con estos 4 modelos entrenados sus respectivas bases de datos. Para probar el sistema final se utilizaron 70 vídeos(277-346), los cuales nunca fueron usados anteriormente. Y solo se calcularon las métricas de precisión, exactitud, recall y f1-score.

Los resultados obtenidos se encuentran en la parte de anexos A.1, A.2,A.3, A.4, A.5, A.6, A.7, y A.8. Después de analizar los resultados se encontraron con varias situaciones. Para los resultados del anexo A.1 se presenta valores de cero en las métricas y una vez analizado cada uno de los vídeos, se encontró que en los vídeos 285, 292, 296, 323, 343 no se presenta ningún peatón. Por otra parte, para el vídeo 346 si se presenta un peatón cruzando, pero las condiciones de contra luz, el sistema no fue capaz de detectar al peatón o de obtener los ángulos.

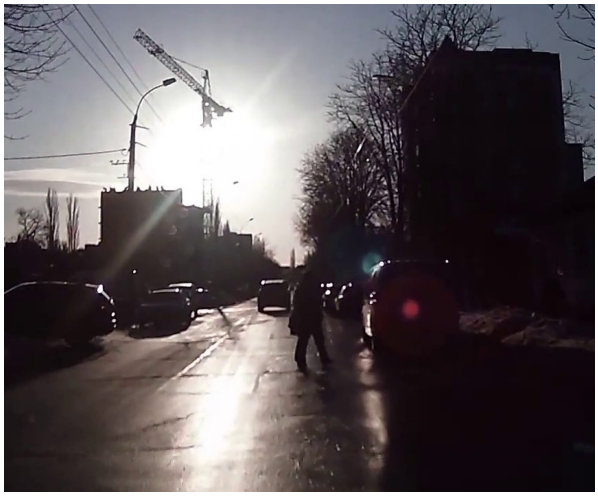


Figura 5.45: Ejemplo de situación compleja de detección a contraluz del sol.

En los anexos A.2 y A.3 se muestran los resultados para los vídeos 284, 288, 289, 300, 304, 308, 309, 318, 329, 335,337, 342, y 344. Claramente las métricas son malas, esto es debido a varias razones. Como se ha visto en el entrenamiento de los modelos de ML, el

sistema tiende a predecir a los peatones como cruzando, y esto explica por qué los resultados en estos vídeos son malos, ya que los peatones detectados en cada vídeo no están cruzando respecto a la base de datos y el sistema dice que si lo están. También esto pudiera ser a que el sistema está detectando al peatón de espalda o de frente (ejemplo en la figura 5.46), lo cual el cálculo bidimensional no es lo mejor para este tipo de tomas.

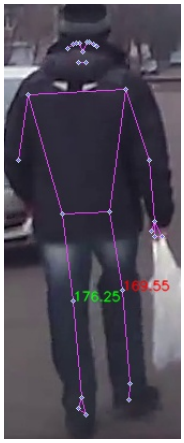


Figura 5.46: Ejemplo de situación donde el peatón se detectó de espaldas en medio de una calle en el vídeo 218.

En los resultados para los anexos A.4, A.5, A.6, A.7, y A.8 se puede apreciar que los valores de las métricas son relativamente buenos y que mantienen el comportamiento presentado por lo modelos SVM. Los parámetros establecidos de confiabilidad para el sistema final fueron para la detección del peatón de 0.90, para las señales de tráfico (semáforos y señales de alto) de 0.60, para los pasos peatonales de 0.60, para la orientación de la cabeza de 0.30, y para la esqueletización de 0.50. Para poder comparar el desempeño de los modelos SVM entrenados con las bases de datos imputada, imputada normalizada(maxmin), imputada normalizada(maxmin) reducida e imputada normalizada(StandardScale) reducida, se calculó el promedio de cada métrica para cada vídeo de su respectivo modelo entrado con un de las cuatro bases de datos ya mencionadas. En las figuras 5.47, 5.48, 5.49, y 5.50 se puede apreciar los resultados del promedio de las métricas de todos los vídeos. Analizando estas gráficas de barras, se puede apreciar claramente que el modelo SVM entrenado con la base de datos imputada presenta menor desempeño para las métricas de exactitud, precisión

y F1-Score, lo que nos dice que esta base de datos imputada que contenía todos los atributos deseados inicialmente no es la mejor. Por otro lado, los modelos de SVM entrenados por la base de datos imputado normalizado(maxmin), imputado normalizado(maxmin) reducido (4 atributos de 6), e imputado normalizado(StandardScale) reducido (4 atributos de 6) presenta valores muy semejantes en todas las métricas. Por lo cual, la selección del tipo de base de datos y su procesamiento para entrenar modelos para el tipo SVM puede ser cualquiera de las dos siguientes, la base de datos imputado normalizado(maxmin) reducido (4 atributos de 6), e imputado normalizado(StandardScale) reducido (4 atributos de 6), esto porque a diferencia de la base de datos imputada reducida (maxmin) tiene o usan dos características menos. Para seleccionar uno de los dos bases de datos "Ganadoras", la base de datos imputado normalizado(maxmin) reducido (4 atributos de 6), e imputado normalizado(StandardScale) reducido (4 atributos de 6), realmente no hay diferencia en utilizar una a la otra, pero si deja en evidencia que normalizar la base de datos fue relevante para este trabajo. El tiempo requerido para las detecciones y clasificación fue en promedio de 0.86 y 0.84 segundos para la base de datos imputado normalizado(maxmin) reducido (4 atributos de 6), e imputado normalizado(StandardScale) reducido (4 atributos de 6) respectivamente. Se utilizo una GPU tesla T4 de colab.

Como manera de cerrar el sistema final, se muestra la tabla 5.11 en la cual se comparan varios trabajos que utilizaron la misma base de datos JAAD contra los resultado de los modelos de SVM entrenados con 4 diferentes variaciones. En esta tabla se comparan las métricas de Exactitud(Exac.), Precisión(Pre.), Recall, y F1-Score, los tiempos de inferencia, la cantidad y el tipo de atributo utilizado y el tipo de hardware utilizado. Vemos que los resultados están dentro del estado del arte, y que es superior en la métricas recall y que además al tener métricas que compiten al mismo nivel pero utilizando menor cantidad de atributos o características para reconocer o predecir la intención del peatón.

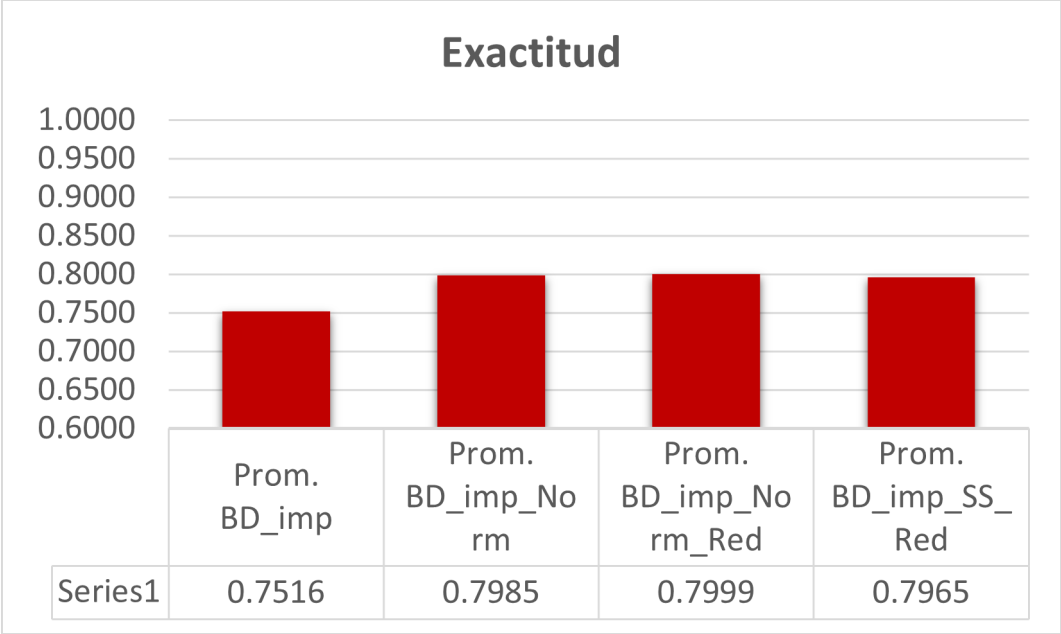


Figura 5.47: Gráfica de barras de la exactitud promedio de los vídeos.

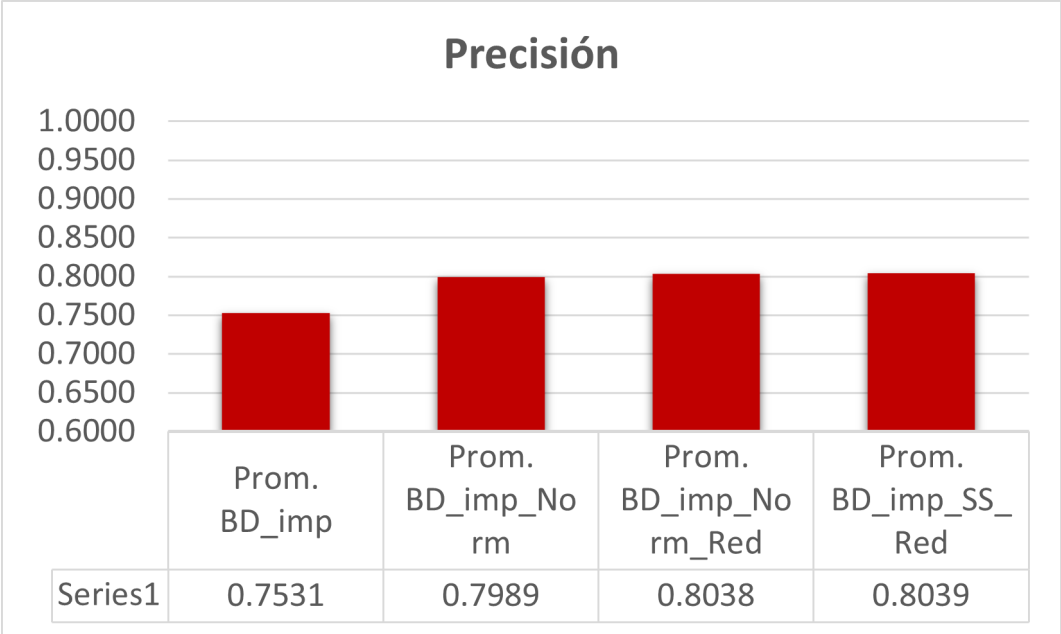


Figura 5.48: Gráfica de barras de la precisión promedio de los vídeos.

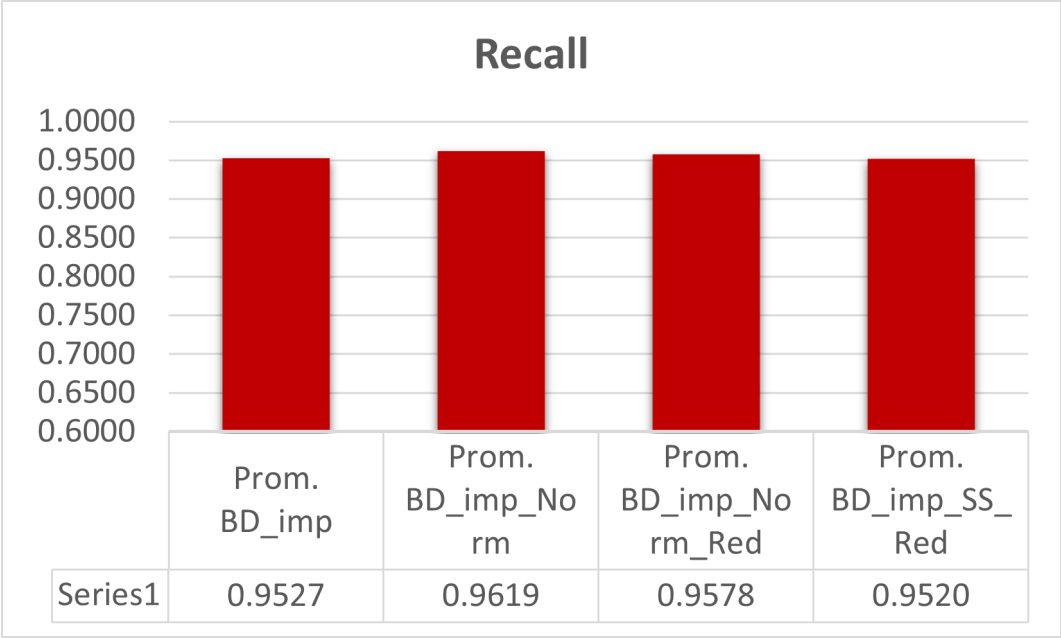


Figura 5.49: Gráfica de barras de la recall promedio de los vídeos.

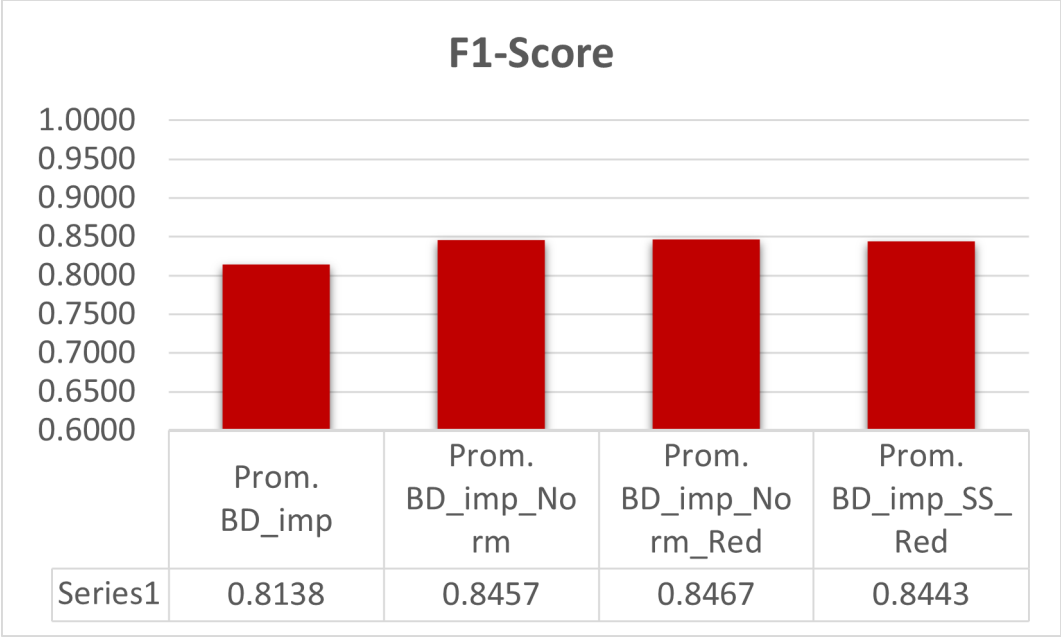


Figura 5.50: Gráfica de barras de la F1-Score promedio de los vídeos.

Tabla 5.11: Comparación de varios trabajos en base a la misma base de datos JAAD

Trabajo	N° Atri.	Atributos	Tiempos de inferencia (s)	Hardware	Exac.	Pre.	Recall	F1-Score
J.Gesnouin,2020[18]	1	PC	0.00025/67	RTX 2080ti/i7 8700K	**0.944**	-	-	-
Z.Fang,2018,[47]	1	PC	-	-	0.88	-	-	-
J.Gesnouin,2021[41]	1	PC	1-2	-	0.85	0.56	0.57	0.55
D. Yang, 2022,[44]	7	Bb,IV, SP, PC,SA,ST,S	1-2	-	0.83	0.51	0.81	0.63
J.A.Abbasi,2022[48]	7	Bb,IV,SP, PC,SA,ST,S	-	RTX™ A4000	0.7676	**0.879**	0.7172	0.7899
Lorenzo,2020,[38]	6	ST,SA,OC, Coo,EM,Bb	-	GTX TITAN X	0.6882	0.7420	0.7703	0.756
Razali 2021,[45]	1	PC	0.2	GTX 1080 Ti	-	-	-	-
Yao 2021,[56]	6	S, PP, SA, T, E, IV	0.006 ±0.0012	Tesla V100	0.82	-	-	**0.88**
Base imputada	6	S, PP, SA, OC, AD,AI	0.8494	Tesla T4 colab	0.7516	0.7531	**0.9527**	0.8138
Base imputada maxmin	6	S, PP ,SA, OC, AD,AI	0.8396	Tesla T4 colab	0.7985	0.7939	**0.9619**	0.8457
Base imputada maxmin reducida	4	S,PP,OC,AD	0.8591	Tesla T4 colab	0.7999	0.8038	**0.9578**	0.8467
ł Base imputada Estándar Scaled reducida	4	S,PP,OC,AI	0.8403	Tesla T4 colab	0.7965	0.8039	**0.9520**	0.8443

En la figura 5.51 se muestra una secuencia de 7 imágenes. En las imagen 0 y 24 solo se observan la detección de pasos peatonales. En la imagen 49 se muestran las detecciones de los pasos peatonales, la ordinación de la cabeza, el cual es incorrecta puesto que pertenece a otro peatón, y el peatón en color rojo y los ángulos de color blanco, dando así la predicción de 1 o cruzando cuando su estado real es no cruzando(FP). En la imagen 74 se muestran las mismas detecciones que en la imagen anterior pero con la orientación de la cabeza correspondiente al peatón, y ahora la predicción es no cruzando (0), mientras que el estado real es No Cruzando un VN. Para la imagen 99 se muestran las misma detecciones para la imagen 49 pero ahora la predicción y el estado real del peatón es 1(C) y C respectivamente y este comportamiento siguen igual para las imágenes 124 y 149. La velocidad de procesamiento por imagen es de 0.84 segundos, debido a que es escogió el sistema entrenado con la Base imputada Estándar Scaled reducida en la tabla 5.11, y es por eso que de el vídeo 305 que consta de 150 imágenes solo se muestran 7.

Figura 5.51: Secuencian de imágenes del sistema final para el vídeo 305 de la base de datos JAAD.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

En base a los resultados obtenidos y analizados en esta tesis se puede concluir los siguientes puntos:

- Debido a las necesidades de este proyecto se requirió crear una base de datos con los 6 atributos deseados. En base a la investigación, se puede decir que esta base de datos es una aportación ya que no se encontró alguna mención o existencia de la base de datos utilizada para entrenar a los modelos de machine learning. Lo cual es muy valioso.

- De acuerdo con los resultados, la preparación de los datos, análisis de los atributos fueron una estratégica altamente recomendada ya que las base a las cuales se les redujo su dimensionalidad (cantidad de atributos, 6 a 4) y la normalización de sus datos con diferentes maneras fueron las que dieron mejores resultados en los modelos de SVM.

- En base a la experimentación, entrenamiento, y resultados del modelo de KNN, SVM, y RF, se seleccionó los modelos de SVM entrenado con diferentes variaciones de la base de datos original. La razón por la cual el modelo RF no funciono se puede deber a que no es bueno cuando el sistema tiene una gran cantidad de instancias y hace la profundidad más grande. Para el modelo KNN no fue el mejor posiblemente por que los vecinos más cercanos no fueron lo suficiente o que los datos estos mezclados. Dado a los dos modelos anteriores el modelo SVM fue mejor muy posiblemente por que los datos que estuvieran mezclados fueron clasificados correctamente por el aumento de dimensionalidad que este modelo ofrece.

- En base a la investigación realizada se encontró que varios trabajos utilizaban varias

y diferentes características ya sea del peatón y mayormente del vehículo, como lo son la orientación de la cabeza, orientación del torso entre otras, y para el vehículo como la velocidad, frenado entro otras más. Pero en este trabajo se centró en lo que el auto "puede ver"mediante una cámara RGB y obtener variables del ambiente y del mismo peatón. Y de las 6 características que se utilizaron (semáforos, alto vehicular, pasos peatonales, orientación de la cabeza, ángulo de la rodilla derecha e izquierda) se puedo determinar que las características mínimas necesarias para reconocer la intención del peatón son la orientación de la cabeza (si el peatón está viendo o no al vehículo), la presencia de pasos peatonales, semáforos y de solo un ángulo de cualquier rodilla, en total 4 de las 6 características iniciales.

- En general para el sistema final realiza todo el proceso de detección y clasificación se le puede considerar bueno ya que detecta muy bien a los peatones que están cruzando. Y que este sistema tiende a ser preventivo lo cual es relativamente bueno, ya que clasifica a un peatón que está cruzando cuando en realidad no lo está. Por lo cual, si solo no limitamos a la tarea de reconocer la intención del peatón para así poder evitar un accidente, este proyecto ha cumplido con su objetivo, y no solo eso, sino demuestra cuales características son realmente necesarias del total inicial.

- Para la parte del sistema es preventivo, esto pudo deberse a al desbalance de clases, debido a que la base de datos se encontraba con mayor cantidad de peatones cruzando. Por lo cual otro método diferente al ADASYN pudiera ser otra opción para mejorar el sistema general en gran medida. También esté comportamiento pudiera ser por la base de datos JAAD, es decir que el modelo lo esté prediciendo bien y que la etiqueta de ese peatón allá sido puesta incorrectamente por una confusión de la definición de cruzar para un peatón. Por lo cual la definición de cuando un peatón está cruzando es fundamental tenerla claramente.

6.2. Trabajo futuro

Para poder mejorar un mas este proyecto se en listan las tareas que se puede realizar para mejorar el sistema en general.

- Buscar modelos pre-entrenados o entrenar modelos para las características deseadas y mencionadas en este proyecto. Por ejemplo, se entrenó el modelo YOLOV8 que fue liberado en enero del 2023 y se podría intentar con el modelo siguiente *YOLO NAS* el cual salió dos o tres meses después.

- Mejorar la detección del peatón para que el sistema sea capaz de realizar esta detección con un valor de con confiabilidad menor a 0.9 para si poder cubrir y analizar más imágenes donde si exista un peatón.

- Probar otros métodos de generación de datos sintéticos para balancear las clases y probarlo con el modelo SVM con la intención de disminuir los falsos positivos y que el modelo sea menos preventivo al clasificar correctamente a los peatones que realmente no están cruzando.

- Probar con más parámetros al entrenar al modelo SVM y esperar su desempeño mejore aún más, sobre todo en las gráficas PR y la gráfica ROC. Esto con la intención de que el modelo sea más capaz de diferenciar mejor las clases cruzando y no cruzando.

- Probar el sistema final, con un poder de cómputo más grande, para así mejorar el tiempo de detección, ya que en este trabajo se utilizó una GPU Tesla T4.

- Una vez que se allá cumplido con lo anterior se podría intercambiar o agregar otra variable para deseadas que involucren al peatón y su entorno para investigar que otras más características pudieran ser igual o más importantes de las ya encontradas en este trabajo.

- Una mejorar sería llevar la esqueletización de un plano 2D a uno en 3D para lograr ser más precisos mediante el uso de vectores pero sin afectar la velocidad de clasificación.

BIBLIOGRAFÍA

[1] Fanta Camara, Nicola Bellotto, Serhan Cosar, Dimitris Nathanael, Matthias Althoff, Jingyuan Wu, Johannes Ruenz, Andre Dietrich, and Charles Fox. Pedestrian Models for Autonomous Driving Part I: Low-Level Models, From Sensing to Tracking. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–21, 2020.

[2] NORMAN ROGELIO MORALES VEGA. *METODOLOGÍA Y DESARROLLO DE UN SISTEMA PARA ANALIZAR EL CICLO DE MARCHA HUMANA*. PhD thesis, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2011.

[3] viso.ia. A guide to openpose in 2022, 2018. `https://viso.ai/deep-learning/openpose/`, Last accessed on 2022-10-07.

[4] Jun Ma and Wenhui Rong. Pedestrian Crossing Intention Prediction Method Based on Multi-Feature Fusion. *World Electric Vehicle Journal*, 13(8):158, 2022.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[6] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.

[7] codecademy. Normalización, 2022. `https://www.codecademy.com/article/normalization`, Ultimo acceso on 05/09/2022.

[8] Praveen Nellihelae. What is k-fold cross validation?, 2022. `https://towardsdatascience.com/`

what-is-k-fold-cross-validation-5a7bb241d82f, Ultimo acceso on 11/12/2022.

[9] Loïc Simon, Ryan Webster, and Julien Rabin. Revisiting precision and recall definition for generative model evaluation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:10174–10183, 2019.

[10] World Health Organisation. Road traffic injuries, 2018. https://www.who.int/health-topics/road-safety#tab=tab_1, Last accessed on 2022-07-03.

[11] Louis Kratz, Student Member, and Ko Nishino. Spatio-Temporal Motion Patterns in Extremely Crowded Scenes. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 34(5):987–1002, 2012.

[12] Jie Yang, Jiarou Fan, Yiru Wang, Yige Wang, Weihao Gan, Lin Liu, and Wei Wu. Hierarchical Feature Embedding for Attribute Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 13052–13061, 2020.

[13] Biao Yang and Rongrong Ni. Vision-based recognition of pedestrian crossing intention in an urban environment. *9th IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, CYBER 2019*, pages 992–995, 2019.

[14] Mudassar Raza, Zonghai Chen, Saeed Ur Rehman, Peng Wang, and Peng Bao. Appearance based pedestrians' head pose and body orientation estimation using deep learning. *Neurocomputing*, 272:647–659, 2018.

[15] Walter Morales-Álvarez, Mara José Gómez-Silva, Gerardo Fernández-López, Fernando Garca-Fernández, and Cristina Olaverri-Monreal. Automatic Analysis of Pedestrian's Body Language in the Interaction with Autonomous Vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June(Iv):1–6, 2018.

[16] Yuanze Wang, Honglong Chen, and Xianghui Cao. Design of a prediction based pedestrian tracking system by UAV. *Proceedings - 2020 35th Youth Academic Annual Conference of Chinese Association of Automation, YAC 2020*, pages 831–836, 2020.

[17] Florin Leon and Marius Gavrilescu. A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics*, 9(6):na, 2021.

[18] Joseph Gesnouin, Steve Pechberti, Guillaume Bresson, Bogdan Stanciulescu, and Fabien Moutarde. Predicting intentions of pedestrians from 2d skeletal pose sequences with a representation-focused multi-branch deep learning network. *Algorithms*, 13(12):1–23, 2020.

[19] V. Onkhar, P. Bazilinskyy, J. C.J. Stapel, D. Dodou, D. Gavrila, and J. C.F. de Winter. Towards the detection of driver–pedestrian eye contact. *Pervasive and Mobile Computing*, 76:101455, 2021.

[20] Steven Seida, David G. Morgenthaler, Mark Podlaseck, Bob Douglas, Jon McSwain, Robert Knourek, and Mark Thomas. Vision-Based Road Following in the Autonomous Land Vehicle. pages 1814–1819, 1987.

[21] Wilfried Enkelmann. Obstacle Detection by Evaluation of Optical Flow Fields from Image Sequences. Number Rehfeld 88, pages 2–6, 1990.

[22] F. Thomanek, E. D. Dickmanns, and D. Dickmanns. Multiple object recognition and scene interpretation for autonomous road vehicle guidance. pages 231–236, 1994.

[23] Fanta Camara, Nicola Bellotto, Serhan Cosar, Florian Weber, Dimitris Nathanael, Matthias Althoff, Jingyuan Wu, Johannes Ruenz, Andre Dietrich, Gustav Markkula, Anna Schieben, Fabio Tango, Natasha Merat, and Charles Fox. Pedestrian Models for Autonomous Driving Part II: High-Level Models of Human Behavior. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5453–5472, 2021.

[24] S. Suriya, Rajesh Harinarayanan Rajasekar, and S. Mercy Shalinie. Understanding deep learning algorithms for object detection and recognition. *Proceedings of the 11th International Conference on Advanced Computing, ICoAC 2019*, pages 79–85, 2019.

[25] J. B. Morrison. The mechanics of the knee joint in relation to normal walking. *Journal of Biomechanics*, 3(1):51–61, 1970.

[26] Technische Universiteit Eindhoven. Pedestrian Intention and State Estimation : Analysis , design and implementation. 2019.

[27] Shile Zhang, Mohamed Abdel-Aty, Yina Wu, and Ou Zheng. Pedestrian Crossing Intention Prediction at Red-Light Using Pose Estimation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–9, 2021.

[28] Raul Quintero Minguez, Ignacio Parra Alonso, David Fernandez-Llorca, and Miguel Angel Sotelo. Pedestrian Path, Pose, and Intention Prediction Through Gaussian Process Dynamical Models and Pedestrian Activity Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1803–1814, 2019.

[29] Amir Rasouli, Iuliia Kotseruba, and John K. Tsotsos. Understanding pedestrian behavior in complex traffic scenes. *IEEE Transactions on Intelligent Vehicles*, 3(1):61–70, 2018.

[30] S. Schmidt and B. Färber. Pedestrians at the kerb – recognising the action intentions of humans. *Transportation Research Part F: Traffic Psychology and Behaviour*, 12(4):300–310, 2009.

[31] Amir Rasouli and Iuliia Kotseruba. Intend-Wait-Cross: Towards Modeling Realistic Pedestrian Crossing Behavior. 2022.

[32] Raiful Hasan and Ragib Hasan. Pedestrian safety using the Internet of Things and sensors: Issues, challenges, and open problems. Technical Report May, 2022.

[33] Tiziana Campisi, Irena Ištoka Otković, Sanja Šurdonja, and Aleksandra Deluka-Tibljaš. Impact Of Social and Technological Distraction on Pedestrian Crossing Behaviour: A Case Study in Enna, Sicily. *Transportation Research Procedia*, 60(2021):100–107, 2022.

[34] Xiaoyuan Zhao, Xiaomeng Li, Andry Rakotonirainy, Samira Bourgeois- Bougrine, and Patricia Delhomme. Predicting pedestrians' intention to cross the road in front of automated vehicles in risky situations. *Transportation Research Part F: Traffic Psychology and Behaviour*, (xxxx), 2022.

[35] Chen Tina Chen. *PEDESTRIAN-ENVIRONMENT INTERACTIONS FOR PREDICTING PEDESTRIAN CROSSING INTENTION FROM THE EGO-VIEW by*. PhD thesis, 2021.

[36] H. Joe Steinhauer Omar Hamed. Pedestrian's Intention Recognition, Fusion of Handcrafted Features in a Deep Learning Approach. *Proceedings - 14th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2018*, 35:15795–15796, 2021.

[37] Tiago Roxo and Hugo Proença. Faces in the Wild: Efficient Gender Recognition in Surveillance Conditions. 2021.

[38] J. Lorenzo, I. Parra, F. Wirth, C. Stiller, D. F. Llorca, and M. A. Sotelo. RNN-based Pedestrian Crossing Prediction using Activity and Pose-related Features. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1801–1806, 2020.

[39] Leticia Gonz, M L Antonio, C Á Juan, and Á Diego. Real-Time Short-Term Pedestrian Trajectory Prediction Based on Gait Biomechanics. *Sensors*, (1), 2022.

[40] Miao Kang, Jingwen Fu, Sanping Zhou, Songyi Zhang, and Nanning Zheng. Learning to predict diverse trajectory from human motion patterns. *Neurocomputing*, 504:123–131, 2022.

[41] Joseph Gesnouin, Steve Pechberti, Bogdan Stanciulcscu, and Fabien Moutarde. TrouSPI-Net: Spatio-temporal attention on parallel atrous convolutions and U-GRUs for skeletal pedestrian crossing prediction. *Proceedings - 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2021*, 2021.

[42] Dimitrios Varytimidis, Fernando Alonso-Fernandez, Boris Duran, and Cristofer Englund. Action and Intention Recognition of Pedestrians in Urban Traffic. *Proceedings - 14th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2018*, pages 676–682, 2018.

[43] Santiago Gerling Konrad, Mao Shan, Favio R. Masson, Stewart Worrall, and Eduardo Nebot. Pedestrian Dynamic and Kinematic Information Obtained from Vision Sensors. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June(June):1299–1305, 2018.

[44] Dongfang Yang, Haolin Zhang, Ekim Yurtsever, Keith Redmill, and Umit Ozguner. Predicting Pedestrian Crossing Intention with Feature Fusion and Spatio-Temporal Attention. *IEEE Transactions on Intelligent Vehicles*, 14(8):1–9, 2021.

[45] Haziq Razali, Taylor Mordan, and Alexandre Alahi. Pedestrian intention prediction: A convolutional bottom-up multi-task approach. *Transportation Research Part C: Emerging Technologies*, 130(June):103259, 2021.

[46] Izaak Van, Seppe Sels, Bart Ribbens, Gunther Steenackers, and Rudi Penne. Accuracy Assessment of Joint Angles Estimated from 2D and 3D Camera Measurements. *Sensors*, 2022.

[47] Zhijie Fang and Antonio M. López. Is the Pedestrian going to Cross? Answering by 2D Pose Estimation. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June:1271–1276, 2018.

[48] Jibran Ali Abbasi, Navid Mohammad Imran, and Myounggyu Won. WatchPed: Pedestrian Crossing Intention Prediction Using Embedded Sensors of Smartwatch. 2022.

[49] Chen Ning, Li Menglu, Yuan Hao, Su Xueping, and Li Yunhong. Survey of pedestrian detection with occlusion. *Complex Intelligent Systems*, 7(1):577–587, 2021.

[50] Suresh Kumaar Jayaraman, Lionel P. Robert, X. Jessie Yang, and Dawn M. Tilbury. Multimodal Hybrid Pedestrian: A Hybrid Automaton Model of Urban Pedestrian Behavior for Automated Driving Applications. *IEEE Access*, 9:27708–27722, 2021.

[51] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213, 2017.

[52] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 264–269, 2017.

[53] Omveer Sharma, N C Sahoo, and Niladri B Puhan. Kernelized convolutional transformer network based driver behavior estimation for conflict resolution at unsignalized roundabout. *ISA Transactions*, (xxxx), 2022.

[54] Sijia Wang, Kun Jiang, Junjie Chen, Mengmeng Yang, Zheng Fu, Tuopu Wen, and Diange Yang. Skeleton-based traffic command recognition at road intersections for intelligent vehicles. *Neurocomputing*, 501:123–134, 2022.

[55] Hongjia Zhang, Yanjuan Liu, Chang Wang, Rui Fu, Qinyu Sun, and Zhen Li. Research on a pedestrian crossing intention recognition model based on natural observation data. *Sensors (Switzerland)*, 20(6), 2020.

[56] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Coupling Intent and Action for Pedestrian Crossing Behavior Prediction. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1238–1244, 2021.

[57] Tiago Roxo and Hugo Proença. Faces in the Wild: Efficient Gender Recognition in Surveillance Conditions. 2021.

[58] Ye Li, Fangyan Shi, Shaoqi Hou, Jipeng Li, Chao Li, and Guangqiang Yin. Feature pyramid attention model and multi-label focal loss for pedestrian attribute recognition. *IEEE Access*, 8(c):164570–164579, 2020.

[59] Carlos Ismael Orozco, Eduardo Xamena, María Elena Buemi, and Julio Jacobo Berlles. Human Action Recognition in Videos using a Robust CNN LSTM Approach. *Ciencia y Tecnología*, pages 21–34, 2020.

[60] Xi Yang, Yingzhi Tang, Nannan Wang, Bin Song, and Xinbo Gao. An End-to-End Noise-Weakened Person Re-Identification and Tracking with Adaptive Partial Information. *IEEE Access*, 7:20984–20995, 2019.

[61] Huiqin Zhan, Yuan Liu, Zhibin Cui, and Hong Cheng. Pedestrian Detection and Behavior Recognition Based on Vision. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pages 771–776, 2019.

[62] Wenxiang Chen, Xiangling Zhuang, Zixin Cui, and Guojie Ma. Drivers' recognition of pedestrian road-crossing intentions: Performance and process. 64(April 2020):552–564, 2019.

[63] Dimitris Nathanael, Evangelia Portouli, Vassilis Papakostopoulos, Kostas Gkikas, and Angelos Amditis. Naturalistic observation of interactions between car drivers and pedestrians in high density urban settings. In Sebastiano Bagnara, Riccardo Tartaglia, Sara Albolino, Thomas Alexander, and Yushi Fujita, editors, *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, pages 389–397, Cham, 2019. Springer International Publishing.

[64] Imed Bouchrika. Datasets for computer vision and image processing on cvonline, 2016. `https://research.com/research/datasets-for-computer-vision-and-image-processing-on-cvonline`, Last accessed on 2021-11-05.

[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[66] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[67] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 868–884, Cham, 2016. Springer International Publishing.

[68] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[69] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking.

[70] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE international workshop on performance evaluation for tracking and surveillance (PETS)*, volume 3, pages 1–7. Citeseer, 2007.

[71] USC Institute of Robotics and Intelligent Systems. Iris computer vision lab, 2018.

[72] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[73] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K. Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *International Conference on Computer Vision (ICCV)*, 2019.

[74] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[75] Yubin DENG, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 789–792, New York, NY, USA, 2014. Association for Computing Machinery.

[76] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004.

[77] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication, 2021.

[78] Eduardo Arnold, Sajjad Mozaffari, and Mehrdad Dianati. Fast and robust registration of partially overlapping point clouds. *IEEE Robotics and Automation Letters*, pages 1–8, 2021.

[79] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[80] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[81] Rajan Parekh. *Fundamentals of IMAGE,AUDIO, and VIDEO PROCESSING Using MATLAB*. Taylor y Francis, primera ed edition, 2021.

[82] Francesco Camastra. *Machine Learning for Audio, Image and Video Analysis*, volume 18. Springer US, segunda ed edition, 2007.

[83] Al Bovik. *The Essential Guie to Video Processing*. 2009.

[84] Wenhua Fang, Jun Chen, Tao Lu, and Ruimin Hu. Pedestrian attributes recognition in surveillance scenarios with hierarchical multi-task cnn models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11165 LNCS(December):758–767, 2018.

[85] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788, 2016.

[86] Kaiming; He, Xiangyu; Zhang, Shaoqing; Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry*, 45(8):1951–1954, 2015.

[87] Riaz Ullah Khan, Xiaosong Zhang, Rajesh Kumar, and Hussain Ahmad Tariq. Analysis of resnet model for malicious code detection. *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2017*, 2018-Febru(November):239–242, 2018.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[89] Vikramaditya Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37, 2006.

[90] Yonas B Dibike, Slavco Velickov, Dimitri Solomatine, and Michael B Abbott. Model induction with support vector machines: introduction and applications. *Journal of Computing in Civil Engineering*, 15(3):208–216, 2001.

[91] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[92] MathWorks. Support vector machine (svm), 2023. https://la.mathworks.com/discovery/support-vector-machine.html#:~:text=Las%20funciones%20de%20kernel%20asignan,lineales%20en%20el%20espacio%20dimensional, Last accessed on 2022-07-03.

[93] Ikbal Gazalba, Nurul Gayatri Indah Reza, et al. Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 294–298. IEEE, 2017.

[94] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[95] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[96] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. Random forests for land cover classification. *Pattern recognition letters*, 27(4):294–300, 2006.

[97] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition*, pages 154–168. Springer, 2012.

[98] Marco Antonio Aceves Fernández. *Inteligencia Artificial para programadores con prisa*. 2021.

[99] Aprendizaje automático. Normalización, 2022. https://developers.google.com/machine-learning/data-prep/transform/normalization, Ultimo acceso on 05/09/2022.

[100] Juan Ignacio Bagnato. Comprende principal component analysis, 2018. https://www.aprendemachinelearning.com/comprende-principal-component-analysis/, Ultimo acceso on 04/11/2022.

[101] Jason Brownlee. A gentle introduction to k-fold cross-validation, 2020. https://machinelearningmastery.com/k-fold-cross-validation/, Ultimo acceso 11/12/2022.

[102] Luis Torres. Curva roc y auc en python". the machine learners, 2023. https://www.themachinelearners.com/curva-roc-vs-prec-recall/, Ultimo acceso 08/06/2023.

[103] Armando silva velazaquez. Crosswalk&pedestrian_looking dataset. https://universe.roboflow.com/tesis-afowb/crosswalk-pedestrian_looking, feb 2023. visited on 2023-06-06.

1. ANEXOS

Video	Base Datos	Accuracy	Precision	Recall	F1_score
285	Metricas_Finales_BD_Imp	0	0	0	0
285	Metricas_Finales_BD_Imp_Norm	0	0	0	0
285	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
285	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0
292	Metricas_Finales_BD_Imp	0	0	0	0
292	Metricas_Finales_BD_Imp_Norm	0	0	0	0
292	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
292	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0
296	Metricas_Finales_BD_Imp	0	0	0	0
296	Metricas_Finales_BD_Imp_Norm	0	0	0	0
296	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
296	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0
323	Metricas_Finales_BD_Imp	0	0	0	0
323	Metricas_Finales_BD_Imp_Norm	0	0	0	0
323	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
323	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0
343	Metricas_Finales_BD_Imp	0	0	0	0
343	Metricas_Finales_BD_Imp_Norm	0	0	0	0
343	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
343	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0
346	Metricas_Finales_BD_Imp	0	0	0	0
346	Metricas_Finales_BD_Imp_Norm	0	0	0	0
346	Metricas_Finales_BD_Imp_Norm_Red	0	0	0	0
346	Metricas_Finales_BD_Imp_SS_Red	0	0	0	0

Figura A.1: Resultados del sistema final para los vídeos que no presentaban un peatón o el sistema nunca logro detectar.

Video	Base Datos	Accuracy	Precision	Recall	F1_score	Time detections
284	Metricas_Finales_BD_Imp	0.154	0.000	0.000	0.000	0.687
284	Metricas_Finales_BD_Imp_Norm	0.231	0.000	0.000	0.000	0.916
284	Metricas_Finales_BD_Imp_Norm_Red	0.231	0.000	0.000	0.000	0.666
284	Metricas_Finales_BD_Imp_SS_Red	0.231	0.000	0.000	0.000	0.678
288	Metricas_Finales_BD_Imp	0.000	0.000	0.000	0.000	0.675
288	Metricas_Finales_BD_Imp_Norm	0.333	0.000	0.000	0.000	0.666
288	Metricas_Finales_BD_Imp_Norm_Red	0.333	0.000	0.000	0.000	0.954
288	Metricas_Finales_BD_Imp_SS_Red	0.333	0.000	0.000	0.000	1.046
289	Metricas_Finales_BD_Imp	0.053	0.000	0.000	0.000	0.746
289	Metricas_Finales_BD_Imp_Norm	0.000	0.000	0.000	0.000	0.780
289	Metricas_Finales_BD_Imp_Norm_Red	0.000	0.000	0.000	0.000	0.707
289	Metricas_Finales_BD_Imp_SS_Red	0.053	0.000	0.000	0.000	0.748
300	Metricas_Finales_BD_Imp	0.244	0.000	0.000	0.000	0.877
300	Metricas_Finales_BD_Imp_Norm	0.000	0.000	0.000	0.000	0.896
300	Metricas_Finales_BD_Imp_Norm_Red	0.000	0.000	0.000	0.000	0.884
300	Metricas_Finales_BD_Imp_SS_Red	0.000	0.000	0.000	0.000	0.868
304	Metricas_Finales_BD_Imp	0.538	0.000	0.000	0.000	0.734
304	Metricas_Finales_BD_Imp_Norm	0.846	0.000	0.000	0.000	0.729
304	Metricas_Finales_BD_Imp_Norm_Red	0.846	0.000	0.000	0.000	0.744
304	Metricas_Finales_BD_Imp_SS_Red	0.846	0.000	0.000	0.000	0.710
308	Metricas_Finales_BD_Imp	0.104	0.000	0.000	0.000	0.817
308	Metricas_Finales_BD_Imp_Norm	0.009	0.000	0.000	0.000	0.785
308	Metricas_Finales_BD_Imp_Norm_Red	0.009	0.000	0.000	0.000	0.794
308	Metricas_Finales_BD_Imp_SS_Red	0.047	0.000	0.000	0.000	0.786
309	Metricas_Finales_BD_Imp	0.211	0.045	0.400	0.082	0.893
309	Metricas_Finales_BD_Imp_Norm	0.474	0.069	0.400	0.118	0.868
309	Metricas_Finales_BD_Imp_Norm_Red	0.474	0.069	0.400	0.118	0.867
309	Metricas_Finales_BD_Imp_SS_Red	0.579	0.048	0.200	0.077	0.897
318	Metricas_Finales_BD_Imp	0.292	0.000	0.000	0.000	0.958
318	Metricas_Finales_BD_Imp_Norm	0.000	0.000	0.000	0.000	0.930
318	Metricas_Finales_BD_Imp_Norm_Red	0.000	0.000	0.000	0.000	0.956
318	Metricas_Finales_BD_Imp_SS_Red	0.000	0.000	0.000	0.000	0.923
329	Metricas_Finales_BD_Imp	0.243	0.000	0.000	0.000	0.930
329	Metricas_Finales_BD_Imp_Norm	0.086	0.000	0.000	0.000	0.910
329	Metricas_Finales_BD_Imp_Norm_Red	0.086	0.000	0.000	0.000	0.984
329	Metricas_Finales_BD_Imp_SS_Red	0.086	0.000	0.000	0.000	0.914
335	Metricas_Finales_BD_Imp	0.214	0.067	1.000	0.126	0.731
335	Metricas_Finales_BD_Imp_Norm	0.189	0.065	1.000	0.122	0.759
335	Metricas_Finales_BD_Imp_Norm_Red	0.189	0.065	1.000	0.122	0.746
335	Metricas_Finales_BD_Imp_SS_Red	0.189	0.065	1.000	0.122	0.740
337	Metricas_Finales_BD_Imp	0.375	0.000	0.000	0.000	0.707
337	Metricas_Finales_BD_Imp_Norm	0.000	0.000	0.000	0.000	0.699
337	Metricas_Finales_BD_Imp_Norm_Red	0.000	0.000	0.000	0.000	0.754
337	Metricas_Finales_BD_Imp_SS_Red	0.000	0.000	0.000	0.000	0.806
342	Metricas_Finales_BD_Imp	0.238	0.000	0.000	0.000	0.796
342	Metricas_Finales_BD_Imp_Norm	0.286	0.000	0.000	0.000	0.772

Figura A.2: Resultados del sistema final para vídeos donde le peatón no cruza parte 1.

342	Metricas_Finales_BD_Imp_Norm_Red	0.286	0.000	0.000	0.000	0.805
342	Metricas_Finales_BD_Imp_SS_Red	0.429	0.000	0.000	0.000	0.812
344	Metricas_Finales_BD_Imp	0.250	0.000	0.000	0.000	0.747
344	Metricas_Finales_BD_Imp_Norm	0.000	0.000	0.000	0.000	0.892
344	Metricas_Finales_BD_Imp_Norm_Red	0.000	0.000	0.000	0.000	0.862
344	Metricas_Finales_BD_Imp_SS_Red	0.000	0.000	0.000	0.000	0.815

Figura A.3: Resultados del sistema final para vídeos donde le peatón no cruza parte 2.

Video	Base Datos	Accuracy	Precision	Recall	F1_score	Time detections
277	Metricas_Finales_BD_Imp	0.599	0.588	0.990	0.738	0.753
277	Metricas_Finales_BD_Imp_Norm	0.686	0.651	0.969	0.779	0.771
277	Metricas_Finales_BD_Imp_Norm_Red	0.692	0.655	0.969	0.782	0.774
277	Metricas_Finales_BD_Imp_SS_Red	0.680	0.646	0.969	0.776	0.756
278	Metricas_Finales_BD_Imp	0.647	0.375	0.750	0.500	0.638
278	Metricas_Finales_BD_Imp_Norm	0.882	0.667	1.000	0.800	0.645
278	Metricas_Finales_BD_Imp_Norm_Red	0.882	0.667	1.000	0.800	0.739
278	Metricas_Finales_BD_Imp_SS_Red	0.882	0.667	1.000	0.800	0.635
279	Metricas_Finales_BD_Imp	0.750	1.000	0.750	0.857	0.732
279	Metricas_Finales_BD_Imp_Norm	0.950	1.000	0.950	0.974	0.722
279	Metricas_Finales_BD_Imp_Norm_Red	0.850	1.000	0.850	0.919	0.772
279	Metricas_Finales_BD_Imp_SS_Red	0.850	1.000	0.850	0.919	0.704
280	Metricas_Finales_BD_Imp	1.000	1.000	1.000	1.000	0.590
280	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.758
280	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.615
280	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.684
281	Metricas_Finales_BD_Imp	1.000	1.000	1.000	1.000	0.702
281	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.688
281	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.789
281	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.598
283	Metricas_Finales_BD_Imp	0.875	1.000	0.875	0.933	0.689
283	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.663
283	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.731
283	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.738
286	Metricas_Finales_BD_Imp	0.669	0.669	1.000	0.802	0.854
286	Metricas_Finales_BD_Imp_Norm	0.669	0.669	1.000	0.802	0.783
286	Metricas_Finales_BD_Imp_Norm_Red	0.669	0.669	1.000	0.802	0.772
286	Metricas_Finales_BD_Imp_SS_Red	0.669	0.669	1.000	0.802	0.771
287	Metricas_Finales_BD_Imp	0.589	0.575	1.000	0.730	0.760
287	Metricas_Finales_BD_Imp_Norm	0.567	0.562	1.000	0.719	0.776
287	Metricas_Finales_BD_Imp_Norm_Red	0.689	0.641	1.000	0.781	0.800
287	Metricas_Finales_BD_Imp_SS_Red	0.656	0.617	1.000	0.763	0.782
290	Metricas_Finales_BD_Imp	0.419	0.419	1.000	0.590	0.790
290	Metricas_Finales_BD_Imp_Norm	0.442	0.429	1.000	0.600	0.805
290	Metricas_Finales_BD_Imp_Norm_Red	0.442	0.429	1.000	0.600	0.793
290	Metricas_Finales_BD_Imp_SS_Red	0.442	0.429	1.000	0.600	0.852
291	Metricas_Finales_BD_Imp	0.722	0.728	0.973	0.833	0.756
291	Metricas_Finales_BD_Imp_Norm	0.861	0.889	0.920	0.904	0.784
291	Metricas_Finales_BD_Imp_Norm_Red	0.867	0.897	0.920	0.908	0.764
291	Metricas_Finales_BD_Imp_SS_Red	0.861	0.889	0.920	0.904	0.734
293	Metricas_Finales_BD_Imp	0.821	0.852	0.958	0.902	0.818
293	Metricas_Finales_BD_Imp_Norm	0.857	0.857	1.000	0.923	0.864
293	Metricas_Finales_BD_Imp_Norm_Red	0.884	0.903	0.969	0.935	0.827
293	Metricas_Finales_BD_Imp_SS_Red	0.884	0.903	0.969	0.935	0.818
294	Metricas_Finales_BD_Imp	0.853	0.879	0.967	0.921	0.741
294	Metricas_Finales_BD_Imp_Norm	0.926	0.923	1.000	0.960	0.780

Figura A.4: Resultados del sistema final para vídeos donde presenta ambos estados parte 1.

294	Metricas_Finales_BD_Imp_Norm_Red	0.926	0.923	1.000	0.960	0.757
294	Metricas_Finales_BD_Imp_SS_Red	0.926	0.923	1.000	0.960	0.735
295	Metricas_Finales_BD_Imp	0.703	0.567	0.958	0.712	0.775
295	Metricas_Finales_BD_Imp_Norm	0.870	0.783	0.915	0.844	0.779
295	Metricas_Finales_BD_Imp_Norm_Red	0.870	0.783	0.915	0.844	0.769
295	Metricas_Finales_BD_Imp_SS_Red	0.870	0.783	0.915	0.844	0.766
297	Metricas_Finales_BD_Imp	0.777	0.792	0.957	0.867	0.891
297	Metricas_Finales_BD_Imp_Norm	0.880	0.878	0.977	0.925	0.924
297	Metricas_Finales_BD_Imp_Norm_Red	0.880	0.878	0.977	0.925	0.901
297	Metricas_Finales_BD_Imp_SS_Red	0.880	0.878	0.977	0.925	0.907
298	Metricas_Finales_BD_Imp	0.841	0.857	0.978	0.914	0.876
298	Metricas_Finales_BD_Imp_Norm	0.860	0.860	1.000	0.925	0.887
298	Metricas_Finales_BD_Imp_Norm_Red	0.860	0.860	1.000	0.925	0.897
298	Metricas_Finales_BD_Imp_SS_Red	0.860	0.860	1.000	0.925	0.910
299	Metricas_Finales_BD_Imp	0.703	0.667	1.000	0.800	0.858
299	Metricas_Finales_BD_Imp_Norm	0.770	0.721	1.000	0.838	0.891
299	Metricas_Finales_BD_Imp_Norm_Red	0.770	0.721	1.000	0.838	0.914
299	Metricas_Finales_BD_Imp_SS_Red	0.743	0.698	1.000	0.822	0.908
301	Metricas_Finales_BD_Imp	0.626	0.602	1.000	0.752	0.745
301	Metricas_Finales_BD_Imp_Norm	0.754	0.707	0.969	0.817	0.788
301	Metricas_Finales_BD_Imp_Norm_Red	0.754	0.707	0.969	0.817	0.795
301	Metricas_Finales_BD_Imp_SS_Red	0.737	0.688	0.979	0.809	0.772
302	Metricas_Finales_BD_Imp	0.917	0.917	1.000	0.957	0.831
302	Metricas_Finales_BD_Imp_Norm	0.917	0.917	1.000	0.957	0.843
302	Metricas_Finales_BD_Imp_Norm_Red	0.917	0.969	0.939	0.954	0.848
302	Metricas_Finales_BD_Imp_SS_Red	0.917	0.941	0.970	0.955	0.881
303	Metricas_Finales_BD_Imp	0.898	0.905	0.987	0.944	0.773
303	Metricas_Finales_BD_Imp_Norm	0.989	0.987	1.000	0.994	0.759
303	Metricas_Finales_BD_Imp_Norm_Red	0.989	0.987	1.000	0.994	0.761
303	Metricas_Finales_BD_Imp_SS_Red	0.989	0.987	1.000	0.994	0.743
305	Metricas_Finales_BD_Imp	0.681	0.679	0.950	0.792	0.887
305	Metricas_Finales_BD_Imp_Norm	0.766	0.764	0.917	0.833	0.870
305	Metricas_Finales_BD_Imp_Norm_Red	0.766	0.764	0.917	0.833	0.865
305	Metricas_Finales_BD_Imp_SS_Red	0.766	0.764	0.917	0.833	0.837
306	Metricas_Finales_BD_Imp	0.667	0.658	0.988	0.790	0.782
306	Metricas_Finales_BD_Imp_Norm	0.762	0.727	1.000	0.842	0.769
306	Metricas_Finales_BD_Imp_Norm_Red	0.762	0.727	1.000	0.842	0.771
306	Metricas_Finales_BD_Imp_SS_Red	0.746	0.714	1.000	0.833	0.751
307	Metricas_Finales_BD_Imp	0.884	0.915	0.963	0.939	0.845
307	Metricas_Finales_BD_Imp_Norm	0.905	0.917	0.985	0.950	0.816
307	Metricas_Finales_BD_Imp_Norm_Red	0.905	0.917	0.985	0.950	0.823
307	Metricas_Finales_BD_Imp_SS_Red	0.912	0.918	0.993	0.954	0.820
310	Metricas_Finales_BD_Imp	0.925	0.941	0.980	0.960	0.875
310	Metricas_Finales_BD_Imp_Norm	0.896	1.000	0.888	0.941	0.819
310	Metricas_Finales_BD_Imp_Norm_Red	0.896	1.000	0.888	0.941	0.859
310	Metricas_Finales_BD_Imp_SS_Red	0.906	1.000	0.898	0.946	0.824
311	Metricas_Finales_BD_Imp	0.864	0.864	1.000	0.927	0.805

Figura A.5: Resultados del sistema final para vídeos donde presenta ambos estados parte 2.

311	Metricas_Finales_BD_Imp_Norm	0.864	0.864	1.000	0.927	0.808
311	Metricas_Finales_BD_Imp_Norm_Red	0.864	0.864	1.000	0.927	0.871
311	Metricas_Finales_BD_Imp_SS_Red	0.864	0.864	1.000	0.927	0.815
312	Metricas_Finales_BD_Imp	0.951	1.000	0.951	0.975	0.894
312	Metricas_Finales_BD_Imp_Norm	0.878	1.000	0.878	0.935	0.798
312	Metricas_Finales_BD_Imp_Norm_Red	0.878	1.000	0.878	0.935	0.889
312	Metricas_Finales_BD_Imp_SS_Red	0.927	1.000	0.927	0.962	0.837
313	Metricas_Finales_BD_Imp	0.644	0.654	0.974	0.783	0.964
313	Metricas_Finales_BD_Imp_Norm	0.653	0.660	0.977	0.787	0.944
313	Metricas_Finales_BD_Imp_Norm_Red	0.653	0.660	0.977	0.787	0.962
313	Metricas_Finales_BD_Imp_SS_Red	0.653	0.660	0.977	0.787	0.920
314	Metricas_Finales_BD_Imp	0.810	0.944	0.850	0.895	0.953
314	Metricas_Finales_BD_Imp_Norm	0.667	0.933	0.700	0.800	0.929
314	Metricas_Finales_BD_Imp_Norm_Red	0.667	0.933	0.700	0.800	0.934
314	Metricas_Finales_BD_Imp_SS_Red	0.667	0.933	0.700	0.800	0.906
315	Metricas_Finales_BD_Imp	0.899	0.930	0.964	0.947	0.896
315	Metricas_Finales_BD_Imp_Norm	0.843	0.926	0.904	0.915	0.886
315	Metricas_Finales_BD_Imp_Norm_Red	0.831	0.925	0.892	0.908	0.907
315	Metricas_Finales_BD_Imp_SS_Red	0.843	0.926	0.904	0.915	0.886
316	Metricas_Finales_BD_Imp	0.900	1.000	0.900	0.947	0.882
316	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.868
316	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.861
316	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.849
317	Metricas_Finales_BD_Imp	0.933	1.000	0.933	0.966	0.925
317	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.837
317	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.838
317	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.878
319	Metricas_Finales_BD_Imp	0.875	1.000	0.875	0.933	0.955
319	Metricas_Finales_BD_Imp_Norm	0.625	1.000	0.625	0.769	0.799
319	Metricas_Finales_BD_Imp_Norm_Red	0.625	1.000	0.625	0.769	0.888
319	Metricas_Finales_BD_Imp_SS_Red	0.625	1.000	0.625	0.769	0.912
320	Metricas_Finales_BD_Imp	0.482	0.448	0.933	0.605	0.965
320	Metricas_Finales_BD_Imp_Norm	0.426	0.423	0.967	0.589	0.923
320	Metricas_Finales_BD_Imp_Norm_Red	0.426	0.423	0.967	0.589	0.956
320	Metricas_Finales_BD_Imp_SS_Red	0.426	0.423	0.967	0.589	0.941
321	Metricas_Finales_BD_Imp	0.933	1.000	0.933	0.966	0.963
321	Metricas_Finales_BD_Imp_Norm	0.933	1.000	0.933	0.966	0.868
321	Metricas_Finales_BD_Imp_Norm_Red	0.933	1.000	0.933	0.966	0.897
321	Metricas_Finales_BD_Imp_SS_Red	0.933	1.000	0.933	0.966	0.940
322	Metricas_Finales_BD_Imp	0.902	0.940	0.957	0.948	0.942
322	Metricas_Finales_BD_Imp_Norm	0.854	1.000	0.843	0.915	0.878
322	Metricas_Finales_BD_Imp_Norm_Red	0.854	1.000	0.843	0.915	1.005
322	Metricas_Finales_BD_Imp_SS_Red	0.837	0.970	0.852	0.907	0.922
324	Metricas_Finales_BD_Imp	0.500	0.500	1.000	0.667	0.998
324	Metricas_Finales_BD_Imp_Norm	0.500	0.500	1.000	0.667	0.902
324	Metricas_Finales_BD_Imp_Norm_Red	0.500	0.500	1.000	0.667	0.982
324	Metricas_Finales_BD_Imp_SS_Red	0.500	0.500	1.000	0.667	0.928

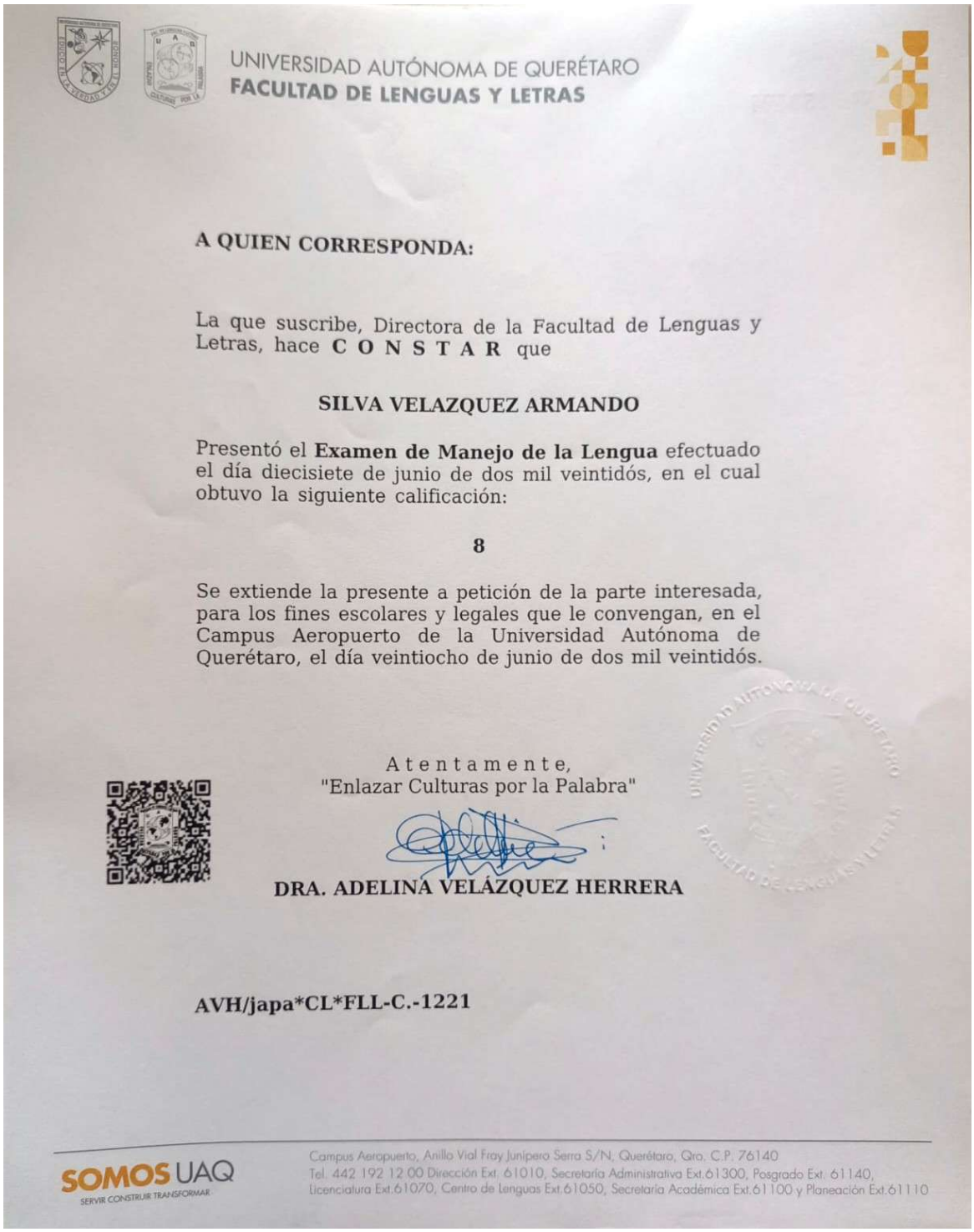
Figura A.6: Resultados del sistema final para vídeos donde presenta ambos estados parte 3.

325	Metricas_Finales_BD_Imp	0.404	0.029	1.000	0.056	0.949
325	Metricas_Finales_BD_Imp_Norm	0.754	0.067	1.000	0.125	0.942
325	Metricas_Finales_BD_Imp_Norm_Red	0.754	0.067	1.000	0.125	1.007
325	Metricas_Finales_BD_Imp_SS_Red	0.754	0.067	1.000	0.125	0.966
326	Metricas_Finales_BD_Imp	0.500	0.474	1.000	0.643	0.964
326	Metricas_Finales_BD_Imp_Norm	0.450	0.450	1.000	0.621	0.937
326	Metricas_Finales_BD_Imp_Norm_Red	0.450	0.450	1.000	0.621	0.953
326	Metricas_Finales_BD_Imp_SS_Red	0.450	0.450	1.000	0.621	0.924
327	Metricas_Finales_BD_Imp	0.750	0.727	0.889	0.800	0.990
327	Metricas_Finales_BD_Imp_Norm	0.875	0.818	1.000	0.900	1.019
327	Metricas_Finales_BD_Imp_Norm_Red	0.875	0.818	1.000	0.900	1.002
327	Metricas_Finales_BD_Imp_SS_Red	0.938	0.900	1.000	0.947	0.967
328	Metricas_Finales_BD_Imp	1.000	1.000	1.000	1.000	
328	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	
328	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	
328	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	
330	Metricas_Finales_BD_Imp	1.000	1.000	1.000	1.000	0.920
330	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.958
330	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	1.120
330	Metricas_Finales_BD_Imp_SS_Red	1.000	1.000	1.000	1.000	0.920
331	Metricas_Finales_BD_Imp	0.657	0.656	0.935	0.771	1.178
331	Metricas_Finales_BD_Imp_Norm	0.609	0.614	0.988	0.757	1.129
331	Metricas_Finales_BD_Imp_Norm_Red	0.609	0.614	0.988	0.757	1.189
331	Metricas_Finales_BD_Imp_SS_Red	0.693	0.700	0.882	0.780	1.119
332	Metricas_Finales_BD_Imp	0.785	0.787	0.937	0.855	0.992
332	Metricas_Finales_BD_Imp_Norm	0.968	0.984	0.968	0.976	0.963
332	Metricas_Finales_BD_Imp_Norm_Red	0.968	0.984	0.968	0.976	1.029
332	Metricas_Finales_BD_Imp_SS_Red	0.968	0.984	0.968	0.976	0.979
333	Metricas_Finales_BD_Imp	0.795	0.793	0.968	0.872	0.925
333	Metricas_Finales_BD_Imp_Norm	0.871	1.000	0.821	0.902	0.928
333	Metricas_Finales_BD_Imp_Norm_Red	0.871	1.000	0.821	0.902	0.965
333	Metricas_Finales_BD_Imp_SS_Red	0.871	1.000	0.821	0.902	0.919
334	Metricas_Finales_BD_Imp	0.710	0.750	0.913	0.824	0.998
334	Metricas_Finales_BD_Imp_Norm	0.742	0.742	1.000	0.852	0.981
334	Metricas_Finales_BD_Imp_Norm_Red	0.742	0.742	1.000	0.852	0.751
334	Metricas_Finales_BD_Imp_SS_Red	0.742	0.742	1.000	0.852	1.002
336	Metricas_Finales_BD_Imp	0.370	0.282	0.917	0.431	0.713
336	Metricas_Finales_BD_Imp_Norm	0.935	0.800	1.000	0.889	0.762
336	Metricas_Finales_BD_Imp_Norm_Red	0.957	0.857	1.000	0.923	0.764
336	Metricas_Finales_BD_Imp_SS_Red	0.957	0.857	1.000	0.923	0.738
338	Metricas_Finales_BD_Imp	0.750	0.747	1.000	0.855	0.733
338	Metricas_Finales_BD_Imp_Norm	0.776	0.767	1.000	0.868	0.754
338	Metricas_Finales_BD_Imp_Norm_Red	0.776	0.767	1.000	0.868	0.757
338	Metricas_Finales_BD_Imp_SS_Red	0.776	0.767	1.000	0.868	0.745
339	Metricas_Finales_BD_Imp	0.960	0.973	0.986	0.980	0.727
339	Metricas_Finales_BD_Imp_Norm	0.973	0.973	1.000	0.986	0.729
339	Metricas_Finales_BD_Imp_Norm_Red	0.973	0.973	1.000	0.986	0.758

Figura A.7: Resultados del sistema final para vídeos donde presenta ambos estados parte 4.

339	Metricas_Finales_BD_Imp_SS_Red	0.973	0.973	1.000	0.986	0.740
340	Metricas_Finales_BD_Imp	0.234	0.200	1.000	0.333	0.823
340	Metricas_Finales_BD_Imp_Norm	0.191	0.191	1.000	0.321	0.802
340	Metricas_Finales_BD_Imp_Norm_Red	0.191	0.191	1.000	0.321	0.848
340	Metricas_Finales_BD_Imp_SS_Red	0.191	0.191	1.000	0.321	0.841
341	Metricas_Finales_BD_Imp	0.928	1.000	0.928	0.963	0.758
341	Metricas_Finales_BD_Imp_Norm	1.000	1.000	1.000	1.000	0.781
341	Metricas_Finales_BD_Imp_Norm_Red	1.000	1.000	1.000	1.000	0.826
341	Metricas_Finales_BD_Imp_SS_Red	0.755	1.000	0.755	0.861	0.804
345	Metricas_Finales_BD_Imp	0.478	0.371	0.867	0.520	0.848
345	Metricas_Finales_BD_Imp_Norm	0.326	0.326	1.000	0.492	0.835
345	Metricas_Finales_BD_Imp_Norm_Red	0.326	0.326	1.000	0.492	0.799
345	Metricas_Finales_BD_Imp_SS_Red	0.304	0.311	0.933	0.467	0.852

Figura A.8: Resultados del sistema final para vídeos donde presenta ambos estados parte 5.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS

A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

SILVA VELAZQUEZ ARMANDO

Presentó el **Examen de Manejo de la Lengua** efectuado el día diecisiete de junio de dos mil veintidós, en el cual obtuvo la siguiente calificación:

8

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día veintiocho de junio de dos mil veintidós.

A t e n t a m e n t e,
"Enlazar Culturas por la Palabra"

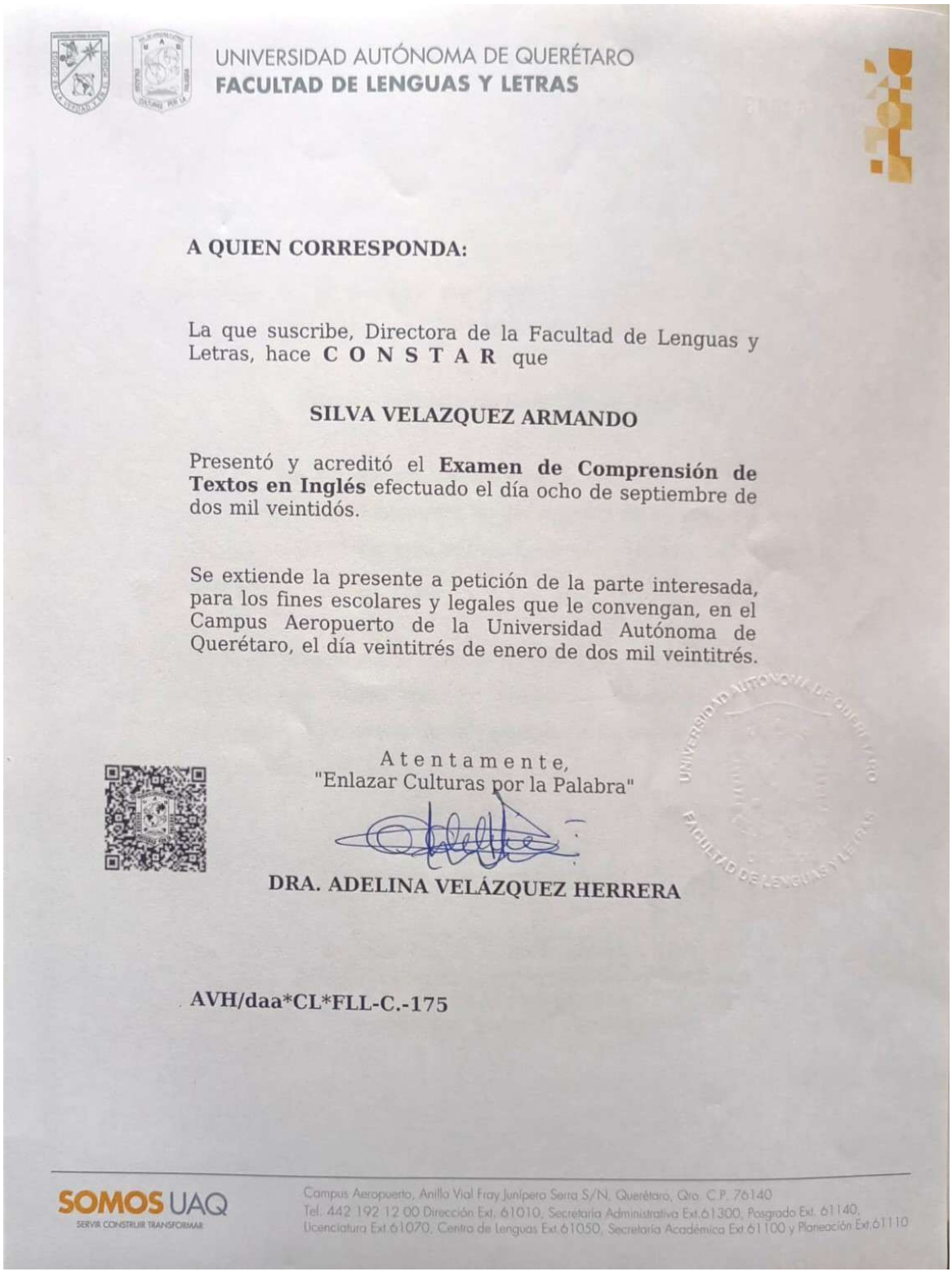
DRA. ADELINA VELÁZQUEZ HERRERA

AVH/japa*CL*FLL-C.-1221

SOMOS UAQ
SERVIR CONSTRUIR TRANSFORMAR

Campus Aeropuerto, Anillo Vial Fray Junípero Serra S/N, Querétaro, Qro. C.P. 76140
Tel. 442 192 12 00 Dirección Ext. 61010, Secretaría Administrativa Ext.61300, Posgrado Ext. 61140,
Licenciatura Ext.61070, Centro de Lenguas Ext.61050, Secretaría Académica Ext.61100 y Planeación Ext.61110

Figura A.9: Constancia de manejo de la lengua.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS

A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

SILVA VELAZQUEZ ARMANDO

Presentó y acreditó el **Examen de Comprensión de Textos en Inglés** efectuado el día ocho de septiembre de dos mil veintidós.

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día veintitrés de enero de dos mil veintitrés.

A t e n t a m e n t e,
"Enlazar Culturas por la Palabra"

DRA. ADELINA VELÁZQUEZ HERRERA

AVH/daa*CL*FLL-C.-175

SOMOS UAQ
SERVIR CONSTRUIR TRANSFORMAR

Campus Aeropuerto, Anillo Vial Fray Junípero Serra S/N, Querétaro, Qro. C.P. 76140
Tel. 442 192 12 00 Dirección Ext. 61010, Secretaría Administrativa Ext.61300, Posgrado Ext. 61140, Licenciatura Ext.61070, Centro de Lenguas Ext.61050, Secretaría Académica Ext.61100 y Planeación Ext.61110

Figura A.10: Constancia de comprensión de textos de lengua extranjera.

Figura A.11: Constancia de producto académico.