UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM VALLE DE MÉXICO

Sistema para la detección de conductas delictivas asociadas con el robo de autopartes aplicando redes neuronales convolucionales 3D

TESIS

Que para obtener el Grado de

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P r e s e n t a

Ing. Bruno Diego Martínez Contreras

Tutor académico:
Dr. Saturnino Job Morales Escobar

Tutores adjuntos:

Dr. Asdrúbal López Chau

Dr. Víctor Manuel Landassuri Moreno

Atizapán de Zaragoza, Edo. de Méx. Octubre de 2025

“Tú tienes poder sobre tu mente,
no sobre los acontecimientos.
Date cuenta de esto y
encontraras tu fuerza.”

-Marco Aurelio-

Dedicatorias

Dedico este trabajo a mi familia, por su amor incondicional y constante apoyo; a mis amigos, por acompañarme en cada etapa del camino; a mis profesores, por su guía y compromiso con mi formación, siendo un pilar fundamental para avanzar y culminar este logro académico.

Agradecimientos

A la Universidad Autónoma del Estado de México por brindarme la oportunidad de formarme profesionalmente y poder alcanzar este logro académico.

Al Dr. Saturnino Job Morales Escobar, mi tutor, por su constante apoyo, y por sus valiosas orientaciones y conversaciones que fueron fundamentales para la culminación de este trabajo.

A mis tutores adjuntos, el Dr. Víctor Manuel Landassuri Moreno y el Dr. Asdrúbal López Chau, por su valioso apoyo, orientación y compromiso durante el desarrollo de este trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el respaldo económico brindado, el cual fue fundamental para cursar y concluir satisfactoriamente mis estudios de posgrado.

Resumen

Este trabajo propone un sistema capaz de identificar conductas delictivas asociadas al robo de autopartes mediante el uso de redes neuronales convolucionales tridimensionales (3D CNN por sus siglas en inglés). El enfoque se basa en el aprendizaje supervisado y tiene como objetivo detectar comportamientos sospechosos a fin de implementar medidas preventivas oportunas. Para el entrenamiento del modelo, se construyó un conjunto de datos con 2000 videos clasificados en dos categorías: “Robo” y “No Robo”. Se aplicaron diversas técnicas de limpieza como limpiado, depurado, recorte y filtrado de videos, para reducir el ruido en los datos y mejorar la calidad del entrenamiento.

La 3D CNN fue entrenada en la plataforma de Google Colab, donde se enfrentaron diversos desafíos con el alto consumo computacional. Para optimizar el rendimiento del sistema, se implementó la conversión de los videos a archivos binarios utilizando TFRecords. Esto permitió segmentar el conjunto de datos en 20 lotes, simplificando el proceso de entrenamiento. Al completar el entrenamiento de cada lote, se generó un modelo, obteniéndose así un total de 20 modelos de la 3D CNN. De estos, seis modelos alcanzaron una precisión del 100%, y otros seis superaron el 90%. Sin embargo, los 8 modelos restantes obtuvieron resultados inferiores al 75%. Lo que indicó áreas de mejora tanto en los datos como en la configuración de los modelos.

Adicionalmente, se implementó una técnica de recorte de frames en los videos, lo cual mejoró significativamente la clasificación, especialmente en términos de F1-score. Finalmente, se comparó el rendimiento de la 3D CNN con otras arquitecturas como memoria a largo y corto plazo (LSTM por sus siglas en inglés), unidad recurrente cerrada (GRU por sus siglas en inglés), unidad recurrente cerrada bidireccional (BiGRU por sus siglas en inglés) y red neuronal convolucional con memoria a largo y corto plazo (CNN-LSTM por sus siglas en inglés). Con base en los resultados, la 3D CNN desarrollada en este proyecto fue la más efectiva al capturar patrones espaciales y temporales en los videos, consolidándose como la opción más adecuada para la detección automática de conductas delictivas en contextos visuales.

Palabras clave: Conducta delictiva, aprendizaje supervisado, Redes Neuronales Convolucionales 3D.

Abstract

This work proposes a system capable of identifying criminal behaviors associated with auto parts theft through the use of three-dimensional convolutional neural networks (3D CNNs). The approach is based on supervised learning and aims to detect suspicious behaviors in order to implement timely preventive measures. For model training, a dataset was built with 2,000 videos classified into two categories: "Theft" and "Non-Theft." Various cleaning techniques were applied—such as scrubbing, refining, trimming, and filtering videos—to reduce data noise and improve training quality.

The 3D CNN was trained on the Google Colab platform, where several challenges related to high computational consumption were encountered. To optimize system performance, the videos were converted into binary files using TFRecords. This allowed the dataset to be divided into 20 batches, simplifying the training process. Upon completing the training of each batch, a model was generated, resulting in a total of 20 3D CNN models. Of these, six models achieved 100% accuracy, and another six exceeded 90%. However, the remaining eight models obtained results below 75%, indicating areas for improvement in both the data and model configuration.

Additionally, a frame reduction technique was implemented in the videos, which significantly improved classification, particularly in terms of F1-score. Finally, the performance of the 3D CNN was compared with other architectures such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional Gated Recurrent Unit (BiGRU), and Convolutional Neural Network with Long Short-Term Memory (CNN-LSTM). Based on the results, the 3D CNN developed in this project proved to be the most effective at capturing spatial and temporal patterns in the videos, establishing itself as the most suitable option for the automatic detection of criminal behavior in visual contexts.

Keywords: Criminal behavior, supervised learning, 3D Convolutional Neural Networks.

Índice

Índice de figuras

Índice de tablas

CAPÍTULO 1

1 Introducción

1.1 Antecedentes

La delincuencia, “fenómeno de delinquir o cometer actos fuera de los estatutos previstos por la sociedad” (Agripino & Felipe, 2011), es uno de los principales problemas sociales alrededor del mundo, según (ONU-Habitat - Violencia e inseguridad en las ciudades, 2022) en los últimos años, la delincuencia ha aumentado en distintos países por la proliferación de armas, el desempleo y el abuso de sustancias tóxicas para la salud. Por ello se han observado y estudiado las conductas delictivas que se definen como “la realización de conductas en contra de las leyes de un país” (Kazdin & Casal, 1999), entre las que se encuentran el vandalismo, el hurto y la venta de drogas, por mencionar algunas. En México existe una clasificación de conductas delictivas que atentan contra el patrimonio, como se muestra en la figura 1.1. Se puede observar que los delitos de violencia familiar, lesiones, robo a casa habitación, robo de vehículo y robo a negocio tienen alta incidencia.

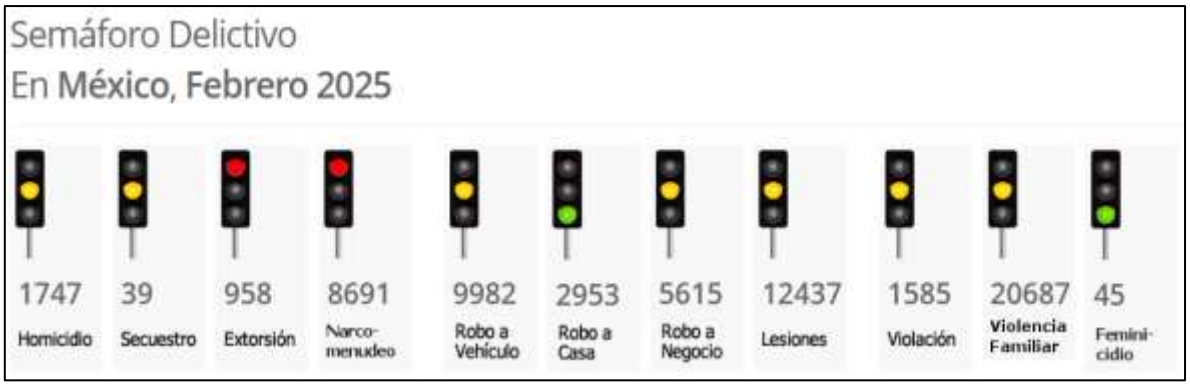

Figura 1.1. Delitos en México con mayor incidencia. (Semáforo Delictivo, 2025)

Estos delitos también son frecuentes en el Estado de México, como se indica en el informe del Semáforo Delictivo, (Semáforo Delictivo, 2025). En la figura 1.2, se presenta un gráfico que muestra la incidencia del robo de vehículos en los últimos 5 años. Delito relacionado al abordado en el presente trabajo. Sin embargo, es importante destacar que la falta de denuncias en el robo de autopartes dificulta el análisis para determinar su incidencia y la afectación que provoca de manera oficial.

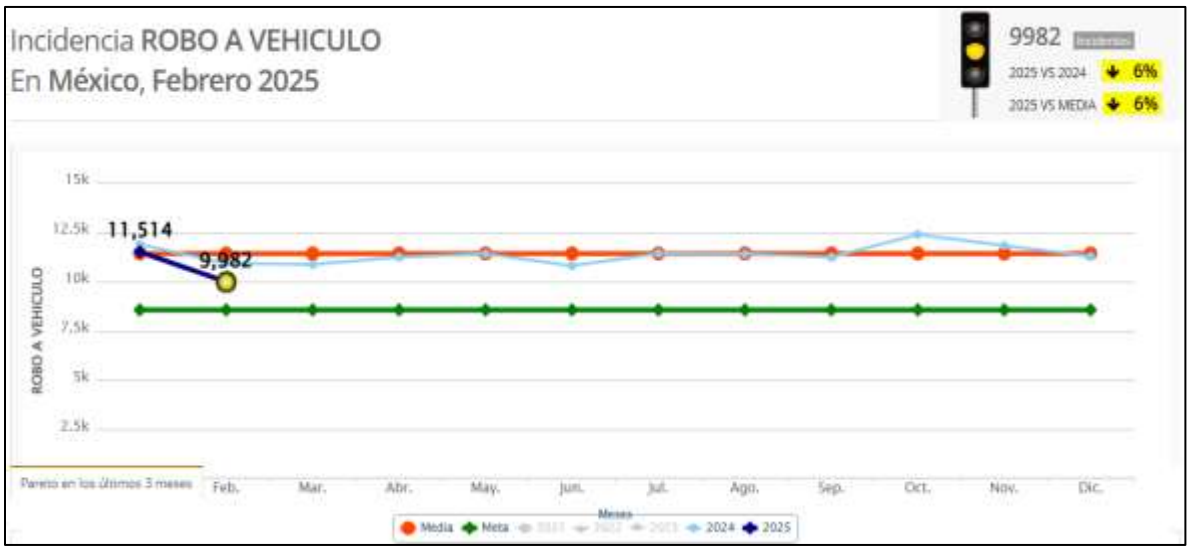

Figura 1.2. Robo a Vehículo, Estado de México. (Semáforo Delictivo, 2025)

Los delitos mencionados, son en esencia, el principal problema de la seguridad ciudadana, aunque actualmente no se ha logrado determinar con precisión el o los factores para estos hechos. Esto motiva que diferentes disciplinas entre ellas la psicología, la sociología, el derecho y la criminología, estudien este gran problema que además de complejo es costoso y dañino para la sociedad.

Por otra parte, el desarrollo tecnológico ha permitido incorporar nuevas herramientas con el fin de incrementar la seguridad en zonas de riesgo al enfocarse en la detección y prevención de delitos ya sea de manera masiva o particular. Es bien sabido que el uso de dispositivos electrónicos para detectar irregularidades que puedan amenazar la seguridad ha aumentado de manera significativa, por ejemplo, detectores de metales en entradas de lugares públicos como aeropuertos, parques de diversión, estadios; o la instalación de cámaras en circuitos cerrados con el fin de identificar personas por medio de reconocimiento facial. Así mismo, se incorporan sensores de movimiento o de proximidad en conjunto con alarmas para enviar alertas a los sectores encargados de la seguridad pública o privada de esos lugares. De esta forma, las herramientas son empleadas en la prevención de actos ilícitos, además, en este sentido, existen investigaciones que, utilizando esos recursos proponen esquemas de aprendizaje y monitoreo en línea para detectar comportamientos sospechosos en videos de vigilancia, lo que permite predecir el ataque de un eventual delincuente antes de que se produzca (Eduardo Francisco Caicedo Bravo et al., 2019); por otro lado, el aprendizaje en

línea es gradual sabiendo que se tiene que alimentar de manera autónoma con videos en tiempo real. En este punto se abre una brecha donde se puede obtener información errónea, por ejemplo, confundir una escena de robo con otra que presenta solo un juego brusco entre amigos.

Existe también un modelo de aprendizaje profundo de extremo a extremo que se basa en el aprendizaje recurrente bidireccional (BiGRU) y una red neuronal convolucional (CNN por sus siglas en inglés) para detectar y prevenir actividades delictivas. La CNN extrae las características espaciales de los fotogramas de video, mientras que BiGRU extrae las características de movimiento temporal y local de múltiples fotogramas de la CNN. Este modelo de aprendizaje es capaz de detectar algunos eventos criminales con una eficacia de 98.86% utilizando el conjunto de datos CAVIAR (Gandapur, 2022). Esto demuestra que las cámaras de videovigilancia en conjunto con algoritmos desarrollados con aprendizaje profundo son capaces de predecir y prevenir delitos detectando las condutas o comportamientos delictivos.

En estas condiciones se aprecia la importancia de la presente investigación dirigida a la prevención de conductas delictivas que pueden derivar en el robo de autopartes usando aprendizaje supervisado; para lograrlo, se tienen videos de diferentes fuentes, como YouTube y cámaras de seguridad particulares, que muestran escenas de robos de autopartes para analizar e identificar el comportamiento inusual de un individuo y de esta manera ofrecer una herramienta que ayude a combatir la inseguridad en esta área que actualmente se vive en el país. Esta investigación se inserta en la línea de generación y aplicación del conocimiento de la Inteligencia Artificial (IA).

1.2 Planteamiento del problema

En la actualidad, la tecnología basada en inteligencia artificial, también se utiliza ampliamente para mejorar la seguridad en diversos entornos, brindando herramientas como YOLO y DeepStream para videovigilancia en tiempo real, así como FaceNet y Mediapipe para reconocimiento facial. Estas tecnologías, junto con OpenCV y modelos de aprendizaje profundo, permiten detectar comportamientos sospechosos y disuadir la actividad delictiva,

optimizando la seguridad física y la respuesta ante incidentes. A pesar de esto, en algunos casos estos dispositivos no resultan suficientes para prevenir el delito, ya que el uso manual provoca un retraso en la identificación de conductas delictivas, lo que dificulta la labor de las autoridades encargadas de garantizar la seguridad pública.

El caso del robo de autopartes en México es una preocupación importante tanto para los dueños de vehículos como para las autoridades de seguridad. A pesar de las medidas de prevención y seguridad implementadas, el problema persiste y no se conoce con certeza qué factores están impulsando este fenómeno. Además, el tiempo es un factor importante para identificar la conducta delictiva, ya que la substracción de autoparte se realiza en aproximadamente 20 segundos, un valor calculado tras analizar los videos del conjunto de datos que registran escenas de robo. Por lo tanto, identificar las conductas que culminen en el robo de autopartes, presenta una oportunidad para investigar y desarrollar nuevas herramientas y tecnologías en beneficio de millones de ciudadanos afectados por este problema.

1.3 Pregunta de investigación

¿Cómo pueden las CNN 3D ser utilizadas para detectar conductas delictivas en el robo de autopartes en sistemas de videovigilancia?

¿Cómo puede reconocer la CNN 3D comportamientos delictivos en escenarios con variaciones de iluminación y ángulos de cámara?

¿Cómo influirá la precisión de la detección de conductas delictivas en el robo de autopartes mediante CNN 3D?

1.4 Objetivos

1.4.1 Objetivo general

El objetivo de este proyecto es desarrollar un sistema basado en CNN 3D que, mediante el uso de aprendizaje supervisado y la clasificación de videos (Conducta delictiva y Conducta no delictiva), permita identificar conductas delictivas asociadas con el robo de autopartes.

Estas conductas representan una amenaza para el patrimonio de las personas y la implementación de este sistema contribuirá a proporcionar mayor seguridad a los ciudadanos y prevenir este tipo de delitos.

1.4.2 Objetivos específicos

- Analizar el funcionamiento de las técnicas de aprendizaje supervisado aplicadas a la identificación de conductas delictivas.
- Definir un método para la generación de un conjunto de datos para la detección de conductas delictivas en el robo de autopartes.
- Recolectar videos de diferentes fuentes, como YouTube y cámaras de seguridad particulares, que muestren escenas de robos de autopartes. Estos videos se utilizarán para crear un conjunto de datos que contendrá alrededor de 2000 videos que se clasificarán en dos tipos (Conducta delictiva y Conducta no delictiva), y cada video tendrá una duración aproximada de 10 segundos.
- Aprovechar la plataforma en la nube de Google Colab para ejecutar el algoritmo con el respaldo de recursos computacionales avanzados, como unidad de procesamiento de gráficos (GPU por sus siglas en inglés) y Unidad de procesamiento tensorial (TPU por sus siglas en inglés), con el objetivo de acelerar el proceso de entrenamiento de modelos de aprendizaje supervisado.
- Entrenar el algoritmo de clasificación de aprendizaje supervisado con videos para realizar la identificación de la conducta delictiva en el robo de autopartes.

- Evaluar el funcionamiento del sistema desarrollado mediante el uso de videos con escenas de robo y no robo de autopartes. De esta manera se validaría su eficacia con base en la correcta clasificación de los videos.

1.5 Delimitación

El sistema se realizará mediante la implementación de aprendizaje profundo usando CNN 3D y será capaz de identificar conductas delictivas específicamente en el robo de autopartes por medio de cámaras de videovigilancia. El sistema será desarrollado en el lenguaje de programación Python. Se obtendrán videos con escenas que presenten la conducta delictiva para posteriormente tomar la información espacial y temporal, la cual permitirá saber qué acción está realizando el individuo y así poder obtener los datos necesarios para su posterior análisis. Cabe mencionar que no se tiene acceso a videos oficiales de manera formal sobre el robo de autopartes, por lo cual se construirá un conjunto de entrenamiento (Dataset) de diferentes fuentes, por ejemplo www.youtube.com y cámaras de seguridad particulares, descartando cualquier otro tipo de delito o conducta que un individuo presente, quedando fuera de alcance del proyecto y por ende sin tratamiento, abriendo una línea para investigaciones futuras.

1.6 Hipótesis

La creación del sistema, fundamentado en el método de aprendizaje supervisado, con el uso de CNN 3D permite identificar incidentes delictivos, enfocados en el robo de autopartes, facultando la detección de una conducta delictiva de un individuo a través de una cámara de videovigilancia.

1.7 Justificación

En la actualidad, se enfrenta a un crecimiento preocupante de la inseguridad, lo cual genera inquietud en la mayoría de los habitantes. Todos corren el riesgo de convertirse en víctimas de la delincuencia, tanto dentro como fuera de nuestros hogares. “El incremento de

estos delitos coincide con la reactivación de algunas actividades económicas y la reducción de medidas de confinamiento o distanciamiento social durante el segundo semestre de 2020. En otras palabras, el incremento en el flujo de personas y la actividad económica crea un entorno favorable para que se produzcan este tipo de delitos." (Evalúa, 2021).

Por otra parte, es bien sabido que los seres humanos pueden detectar el comportamiento de una persona por medio de lenguaje no verbal, este incluye los comportamientos de las personas mostrados a través de gestos, movimientos, la mirada, las expresiones faciales (Carrera-Levillain & Fernandez-Dols, 1994), dando pauta a que con ayuda de la IA se pueda detectar ese comportamiento por medio de algoritmos que faciliten la predicción de dichas conductas.

Así, el presente trabajo, se centra en el desarrollo de un sistema que aprenda a identificar conductas delictivas para la prevención del robo de autopartes empleando aprendizaje supervisado. Específicamente, se entrena una CNN 3D para posteriormente realizar la identificación del comportamiento de un individuo y determinar si puede ser una conducta delictiva orientada al robo de autopartes. Para el entrenamiento, se creó un conjunto de videos que representen tanto escenas de robo de autopartes como videos que no tienen estas escenas. Luego, se clasifican estos videos en dos categorías: conducta delictiva y conducta no delictiva. De esta forma, se puede identificar la conducta delictiva y obtener resultados que permitan desarrollar estrategias para mejorar la seguridad y combatir el robo de autopartes. Este enfoque de clasificación mediante aprendizaje supervisado se considera adecuado para predecir y prevenir la actividad delictiva relacionada con el robo de autopartes.

1.8 Fundamentación inicial

En la actualidad, los desarrollos tecnológicos avanzan rápidamente, particularmente en el campo de las ciencias computacionales. Como resultado, en los últimos años se han perfeccionado sistemas basados en Inteligencia Artificial (IA por sus siglas en inglés) que pueden automatizar aspectos de la vida cotidiana de las personas. Además, estos sistemas han demostrado la capacidad de identificar conductas asociadas a actividades delictivas.

Entre las investigaciones más recientes (Martínez-Mascorro et al., 2020) proponen un método para obtener segmentos de video que sirven para entrenar una CNN 3D y clasificar un comportamiento sospechoso. Los videos con los que se alimenta la red son escenas de acciones diarias y muestras de robo en tiendas, teniendo un enfoque orientado a la prevención de la delincuencia. De igual forma describe una metodología basada en el método PCB por sus siglas en inglés (Pre-Crime Behavior), para unificar el procesamiento y la división de las muestras de vídeo delictivo en segmentos útiles que posteriormente pueden ser utilizados para alimentar la CNN 3D. Cabe mencionar que implementan el uso de la psicología combinada con el aprendizaje profundo para identificar las intenciones criminales cuando una persona muestre un comportamiento sospechoso.

En uno de los estudios más recientes se ha implementado el uso de Deep Learning para identificar el comportamiento de los seres humanos usando un modelo de aprendizaje preentrenado llamado VGG-16 (Visual Geometry Group), el modelo de aprendizaje se encuentra en desarrollo para predecir comportamientos humanos sospechosos en grabaciones de video para ayudar en el proceso de vigilancia (Amrutha et al., 2020). Esta predicción se realiza con base en las imágenes tomadas de los mismos videos.

En el contexto de guarderías, se ha implementado un enfoque innovador que utiliza el aprendizaje profundo para garantizar la seguridad de los niños y evitar posibles abusos por parte del personal (Vallathan et al., 2021). Este método se basa en el análisis de imágenes recopiladas de sistemas de vigilancia en red y mediante técnicas de aprendizaje profundo puede predecir la aparición de eventos anormales. Además, notifica a los usuarios sobre estos eventos en un entorno de Internet de las cosas (IoT). Para lograr esto, se utiliza un multiclasificador dentro de una red neuronal profunda, combinado con funciones de densidad del kernel con el fin de clasificar las secuencias de entrada de videos. La red neuronal profunda se usa para aprender y entrenar, la densidad del kernel se usa para agrupar y predecir, dando como resultado la predicción de la actividad.

Por otro lado, se encontraron artículos relacionados con la detección de patrones de comportamiento anómalos en sistemas de vigilancia dentro de un entorno universitario con base en el rendimiento de una red neuronal convolucional con memoria a corto y largo plazo (CNN-LSTM por sus siglas en inglés). Se observo que el trabajo funciona con la extracción

de imágenes usando CNN, la diferencia es que se utiliza adicionalmente LSTM. Los experimentos se realizaron con el conjunto de datos de la Universidad de California en San Diego utilizando el sistema de detección de patrones de comportamiento anómalo, dichos resultados muestran que el sistema obtuvo un 86% de exactitud (Esan et al., 2020).

En el artículo (Kirichenko et al., 2022), también se aborda el problema de los robos en tiendas utilizando grabaciones de video que se clasifican y procesan mediante una red neuronal híbrida, combinando redes neuronales convolucionales y redes neuronales recurrentes. La red convolucional se utilizó para extraer características de los fotogramas de vídeo. La red recurrente procesó la secuencia temporal de las características de los fotogramas de vídeo y clasificó los fragmentos de vídeo. Se usó el conjunto de datos UCF-Crime para formar los conjuntos de datos de entrenamiento y prueba. Los resultados de clasificación mostraron una exactitud del 93% siendo superior a la de los clasificadores considerados en esa revisión.

Otro estudio propone un modelo de aprendizaje entrenado con el conjunto de datos VOC2012 que consiste de fotografías macro para la detección de objetos. Haciendo uso de una CNN obtiene resultados de 50.2% de exactitud, con el método FA-SVM (Máquina de soporte vectorial con algoritmo firefly). Cabe mencionar que la evaluación se implementó con imágenes binarias y de objetos múltiples (Kuppusamy & Hung, 2021).

Por otra parte, se encontraron trabajos que proponen una solución para detectar conductas delictivas, por ejemplo (Chackravarthy et al., 2018), se basaron en redes neuronales diseñadas con el algoritmo Hybrid Deep Learning (HDL) que utiliza fragmentos de video (frames) para modelar patrones de comportamiento. Trabajando en conjunto con una Red Neural Convolucional Profunda (DCNN) aprende a identificar objetos y personas. Con la combinación de estas tecnologías y modelos de aprendizaje, lograron un sistema preciso de clasificación para la detección de delitos.

Por si fuera poco (Navalgund & Priyadharshini, 2018), desarrollaron un modelo de aprendizaje profundo preentrenado VGGNet-19 capaz de detectar por medio de cámaras de circuito de televisión (CCTV), si una persona porta arma, y con ayuda de Redes Neuronales Convolucionales Recurrentes Rápidas (Fast RCNN por sus siglas en inglés) y Redes Neuronales Convolucionales Recurrentes (RCNN por sus siglas en inglés), se puede dibujar

un cuadro delimitador de imágenes tales como personas y objetos (pistolas, cuchillos y algunos objetos no entrenados). Permitiendo la detección y clasificación de estos elementos.

En el trabajo de (Kuppusamy & Bharathi, 2022), se presenta un sistema que registra los incidentes que contienen los patrones de diversos comportamientos humanos, y verifica videos para identificar un incidente manualmente, sin embargo, este proceso lleva mucho tiempo ya que necesita un sistema de automatización para procesar videos largos. Para ello se utilizan CNN 3D donde se observa que se trabaja de mejor manera que con aprendizaje automático. La red alcanzó una precisión del 87.5%, con una tasa de recall del 85.2% y un F1-score del 86.3%, superando SVM y Modelos Ocultos de Markov (HMM por sus siglas en inglés).

1.9 Publicaciones derivadas de este trabajo

- **Ponencia:** "Creación de un conjunto de datos para la detección de conductas delictivas asociadas con el robo de autopartes", 2023 12vo. Congreso Internacional de Matemáticas de Procesos de Software (CIMPS) – 19 de octubre 2023.

- **Artículo: B. D. Martínez Contreras, S. J. Morales Escobar, V. M. Landassuri Moreno, A. López-Chau**, "Creación de un conjunto de datos para la detección de conductas delictivas asociadas con el robo de autopartes", 2023 12vo. Congreso Internacional de Matemáticas de Procesos de Software (CIMPS) | ISBN:979-8-3503-5856-8/23/©2023 IEEE | DOI: 10.1109/CIMPS61323.2023.10528822.

- **Artículo: B. D. Martínez Contreras, S. J. Morales Escobar, V. M. Landassuri Moreno, A. López-Chau**, "Overcoming Obstacles: Training a 3DCNN to Detect Car Part Theft in Video". Enviado a la revista Computación y Sistemas (CyS). En revisión.

1.10 Organización del capitulado

La estructura del trabajo de esta tesis es descrita a continuación:

En el Capítulo 2, se presenta el marco teórico relacionado con el lenguaje no verbal y las conductas delictivas, centrándose en la identificación e interpretación del comportamiento de un individuo. Además, se presentan técnicas empleadas en el problema de clasificación supervisada, con énfasis en los principios fundamentales de las redes neuronales artificiales, perceptrón simple, así como en el estudio de las redes neuronales recurrentes, redes neuronales convolucionales y CNN 3D.

En el Capítulo 3, se ofrece la metodología y desarrollo del sistema, se presentan las herramientas utilizadas y se explica la configuración del algoritmo empleado para la clasificación de conductas delictivas, centrándonos en un caso de estudio específico: el robo de autopartes. De esta forma en el Capítulo 4, se muestran los resultados obtenidos, el desenlace del desarrollo del sistema. Finalmente, en el Capítulo 5, se ofrecen las conclusiones y propuestas para el trabajo futuro.

CAPÍTULO 2

2 Marco Teórico

2.1 Lenguaje no verbal

El lenguaje no verbal se refiere a la forma de comunicación que no utiliza palabras habladas o escritas. Engloba gestos, expresiones faciales, posturas corporales, tono de voz y contacto visual. A través del lenguaje no verbal se pueden transmitir información, emociones, actitudes e intenciones, y puede complementar o contradecir el lenguaje verbal. Por ejemplo, las expresiones faciales revelan emociones, los gestos transmiten significados y la postura corporal comunica confianza o tensión (Carrera-Levillain & Fernandez-Dols, 1994) . El lenguaje no verbal es fundamental en la comunicación humana y su comprensión mejora la interpretación de los mensajes y la interacción entre las personas, en la figura 2.1, se representan 3 lenguajes no verbales.

Figura 2.1. Representación de lenguajes no verbales. (Comunicación no verbal, 2023)

2.2 Conductas delictivas

En (Piquero et al., 2003), se introduce el concepto de la "carrera delictiva", que se enfoca en cómo la participación en actividades delictivas puede cambiar y evolucionar a lo largo de la vida. Examina los diferentes patrones de comportamiento delictivo de los criminales, así

como los factores de riesgo que pueden influir en este tipo de participación. En recapitulación el artículo aborda la participación de personas en actividades delictivas como:

- Delitos violentos: homicidio, asalto, robo con violencia y violación.
- Delitos contra la propiedad: robo, hurto, robo de vehículos y allanamiento de morada.
- Delitos relacionados con drogas: tráfico y posesión de drogas.

Estas pueden variar dependiendo el curso del tiempo y una determinada influencia en la actividad criminal.

2.3 Lenguaje no verbal en conductas delictivas

El lenguaje no verbal puede manifestarse en el contexto de conductas delictivas. Sin embargo, es importante tener en cuenta que el lenguaje no verbal no constituye una indicación concreta de una conducta delictiva en sí misma. En cambio, ciertos comportamientos no verbales pueden sugerir actitudes o intenciones relacionadas con actividades delictivas. Algunos ejemplos de estos comportamientos incluyen:

- Evitar el contacto visual: Se presenta con desvíos de la mirada, evitan el contacto visual o mirar hacia abajo.
- Comportamiento nervioso: Indica ansiedad o nerviosismo, jugueteo con las manos, dedos e inquietud.
- Movimientos de manipulación: Gestos como tocarse la cara, frotarse las manos o llevarse las manos a la boca pueden indicar que una persona esta nerviosa y pueda presentar un acto delictivo.
- Distancia: La manera en que las personas utilizan y ocupan el espacio, así como la distancia que mantienen al comunicarse entre sí.
- Manera: El modo en que se realiza un gesto puede ser consciente o inconsciente y se aprende socialmente según cada contexto.

Es crucial tener en cuenta que estos comportamientos no verbales son indicios y no constituyen pruebas concluyentes de conducta delictiva (Duarte Duarte, 2022). La

interpretación del lenguaje no verbal debe complementarse con otras pruebas y circunstancias para obtener una comprensión completa de la conducta.

2.4 Aprendizaje automático

El aprendizaje automático (ML por sus siglas en inglés), es una rama de la inteligencia artificial, en gran parte inspirada en la deducción humana, este se clasifica generalmente en tres tipos principales según las características y el tipo de datos empleados durante el proceso de entrenamiento (Sandoval, 2018). Para comprender el aprendizaje automático, resulta fundamental familiarizarse con la clasificación de los distintos tipos de problemas que se detallan a continuación:

- Aprendizaje supervisado: En este tipo de problemas, se instruyen los algoritmos sobre cómo llevar a cabo su tarea mediante un conjunto de datos previamente clasificados según ciertos criterios o conceptos de especialistas en el dominio del problema. El objetivo es identificar patrones que puedan ser aplicados en un análisis (Mueller & Massaron, 2021) y generar una salida a partir de las que ya conocidas.
- Aprendizaje no supervisado: En estos problemas se implica la formación de un modelo de aprendizaje predictivo de manera análoga al aprendizaje supervisado. La distinción radica en que la comprensión se desarrolla a partir de datos no clasificados o etiquetados, revelando patrones de similitud entre diferentes grupos de datos (Bishop, 2006).
- Aprendizaje híbrido: Es una combinación de los dos previos, donde algunas capas adoptan un método de aprendizaje supervisado, mientras que otras capas se rigen por un enfoque de tipo no supervisado (Russo et al., 2016) .
- Aprendizaje reforzado: Este tipo de aprendizaje automático no implica entrenamiento con datos clasificados. En lugar de ello, el sistema aprende en un entorno donde no dispone de información sobre la posible salida. Este método se basa en acciones y los resultados obtenidos, como destaca (López Boada et al., 2005). Además, el modelo de aprendizaje se refuerza con un agente que, a través de prueba y error, ajusta su

estrategia para maximizar la recompensa acumulada, utilizando modelos como los procesos de decisión de Markov (MDP por sus siglas en inglés).

Los métodos de aprendizaje supervisado para la detección de objetos, clasificación de acciones, segmentación de imágenes y objetos, se basan en datos de entrenamiento etiquetados, donde cada muestra tiene una clase conocida. A continuación, se destacan los más utilizados en este contexto (Martínez et al., 2022):

1. Redes neuronales artificiales (RNA)
2. Redes neuronales recurrentes (RNN)
3. Redes de memoria a largo y corto plazo (LSTM)
4. Redes neuronales convolucionales (CNN)
5. Redes Combinadas (CNN + LSTM)
6. Redes neuronales convolucionales profundas (DCNN)
7. Redes neuronales convolucionales 3D (CNN3D)

Estas técnicas han demostrado ser positivas en el procesamiento y análisis de videos, pueden aprender patrones complejos y mejorar su capacidad en nuevos escenarios, siendo las mejores para la extracción de características espaciales permitiendo capturar secuencias de video temporales.

2.4.1 Redes neuronales artificiales (RNA)

Una RNA, es un modelo de aprendizaje matemático inspirado en el comportamiento biológico de una neurona real caracterizadas por el aprendizaje a través de la experiencia y la extracción de conocimiento genérico a partir de un conjunto de datos (Díez et al., 2001), además (Ivan et al., 2013) afirman que las RNA son una familia de técnicas de procesamiento de información inspirado por la forma de procesar información del sistema nervioso biológico de un ser vivo, tratando de “emular el comportamiento del cerebro”. La estructura biológica de la red neuronal, se ilustra en la figura 2.2.

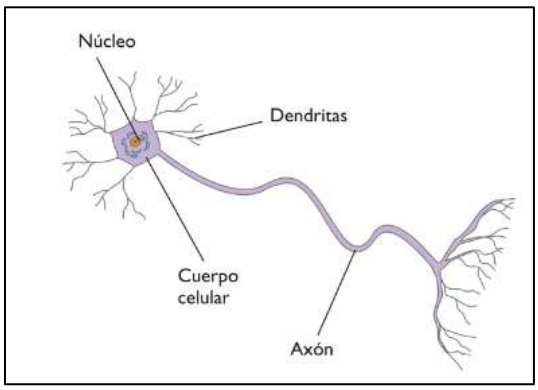

Figura 2.2. Estructura de una neurona biológica.
(Díez et al., 2001)

La RNA, cuyo modelo de aprendizaje profundo puede verse en la figura 2.3, se utiliza en aprendizaje supervisado, están organizadas en capas las cuales juegan un papel esencial al organizar y estructurar de manera significativa la información.

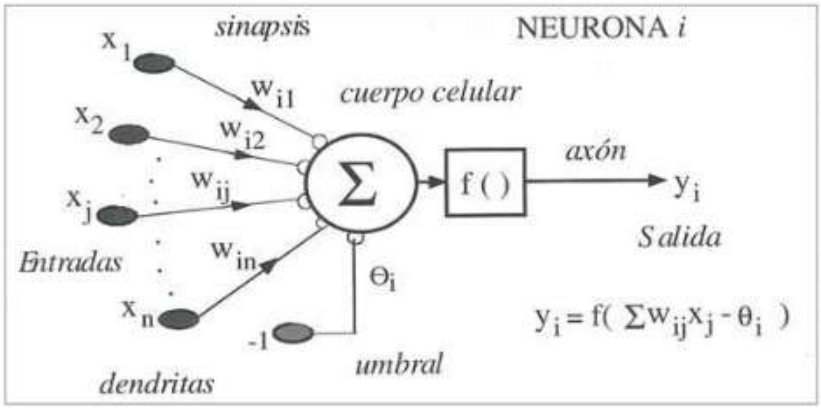

Figura 2.3. Modelo para una neurona artificial. (Nacelle & Mizraji, 2009)

Los símbolos como $\boldsymbol{x_{ij}}$, $\boldsymbol{w_{ij}}$, $\boldsymbol{y_i = f(x)}$ muestran los siguientes significados relacionados con el procesamiento de datos y el aprendizaje.

- w_{ij} (Peso o Weight): Define la importancia de cada entrada en la neurona. Se ajusta durante el entrenamiento mejorando la precisión del modelo (LeCun et al., 2015a).
- x_{ij} : entradas de la red neuronal.

- $y_i = f(x)$ (Función de activación): Función matemática aplicada a la suma ponderada para introducir no linealidad en la RNA. Las funciones más usadas son ReLU, sigmoide y tangente hiperbólica (LeCun et al., 2015a).

$$y_i = f(\sum w_{ij} \cdot x_{ij} - \theta_i)$$

La RNA está compuesta por tres capas principales: la capa de entrada, la capa oculta y la capa de salida, las cuales se describen a continuación.

- Capa de entrada (Input Layer): La capa iniciales la encargada de recibir los datos de entrada. Cada nodo en esta capa representa una característica de entrada específica. La cantidad de nodos en esta capa suele encajar generalmente con la cantidad de características presentes en los datos de entrada (Nacelle & Mizraji, 2009).
- Capas ocultas (Hidden Layers): La capa de entrada y la capa de salida **no están directamente conectadas entre sí**; en su lugar, la información pasa a través de las capas ocultas. En cada nodo de las capas ocultas, se llevan a cabo cálculos basados en los datos de entrada, y reciben la información de la capa de entrada a través de las conexiones y pesos asignados (w_i). La existencia de múltiples capas ocultas habilita a la red para adquirir representaciones más refinadas y abstractas de los datos (Díez et al., 2001).
- Capa de salida (Output Layer): Es la encargada de generar el resultado o la predicción de la red. La cantidad de nodos en esta capa suele alinearse con la cantidad de clases o variables a predecir que se definieron en el diseño de la red (López & Fernández, 2008).

En la figura 2.4, se muestra la estructura de una red neuronal artificial, incluyendo las capas de entrada, las capas ocultas y las capas de salida.

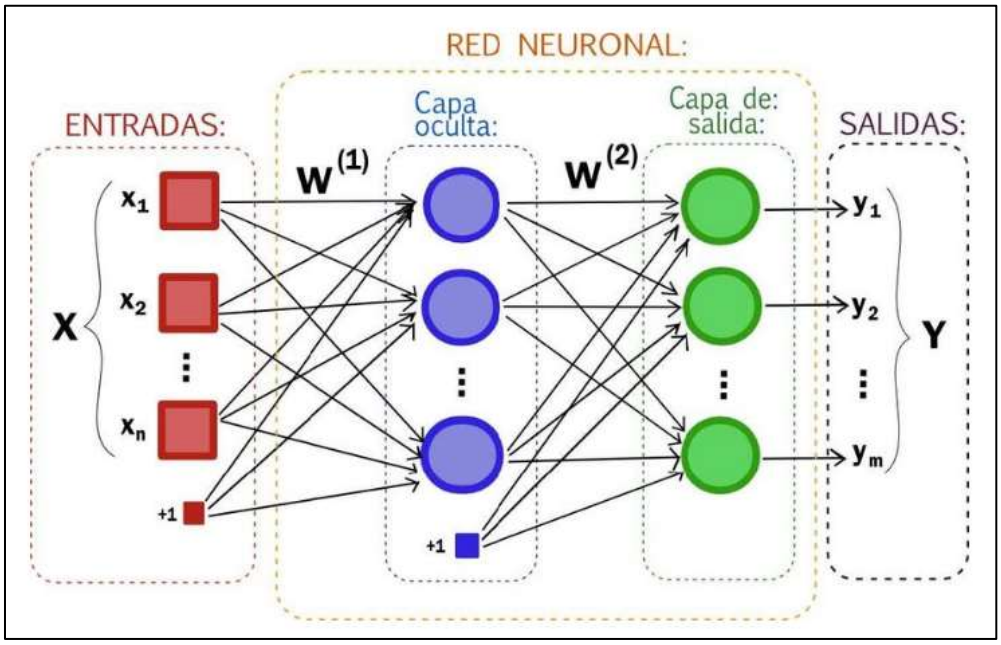

Figura 2.4. Red neuronal artificial. (Díez et al., 2001)

Las RNA tienen la capacidad de aprender patrones y características a partir de un conjunto de datos de entrenamiento. El funcionamiento de una RNA se logra interconectando nodos y estableciendo relaciones entre ellos, la información fluye desde la capa de entrada a través de las capas ocultas y finalmente llega a la capa de salida. Durante el entrenamiento de una RNA, se ajustan los pesos de las conexiones entre las neuronas para que la red pueda aprender a reconocer patrones y realizar predicciones precisas. Esto se logra mediante algoritmos de optimización que buscan minimizar la diferencia entre las salidas reales y las salidas deseadas (Nacelle & Mizraji, 2009). Una vez entrenadas, estas redes pueden aplicar ese conocimiento para realizar predicciones en nuevos conjuntos de datos.

2.4.1.1 Perceptrón simple (SLP)

El SLP (por sus siglas en inglés, Single Layer Perceptron), es el modelo matemático más elemental de una neurona, captura la esencia de esta célula especializada. La neurona, con dendritas que funcionan como sensores y un axón como canal de salida. La neurona recoge información de su entorno a través de las dendritas, transmitiéndola al cuerpo de la neurona.

La respuesta de la neurona se produce mediante una sinapsis, enviando una señal al cerebro (Ramírez, 2018). En la figura 2.5, se muestra la representación de un perceptrón simple.

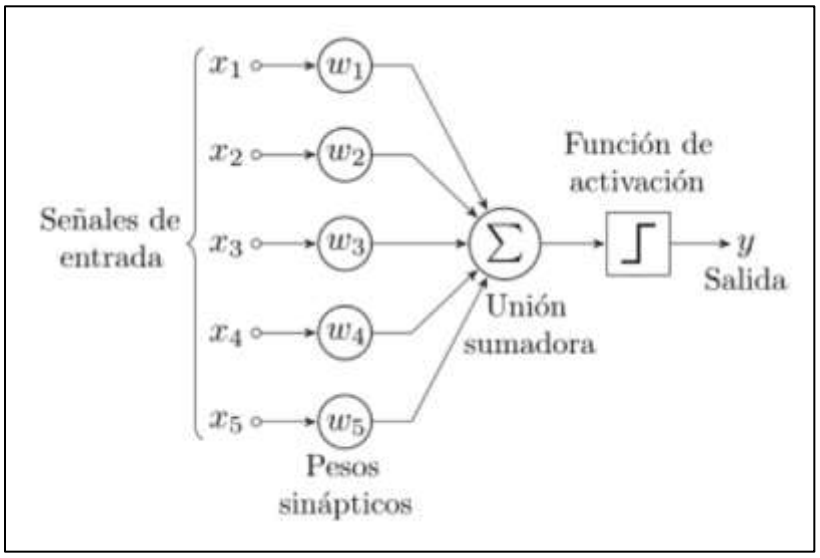

Figura 2.5. Perceptrón simple. (Munt, 2018)

El SLP tiene la capacidad de discriminar únicamente entre dos clases que pueden ser separadas por una línea recta o un hiperplano, esto es, se divide el espacio en dos regiones distintas que corresponden a dos clases diferentes de patrones (Larranaga et al., 1997). En este caso se dice que la frontera de decisión entre las clases es lineal. Véase un ejemplo en la figura 2.6. Se aprecia que, al aplicar una condición lineal, se separa el espacio en dos regiones distintas que corresponden a dos clases diferentes.

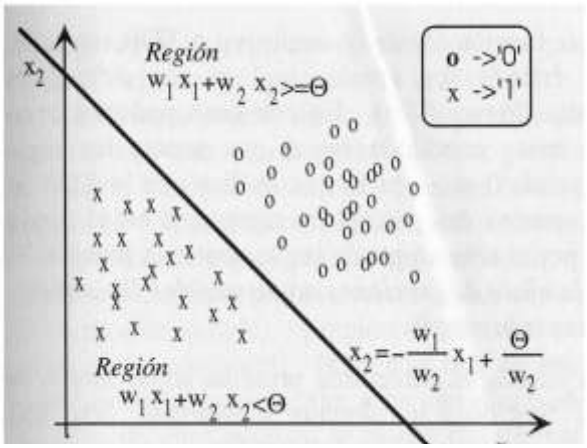

Figura 2.6. Región de decisión correspondiente a un perceptrón simple con dos neuronas de entrada. (Larranaga et al., 1997)

La estructura del SLP se muestra en la figura 8, la cual consiste de:

- Entradas (x_1, x_2, ..., x_n): Los datos de entrada.
- Pesos (W_1, W_2, ..., W_n): Los coeficientes asociados a cada entrada.
- Sesgo (θ): Un valor constante añadido para ajustar la función de activación.
- La unión sumadora de entradas y pesos, más el sesgo, se calcula como:

$$y_i = f\left(\sum_{j=1}^{n} x_{ij} w_{ij} - \theta_i\right)$$

Donde:

y_{ij} Suma ponderada de las entradas.

x_{ij} Entradas.

W_i Pesos correspondientes a las entradas.

θ Sesgo (bias).

f Función de activación.

- Función de activación: La función de activación convierte la suma ponderada en una salida binaria. Las funciones de activación comúnmente usadas en un perceptrón simple son la función escalón mostrada en la figura 2.7 y la función sigmoide mostrada en la figura 2.8, ambos modelos de aprendizaje no lineales (Izaurieta & Saavedra, 2000).

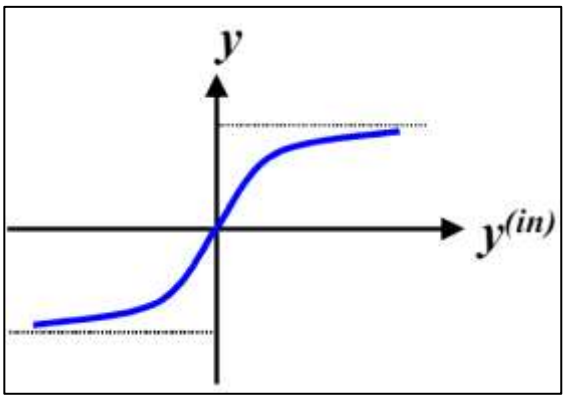

Figura 2.7. Función Sigmoidea.
(Izaurieta & Saavedra, 2000)

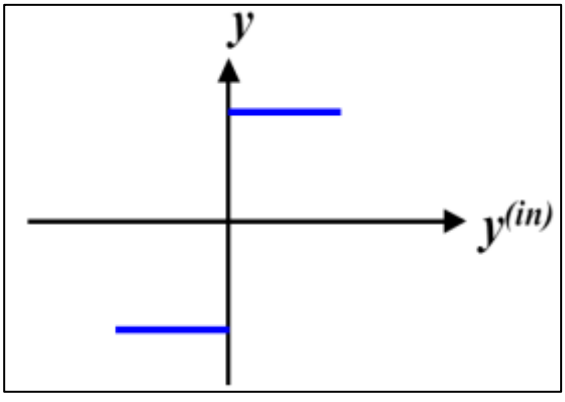

Figura 2.8. Función Escalón.
(Izaurieta & Saavedra, 2000)

Dado que x_1 y x_2 representan las neuronas de entrada, la operación realizada por el perceptrón simple consiste en:

$$y = \begin{cases} 1 \; si \; w_1 x_1 + w_2 x_2 \; \geq 0 \\ 0 \; si \; w_1 x_1 + w_2 x_2 \; < 0 \end{cases}$$

Si tomamos x_1 y x_2 como las coordenadas en los ejes de abscisas y ordenadas respectivamente, la condición es equivalente a:

$$w_1 x_1 \; + w_2 x_2 - \theta = 0$$

$$x_2 = \frac{w_1}{w_1} x_1 + \frac{\theta}{w_2}$$

Esta línea define la región de decisión establecida por el perceptrón simple, actuando como un discriminador lineal (Larranaga et al., 1997).

2.4.1.2 Perceptrón multicapa (MPL)

El MLP (por sus siglas en inglés, Multi-Layer Perceptron). Este modelo de aprendizaje se compone de una capa de entrada, varias capas ocultas y una capa de salida, como se muestra en la figura 2.9. Cada capa está formada por neuronas que reciben, procesan y transmiten datos a otras neuronas, utilizando diversas funciones matemáticas para procesar la información (Bravo, 2009). La estructura presenta una arquitectura de tipo propagación hacia delante (en inglés, Forward Propagation) donde las entradas se propagan hacia adelante a través de las capas ocultas hasta la capa de salida.

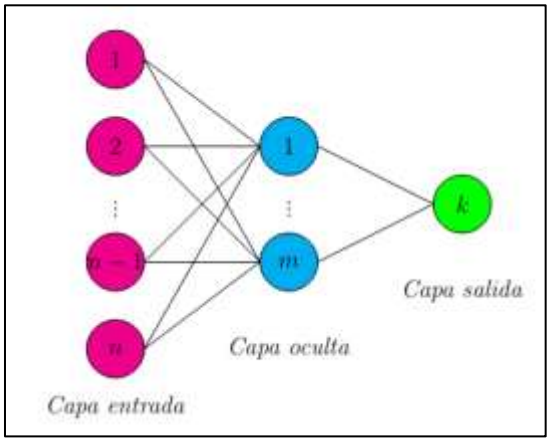

Figura 2.9. Perceptrón Multicapa (MPL). (Vivas,2014)

La estructura del MPL consta de:

- Capa de Entrada: Captura los datos iniciales.
- Capas Ocultas: Una o más capas donde se realiza el procesamiento.
- Capa de Salida: Produce el resultado final de la red.
- Funciones de Activación: Se emplean funciones como ReLU (Rectified Linear Unit) en las capas ocultas y softmax o sigmoide en la capa de salida, según el tipo de tarea (Heaton, 2018).
- Cálculo de Pérdida y Optimización: Se determina una función de pérdida (como el error cuadrático medio para problemas de regresión o la entropía cruzada para problemas de clasificación) y se minimiza utilizando algoritmos como el descenso de gradiente (Rumelhart et al., 1986).

2.4.2 Red neuronal recurrente (RNN)

Las redes neuronales recurrentes (RNN) son una clase de modelos de aprendizaje profundo que se basan en los trabajos de David Rumelhart de 1986. Las RNN resaltan por su capacidad para procesar y extraer información de datos secuenciales y temporales, siendo ideales para el análisis de vídeo, generación de subtítulos para imágenes, procesamiento del lenguaje natural, entre otros. A diferencia de las redes neuronales tradicionales que dependen de los datos de entrada, las RNN capturan y mejoran las dependencias secuenciales y temporales presentes en los datos (Arana, 2021).

La RNN no es un modelo de aprendizaje neuronal clásico de avance hacia adelante, donde las neuronas transmiten señales únicamente a las capas siguientes. Las RNN, en cambio, incluyen conexiones repetitivas entre sus nodos, lo que les permite extender información tanto a las capas siguientes como propias, permitiendo utilizar la información para realizar predicciones futuras y modelar datos secuenciales de video (Guardia Vaca & Sandoval Alcocer, 2018). Una RNN es una neurona A, que procesa la entrada de datos x_t y genera un valor de salida h_t . Repitiéndose interiormente, transfiriendo la información a

la siguiente red de manera periódica. En la figura 2.10, se puede observar la estructura de una RNN (Powell González, 2021).

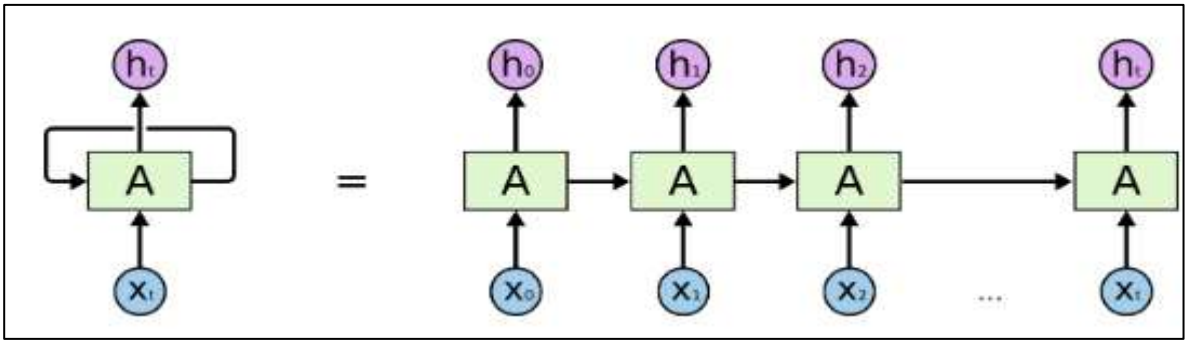

Figura 2.10. Red neuronal recurrente (RNN). (Powell González, 2021)

2.4.3 Redes de memoria a largo y corto plazo (LSTM)

Las LSTM es derivada de las RNN, estructurada para modelar una sucesión de datos a largo plazo, fueron incorporadas por Hochreiter y Schmidhuber, las LSTM fueron construidas para resolver problemas como el desvanecimiento y mejora del gradiente, siendo difícil para las RNN realizar el aprendizaje a largo plazo (Hochreiter & Schmidhuber, 1997). La unidad básica de una red LSTM es la memoria de bloque, que contiene una o más celdas de memoria y un par de células adaptativas, cada celda de memoria tiene en su núcleo una línea auto conectada de forma recurrente llamada carrusel de error constante (CEC), (Graves & Schmidhuber, 2005), véase en la figura 2.11, la estructura de una Red de LSTM.

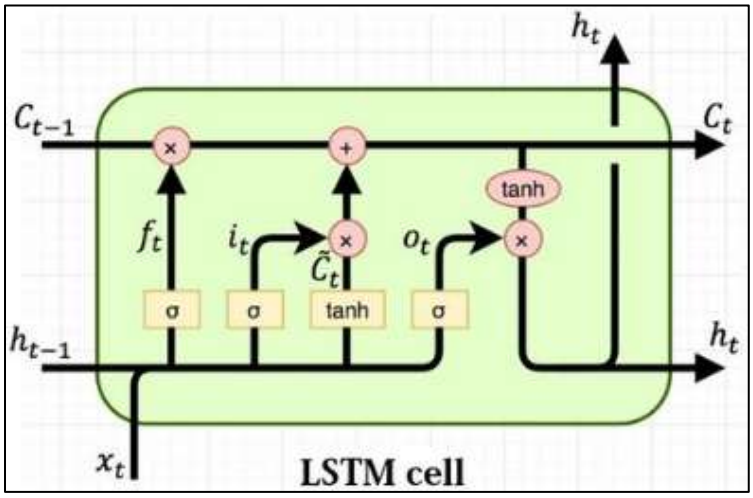

Figura 2.11. Red de LSTM. (Varsamopoulos et al., 2018)

La estructura de una LSTM consta de:

- Puerta de olvido (en inglés, Forget Gate) decide qué información de la celda debe ser descartada C_{t-1}, . El valor se fundamenta en la entrada actual x_t y el estado oculto previo h_{t-1}, (Gers et al., 2000):

$$f_t = \sigma\ (w_f \cdot [\ h_{t-1}, x_t\ ] + \ b_f\ )$$

donde:

- f_t Vector de la puerta de olvido.
- w_f Pesos de la puerta de olvido.
- b_f Sesgo de la puerta de olvido.
- σ Función sigmoide.

- Puerta de entrada (en inglés, Input Gate) indica que la nueva información sea agregada al estado de la celda, el proceso consta de dos pasos que son calcular el vector de la puerta de entrada i_t y la generación del contenido candidato de la celda $\hat{C}_t$ (Gers et al., 2000):

$$i_t = \sigma\ (\ w_i\ [\ h_{t-1}, x_t\ ] + \ b_i\ )$$

$$\hat{C}_t = tanh\ (\ w_c\ [\ h_{t-1}, x_t\ ] + \ b_c\ )$$

donde:

- i_t Vector de la puerta de entrada.
- w_i Pesos de la puerta de entrada.
- b_i Sesgo de la puerta de entrada.
- $\hat{C}_t$ Nuevo contenido para la celda.

- w_c Pesos para el nuevo contenido de la celda.
- b_c Sesgo para el nuevo contenido de la celda.
- $tanh$ Función tangente hiperbólica.

- Estado de la celda C_t se actualiza mezclando la información que se debe olvidar y agrega la nueva información (Gers et al., 2000):

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

- Puerta de salida **(**en inglés, Output Gate) toma la decisión de que parte del estado de la celda se va a utilizar para la salida actual. La salida se depura por una función de tangente hiperbólica (Gers et al., 2000):

$$o_t = \sigma \left(w_o \left[h_{t-1}, x_t \right] + b_o \right)$$

$$h_t = o_t * tanh \left(C_t \right)$$

donde:

- o_t Vector de la puerta de salida.
- w_o Pesos de la puerta de salida.
- b_o Sesgo de la puerta de salida.

Las LSTM son una herramienta poderosa para modelar secuencias de datos a largo y corto plazo. Su diseño permite manejar el flujo de información de manera eficiente, haciéndolas útiles para realizar la detección de peleas en videos, poses de personas o detección de lenguaje natural (Powell González, 2021).

2.4.4 Redes neuronales convolucionales (CNN)

Una red neuronal convolucional (CNN) es una de red neuronal diseñada para detectar imágenes y patrones visuales de forma eficiente, han mostrado ser efectivas en la clasificación de imágenes, la detección de objetos y el reconocimiento facial (LeCun et al., 1998).

En la figura 2.12, se aprecia la arquitectura básica de una red neuronal convolucional, donde la entrada es una imagen a color RGB, conocida comúnmente de 3 canales de 254x254 pixeles seguido de las capas básicas de una (CNN) incluyen las capas de convolución, las capas de agrupación, las capas de activación, las capas completamente conectadas y la capa de salida (Vizcaya Cárdenas et al., 2017).

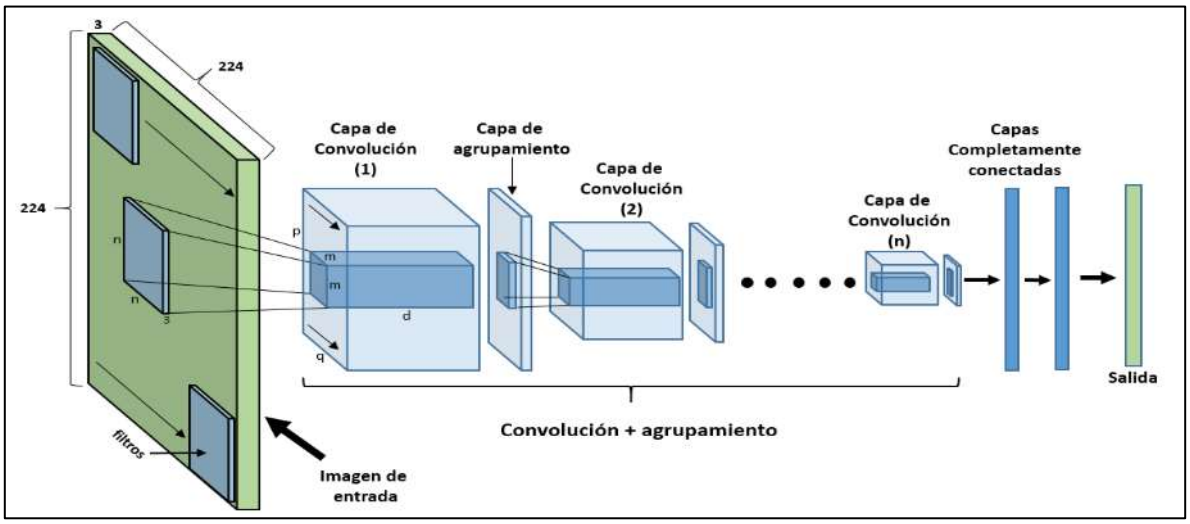

Figura 2.12. Arquitectura típica de una CNN. (Vizcaya Cárdenas et al., 2017)

La estructura de una CNN se describe a continuación:

- Capas de convolución, consisten en tomar un conjunto de píxeles de la imagen de entrada e ir operando matemáticamente contra una matriz más pequeña llamada kernel y conocida comúnmente como filtro (Ayora, 2024).

El kernel recorre todas las neuronas de entrada de izquierda a derecha y de arriba hacia abajo y genera una nueva matriz de salida, siendo la nueva capa de neuronas ocultas conocida como matriz de activación. En la imagen de entrada se debe rellenar los bordes con ceros,

para ajustar al tamaño de los filtros y se pueda hacer el barrido completo, en la figura 2.13, se muestra la capa de convolución (Vizcaya Cárdenas, 2017) .

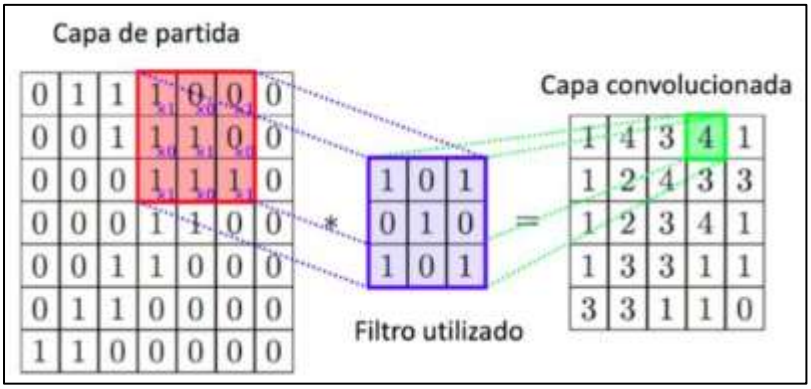

Figura 2.13. Capa de convolución "Filtro o Kernel". (Ayora, 2024)

- Capas de agrupamiento, reducen la dimensión de los mapas de características, capas de entrada o imágenes de entrada, conservando la información más relevante, se realiza un muestreo cromático (en inglés, Subsampling) el más utilizado es Max-pooling.
- Capas de activación, después de cada operación convolucional, se aplica una función de activación, la unidad lineal rectificada (en inglés, Rectified Linear Unit (ReLu)) es la función de activación más frecuente en modelos de aprendizaje profundo (LeCun et al., 1998). La función regresa 0 si recibe una entrada negativa, y para cualquier valor positivo x devuelve el mismo valor, véase en la figura 2.14.

$$f(x) = \max(0, x)$$

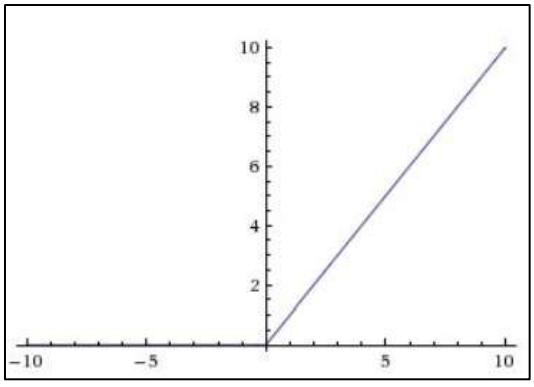

Figura 2.14. Gráfica Función ReLu.
(Krizhevsky et al., 2017)

- Capas completamente conectadas, Estas capas procesan las características extraídas y las combinan para hacer una predicción. Estas capas actúan en la clasificación de imágenes (Krizhevsky et al., 2017).
- Capa de salida, es el resultado del procesamiento con la CNN, dando la probabilidad de obtener un etiquetado correcto por cada clasificación.

Otros conceptos fundamentales de las CNN son:

- Max-pooling, es una técnica de reducción de dimensionalidad que reduce las dimensiones espaciales de los mapas de características producidos por las capas convolucionales. En la figura 2.15, se muestra una matriz donde se aplica el Max-pooling de un tamaño de 2x2 recorriendo una imagen de izquierda a derecha y arriba-abajo, tomando 2x2 pixeles preservando el valor más alto, reduciendo la imagen a la mitad (Ayora, 2024).

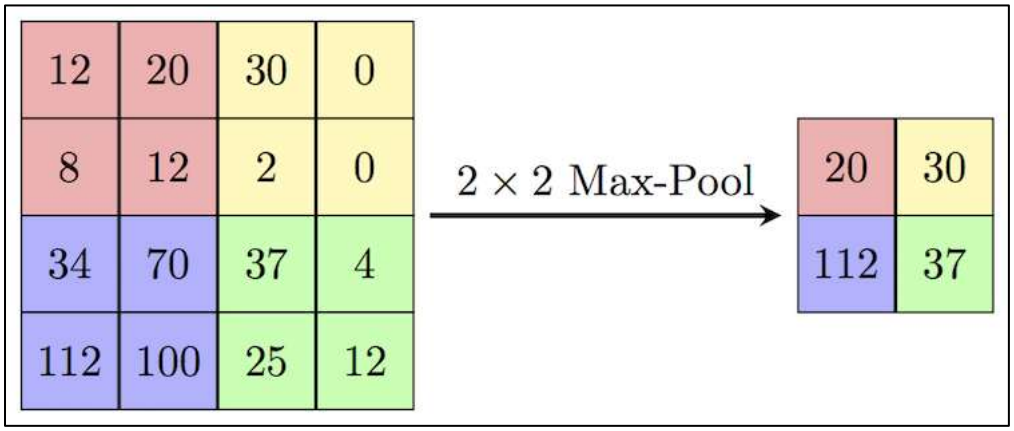

Figura 2.15. Max-pooling. (Ayora, 2024)

- Retropropagación (en inglés, Backpropagation), ajusta los pesos minimizando la función de pérdida a través de la propagación del error, desde la capa de salida hasta las capas de entrada, en la figura 2.16 muestra, como actúa el backpropagation realizando el cálculo de gradiente de la función de pérdida con respecto a los pesos en la red, ajustando los pesos para minimizar la pérdida (Srivastava et al., 2024).

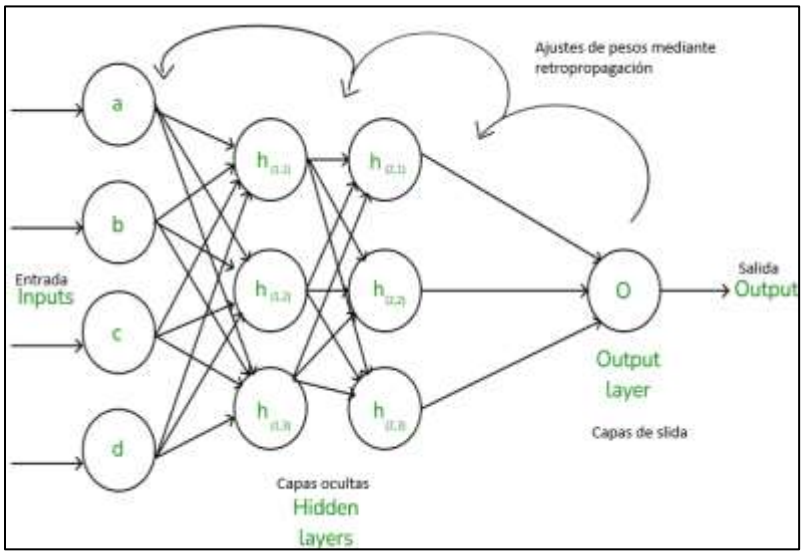

Figura 2.16. Backpropagatio. (Lillicrap et al., 2020)

Abandono (en inglés, Dropout) es un método que se usa en la fase de entrenamiento de la red, donde se desactiva un número definido de neuronas en forma aleatoria. Las neuronas desactivadas no se toman en cuenta para la propagación hacia delante ni para atrás, por ello las neuronas no dependen de las neuronas desactivadas más cercanas a ellas, ayudando a reducir el sobreajuste (en inglés, Overfitting), Durante el entrenamiento, las neuronas seleccionadas se desactivan usando todas las neuronas ajustando los pesos (Khalifa & Frigui, 2016). En la figura 2.17, se muestra la representación del Dropout.

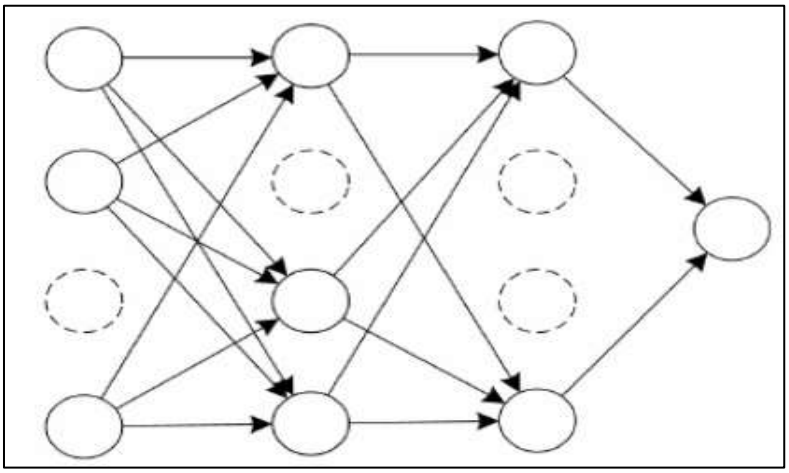

Figura 2.17. Dropout. (Khalifa & Frigui, 2016)

- Sobreajuste (en inglés, Overfitting) ocurre cuando una red neuronal se ajusta demasiado a los datos de entrenamiento, capturando ruido y patrones específicos que no generalizan bien a nuevos datos, véase en la figura 2.18 (Ayora, 2024).

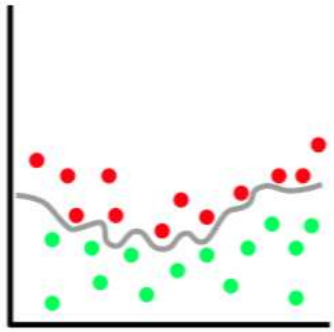

Figura 2.18. Overfitting. (Lin, 2023)

- Subajuste (en inglés, Underfitting) ocurre cuando una red neuronal no captura adecuadamente las tendencias subyacentes en los datos de entrenamiento, véase en la figura 2.19 (Ayora, 2024).

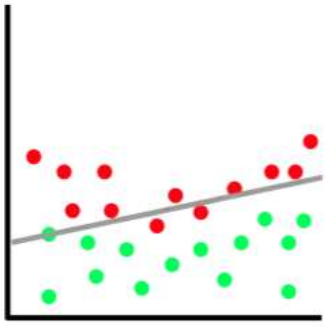

Figura 2.19. Underfitting. (Lin, 2023)

2.4.5 Redes neuronales convolucionales profundas (DCNN)

Las DCNN (por sus siglas en inglés) se utilizan en la clasificación de imágenes y video, como AlexNet, LeNet, VGGNet y otros modelos de aprendizaje avanzados. Se exploran los principios básicos y los conceptos esenciales detrás de estas redes neuronales, así como las técnicas de entrenamiento y optimización que se aplican (Rawat & Wang, 2017). El objetivo es analizar la evolución de estas arquitecturas y los enfoques empleados para mejorar la precisión y el rendimiento en la clasificación de imágenes y videos.

- AlexNet, fue desarrollada por Alex Krizhevsky, Ilya Sutskever y Geoffrey Hinton, AlexNet es una CNN que revolucionó el campo del aprendizaje profundo, teniendo un impacto importante en aprendizaje automático (Krizhevsky et al., 2017), especialmente en la visión por computadora. Su arquitectura es similar a LeNet, con un mayor número de filtros por capa y con capas convolucionales apiladas (Ayora, 2024). Véase en la figura 2.20.

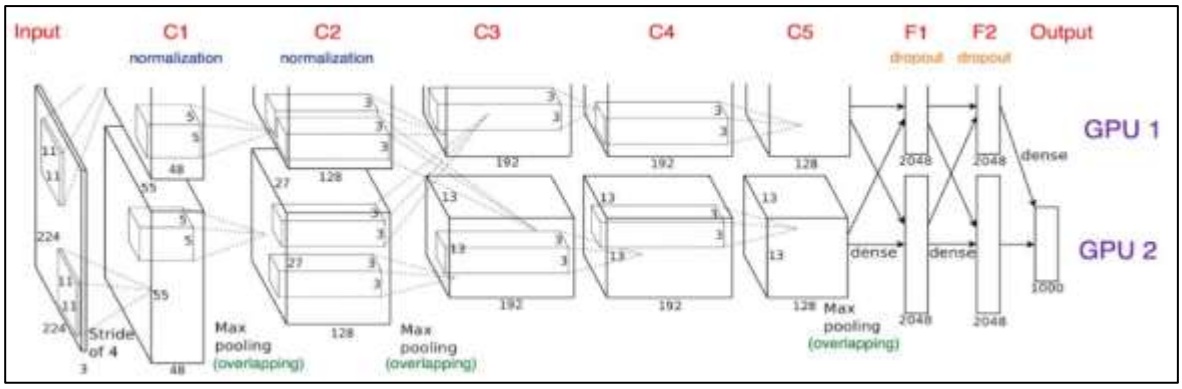

Figura 2.20. AlexNet. (Díaz-Ramírez, 2021)

- LeNet, trata de una red neuronal convolucional constituida por múltiples capas dispuestas en un patrón específico para procesar y clasificar imágenes. Esta red es capaz de detectar caracteres utilizando los conceptos de backpropagation y feedforward (Pechyonkin, 2018). Véase en la figura 2.21.

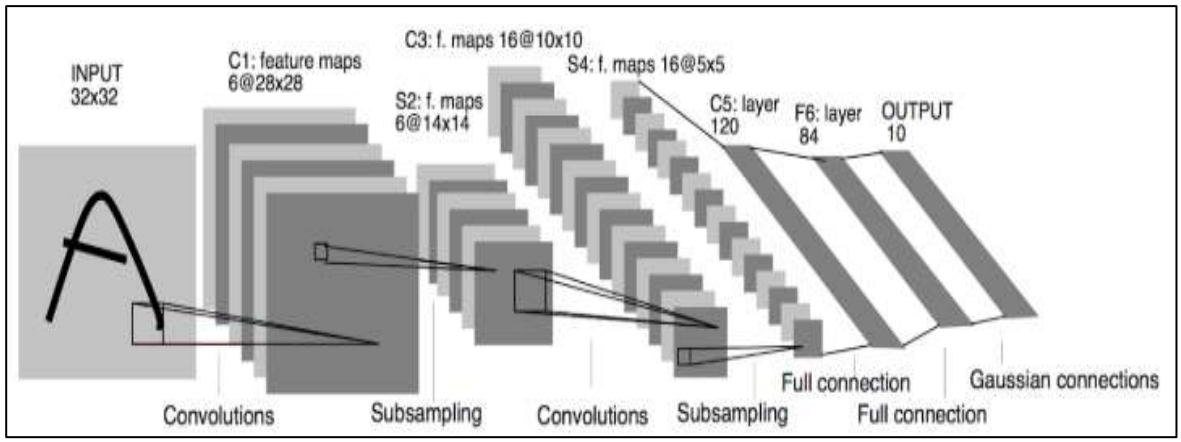

Figura 2.21. LeNet. (Ayora, 2024)

- VGGNet, es una CNN desarrollada por el grupo de investigación Visual Geometry Group (VGG) de la Universidad de Oxford. Los modelos de aprendizaje más populares de la familia VGG son VGG16 y VGG19, que tienen 16 y 19 capas de aprendizaje profundo (Simonyan & Zisserman, 2015).
- VGG-16 se compone por 16 capas, 13 capas de convolución, 2 capas totalmente conectadas, la capa final y 1 softmax para clasificar (Hassan, 2018), en la figura 2.22, se muestra la arquitectura de una VGG16.

Figura 2.22. VGG16. (Hassan, 2018)

- VGG-19 se compone por 19 capas, 16 capas de convolución, 2 de ellas totalmente conectadas y la capa final una softmax para clasificar (Jaworek-Korjakowska et al., 2019), en la figura 2.23, se observa la arquitectura de un VGG19.

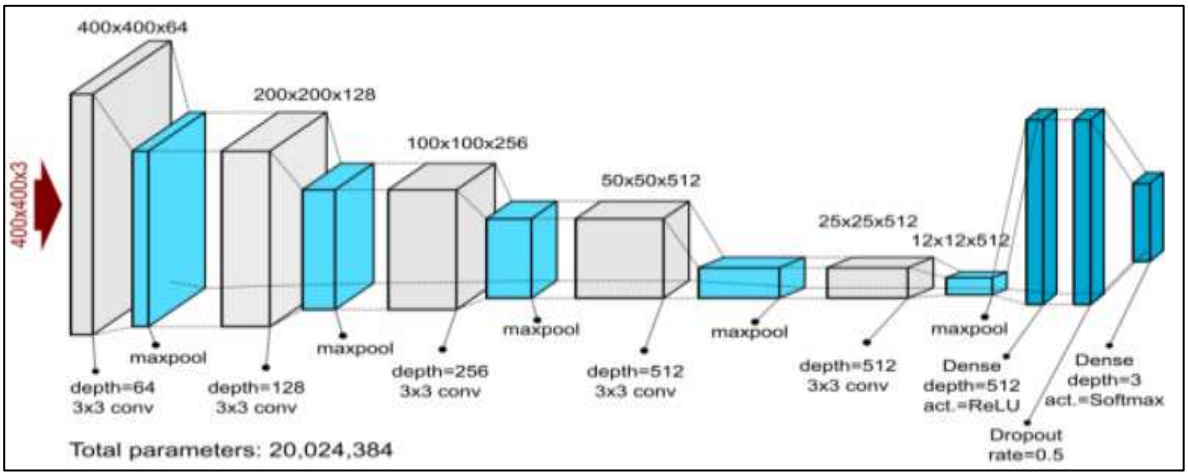

Figura 2.23. VGG19. (Jaworek-Korjakowska et al., 2019)

2.4.6 Redes neuronales convolucionales 3D (CNN 3D)

Las CNN 3D son parte de las CNN diseñadas para trabajar con datos tridimensionales, como videos o imágenes médicas volumétricas. A diferencia de las CNN 2D que se aplican a imágenes bidimensionales, las CNN 3D consideran tanto las dimensiones espaciales (ancho y alto) como la dimensión temporal (Esan et al., 2020). Esto las hace especialmente útiles para tareas que involucran secuencias temporales en datos tridimensionales, como el análisis de videos, imágenes médicas y reconocimiento de actividades humanas (Ji et al., 2012).

Las CNN 3D trabajan de igual forma que una CNN convencional, la diferencia es que agregan capas de convolución 3D. En la figura 2.24, se presenta la arquitectura de la CNN 3D.

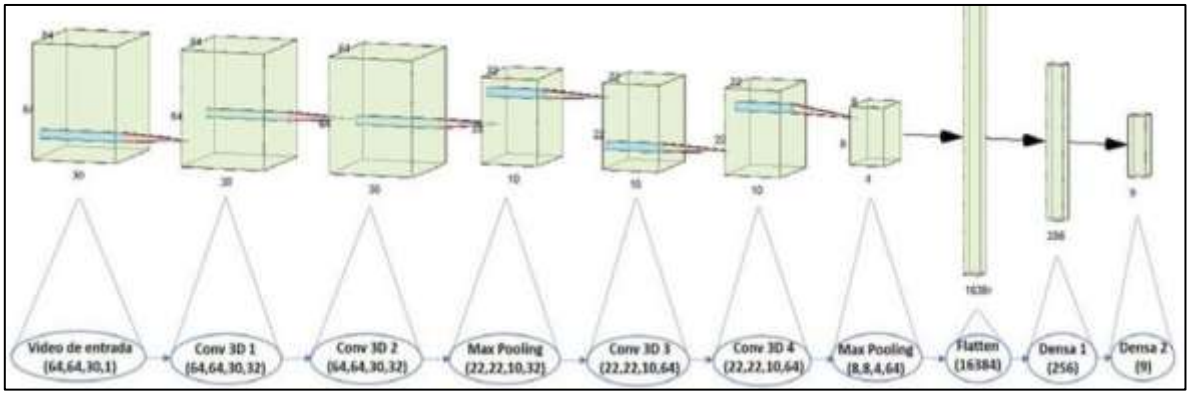

Figura 2.24. CNN 3D. (Lin et al., 2020)

- Capas de convolución 3D (Conv 3D) ejecuta operaciones de convolución en tres dimensiones, usando filtros tridimensionales (Lin et al., 2020).

Se capturan las características volumétricas de los datos de entrada mediante la reducción dimensional de un volumen de características, dependiendo del tamaño del filtro y el stride empleados. Se lleva a cabo la detección y clasificación de acciones humanas en secuencias de video, así como el análisis de contenido, que incluye la clasificación de escenas y eventos en los videos (Zhou et al., 2016).

CAPÍTULO 3

3 Metodología

En este capítulo se presenta el proceso de desarrollo del sistema para detectar conductas delictivas en el robo de autopartes, utilizando CNN 3D. En el desarrollo del sistema, se consideran 3 etapas como se puede observar en la figura 3.1. En principio, se desarrolló el conjunto de datos para entrenamiento y pruebas de un sistema que por medio de cámaras e vigilancia pueda detectar conductas delictivas relacionadas con el robo de autopartes mediante el uso de CNN 3D.

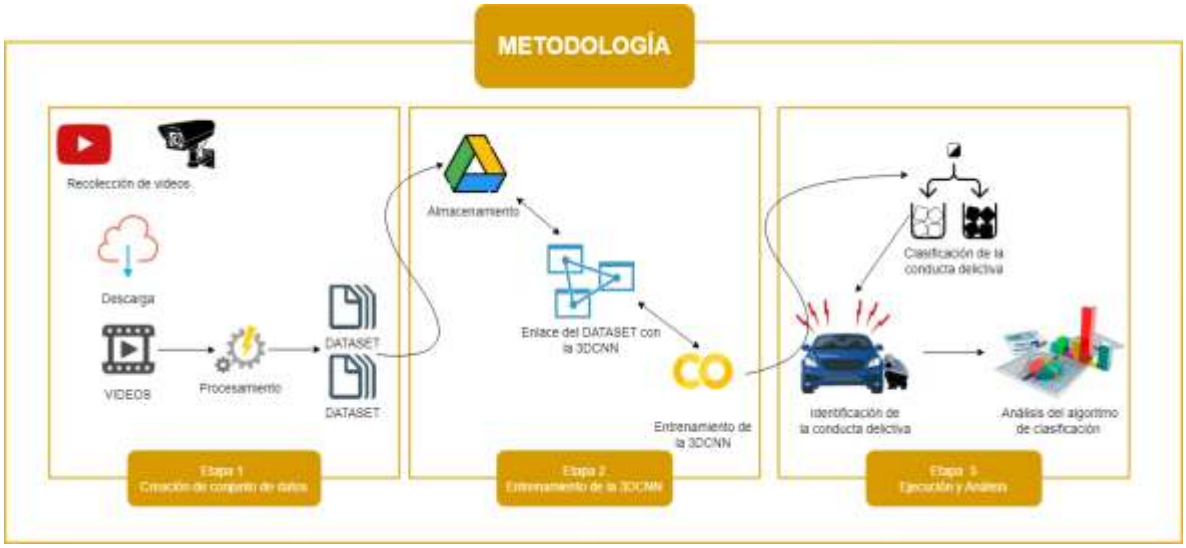

Figura 3.1. Etapas de la metodología. Elaboración propia.

3.1 Etapa 1. Preparación y creación de un conjunto de datos

La fase de preparación de datos involucró la recopilación y preprocesamiento de un conjunto de videos de vigilancia que contienen incidentes de robo de autopartes. Como se mencionó en la delimitación de la tesis, los videos fueron obtenidos de diferentes fuentes, principalmente de la plataforma www.youtube.com y de cámaras de seguridad particulares. Posteriormente fueron segmentados, y etiquetados manualmente para identificar la conducta delictiva en el robo de autopartes, de esta manera se garantiza la veracidad de la muestra de entrenamiento.

El conjunto de datos creado incluye videos con escenas de robo de autopartes y otros videos que no presentan este delito. Cada video se analizó cuidadosamente para identificar las escenas que presentan conductas delictivas, recortando los videos con dichas escenas en un lapso entre 3 y 40 segundos. Este proceso se repitió hasta obtener 1000 videos de conductas delictivas y 1000 videos sin conductas delictivas, este conjunto de datos se almacenó localmente en una carpeta denominada 'C: \Conductas Delictivas\', ubicada en un ordenador personal, con un uso de espacio de 3.95 GB. Es importante destacar que, al recopilar los videos, se presentaron una gran variedad de resoluciones que van desde 144p hasta 1080p. En la figura 3.2, se muestra un ejemplo de cómo son las grabaciones para identificar casos de robo y no robo de autopartes.

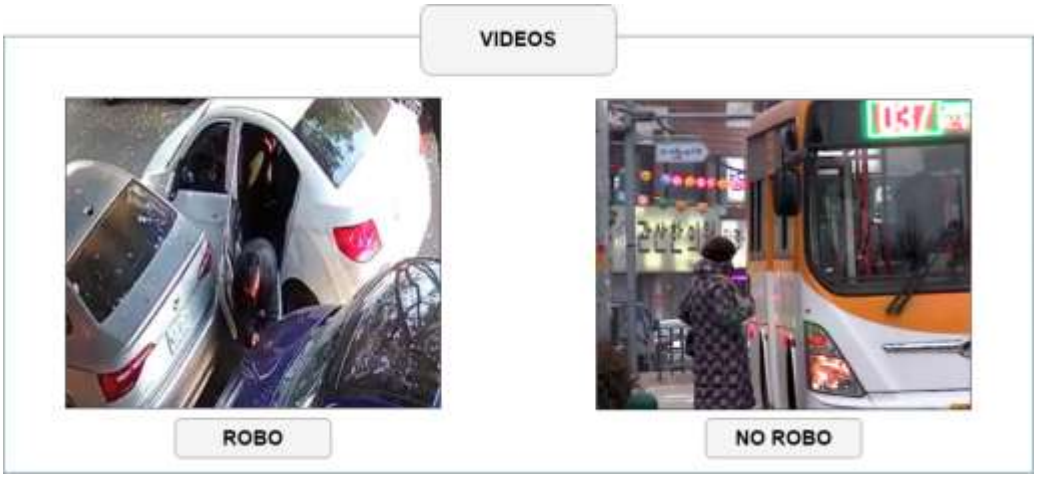

Figura 3.2. Videos obtenidos de cámaras de seguridad y plataforma YouTube que presentan escenas de robo y no robo de autopartes. Elaboración propia.

3.1.1 Descargar videos

Se realizó la búsqueda de videos en el sitio web YouTube que contenían escenas de robo de autopartes como se muestra en la figura 3.3, copiando la URL de los videos que se deseaban descargar desde la barra de direcciones del navegador. Luego, se utilizó el sitio web savefrom.net para descargar los videos de robos de autopartes, para el proceso de videos obtenidos de CCTV fue por medio de descarga de DVR el cual pudimos extraer videos de un consultorio médico y casa habitación de algunos familiares y conocidos. Estos archivos se almacenaron en un ordenador personal en una carpeta denominada 'C:\Conductas Delictivas', la cual contiene subcarpetas organizadas por clase: 'Robo' y 'No Robo'.

Figura 3.3. Descarga de videos.
Elaboración propia.

3.1.2 Recortar videos

Se utilizó el software de edición Macrosoft Clipchamp para recortar los videos que contenían escenas de 'Robo' y 'No Robo' de autopartes, véase en la figura 3.4. En la línea de tiempo de los videos, se localizó y seleccionó el punto inicial y final deseado, enfocándose en las escenas específicas del robo de autopartes, y se procedió a realizar los recortes necesarios. Los videos recortados se exportaron en formato (.mp4) y se almacenaron en la ruta 'C:\Conductas Delictivas\Videos-Pre-Procesados'.

Figura 3.4. Recorte de videos respecto a escenas de Robo y No Robo. Elaboración propia.

3.1.3 Depurado de videos

Se empleó el software de edición de video avanzado online-video-cutter.com para depurar los videos realizando el análisis minucioso de las características de marcas, mostrado

en la figura 3.5, etiquetas y logotipos, considerando su posición, tamaño y apariencia. Posteriormente, se almacenaron en la ruta 'C: \Conductas Delictivas\Videos-Pre-Procesados'.

Figura 3.5. Depuración de videos. Elaboración propia.

3.1.4 Carga de conjunto de datos a Google Drive

Se creo una cuenta de Google llamada 'Videos-Procesados' para poder almacenar el conjunto de datos creado, se cargaron desde la PC donde se tenían almacenados con la ruta 'C:\Conductas Delictivas\Videos-Pre-Procesados' a Google Drive, se crearon dos subcarpetas adicionales llamadas 'Conducta Delictiva' y 'Conducta No Delictiva' y se agregaron 1000 videos por cada conducta siendo un total de 2000 videos, se muestra en la figura 3.6.

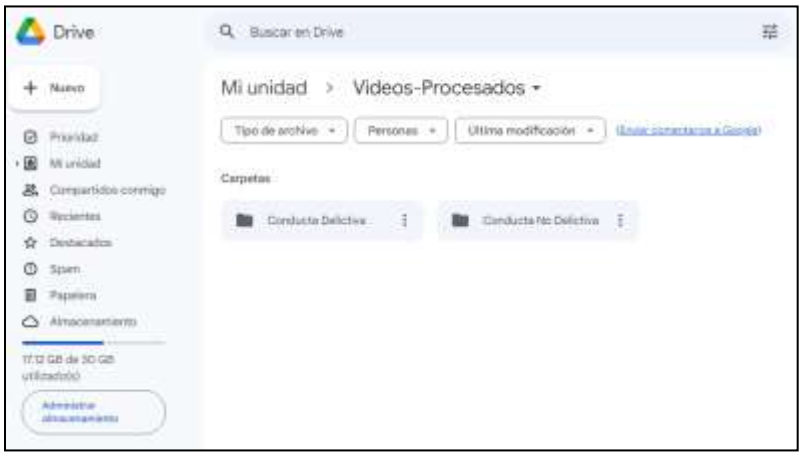

Figura 3.6. Carga del nuevo conjunto de datos a Google Drive. Elaboración propia.

3.2 Etapa 2. Entrenamiento de la CNN 3D

Fue imprescindible transferir el conjunto de datos a un servicio de almacenamiento en la nube. Esto se debió a que había una gran cantidad de videos recolectados, lo que resulta en un peso considerable y difícil de manejar en un ordenador de escritorio por falta de recursos. Principalmente Memoria RAM, CPU y GPU. La solución consistió en subir todos los videos a 'Google Drive' para su almacenamiento y utilizar Google Colab para desarrollar el programa en Python. Como ambas plataformas son de Google su integración facilitó la conexión entre el conjunto de datos y el programa desarrollado para la detección de conductas delictivas en el robo de autopartes con CNN 3D. Una representación simplificada de este proceso se muestra en la figura 3.7.

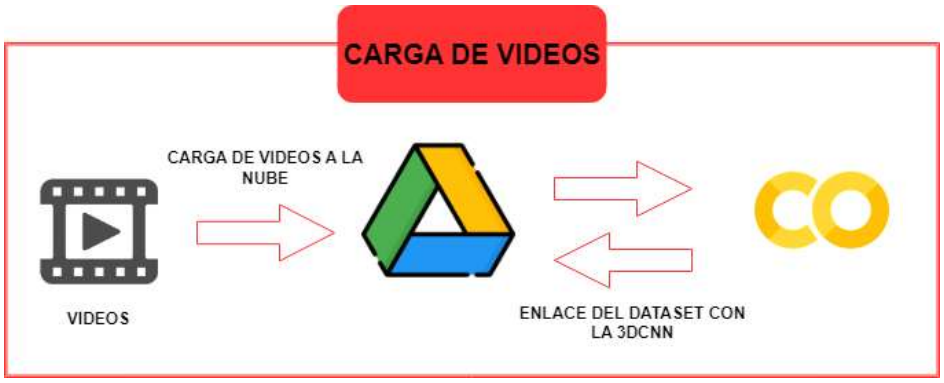

Figura 3.7. Almacenamiento y enlace en la nube con Google Drive. Elaboración propia.

3.2.1 Recursos de computacionales por Google Colab

Google Colab es una plataforma en la nube que permite desarrollar y ejecutar código en Python, ofreciendo acceso a recursos computacionales como MEMORIA RAM y GPU sin necesidad de configuraciones locales. Entre sus principales recursos destacan:

- Memoria RAM - 12.7 GB
- Memoria RAM de GPU - 15 GB
- Almacenamiento - 78.2 GB

3.2.2 Representación de la detección de conductas delictivas con la CNN 3D

En la figura 3.8, se muestra de manera visual cómo se llevó a cabo el procesamiento del conjunto de datos y su posterior clasificación utilizando una CNN 3D. Inicialmente, cada video se dividió en frames, y se obtuvieron vectores de características convolucionales. Después se concatenaron para formar una matriz vectorial, donde se acomodaron con respecto a la información en tiempo y espacio, mismas que representan las características convolucionales, convirtiéndose en una matriz vectorial de varias dimensiones representada cómo una nueva imagen, permitiendo procesarla una vez más por otra CNN 3D.

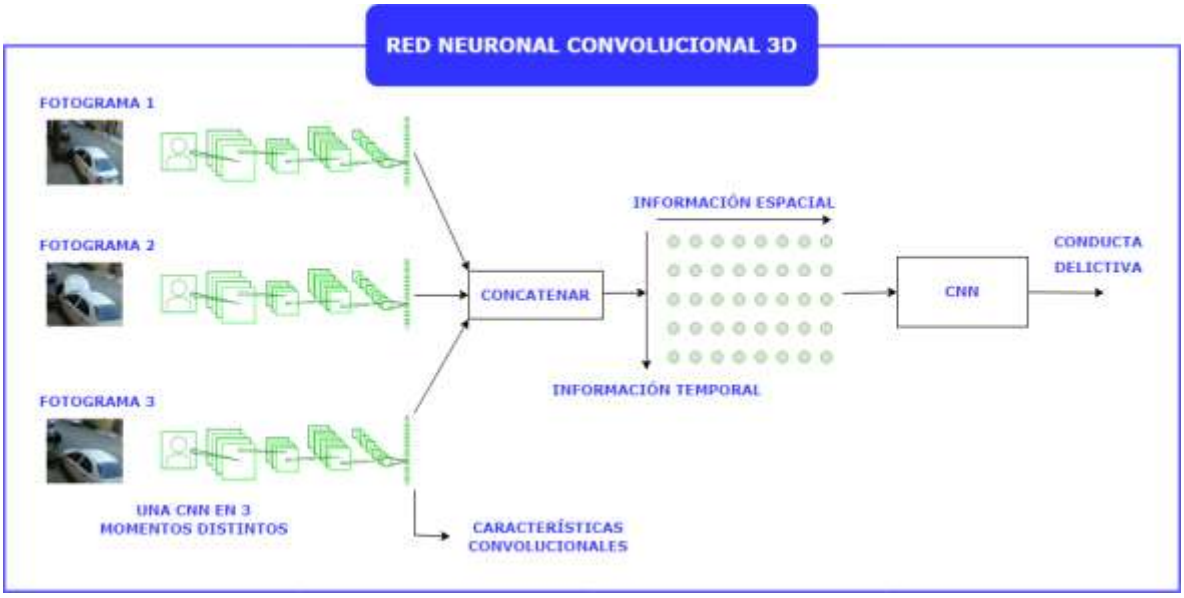

Figura 3.8. Representación para el entrenamiento de la CNN 3D. Elaboración propia.

3.2.3 Extracción de frames de video

Las CNN 3D están diseñadas para procesar imágenes, por lo tanto, los videos deben ser procesados en secuencias de imágenes por ello es pertinente realizar la extracción de los frames de cada video almacenado en el conjunto de datos, por tal motivo, se estableció una conexión entre 'Google Colab' y 'Google Drive' utilizando la biblioteca 'drive'. Posteriormente, se empleó un algoritmo que aprovecha las bibliotecas 'OpenCV y matplotlib' en Python para leer los archivos de video, ubicados en las rutas '/content/drive/MyDrive/Clasificación-Procesados/Robo/' y

'/content/drive/MyDrive/Clasificación-Procesados/No-Robo/' para procesar cada video con sus respectivos frames y mostrarlos como una secuencia de imágenes. De la misma manera se ocupó la librería 'cv2' de 'OpenCV' la cual permitió realizar el recorte estandarizado de cada video almacenado en el conjunto de datos ya clasificado. Esto implicó extraer y guardar cada uno de los frames de los videos en archivos de imagen separados, convirtiendo los frames en imágenes de 224 x 224 píxeles, como de observa en la figura 3.9. Esto, facilitó a la CNN 3D extraer patrones, características, bordes y texturas de los cuadros de video.

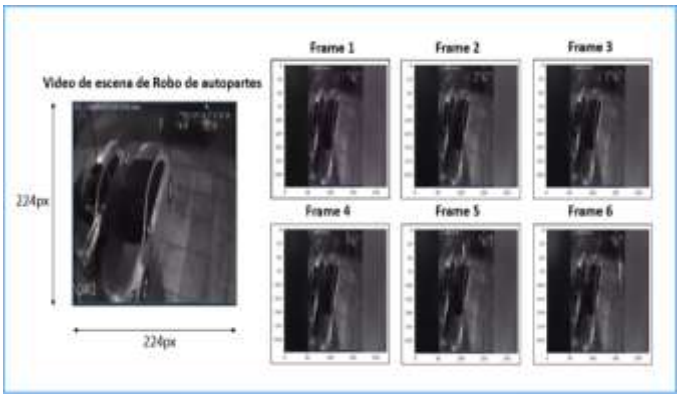

Figura 3.9. Extracción de frames por video del conjunto de datos.
Elaboración propia.

3.2.4 Problemas encontrados por alto consumo de recursos computacionales y soluciones

Durante el entrenamiento de la CNN 3D surgieron diversos desafíos, como el alto consumo computacional, errores en el etiquetado, variaciones en la resolución de los videos y diferencias en la cantidad de frames por video. A continuación, se describen estos problemas junto con las soluciones implementadas.

3.2.4.1 Resolución de videos

El conjunto de datos es esencial para la detección de conductas delictivas en el robo de autopartes con la CNN 3D. Por ello, se tomó en cuenta la resolución de los videos a trabajar con este tipo de modelo de aprendizaje, ya que influyó significativamente en su rendimiento

y en el proceso de entrenamiento. Así, fue imprescindible tomar en cuenta los siguientes aspectos al determinar la resolución de los videos en el conjunto de datos:

- Detalle y contexto temporal: es esencial decidir la resolución para la detección de conductas delictivas. Una resolución alta facilita la captura de detalles minuciosos en cada frame del video, siendo importante en tareas que implican el análisis de características específicas. Por otro lado, una resolución baja puede resultar suficiente cuando la tarea se enfoca en patrones de movimiento general y no demanda la identificación precisa de detalles.
- Generalización: Una resolución alta tiene el potencial de mejorar la capacidad del modelo de aprendizaje para generalizar en diversas situaciones y condiciones. Esto es especialmente valioso al enfrentar escenarios variados y cambios en la iluminación y contrastes de los videos, permitiendo al modelo de aprendizaje adaptarse de manera efectiva. En contraste, una resolución baja puede hacer que el modelo de aprendizaje sea más propenso al sobreajuste a situaciones específicas, limitando su capacidad para adaptarse a variaciones en los datos. La elección entre alta y baja resolución debe considerarse cuidadosamente, buscando un equilibrio que promueva la generalización del modelo de aprendizaje sin comprometer su capacidad de adaptación a la diversidad de condiciones presentes en la detección de conductas delictivas.
- Memoria y el tamaño del modelo de aprendizaje: En el caso de una resolución alta, se requiere una mayor cantidad de memoria, tanto para almacenar los videos como para los parámetros del modelo de aprendizaje. Esto puede generar desafíos en términos de capacidad de almacenamiento y recursos computacionales. Por otro lado, optar por una resolución baja disminuye significativamente los requisitos de memoria y reduce el tamaño del modelo de aprendizaje. Esta elección puede ser estratégica para optimizar la eficiencia del sistema, especialmente cuando se enfrentan limitaciones de recursos. En última instancia, la decisión entre alta y baja resolución debe equilibrarse cuidadosamente, considerando la capacidad de almacenamiento disponible y los recursos computacionales para lograr un rendimiento óptimo del modelo de aprendizaje.

- Tiempo de entrenamiento: En el caso de una resolución alta, es probable que se incremente considerablemente el tiempo de entrenamiento debido al mayor número de píxeles y cuadros por video. Este aumento en la complejidad computacional puede ser una consideración importante. Por el contrario, optar por una resolución baja tiene el beneficio de reducir el tiempo de entrenamiento, lo que puede ser ventajoso en situaciones donde se busca una implementación eficiente y rápida.

Con el fin de comprender las resoluciones que se presentan en los videos, en la tabla 3.1, se muestran las distintas resoluciones de videos. Esta resolución se refiere a la cantidad de píxeles que compone una imagen de video, por ejemplo "1920x1080", representa que la imagen está compuesta por 1920 píxeles en el eje horizontal (ancho) y 1080 píxeles en el eje vertical (alto). A continuación, se muestra la amplia variedad de resoluciones, desde la estándar de 720x480 (SD) hasta definiciones de alta calidad como 7680x4320 (8K).

Tabla 3.1. Resolución de videos. Elaboración propia.

Resolución	**Tamaño (pixels)**
SD (Standard Definition)- NTCS	720x480
SD (Standard Definition)- PAL	720x576
HD (High Definition) - 720p	1280x720
HD (High Definition) - 1080p	1920x1080
Full HD (High Definition) - 1080p	1920x1080
2K	2048x1080
4K UHD (Ultra High Definition) - 4320p	3840x2160
8K	7680x4320

En la figura 3.10, se presenta la distribución de videos del conjunto de datos según sus resoluciones en la clase "Robo", mientras que en la figura 3.11, se ilustra la cantidad de videos por resoluciones clasificados como "No Robo". Se aprecia una notable diversidad en la resolución de los videos, marcando notables desafíos como diferentes niveles de calidad, alto consumo de recursos computacionales, diferente tamaño de datos de entrada para la CNN 3D, afectando el proceso de entrenamiento del modelo de aprendizaje.

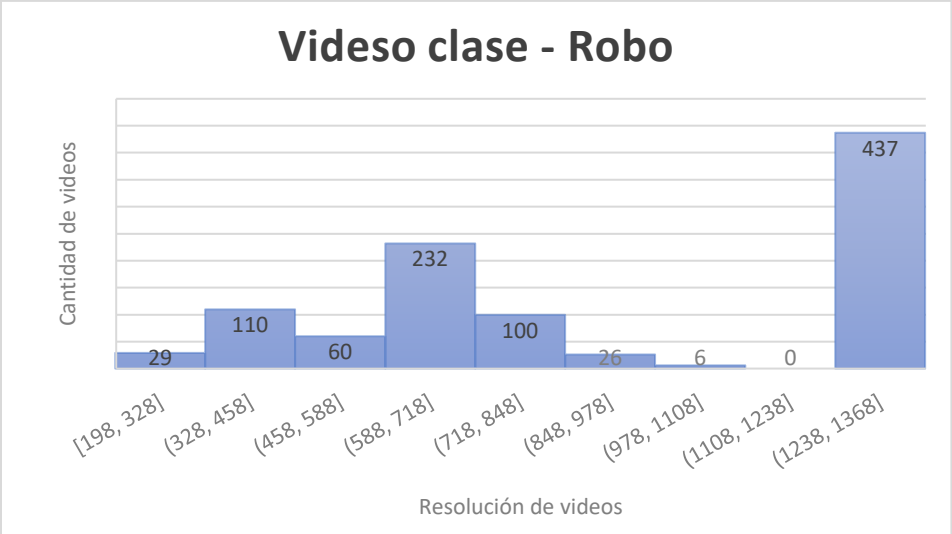

Figura 3.10. Resoluciones de video para la clase – Robo. Elaboración propia.

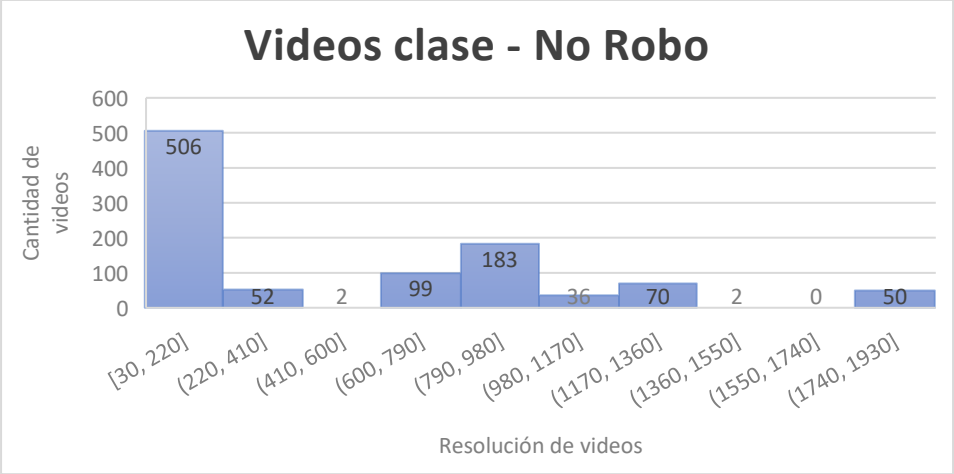

Figura 3.11. Resoluciones de video para la clase - No Robo. Elaboración Propia.

Para ello, se realizaron una serie de pruebas enfocadas en el cambio de formato y resolución de video. Los resultados de cada prueba se presentan en la tabla 3.2, mostrando 3 pruebas realizadas a un video en específico dentro del conjunto de datos, en el primer renglón se muestra el video denominado "Robo (09).mp4" siendo este el original, posteriormente se procedió a cambiar el formato y la resolución del video, lo cual se observa en los renglones 2, 3 y 4. Sin embargo, durante estas pruebas, se observó que mantuvo la misma cantidad de frames, mostrando solamente un cambio respecto a tiempo total de video y aumentando la cantidad de frames por segundo, dando como resultado el mismo total de frames.

Tabla 3.2. Resultados del cambio de formato de un vídeo. Elaboración propia.

Nombre del video	Frames por seg. (fps)	Total de frames	Ancho de video	Altura de video	Tiempo total de video
Robo (09).mp4	340.4111704	1303	1280	720	3.82772398
Robo (09).mp4	679.4309466	1303	854	480	1.917781353
Robo (09).mpg	731.0613932	1303	720	576	1.78234005
Robo (09).avi	951.0334015	1303	720	480	1.370088577

Al realizar el cambio de la resolución del video, la cantidad de frames por segundo (fps) generalmente se mantiene constante porque la resolución y la tasa de frames, son dos características independientes de un video. La resolución se refiere a las dimensiones del video en términos de píxeles y la tasa de frames es la cantidad de frames que se muestran por segundo del video, esto es lo que determina la calidad y fluidez del video.

Los resultados, indican que se puede descartar la opción de realizar cambios en la resolución de los videos. La razón es que, al mantener la misma cantidad de frames en cada video, no se logra reducir la carga computacional para la fase de entrenamiento del conjunto de datos. La CNN 3D usa técnicas de filtrado para reducir el tamaño de cada frame pero la cantidad de frames por video es demasiado grande, cargando el peso de procesamiento en memoria RAM, CPU y GPU afectando el procesamiento de la misma CNN 3D.

3.2.4.2 Cantidad de frames

Los frames de video al reproducirse rápidamente, crean la ilusión de movimiento. Cada frame contiene información visual detallada para un momento específico en el tiempo, siendo de suma importancia para adquirir las características que nos ayudaron a identificar el comportamiento delictivo y no delictivo, al realizar el entrenamiento en la CNN 3D.

En la figura 3.12, se muestra la cantidad de frames correspondientes a la clase 'Robo', mientras que en la figura 3.13, se presenta la cantidad de frames para la clase 'No Robo'. En el eje Y se indica el número de videos y en el eje X, la cantidad de frames. Se observa una amplia variabilidad en el número de frames. Al sumar todos los frames del conjunto de datos, se obtiene un total de 449,699. Esta variabilidad representa un factor crítico en la fase de

entrenamiento de la CNN 3D, ya que implicó un alto consumo de recursos computacionales saturando la plataforma Google Colab.

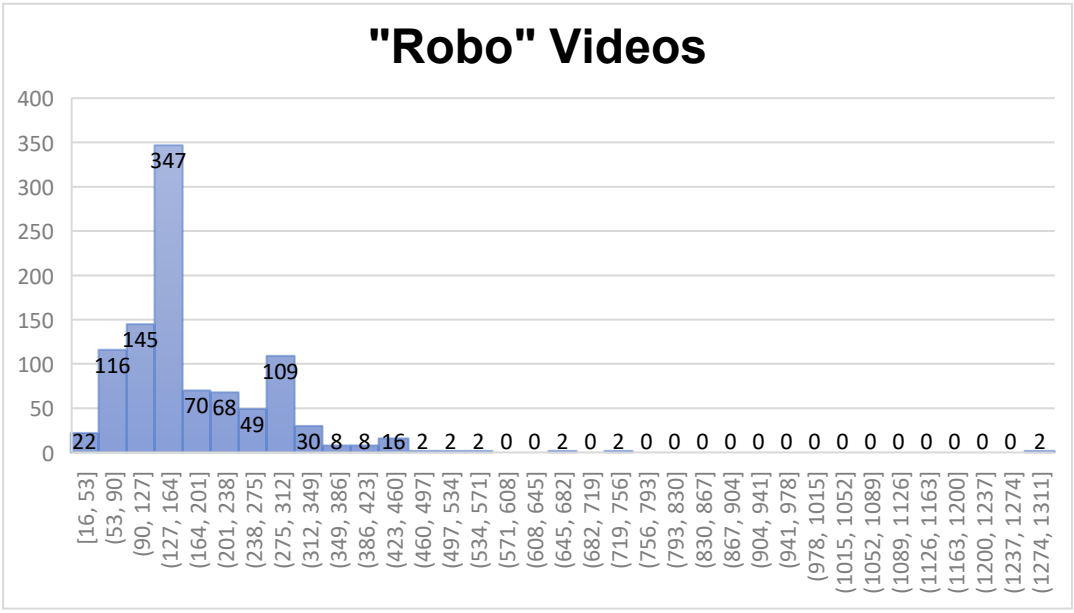

Figura 3.12. Cantidad de frames del conjunto de datos por clase Robo. Elaboración propia.

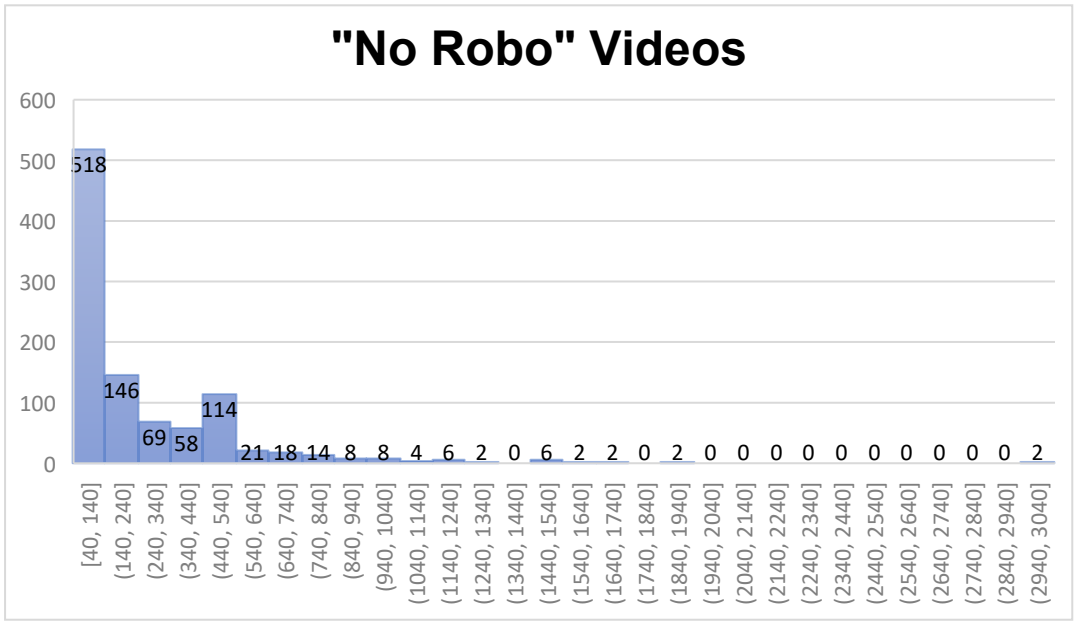

Figura 3.13. Cantidad de frames del conjunto de datos por clase No Robo. Elaboración propia.

3.2.4.3 Importancia del conjunto de datos y su etiquetado

La calidad del conjunto de datos resultó crucial en el proceso del entrenamiento de la CNN 3D. Su importancia se centró en asegurar la diversidad de datos de entrada, permitiendo que la CNN 3D extrajera características y patrones más relevantes. Uno de los aspectos fundamentales del conjunto de datos en el entrenamiento de la CNN 3D, como se mencionó anteriormente, es asegurar el equilibrio entre ambas clases en cuestión a la cantidad de videos por cada clase.

Del conjunto de datos almacenado en Google Drive, en la carpeta denominada "Clasificación de Videos", se accedió a las subcarpetas que contienen los videos organizados por clase llamada "No_Robo" que está etiquetada en el algoritmo de Python con el valor (0), mientras que la clase "Robo" se representa con el valor (1). Esta clasificación permite que el programa identifique y cargue automáticamente los videos correspondientes a cada categoría durante el proceso de entrenamiento y evaluación del modelo. Inicialmente los videos presentaron un etiquetado poco ordenado y con muchas discrepancias, esto se observa en la Tabla. 3.3, siendo uno de los errores más frecuentes, propiciando diversidad de etiquetas para cada clase de conducta delictiva. Los videos que contienen escenas de robo comienzan con la cadena de texto "Robo", mientras que los videos que no contienen este tipo de escenas inician con la cadena "No_Robo". Además, ambas clases del conjunto de datos son enumeradas entre paréntesis, en este caso se invoca el primer lote de 10 videos por cada clase de forma ascendente, mostrando una variación notable de enumeración de los mismos.

Tabla 3.3. Etiquetado original de los videos. Elaboración propia.

1 Lote_0[0]	**1 Lote_1[0]**
No_Robo (1).mp4'	Robo (1).mp4'
No_Robo (10).mp4'	Robo (10).mp4'
No_Robo (100).mp4'	Robo (100).mp4'
No_Robo (1000).mp4'	Robo (1000).mp4'
No_Robo (1001).mp4'	Robo (1001).mp4'
No_Robo (1002).mp4'	Robo (101).mp4'
No_Robo (1003).mp4'	Robo (102).mp4'
No_Robo (1004).mp4'	Robo (103).mp4'
No_Robo (1005).mp4'	Robo (104).mp4'
No_Robo (1006).mp4'	Robo (105).mp4'

Aunque este etiquetado no afecta al entrenamiento o el desempeño de la CNN 3D, se decidió cambiar el nombre de la etiqueta de los videos en el conjunto de datos para una mejor organización y control. En la tabla. 3.4, se presentan las modificaciones del etiquetado en el conjunto de datos en su clase correspondiente, dando un mejor orden al invocar los videos por bloques.

Tabla 3.4. Actualización del etiquetado de los videos. Elaboración propia.

1 Lote_0[0]	1 Lote_1[0]
No_Robo (1).mp4'	Robo (1).mp4'
No_Robo (2).mp4'	Robo (2).mp4'
No_Robo (3).mp4'	Robo (3).mp4'
No_Robo (4).mp4'	Robo (4).mp4'
No_Robo (5).mp4'	Robo (5).mp4'
No_Robo (6).mp4'	Robo (6).mp4'
No_Robo (7).mp4'	Robo (7).mp4'
No_Robo (8).mp4'	Robo (8).mp4'
No_Robo (9).mp4'	Robo (9).mp4'
No_Robo (10).mp4'	Robo (10).mp4'

3.2.4.4 Mejoramiento de consumo computacional alto (TFRecords)

Durante el entrenamiento con el conjunto de datos, se identificó un alto consumo de memoria, lo que representó un desafío constante debido a las limitaciones de recursos en Google Colab. Tras investigar posibles soluciones, se encontró la librería TFRecords de TensorFlow, la cual permite convertir grandes volúmenes de datos en archivos binarios optimizados. Estos archivos pueden ser almacenados de manera eficiente en disco en la nube, reduciendo el uso de memoria y mejorando el rendimiento en la carga y procesamiento de datos, como se muestra en la figura 3.14.

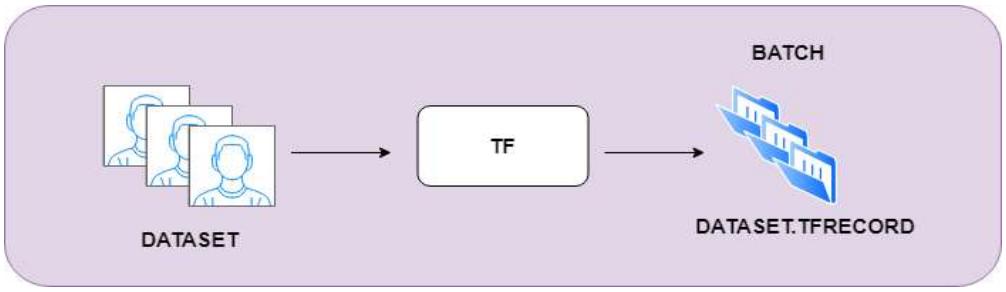

Figura 3.14. Representación de cómo funciona librería TFRecords de TensorFlow. Elaboración propia.

En este proceso, el conjunto de datos fue convertido a archivos binarios y almacenado en disco en lugar de la memoria RAM. Esto se debe a que, durante el entrenamiento de la CNN 3D, cada video o en este caso cada archivo binario, se carga directamente desde el disco cuando es necesario, en lugar de mantenerse en la RAM. Esta estrategia reduce significativamente el consumo de memoria, evitando la saturación de los recursos computacionales y haciendo el entrenamiento más ligero y eficiente.

Además, el uso de TFRecords optimiza la lectura secuencial de datos, mejorando la velocidad y estabilidad del proceso de entrenamiento. Gracias a esta optimización, se aprovecha mejor el uso de GPU, permitiendo acelerar el procesamiento de los datos y el ajuste de los parámetros del modelo maximizando el rendimiento en Google Colab, donde los recursos son limitados. En la figura 3.15, se muestra una representación de cómo trabaja la librería en relación con los recursos proporcionados por Google Colab.

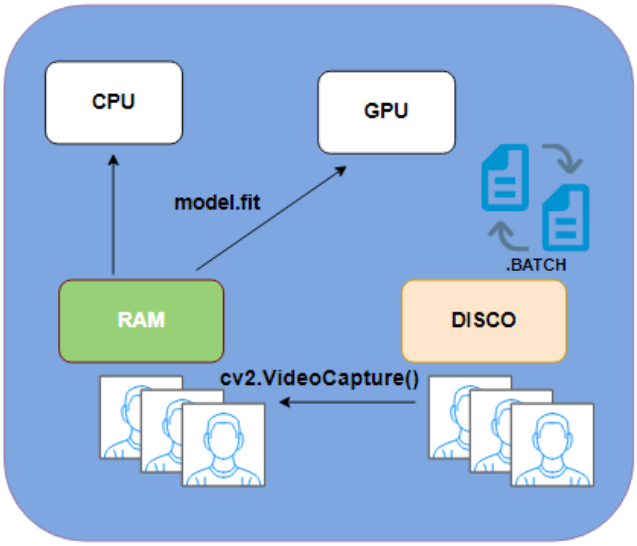

Figura 3.15. Representación de almacenamiento en disco en la nube y uso de memoria RAM en Google Colab usando TFRecords de TensorFlow. Elaboración propia.

A pesar de implementar la librería TFRecords para optimizar el almacenamiento y la carga de datos, el problema de alto consumo computacional persistió. Esto dejó una brecha significativa por resolver, impulsando la búsqueda de nuevas soluciones y mejorar la eficiencia del entrenamiento con la CNN 3D, como respuesta, se consideró la estrategia de entrenamiento por lotes, cuya metodología y beneficios se detallan en el siguiente capítulo.

3.2.4.5 Separación por lotes

Tras un análisis exhaustivo, se consideró el procesamiento por lotes (Batches), siendo una alternativa para la gran cantidad de datos. En este caso, al usar videos para el entrenamiento de las CNN 3D, resulta necesario cargar los datos de manera incremental, es decir, por lotes. Esto resulta importante si la cantidad de recursos computacionales es limitada, como es el caso de la plataforma Google Colab que es donde se realizó el entrenamiento de modelos de aprendizaje para la detección de conductas delictivas, siendo estos mismos limitados para el procesamiento y entrenamiento de la CNN 3D. La ventaja de trabajar con lotes, es realizar los cálculos de manera más eficiente y aprovechar mejor los recursos computacionales, como la memoria RAM, almacenamiento y procesadores CPU o GPU y reducir su costo para el entrenamiento de estos modelos de aprendizaje.

En la tabla 3.5, se presenta de manera detallada la división por lotes del conjunto de datos. Los 2000 videos, se separaron en 10 subconjuntos de 100 videos por cada lote, dando en total de 50 videos de Robo y 50 videos de No Robo por categoría.

Tabla 3.5. Separación por lotes del conjunto de datos para el entrenamiento de la CNN 3D. Elaboración propia.

# Lotes	**Lote_1**	**Lote_2**	**Lote_3**	**Lote_4**	**Lote_5**	**…**	**…**	**…**	**Lote_20**	**Total**
Lote %	5%	5%	5%	5%	5%	…	…	…	5%	100%
# Lotes de videos	100	100	100	100	100	…	…	…	100	2000

3.2.5 Arquitectura y entrenamiento del conjunto de datos

Se eligió la CNN 3D por su capacidad de analizar tanto características espaciales como temporales en secuencias de video, lo que es esencial para identificar conductas como el robo de autopartes. A diferencia de las CNN 2D, esta arquitectura permite identificar patrones de movimiento entre frames consecutivos, mejorando la precisión en la detección de eventos sospechosos. En la figura 3.16, se muestra la arquitectura de la CNN 3D diseñada para la identificación de la conducta delictiva.

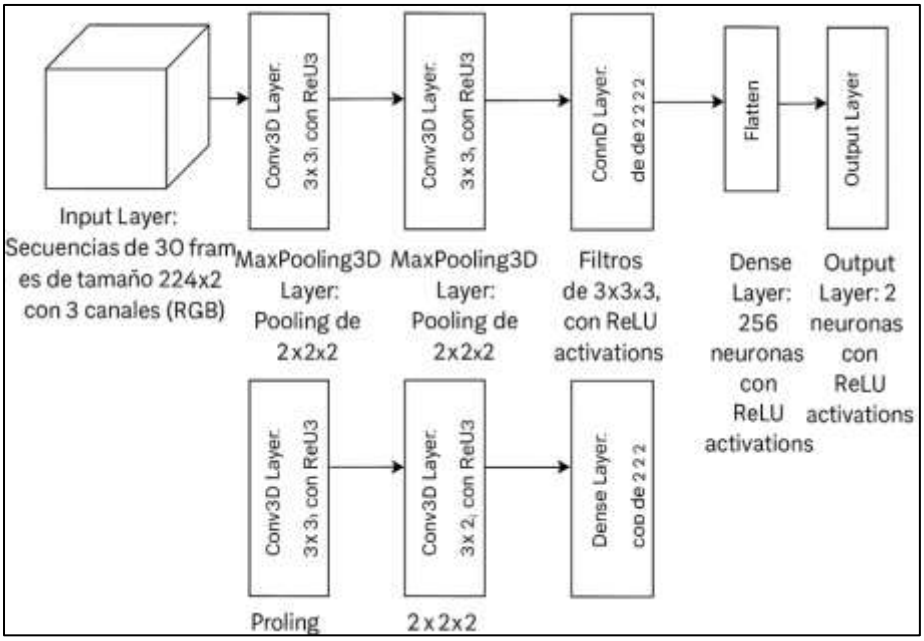

Figura 3.16. Arquitectura de la CNN 3D. Elaboración propia.

Arquitectura de la CNN 3D:

- Input Layer: Secuencias de 30 frames de tamaño 224x224 con 3 canales (RGB)
- Conv3D Layer: Filtros de 3x3x3, con ReLU activations.
- MaxPooling3D Layer: Pooling de 2x2x2
- Conv3D Layer: Filtros de 3x3x3, con ReLU activations.
- MaxPooling3D Layer: Pooling de 2x2x2
- Conv3D Layer: Filtros de 3x3x3, con ReLU activations
- MaxPooling3D Layer: Pooling de 2x2x2
- Flatten Layer
- Dense Layer: 256 neuronas con ReLU activations
- Dense Layer: 128 neuronas con ReLU activations
- Output Layer: 2 neuronas con softmax activation

Entrenamiento

- División del conjunto de datos: 80% para entrenamiento y 20% para prueba.
- Optimización: Adam optimizer
- Función de Pérdida: Categorical Crossentropy
- Épocas: 10
- Batch Size**:** 15

3.3 Creación de conjuntos de datos para bajar consumo computacional alto

Si del conjunto de datos, se obtiene una alta cantidad de frames, se podrán obtener más detalles y características que ayuden a tener una clasificación más precisa de los videos, a cambio de un mayor costo computacional. Por otro lado, si se obtiene una cantidad de frames mucho menor, se obtendrá una inferior porción de detalles y características, dando como resultado una deficiencia considerable para el entrenamiento del conjunto de datos con la CNN 3D.

La elección precisa de la cantidad de frames fue considerada meticulosamente, buscando un equilibrio óptimo entre la información temporal capturada y la eficiencia computacional. Este proceso de selección fue guiado por las particularidades de los recursos disponibles, asegurando así el rendimiento efectivo con una cantidad adecuada de frames para el modelo de aprendizaje.

Para mejorar el rendimiento del entrenamiento en el conjunto de datos de la CNN 3D, se creó un programa en Python que emplea las bibliotecas OpenCV para recortar videos y OS para creación de nuevas rutas de librerías donde se almacenaron los nuevos videos obtenidos por dichos recortes, se representan en la figura 3.17. En la primera fase, se eliminaron los frames pares del conjunto de datos, reduciendo el total de frames por video al 50%. En la segunda etapa, se implementó un recorte de 2 frames, resultando en un 36% de frames por video. En el tercer y último paso, se aplicó un recorte de 3 frames, alcanzando así un 27% de frames por video, obteniendo 3 conjuntos de datos extras para el entrenamiento de la CNN 3D.

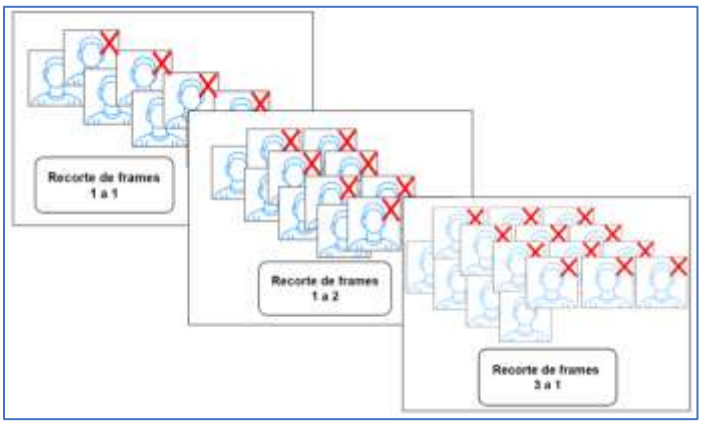

Figura 3.17. Representación del conjunto de datos por frame de video.
Elaboración propia.

En las figuras 3.18 y 3.19, se presenta la cantidad total de frames de video para cada elemento del conjunto de datos. Estos resultados revelan una disminución significativa en el número de frames, indicando así un impacto notable en la cantidad total de información visual registrada en los videos. Este fenómeno fue crucial al considerar la eficiencia computacional durante el entrenamiento del modelo de aprendizaje, ya que la reducción de frames contribuyó a una carga de procesamiento más liviana y, en consecuencia, a un rendimiento más eficaz en el análisis de datos mediante la CNN 3D.

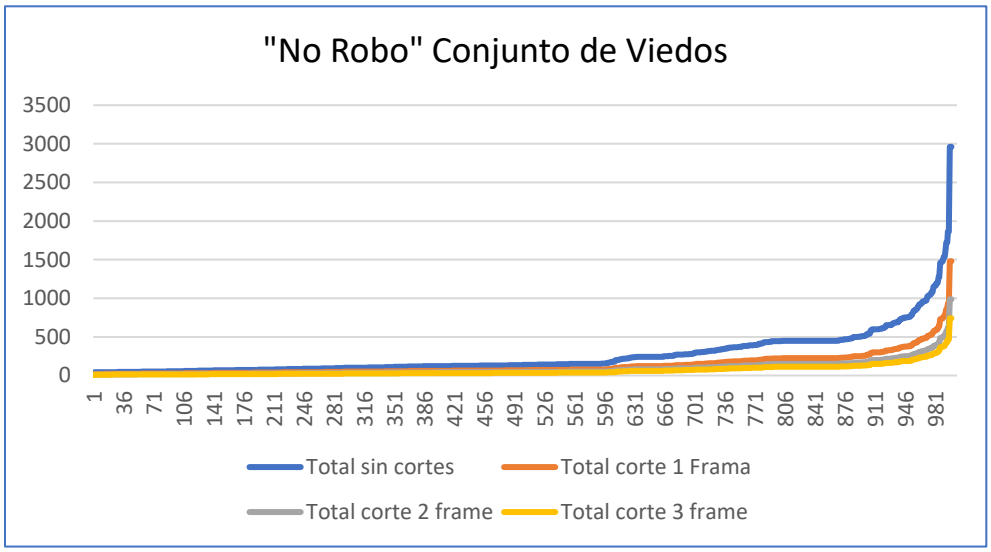

Figura 3.18. Total de frames por los conjuntos de datos creados por los recortes de frame para la clase No robo. Elaboración propia.

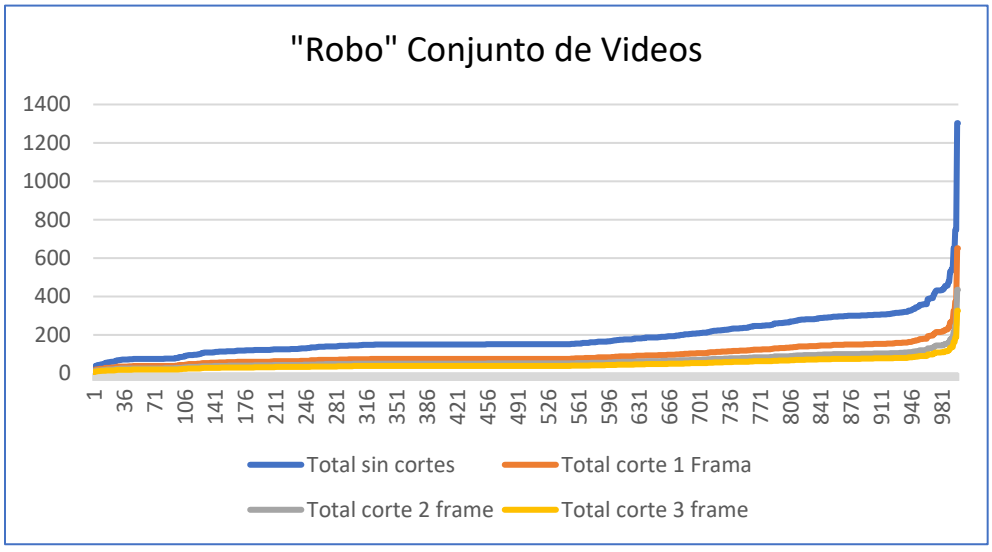

Figura 3.19. Total de frames por los conjuntos de datos creados por recortes de frame para la clase Robo. Elaboración propia.

Los efectos de recortar distintas cantidades de frames en un video pueden ser valiosos para reducir tanto el espacio de almacenamiento como el tiempo de procesamiento con la CNN 3D. Al seleccionar específicamente ciertos frames, se logra dirigir la atención hacia las partes más importantes del video, al mismo tiempo que se eliminan fracciones de video que no son perceptibles para el ojo humano, esto ayuda a optimizar el uso de recursos, mejorar el rendimiento del sistema y facilitar la interpretación del contenido de los videos en el conjunto de datos.

3.4 Etapa 3. Ejecución y análisis

El entrenamiento de la CNN 3D se llevó a cabo utilizando el conjunto de datos creado para la detección de conductas delictivas en el robo de autopartes. La ejecución comenzó con la preparación de los datos, que fueron procesados y almacenados en archivos binarios mediante la librería TFRecords para optimizar el uso de memoria y recursos computacionales. Se empleó el procesamiento por lotes mejorando la gran cantidad de datos sin saturar la memoria RAM. Durante el proceso de entrenamiento, se utilizaron recursos limitados de la plataforma Google Colab, lo que impuso restricciones en la velocidad y eficiencia del modelo. Sin embargo, el uso de GPU ayudó a acelerar los cálculos, permitiendo un entrenamiento más rápido y efectivo. La arquitectura de la CNN 3D fue diseñada para capturar patrones espaciales y temporales en los videos, lo que facilitó la clasificación entre los dos tipos de clases.

3.4.1 Análisis del entrenamiento de la CNN 3D

El análisis de los resultados de la CNN 3D para identificar conductas delictivas en el robo de autopartes implicó evaluar qué tan efectivo y preciso es el modelo de aprendizaje en la tarea de clasificación. Para ello se consideraron las siguientes métricas:

- Exactitud (Accuracy): Son las predicciones correctas entre el total de muestras analizadas. Se calcula con la suma de verdaderos positivos (VP) y verdaderos negativos (VN) dividida entre el total de observaciones. Es útil cuando las clases están equilibradas. Pueden presentar errores si contienen un desequilibrio de datos (Powers, 2020).

- Precisión (Precision): Señala que cantidad de predicciones son seguras y positivas del modelo. Se obtiene dividiendo los verdaderos positivos entre la suma de verdaderos positivos y falsos positivos (FP). La alta precisión indica que el modelo tiene pocos falsos positivos, siendo importante en escenarios donde las falsas alarmas deben de ser mínimas (Powers, 2020).
- Recall (Sensibilidad): Mide la capacidad del modelo para identificar correctamente las instancias de la clase positiva. Se calcula dividiendo los verdaderos positivos entre la suma de verdaderos positivos y falsos negativos (FN). Un alto recall indica que el modelo es altamente efectivo para detectar las clases, incluyendo si aumenta el número de falsos positivos (Sokolova & Lapalme, 2009).
- F1-Score: Es la media armónica entre precisión y recall, proporcionando un equilibrio entre ambas métricas (Powers, 2020). Se calcula como:

$$F1 = 2\ X\ \ (Precisión\ x\ Recall)/(Precisión + Recall)$$

Estas métricas ayudan a evaluar el rendimiento del modelo y desempeño de la CNN 3D en la detección de conductas delictivas. Además, permiten comparar distintos modelos de aprendizaje y técnicas con el objetivo de identificar aquellos que ofrecen el mejor rendimiento y la mayor precisión en esta tarea específica. Los resultados obtenidos se detallarán en el siguiente apartado.

3.5 Desarrollo de la interfaz para la detección de conductas delictivas en el robo de autopartes

El presente trabajo describe el desarrollo de la interfaz para la detección de conductas delictivas en el robo de autopartes, cuyo objetivo es ofrecer una herramienta interactiva, intuitiva y segura que permita a administradores y analistas visualizar, gestionar y evaluar eventos detectados por modelos de visión basados en redes neuronales 3D.Para orientar el diseño del sistema, se presenta la figura 3.20, correspondiente al diagrama de elaboración del sistema, el cual ilustra de manera clara el flujo de información y las interacciones entre los módulos principales: recolección de video, preprocesamiento, entorno de entrenamiento, librerías, almacenamiento de datos y la capa de interfaz y visualización.

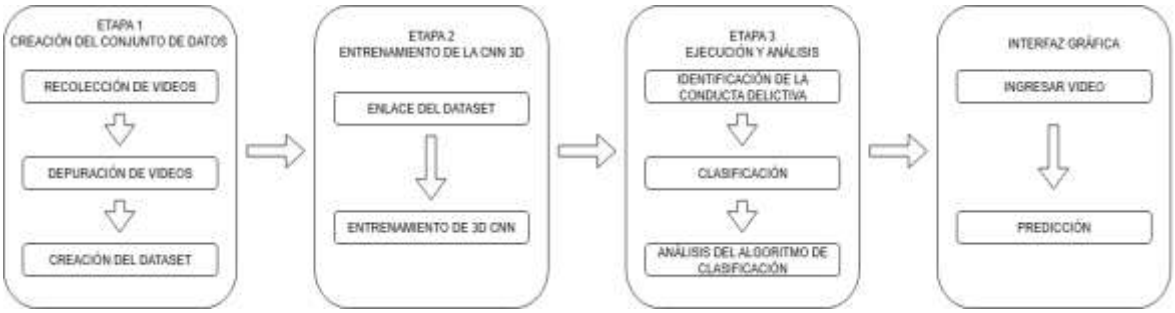

Figura 3.20. Diagrama de elaboración del sistema. Elaboración propia.

3.5.1 Arquitectura empleada

Se presenta en la figura 3.21, la arquitectura utilizada para el desarrollo del sistema de detección de conductas delictivas en el robo de autopartes, incluyendo tanto los componentes de software como el hardware necesario para su correcta implementación. Entre los principales requerimientos se incluyen: procesamiento de videos de diferentes resoluciones, almacenamiento eficiente de grandes volúmenes de datos, uso de librerías de análisis y aprendizaje profundo como TensorFlow, OpenCV, NumPy y Pandas, entrenamiento de la 3D CNN en el entorno de Google Colab, y capacidad de visualización interactiva para la detección de las conductas delictivas.

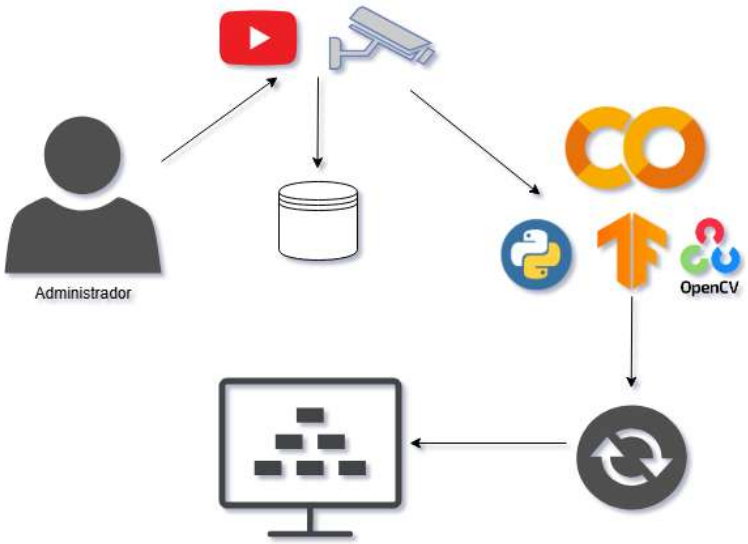

Figura 3.21. Arquitectura del sistema. Elaboración propia.

3.5.2 Descripción funcional de los casos de uso para la detección de conductas delictivas en el robo de autopartes

Se describe, las interacciones principales entre el usuario y el sistema. En primera instancia, el usuario ingresa al sistema, accede al panel de control. Posteriormente, se realiza la carga de video, donde se pueden incorporar archivos desde el almacenamiento local. Una vez cargado, el usuario puede iniciar la predicción, activando el modelo entrenado para analizar el video y generar la detección. Finalmente, el sistema muestra la predicción, indicando si se detecta o no una conducta delictiva de robo de autopartes, permitiendo al usuario tomar decisiones informadas y registrar el evento para su seguimiento y posterior análisis, véase en la figura 3.22, el diagrama de casos de uso del sistema.

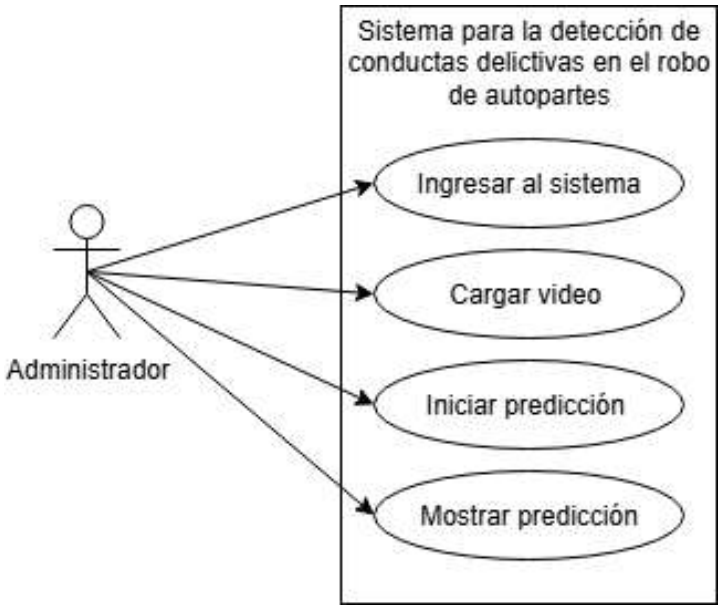

Figura 3.22. Diagrama de caso de uso del sistema. Elaboración propia.

3.5.3 Entorno y herramientas de desarrollo

El proceso se inicia cuando el usuario monta Google Drive, estableciendo la conexión necesaria para acceder a los archivos. Una vez confirmada la conexión, el usuario especifica las rutas y los lotes de videos a procesar.

El sistema toma el control, realizando una serie de pasos secuenciales. En primer lugar, procesa los videos para extraer fotogramas. Posteriormente, utiliza el modelo VGG19 para extraer las características relevantes de cada uno de esos fotogramas. Estas características, junto con sus etiquetas correspondientes, se guardan en el formato TFRecords, optimizado para el entrenamiento de modelos de aprendizaje profundo.

Con los datos listos, en la figura 3.23, se muestra el diagrama de flujo que continúa con la etapa de entrenamiento. Se entrena un modelo 3D CNN utilizando el conjunto de datos preparado. Después del entrenamiento, el modelo se somete a una evaluación del conjunto de datos de prueba, lo que permite verificar su rendimiento y precisión. Finalmente, el usuario puede solicitar la predicción de un nuevo video. El sistema lo procesa de la misma manera que los datos de entrenamiento para que el modelo clasifique los videos brindados por el usuario como "ROBO" o "NO ROBO".

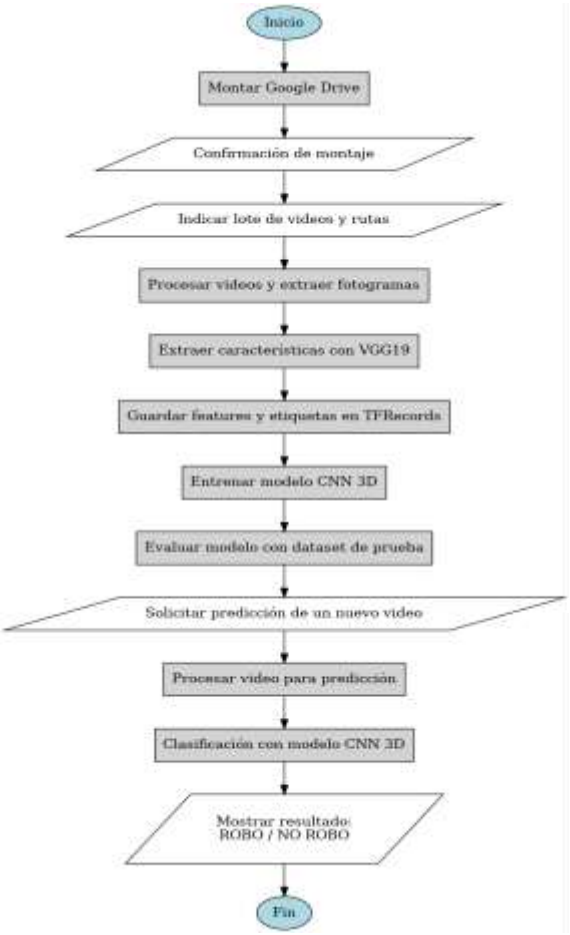

Figura 3.23. Diagrama de flujo del sistema. Elaboración propia.

3.5.4 Interfaz gráfica

La interfaz gráfica mostrada en la figura 3.24, corresponde al sistema de detección de conductas delictivas en el robo de autopartes con CNN 3D. Su diseño es sencillo y funcional, orientado a la interacción directa con el usuario para la visualización de resultados.

En la parte superior se encuentra el título “Detección de robo de autopartes”, resaltado en color rojo para captar la atención. Debajo, se muestra el resultado del análisis del modelo con un mensaje en letras mayúsculas y color verde oscuro: “EL VIDEO NO ES ROBO.”, lo que comunica de manera clara y rápida la conclusión del sistema tras procesar el video.

En el centro de la interfaz se presenta un recuadro de reproducción de video, donde se puede observar el video analizado por la CNN 3D. Este componente permite al usuario visualizar el contenido y validar la predicción del sistema, reforzando la transparencia y confiabilidad del análisis.

En la parte inferior, se incluyen dos botones de interacción:

- Reproducir (en color verde), para iniciar la visualización del video.
- Detener (en color rojo), para pausar o finalizar la reproducción.

El esquema de colores está diseñado para transmitir mensajes claros: el verde asociado a acciones seguras o de confirmación, y el rojo para acciones de detención o advertencia. En conjunto, la interfaz logra un equilibrio entre usabilidad y simplicidad, permitiendo al usuario identificar de forma inmediata si el sistema ha detectado o no una conducta delictiva en el video procesado.

Figura 3.24. Interfaz gráfica del sistema.
Elaboración propia.

CAPÍTULO 4

4 Resultados

4.1 Resultados de los 20 modelos obtenidos del entrenamiento por lotes de la CNN 3D

Como se muestra en la tabla. 4.1, se obtuvieron 20 modelos a partir del proceso de entrenamiento con la CNN 3D. Los modelos fueron capaces de detectar patrones y características temporales importantes en los videos para distinguir entre conductas de Robo y No Robo. Estos modelos mostraron una mejora en la capacidad y precisión en la detección de conductas delictivas en el robo de autopartes.

Tabla 4.1. Métricas obtenidas por los 20 entrenamientos de la CNN 3D. Elaboración propia.

Entrenamientos	1		2		3		4		5	
Clases	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo
Precision	0.91	1.00	1.00	1.00	0.83	0.64	1.00	1.00	1.00	1.00
Recall	1.00	0.89	1.00	1.00	0.5	0.9	1.00	1.00	1.00	1.00
F1-Score	0.95	0.94	1.00	1.00	0.62	0.75	1.00	1.00	1.00	1.00
Support	10	10	10	10	10	10	9	10	10	10
Accuracy	0.95		1.00		0.71		1.00		1.00	
Entrenamientos	6		7		8		9		10	
Clases	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo
Precision	0.69	1.00	1.00	1.00	0.91	1.00	1.00	1.00	0.83	1.00
Recall	1.00	0.50	1.00	1.00	1.00	0.88	1.00	1.00	1.00	0.80
F1-Score	0.82	0.67	1.00	1.00	0.95	0.93	1.00	1.00	0.91	0.89
Support	10	9	10	6	10	6	10	10	10	10
Accuracy	0.78		1.00		0.95		1.00		0.91	
Entrenamientos	11		12		13		14		15	
Clases	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo
Precision	0.91	1.00	1.00	1.00	0.77	1.00	0.88	0.82	0.90	0.90
Recall	1.00	0.90	1.00	1.00	1.00	0.70	0.78	0.90	0.90	0.90
F1-Score	0.95	0.95	1.00	1.00	0.87	0.82	0.90	0.86	0.90	0.90
Support	10	10	10	10	10	10	9	10	10	10
Accuracy	0.95		1.00		0.86		0.86		0.90	
Entrenamientos	16		17		18		19		20	
Clases	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo	No Robo	Robo
Precision	0.80	0.75	0.67	1.00	1.00	0.91	0.80	0.78	1.00	0.50
Recall	0.80	0.75	1.00	0.38	0.90	1.00	0.80	0.78	0.40	1.00
F1-Score	0.80	0.75	0.80	0.55	0.95	0.95	0.80	0.78	0.57	0.67
Support	10	8	10	9	9	10	10	8	10	6
Accuracy	0.78		0.73		0.95		0.79		0.69	

Modelos y su precisión:

- Modelos con precisión perfecta (Accuracy = 1.00): 2, 4, 5, 7, 9, 12
- Modelos con alta precisión (0.90 - 0.99): 1, 8, 10, 11, 15, 18
- Modelos con precisión media (0.75 - 0.89): 6, 13, 14, 16, 17, 19
- Modelos con baja precisión (< 0.75): 3, 20

En general, 6 modelos lograron una precisión perfecta, pero al analizar estos modelos podrían estar en sobreajuste, lo que significa que podrían haber aprendido demasiado bien los datos de entrenamiento y no pudieron procesar los nuevos datos. Mientras que otros 6 tuvieron una precisión superior al 90% dando un buen entrenamiento de estos modelos que se usaron para revisar la clasificación de los videos e identificar las conductas delictivas en el robo de autopartes. Los modelos con menor rendimiento fueron el 3 y el 20, con precisiones de 0.71 y 0.69, respectivamente. Esto podría deberse a una falta de información del conjunto de datos seleccionado, lo que limita la capacidad del modelo para extraer características relevantes y genera un aprendizaje insuficiente para la detección de las conductas delictivas.

4.2 Resultados de los modelos obtenidos de la creación de los conjuntos de datos para bajar el consumo computacional alto

Para comparar el desempeño de los cuatro conjuntos de datos obtenidos en la clasificación de conductas delictivas relacionadas con el robo de autopartes utilizando CNN 3D, se aplicaron tres métricas: precisión (precision), exhaustividad (recall) y F1-score. Estas métricas se reportan en las tablas 4.2, 4.3, 4.4 y 4.5 calculadas para cada modelo de aprendizaje generado a partir de los diferentes conjuntos de datos. Los mejores resultados, en términos de F1-score, se destacan en negrita.

Tabla 2.2. Rendimiento de clasificación del primer conjunto de datos - Original.

Clases	Precision	Recall	F1-Score
No Robo	0.91	1.00	**0.95**
Robo	1.00	0.89	0.94

Tabla 4.3. Rendimiento de clasificación del segundo conjunto de datos: corte de frame 1 a 1.

Clases	Precision	Recall	F1-Score
No Robo	0.88	0.70	0.78
Robo	0.73	0.89	**0.80**

Tabla 4.4. Rendimiento de clasificación del tercer conjunto de datos: corte de 1 a 2 frames.

Clases	Precision	Recall	F1-Score
No Robo	0.80	0.80	**0.80**
Robo	0.80	0.80	**0.80**

Tabla 4.5. Rendimiento de clasificación del tercer conjunto de datos: corte de 1 a 3 frames.

Clases	Precision	Recall	F1-Score
No Robo	1.00	0.70	0.82
Robo	0.77	1.00	**0.87**

Según los resultados presentados en la tabla 3.5, se observa el desempeño de la clasificación para las clases 'Robo' y 'No Robo'. La técnica de recorte de 1-3 frames por video contribuyó a mejorar significativamente el rendimiento del modelo de aprendizaje.

4.3 Mejoramiento de la carga computacional al entrenar la CNN 3D con los nuevos conjuntos de datos

Por otro lado, se analizó la carga computacional de cada conjunto de datos obtenidos mediante el recorte por frames como se muestra en la figura 4.1. Se observó una reducción en el uso de RAM, RAM de la GPU y espacio en disco. El conjunto de datos con recorte de 1-1 frames mostró el mejor rendimiento en términos de carga computacional, mientras que el conjunto de datos con recorte de 1-2 frames presentó el mejor desempeño en la clasificación del modelo de aprendizaje.

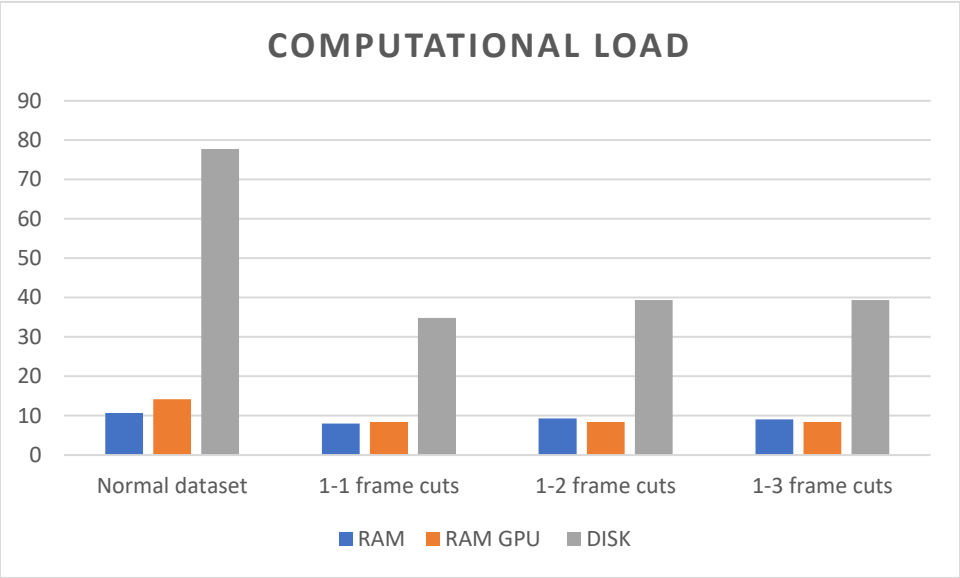

Figura 4.1. Carga computacional de los conjuntos de datos entrenados con CNN 3D en Google Colab. Elaboración propia.

Determinar cuál modelo de aprendizaje es mejor entre uno de los conjuntos de datos entrenados con respecto a métricas o carga computacional es ambigua. Un modelo de aprendizaje con alta precisión como 0.95 es preferible, pero en este caso presentó un mayor costo computacional, el cual limitaba los recursos presentando una saturación del mismo.

Por otro lado, los conjuntos de datos obtenidos de los recortes de frames y los resultados encontrados en los modelos de aprendizaje posterior a su entrenamiento, presentan una disminución considerable en el costo computacional, logrando una mayor rendimiento. Además permite trabajar con un mayor número de lotes de datos. Esto, a su vez, facilita la obtención de un nuevo modelo de aprendizaje capaz de identificar conductas delictivas en el robo de autopartes de manera más efectiva.

Es de destacar la importancia de asegurar que los recursos de datos necesarios estén adecuadamente preparados y accesibles para cualquier etapa del procesamiento. Al garantizar una correcta gestión de archivos y configuraciones, se pueden mitigar estos errores y mantener la eficiencia en el flujo de trabajo.

4.4 Comparativa con otros modelos o arquitecturas

Se llevó a cabo el entrenamiento de un conjunto de datos con el objetivo de comparar la eficiencia de diferentes modelos de clasificación en la detección de conductas delictivas. Para ello, se exploraron arquitecturas de redes neuronales, incluyendo CNN 3D, LSTM, GRU, BiGRU y CNN-LSTM, cada una con ventajas específicas en la extracción y modelado de características espaciales y temporales.

El análisis comparativo del desempeño de estos modelos permitió evaluar su capacidad para clasificar con precisión las escenas y detectar patrones relevantes en el comportamiento delictivo. Como resultado, se identificó el modelo más efectivo siendo este la CNN 3D, que mostró un mayor rendimiento con base en las métricas elegidas.

En la tabla 4.6, se presentan los modelos generados para cada arquitectura junto con sus respectivas métricas de evaluación para determinar el enfoque más adecuado en la clasificación de las conductas delictivas en el robo de autopartes.

Tabla 4.6. Resultados de la comparativa del entrenamiento del conjunto de datos con la CNN 3D con otras arquitecturas. Elaboración propia.

	Clasificador	Precision	Recall	F1-Score	Accuracy
	CNN 3D	1.00	0.89	0.95	0.95
	LSTM	1.00	0.78	0.88	0.89
Robo	GRU	0.47	1.00	0.64	0.47
	BiGRU	0.53	1.00	0.69	0.37
	CNN-LSTM	0.83	1.00	0.91	0.90
	Clasificador	**Precision**	**Recall**	**F1-Score**	
	CNN 3D	0.91	1.00	0.95	
No Robo	LSTM	0.83	1.00	0.91	
	GRU	0.00	0.00	0.00	
	BiGRU	0.00	0.00	0.00	
	CNN-LSTM	0.91	1.00	0.95	

Los resultados mostraron que la CNN 3D, CNN-LSTM y LSTM fueron las arquitecturas más efectivas para la clasificación de videos de "Robo" y "No Robo", alcanzando los mejores valores en precisión, recall y F1-Score. CNN 3D destacando con un F1-Score de 0.95 y una exactitud de 0.95, posteriormente la CNN-LSTM obtuvo un F1-Score de 0.91 y exactitud de 0.90. LSTM, aunque con una precisión de 1, presentó un recall de 0.78, lo que redujo su F1-

Score a 0.88 y su exactitud a 0.89. En comparación, los modelos GRU y BiGRU mostraron un rendimiento inferior, especialmente en la clasificación de "No Robo", donde sus métricas fueron nulas, dejando limitantes en la detección de esta clase.

CAPÍTULO 5

5 Conclusiones

El desarrollo de la presente tesis permitió la implementación de una metodología sólida para llevar a cabo un sistema para la detección de conductas delictivas en el robo de autopartes utilizando CNN 3D. A través de una serie de etapas que involucraron desde la creación y optimización del conjunto de datos hasta la ejecución y análisis de los modelos entrenados, se logró alcanzar resultados significativos tanto en términos de precisión como de un óptimo consumo computacional.

Una de las principales dificultades que se presentó en el proceso fue el alto consumo computacional, principalmente debido a las limitaciones de memoria en la plataforma Google Colab. Este problema se abordó de manera efectiva mediante la implementación de TFRecords, una librería de TensorFlow que permitió convertir los videos en archivos binarios optimizados, almacenados en disco en lugar de la memoria RAM. Esta solución no solo redujo el uso de memoria, sino que también mejoró el rendimiento de la carga y procesamiento de los datos, lo que permitió realizar un entrenamiento más eficiente de la CNN 3D. El uso de esta estrategia también favoreció el aprovechamiento de la GPU disponible, acelerando el procesamiento y optimizando la capacidad de cómputo.

El procesamiento por lotes, permitió una carga más equilibrada y controlada haciendo más eficiente los recursos computacionales brindados por Google Colab. Al dividir los 2000 videos en 10 subconjuntos de 100 videos cada uno, se evitó la saturación de la memoria y se mejoró el tiempo de procesamiento, permitiendo entrenar el modelo de manera más eficaz.

El reto de la gran cantidad de datos generados por los videos se abordó mediante el recorte de frames lo que permitió optimizar el consumo computacional y mejorar el rendimiento del modelo. Se realizó mediente una reducción progresiva del número de frames, eliminando frames pares, y luego aplicando recortes de 2 y 3 frames. Este ajuste resultó en una disminución significativa del número de frames, lo que permitió reducir la carga computacional sin sacrificar la precisión del modelo en gran medida.

El análisis de los modelos entrenados mostró que la reducción de frames mejoró la eficiencia computacional, lo que se reflejó en una menor saturación de recursos durante el entrenamiento. Sin embargo, aunque se logró un balance entre el costo computacional y la precisión del modelo, se observó que los modelos con precisión perfecta indican un sobreajuste en los datos de entrenamiento, lo que evidenció que no se podrían generalizar bien en datos nuevos. Esta observación resalta la necesidad de seguir explorando y ajustando las técnicas de preprocesamiento de datos para evitar el sobreajuste.

Los resultados obtenidos del entrenamiento de los 20 modelos de CNN 3D mostraron que fueron capaces de clasificar correctamente entre las clases "Robo" y "No Robo". De los 20 modelos, seis lograron una precisión perfecta, mientras que otros seis alcanzaron una precisión superior al 90%. No obstante, los modelos con menor rendimiento presentaron precisiones por debajo del 75%, lo que sugiere que la calidad del conjunto de datos o la configuración del modelo podría ser optimizada aún más. La clasificación de los videos de "Robo" y "No Robo" mostró una mejora significativa al utilizar técnicas de recorte de frames, con el modelo que recortó de 1-2 frames alcanzando el mejor desempeño en términos de F1-score.

Además de la CNN 3D, se evaluaron otras arquitecturas, como LSTM, GRU, BiGRU y CNN-LSTM, en un esfuerzo por determinar la arquitectura más eficaz para la detección de conductas delictivas. La CNN 3D mostró ser la más efectiva, superando a las otras arquitecturas en términos de precisión y capacidad para capturar patrones espaciales y temporales en los videos. La capacidad de la CNN 3D para analizar secuencias de frames en su dimensión temporal y espacial la hace particularmente adecuada para esta tarea de detección de eventos delictivos en videos.

En conclusión, el uso de CNN 3D para la detección de conductas delictivas en el robo de autopartes ha demostrado ser una metodología prometedora. La optimización del consumo computacional mediante TFRecords y la reducción de frames han sido esenciales para permitir entrenar modelos eficientes y precisos a pesar de las limitaciones de recursos. Sin embargo, se debe seguir explorando estrategias para mejorar aún más la generalización de los modelos, evitando el sobreajuste y asegurando que los modelos puedan adaptarse a nuevas situaciones o tipos de datos.

Los resultados obtenidos en la clasificación de videos mediante CNN 3D demuestran el cumplimiento de los objetivos planteados en este proyecto. Se logró desarrollar un sistema capaz de identificar conductas delictivas relacionadas con el robo de autopartes, alcanzando una alta precisión en la clasificación. De los 20 modelos entrenados, seis obtuvieron una precisión perfecta (Accuracy = 1.00), mientras que otros seis superaron el 90% de precisión, lo que confirma la efectividad de la metodología aplicada.

La implementación de estrategias como el uso de TFRecords para optimizar el consumo computacional, la reducción de frames para mejorar la eficiencia y el procesamiento por lotes permitió entrenar los modelos de manera eficiente en Google Colab, aprovechando sus recursos computacionales avanzados. Además, la recolección y clasificación de un conjunto de datos representativo con 2000 videos facilitó la correcta identificación de las categorías "Robo" y "No Robo".

A pesar de estos avances, algunos modelos presentaron una precisión inferior al 75%, lo que sugiere la necesidad de seguir explorando ajustes en la arquitectura y preprocesamiento de los datos para mejorar la generalización. No obstante, los resultados obtenidos confirman que el sistema desarrollado tiene un alto potencial para contribuir a la detección de robos de autopartes, proporcionando una herramienta útil para la seguridad ciudadana.

5.1 Trabajo futuro

Futuras investigaciones podrían mejorar los modelos de detección de conductas delictivas en el robo de autopartes utilizando técnicas como Bagging y Stacking las cuales permiten combinar varios modelos entrenados en subconjuntos de datos, mejorando la precisión y reduciendo el sobreajuste, mejorando la generalización al combinar predicciones de modelos base. Además, el uso de técnicas de data augmentation podría generar datos sintéticos, aumentando la diversidad del conjunto de entrenamiento y optimizando el rendimiento de los modelos en contextos dinámicos

6 Referencias

Agripino, G., & Felipe, L. (2011). Delincuencia organizada: Una amenaza emergente para el Estado mexicano. *Letras Jurídicas (Ocotlán)*, *12*, Article 12.

Amrutha, C. V., Jyotsna, C., & Amudha, J. (2020). Deep learning approach for suspicious activity detection from surveillance video. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 335-339.

Anacona, C. A. R. (2022). *Trastorno disocial: Evaluación, tratamiento y prevención de la conducta antisocial en niños y adolecentes*. Editorial El Manual Moderno.

Andreu, J. M., & Peña, M. E. (2013). Propiedades psicométricas de la Escala de Conducta Antisocial y Delictiva en adolescentes. *Anales de psicología*, *29*(2), 516-522.

Arana, C. (2021). *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales*. Serie Documentos de Trabajo. https://www.econstor.eu/bitstream/10419/238422/1/797.pdf

Ayora, M. J. M. (2024). *Clasificación de imágenes usando redes neuronales convolucionales en Python*.

Bishop, C. M. (2006). *Pattern recognition and machine learning: Springer New York*.

Bravo, E. F. C. (2009). *Una aproximación práctica a las redes neuronales artificiales*. Universidad del Valle. https://books.google.es/books?hl=es&lr=&id=ZX2nEAAAQBAJ&oi=fnd&pg=PA5&dq=.+Una+aproximaci%C3%B3n+Pr%C3%A1ctica+a+las+Redes+Neuronales+Artificiales&ots=3K53MpnEse&sig=oeOOmEzbgoYGKK4HS_o4hhzChw0

Canter, D. (2004). Offender profiling and investigative psychology. *Journal of Investigative Psychology and Offender Profiling*, *1*(1), 1-15. https://doi.org/10.1002/jip.7

Canter, D. V. (2011). Resolving the Offender "Profiling Equations" and the Emergence of an Investigative Psychology. *Current Directions in Psychological Science*, *20*(1), 5-10. https://doi.org/10.1177/0963721410396825

Carrera-Levillain, P., & Fernandez-Dols, J.-M. (1994). Neutral faces in context: Their emotional meaning and their function. *Journal of Nonverbal Behavior*, *18*(4), 281-299. https://doi.org/10.1007/BF02172290

Chackravarthy, S., Schmitt, S., & Yang, L. (2018). Intelligent crime anomaly detection in smart cities using deep learning. *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, 399-404.

Code Generation for Object Detection by Using YOLO v2—MATLAB & Simulink—MathWorks América Latina. (s. f.). Recuperado 24 de junio de 2023, de https://la.mathworks.com/help/gpucoder/ug/code-generation-for-object-detection-using-YOLOv2.html?searchHighlight=videofile&s_tid=srchtitle_videofile_1

Comunicación no verbal: En qué consiste y cómo interpretarla | Vistaprint. (2023, julio 3). *El blog de Vistaprint*. https://www.vistaprint.es/hub/comunicacion-no-verbal-en-que-consiste-y-como-interpretarla/

Create object to write video files—MATLAB - MathWorks América Latina. (s. f.). Recuperado 24 de junio de 2023, de https://la.mathworks.com/help/matlab/ref/videowriter.html?searchHighlight=VideoWriter&s_tid=srchtitle_VideoWriter_1

Create table from file—MATLAB readtable—MathWorks América Latina. (s. f.). Recuperado 24 de junio de 2023, de https://la.mathworks.com/help/matlab/ref/readtable.html?searchHighlight=readtable&s_tid=srchtitle_readtable_1

Díez, R. P., Gómez, A. G., & Martínez, N. de A. (2001). *Introducción a la inteligencia artificial: Sistemas expertos, redes neuronales artificiales y computación evolutiva*. Universidad de Oviedo.

Duarte Duarte, E. (2022). *Tras la pista de un gesto: La comunicación no verbal y la delincuencia común*. http://repository.unimilitar.edu.co/handle/10654/40596

Eduardo Francisco Caicedo Bravo, Humberto Loaiza Correa, & Duber Martinez Torres. (2019). Online learning of contexts for detecting suspicious behaviors in surveillance videos. *Image and Vision Computing*, *89*, 197-210.

Esan, D. O., Owolawi, P. A., & Tu, C. (2020). Detection of Anomalous Behavioural Patterns In University Environment Using CNN-LSTM. *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 1-8.

Evalúa, M. (2021, enero 14). ENSU: Los robos y la extorsión repuntan en el segundo semestre de 2020. *México Evalúa*. https://www.mexicoevalua.org/ensu-los-robos-y-la-extorsion-repuntan-en-el-segundo-semestre-de-2020/

Fan Zhou, Shupei Chen, Jin Wu, Chengtai Cao, & Shengming Zhang. (2021, enero 1). *Trajectory-User Linking via Graph Neural Network*. ICC 2021 - IEEE International Conference on Communications, Place of Publication: Piscataway, NJ, USA; Montreal, QC, Canada. Country of Publication: USA. https://doi.org/10.1109/ICC42927.2021.9500836

Figura 1. Conceptualización de una red neuronal artificial como un... (s. f.). ResearchGate. Recuperado 3 de julio de 2023, de https://www.researchgate.net/figure/Figura-1-Conceptualizacion-de-una-red-neuronal-artificial-como-un-sistema-Figure-1_fig1_335360392

Figura 2. Representación de la red neuronal convolucional utilizada en... (s. f.). ResearchGate. Recuperado 3 de julio de 2023, de https://www.researchgate.net/figure/Figura-2-Representacion-de-la-red-neuronal-convolucional-utilizada-en-el-modelo-Fuente_fig2_348825166

Gandapur, M. Q. (2022). E2E-VSDL: End-to-end video surveillance-based deep learning model to detect and prevent criminal activities. *Image and Vision Computing*, *123*, 104467. https://doi.org/10.1016/j.imavis.2022.104467

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, *12*(10), 2451-2471.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, *4*, 2047-2052. https://doi.org/10.1109/IJCNN.2005.1556215

Guardia Vaca, D. L., & Sandoval Alcocer, J. P. (2018). Una técnica de muestreo para categorizar videos. *Acta Nova*, *8*(4), 631-650.

Hassan, M. ul. (2018, noviembre 20). *VGG16—Convolutional Network for Classification and Detection*. https://neurohive.io/en/popular-networks/vgg16/

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. *Genetic Programming and Evolvable Machines*, *19*(1-2), 305-307. https://doi.org/10.1007/s10710-017-9314-z

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Ivan, C., Juan, P., Juan, R., Ferney, H., & Fabio, D. (2013). Implementación de una red neuronal artificial tipo SOM en una FPGA para la resolución de trayectorias tipo laberinto. *2013 II International Congress of Engineering Mechatronics and Automation (CIIMA)*, 1-6. https://ieeexplore.ieee.org/abstract/document/6682790/

Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. *Departamento de Física, Universidad de Concepción Chile*. https://www.academia.edu/download/36957207/Redes_neuronales.pdf

Jaramillo, C. D. (2021). Utilización del sistema de reconocimiento facial para preservar la seguridad ciudadana. *El Criminalista Digital. Papeles de Criminología*, *9*, 20-37.

Jaworek-Korjakowska, J., Kleczek, P., & Gorgon, M. (2019). Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0-0. http://openaccess.thecvf.com/content_CVPRW_2019/html/ISIC/Jaworek-Korjakowska_Melanoma_Thickness_Prediction_Based_on_Convolutional_Neural_Network_With_VGG-19_CVPRW_2019_paper.html

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, *35*(1), 221-231.

Jiang, Y.-G., Wu, Z., Tang, J., Li, Z., Xue, X., & Chang, S.-F. (2018). Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, *20*(11), 3137-3147.

Jindal, S., Sachdeva, M., & Kushwaha, A. K. S. (2022). Deep Learning for Video Based Human Activity Recognition: Review and Recent Developments. En R. C. Bansal, A. Zemmari, K. G. Sharma, & J. Gajrani (Eds.), *Proceedings of International Conference on Computational Intelligence and Emerging Power System* (pp. 71-83). Springer. https://doi.org/10.1007/978-981-16-4103-9_7

Kazdin, A. E., & Casal, G. B. (1999). *Conducta antisocial: Evaluación, tratamiento y prevención en la infancia y adolescencia*. Pirámide. https://dialnet.unirioja.es/servlet/libro?codigo=140340

Khalifa, A. B., & Frigui, H. (2016). *Multiple Instance Fuzzy Inference Neural Networks* (arXiv:1610.04973). arXiv. http://arxiv.org/abs/1610.04973

Kirichenko, L., Radivilova, T., Sydorenko, B., & Yakovlev, S. (2022). Detection of Shoplifting on Video Using a Hybrid Network. *Computation*, *10*(11), 199.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90. https://doi.org/10.1145/3065386

Kuppusamy, P., & Bharathi, V. C. (2022). Human abnormal behavior detection using CNNs in crowded and uncrowded surveillance–A survey. *Measurement: Sensors*, *24*, 100510.

Kuppusamy, P., & Hung, C.-L. (2021). Enriching the Multi-Object Detection using Convolutional Neural Network in Macro-Image. *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 1-5.

Laughlin, B., Sankaranarayanan, K., & El-Khatib, K. (2020). A service architecture using machine learning to contextualize anomaly detection. *Journal of Database Management (JDM)*, *31*(1), 64-84.

LeCun, Y., Bengio, Y., & Hinton, G. (2015b). Deep learning. *nature*, *521*(7553), 436-444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*(6), 335-346.

Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P., & Dollar, P. (2020). IEEE Transactions on Pattern Analysis and Machine Intelligence. *IEEE Int. Conf. Comp. Vis.*, *42*, 318-327.

López, R. F., & Fernández, J. M. F. (2008). *Las Redes Neuronales Artificiales*. Netbiblo.

Maqsood, R., Bajwa, U. I., Saleem, G., Raza, R. H., & Anwar, M. W. (2021). Anomaly recognition from surveillance videos using 3D convolution neural network. *Multimedia Tools and Applications*, *80*(12), 18693-18716. https://doi.org/10.1007/s11042-021-10570-3

Martínez, M. Y., Molina, M. M., García, N. M., & López, E. V. (2022). Técnicas de aprendizaje supervisado para la detección y clasificación de enfermedades y defectos en imágenes de frutas: Revisión. *Magazine de las Ciencias: Revista de Investigación e Innovación*, *7*(1), 1-16.

Martínez-Mascorro, G. A., Abreu-Pederzini, J. R., Ortiz-Bayliss, J. C., & Terashima-Marín, H. (2020). Suspicious behavior detection on shoplifting cases for crime prevention by using 3D convolutional neural networks. *arXiv preprint arXiv:2005.02142*.

Mueller, J. P., & Massaron, L. (2021). *Machine Learning For Dummies*. John Wiley & Sons.

Nacelle, A., & Mizraji, E. (2009). Redes neuronales artificiales. *Núcleo de ingeniería biomédica–Universidad de la Republica Uruguay*.

Navalgund, U. V., & Priyadharshini, K. (2018). Crime intention detection system using deep learning. *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 1-6.

ONU-Habitat—Violencia e inseguridad en las ciudades. (2022, noviembre 20). https://onuhabitat.org.mx/index.php/violencia-en-inseguridad-en-las-ciudades

Pechyonkin, M. (2018, diciembre 18). Key Deep Learning Architectures: LeNet-5. *Medium*. https://medium.com/@pechyonkin/key-deep-learning-architectures-lenet-5-6fc3c59e6f4

Piquero, A. R., Farrington, D. P., & Blumstein, A. (2003). The criminal career paradigm. *Crime and justice*, *30*, 359-506.

Powell González, J. E. (2021). *Detección de peleas en videos usando estimación de postura y bi-lstm*. https://repositorioinstitucional.buap.mx/items/343d2184-c670-4d24-b50b-2f71ee2ebcb4

Powers, D. M. W. (2020). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation* (arXiv:2010.16061). arXiv. https://doi.org/10.48550/arXiv.2010.16061

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

Red neuronal de ajuste de funciones—MATLAB fitnet—MathWorks América Latina. (s. f.). Recuperado 24 de junio de 2023, de https://la.mathworks.com/help/deeplearning/ref/fitnet.html?searchHighlight=fitnet&s_tid=srchtitle_fitnet_1

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Russo, C., Ramón, H., Alonso, N., Cicerchia, B., Esnaola, L., & Tessore, J. P. (2016). Tratamiento masivo de datos utilizando técnicas de machine learning. En *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*. Universidad Nacional de Entre Ríos. http://repositorio.unnoba.edu.ar/xmlui/handle/23601/107

Sandoval, L. J. (2018). *ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA ANÁLISIS Y PREDICCIÓN DE DATOS*.

Semáforo Delictivo. (2025, noviembre 20). http://www.semaforo.com.mx/

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (arXiv:1409.1556). arXiv. http://arxiv.org/abs/1409.1556

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2024). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*.

UCI Machine Learning Repository: Wine Data Set. (s. f.). Recuperado 31 de marzo de 2023, de https://archive.ics.uci.edu/ml/datasets/wine

Vallathan, G., John, A., Thirumalai, C., Mohan, S., Srivastava, G., & Lin, J. C.-W. (2021). Suspicious activity detection using deep learning in secure assisted living IoT environments. *The Journal of Supercomputing*, *77*(4), 3242-3260.

VIZCAYA CARDENAS, R., FLORES ALBINO, J., LANDASSURI MORENO, V., & LAZCANO SALAS, S. (2017). *Desempeño de una Red Neuronal Convolucional para Clasificación de Señales de Tránsito*. https://core.ac.uk/download/pdf/154796764.pdf

Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., & Xue, X. (2015). Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. *Proceedings of the 23rd ACM international conference on Multimedia*, 461-470. https://doi.org/10.1145/2733373.2806222

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921-2929. http://openaccess.thecvf.com/content_cvpr_2016/html/Zhou_Learning_Deep_Features_CVPR_2016_paper.html