Desarrollo de un clasificador de video para la detección automática de eventos de asalto a peatones basado en algoritmos de aprendizaje profundo

Terán Zambrano, Cristhian Daniel

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización Y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica Automatización y Control

Ing. Silva Tapia, Rodrigo

04 de agosto del 2022

COPYLEAKS

Trabajo_Titulacion_CTeran_Final.pdf

Scanned on: 21:53 August 4, 2022 UTC

Identical Words	254
Words with Minor Changes	49
Paraphrased Words	246
Omitted Words	0

Ing. Rodrigo Silva Tapia
C.I. 0602199523

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Certificación

Certifico que el trabajo de titulación: "**Desarrollo de un clasificador de video para la detección automática de eventos de asalto a peatones basado en algoritmos de aprendizaje profundo** " fue realizado por el señor **Terán Zambrano, Cristhian Daniel;** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 05 de Agosto de 2022

Firma:

Firmado electrónicamente por:
RODRIGO SILVA TAPIA

.....

Silva Tapia, Rodrigo

C. C. 060219952-3

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Responsabilidad de Autoría

Yo, **Terán Zambrano, Cristhian Daniel**, con cédula de ciudadanía n°172358595-4, declaro que el contenido, ideas y criterios del trabajo de titulación**: Desarrollo de un clasificador de video para la detección automática de eventos de asalto a peatones basado en algoritmos de aprendizaje profundo** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 05 de Agosto de 2022

Firma

.....

Terán Zambrano, Cristhian Daniel

C.C. 172358595-4

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Autorización de Publicación

Yo, **Terán Zambrano, Cristhian Daniel**, con cédula de ciudadanían°172358595-4, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de un clasificador de video para la detección automática de eventos de asalto a peatones basado en algoritmos de aprendizaje profundo** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 05 de Agosto de 2022

Firma

.....

Terán Zambrano, Cristhian Daniel

C.C.: 172358595-4

Dedicatoria

Dedico este trabajo de manera especial a mi madre, Sandra Zambrano, que es la persona que ha confiado en mis capacidades y las ha fortalecido durante toda mi vida, sin ella habría sido muy difícil cumplir cada una de las metas que me he propuesto. Gracias a su ejemplo, apoyo, cariño y amor he podido convertirme en la persona que soy ahora.

Agradecimiento

A mis padres, Sandra Zambrano y Juan Terán, y a mi hermano, Francisco Terán, que han sido un gran soporte durante toda mi vida, compartiendo conmigo incondicionalmente tantos buenos y malos momentos.

A todas las personas que conocí y con las que compartí durante toda mi carrera universitaria, especialmente a mis amigos que sin ellos el camino habría sido más complicado y sobre todo aburrido, gracias por ser un apoyo.

A mis profesores que han sido una guía en mi proceso de aprendizaje, que son un referente para mi presente y mi futuro, a mi tutor, el Ing. Rodrigo Silva, que me ha apoyado en la culminación de mi carrera y me ha abierto la mente a temáticas de vanguardia que antes no las hubiera considerado.

Índice

Índice de Tablas

Índice de Figuras

Resumen

En la actualidad es común encontrar sistemas de videovigilancia gubernamentales operando en distintos lugares públicos tales como calles y plazas con la intención de monitorear de manera permanente algún incidente entre personas o vehículos que transitan por dichos lugares y de ser el caso, actuar oportunamente coordinando con las unidades de auxilio inmediato. Estos sistemas son operados por personal entrenado de evaluadores que trabajan muchas horas frente a monitores, quienes pueden sufrir distracciones motivadas por el cansancio mental y físico en sus largas jornadas diarias de trabajo. La tecnología puede ser un gran aliado para apoyar el trabajo de los evaluadores, ya que es posible realizar la detección de un incidente típico como el asalto a peatones, utilizando técnicas de visión por computadora, implementadas con modelos de redes neuronales convolucionales y recurrentes de aprendizaje profundo. En este trabajo, se ha implementado un clasificador de video haciendo algunos ajustes de un modelo de aprendizaje supervisado, combinando VGG16 y LSTM para la detección de incidentes de asalto a peatones. En la implementación del clasificador se hace uso de cientos de videoclips de corta duración y herramientas computacionales tales como Tensorflow, OpenCV, Google Colab y Paperspace Gradient para el entrenamiento, validación y pruebas del modelo.

Palabras clave: videovigilancia, visión por computadora, asalto a peatones, aprendizaje profundo, clasificador de video

Abstract

Nowadays, it is common to find government video surveillance systems in different public places such as streets and squares, with the purpose of permanently monitoring any incident between people or vehicles that pass through those places and if this is the case, act timely in coordination with immediate relief units. These systems are operated by trained staff of evaluators who work long time in front of monitors, who can suffer distractions due to mental and physical exhaustion in their daily long working hours. Technology could be a great ally to support the work of evaluators, since it is possible to detect a typical incident such as pedestrian assaults, using computer vision techniques, implemented with convolutional and recurrent deep learning models. In this work, a video classifier has been implemented by making some adjustments to a supervised learning model, combining VGG16 and LSTM for the detection of pedestrian assault incidents. In the implementation of the classifier, hundreds of short video clips, and computational tools such as Tensorflow, OpenCV, Google Colab and Paperspace Gradient are used for model training, validation, and testing.

Keywords: video surveillance, computer vision, pedestrians assaults, deep learning, video classifier.

Capítulo I

Introducción

Antecedentes

Los incidentes de asaltos a peatones están creciendo con muchas víctimas que quedan a merced de los delincuentes, mientras que las instituciones encargadas de la seguridad, en la mayoría de las ocasiones, no pueden reaccionar de manera oportuna ante la delincuencia, quedando el delito impune y una sensación de inseguridad en la ciudadanía (Fiscalía General del Estado, 2021).

Nuestro país cuenta con el Servicio Integrado de Seguridad SIS-ECU 911 como entidad pública preparada para coordinar la atención de organismos del Estado tales como la Policía Nacional, Fuerzas Armadas, Cuerpo de Bomberos, Comisión Nacional de Tránsito, Ministerio de Salud Pública, Instituto Ecuatoriano de Seguridad Social, Servicio Nacional de Gestión de Riesgos y Emergencias, Cruz Roja Ecuatoriana y otros organismos locales (ECU 911, 2021) ante emergencias provocadas por incidentes ocurridos en cualquier parte del territorio ecuatoriano.

La red nacional de videovigilancia del ECU 911 cuenta con alrededor de 6500 cámaras para monitorear en tiempo real y permanente dentro del territorio ecuatoriano distintos tipos de incidentes ocurridos en calles, parques, escuelas, colegios y medios de transporte público. Para realizar el monitoreo permanente de las cámaras, el ECU 911 mantiene personal de evaluadores trabajando durante las 24 horas del día y los 365 días del año (ECU 911, 2016). El trabajo prolongado de los evaluadores produce cansancio físico y mental que pueden causar distracciones de atención sobre los monitores en los cuales se muestran simultáneamente vídeos de más de 10 cámaras (Fernández Carrobles et al., 2019). Es lógico por tanto, que existan incidentes que escapen de la atención de los evaluadores y por consiguiente se hace necesario apoyarse en la tecnología capaz de detectar automáticamente cierto tipo de eventos y producir algún mensaje de alerta en la pantalla que observa el evaluador facilitando su tarea.

Justificación e Importancia

El desarrollo tecnológico en los sistemas de videovigilancia está promoviendo la instalación masiva de cámaras de vídeo en lugares de acceso público tales como parques, aeropuertos, calles, escuelas, unidades de transporte, entre otros. La videovigilancia manual requiere de gran cantidad de tiempo y de personal especializado a cargo, es por ello que se ha venido investigando métodos y técnicas que permitan realizar esta tarea de una forma más óptima (Wu et al., 2019).

La investigación en el área de inteligencia artificial aplicada a la visión por computadora permitirá desarrollar medios tecnológicos para realizar tareas automáticas de seguimiento de peatones, análisis de comportamiento de multitudes, estadísticas de tráfico, detección de actividad, y otros (Wu et al., 2019).

La tecnificación de la videovigilancia muestra una tendencia de crecimiento, y se ha venido desarrollando en conjunto con la diversidad de tecnología, desarrollo e innovación que tienen como propósito atender ciertos problemas y situaciones. Como ejemplo se tiene sistemas de información geográfica utilizados en el análisis de la incidencia delictiva y generación de inteligencia, la atención de incidentes y delitos en tiempo real mediante geolocalización, revisión de personas y vehículos, controles de acceso, alarmas instaladas en comunidades y la vía pública, incluso aplicaciones móviles para evitar el acoso y otras tantas situaciones de interés público (Jasso López, Seguridad ciudadana y tecnología: uso, planeación y regulación de la videovigilancia en Latinoamérica, 2020).

Las técnicas de aprendizaje profundo, sus métodos y algoritmos, han logrado resultados importantes para la detección de eventos a través de la videovigilancia, permitiendo abordar problemas como la identificación de robos, situaciones de violencia, posibilidades de explosión, entre otras clases de eventos (Sreenuu & Saleem, 2019).

En este contexto, el proyecto propone una solución de ingeniería basada en técnicas de clasificación de vídeo para realizar la detección automática de asaltos a peatones en videos

capturados por cámaras de videovigilancia. Estas técnicas involucran la implementación de algoritmos de redes neuronales convolucionales que pueden implementarse posteriormente sobre pequeños dispositivos computacionales de alta integración.

Alcance del Proyecto

El proyecto consiste en el desarrollo de un clasificador de video basado en técnicas de aprendizaje profundo, con el fin de identificar un evento de asalto a peatones. Para la detección del evento se implementará algoritmos sobre redes neuronales convolucionales y también redes neuronales recurrentes.

Para las redes neuronales convolucionales se utilizará un modelo pre entrenado VGG16; que mediante transfer learning (transferencia de aprendizaje) extraerá las características de fotogramas tomados de un set de videos que muestran eventos de asalto en la calle. Luego mediante redes neuronales recurrentes, LSTM se encontrarán los patrones temporales de las características extraídas anteriormente para determinar automáticamente si en el video existe o no un evento de asalto a peatones y emitirá un mensaje de alerta en caso positivo.

La programación del modelo será realizada en lenguaje Python con sus librerías Tensorflow y Keras, para el desarrollo de redes neuronales. Así también para el procesamiento del video y extracción de los paquetes de información requeridos se utilizará OpenCV, herramienta de código abierto diseñada para aplicaciones de visión artificial.

La red neuronal del clasificador de video será entrenada, validada y probada en un computador utilizando videos de la base de datos UCF-Crime (Waqas et al., 2018) y también con videos relacionados con asalto a peatones recopilados del internet.

Objetivos

Objetivo General

Desarrollar un clasificador de video para la detección automática de eventos de asalto de peatones basado en algoritmos de aprendizaje profundo.

Objetivos Específicos

- Investigar los algoritmos de aprendizaje profundo para la clasificación videos
- Generar la base de datos con videos de eventos de asalto a peatones para el entrenamiento, validación y pruebas de un modelo de aprendizaje profundo.
- Seleccionar, configurar y entrenar un modelo de aprendizaje profundo para que realice la detección de eventos de asalto a peatones.
- Evaluar el desempeño del modelo.

Capítulo II

Marco Teórico

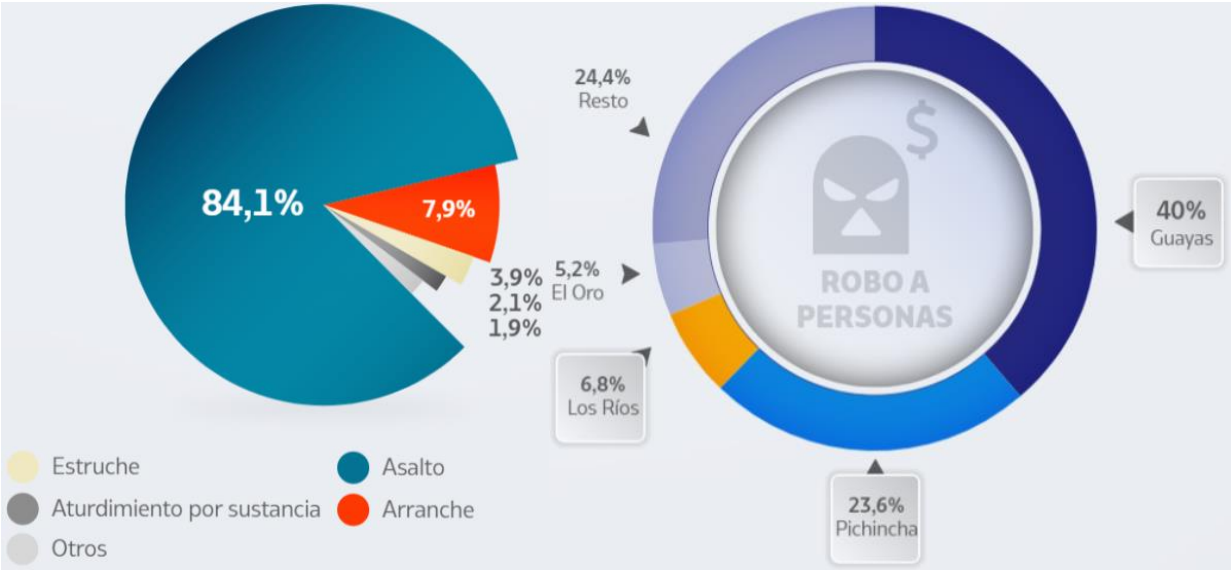
En este capítulo se presenta el fundamento teórico del trabajo realizado, el cual inicia con una breve explicación acerca del robo a personas, centrándose en la modalidad de asaltos y su presencia en el Ecuador, también se dará un repaso sobre detección de eventos anómalos y su aplicación, y se abordarán temas relacionados con el estudio de los elementos básicos de las redes neuronales artificiales y Deep Learning o Aprendizaje Profundo, que es la base de la solución propuesta. Finalmente se presentarán las principales herramientas de software utilizadas para el desarrollo del trabajo, principalmente Tensorflow y OpenCV además de una descripción breve de los entornos virtuales utilizados para el entrenamiento de las redes neuronales es decir Google Colaboratory y Paperspace Gradient.

Robo a personas

Los robos a personas son eventos en los que a una o varias personas, mediante amenazas o violencia, se les sustraiga o se apodere de un bien personal o del que sea custodio, que porte en el momento del hecho, sea en un lugar público o privado. El robo a personas en el periodo de Enero a Noviembre 2021 ascendió a la cifra de 22.614, sufriendo un aumento del 24,4% con respecto al 2020. El asalto es la modalidad que se encuentra con mayor frecuencia en el Ecuador con un 84.1% de los robos a personas siendo Guayas y Pichincha las provincias de mayor preocupación como se puede observar en la Figura 1 (Fiscalía General del Estado, 2021).

Figura 1

Robo a personas, modalidades y su presencia en el Ecuador



Nota. Adaptada de *Cifras de Robos*, de Fiscalía General del Estado, 2021.

Según la Real Academia Española (2021), asaltar es acometer repentinamente y por sorpresa, por otro lado, Oxford Languages (s.f.)define al asalto como un ataque contra una persona o entrada en una propiedad con intención de robar.

Seguridad Ciudadana

Según la Comisión Internacional de Derechos Humanos (2009), la seguridad es y ha sido una de las funciones principales de los Estados. Hoy en día se buscan modelos policiales considerando a la protección ciudadana, en un marco de respeto a las leyes y los derechos fundamentales, poniendo especial atención en las labores de prevención y control, en lugar de actividades represivas o reactivas frente a cualquier forma de violencia que afecte a la ciudadanía. En su análisis, Páez, Peón y Ramírez (2018) concluyen que el concepto de seguridad ciudadana ha evolucionado a través del concepto de seguridad nacional centrándose en la protección a la ciudadanía que conforma un país, y se plantea la generación de estrategias programas y planes en un trabajo conjunto del estado, la policía y la ciudadanía.

De acuerdo con el Ministerio del Interior (2019), la seguridad ciudadana se refiere a la protección de orden público, y busca dirigir políticas, planificación, regulación y gestión, apoyados en instituciones como la Policía Nacional, con el fin de garantizar los derechos de la ciudadanía, mantener la convivencia pacífica a través de la detección, disuasión, investigación y control del delito y la violencia, además de proteger de cualquier riesgo o amenaza a las personas y bienes jurídicos.

En Latinoamérica se han venido utilizando diversas técnicas para la implementación de programas y acciones que ayuden a mejorar la seguridad ciudadana, son los gobiernos los que hacen mayor uso de innovación y tecnología en este campo con el objetivo de atender problemas o situaciones específicas. Así se encuentran ejemplos como sistemas de información geográfica, generación de sistemas de inteligentes, videovigilancia, geolocalización de delitos e incidentes en tiempo real, control automático de acceso, detección de personas mediante escáneres, alarmas comunitarias, entre otras (Jasso López, 2020).

Videovigilancia

Una de las herramientas principales que se ha adoptado para la seguridad ciudadana es la videovigilancia, que permite monitorear eventos que pueden ser una amenaza o riesgo para la ciudadanía a través de cámaras situadas en lugares estratégicos.

Existen varios tipos de sistemas de videovigilancia que cuentan con características diferentes, y con ayuda de la inteligencia artificial permiten acciones como escaneo de rostros, identificación de patrones de comportamiento, registro de eventos, entre otros, éstos diferentes sistemas se han adoptado como parte de las estrategias de seguridad ciudadana con el fin de evitar el delito y disuadir a los criminales potenciales de cometerlos (Jasso López, 2020).

La videovigilancia a través de la tecnología ha permitido, mediante un conjunto de algoritmos, evaluar las diferentes situaciones y eventos que se presentan en una imagen o conjunto de imágenes, obtener información de estas para la toma de decisiones o la ejecución de órdenes dependiendo de la necesidad. Mediante el uso de la visión artificial se logra

detectar toda clase de objetos, animales y personas para evaluar sus características, patrones de comportamiento, particularidades que refuerzan el monitoreo con el objetivo de clasificar, seleccionar, y evaluar la trazabilidad de las personas u objetos observados. La automatización de estos procesos permite anticiparse a ciertas conductas o eventos anómalos, es decir, que se alejan de lo que “debería ser” y pueden alertar a los organismos responsables de la vigilancia sobre un posible cometimiento de algún delito o infracción (Pérez Esquivel, 2021).

Eventos Anómalos

Los eventos anómalos son hechos que ocurren sin previsión, de una forma repentina y que no siguen el patrón habitual de comportamiento (García Carretero & Parada Medina, 2020).

La detección de eventos anómalos puede ser usada para la solución de diferentes problemas de ingeniería, problemas económicos, sociales entre otros tantos. Uno de los retos es aplicar la detección de eventos anómalos en la videovigilancia, especialmente por su potencial de aplicación en cualquier tema de seguridad pública, lo que se viene logrando mediante la visión artificial y el uso de métodos basados en redes neuronales de aprendizaje profundo (Lin et al., 2021).

En la visión artificial, el aprendizaje de las características de un video en tiempo y espacio es un problema fundamental, el entendimiento del video y su clasificación pueden tener grandes aplicaciones, útiles para el ser humano. Tomando en cuenta que el video es una secuencia de imágenes o fotogramas que evoluciona en el tiempo, se vuelve relevante el estudio de dos señales visuales: las características de cada uno de los fotogramas y la relación temporal entre dichos fotogramas. (Wang et al., 2018).

Visión Artificial

La visión artificial es un campo de estudio en el que el objetivo principal es proveer, a una máquina, la capacidad de entender una imagen o conjunto de imágenes mediante su adquisición, procesamiento y análisis. Esta capacidad se consigue mediante el uso de algoritmos, modelos matemáticos, redes neuronales y diferentes técnicas computacionales que ayuden al análisis de las características de imágenes (Dynatec, 2021).

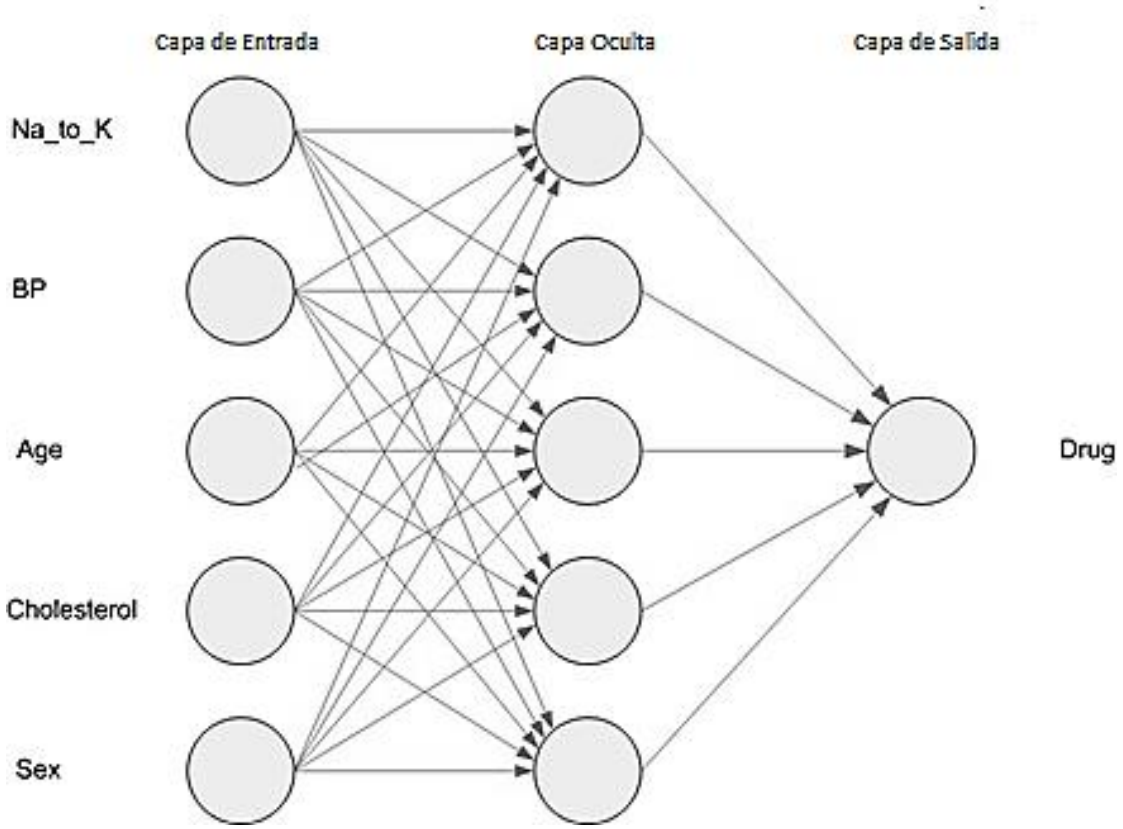
Uno de los grandes problemas de la visión artificial es la capacidad de procesamiento del ordenador, lo que en ocasiones impide crear aplicaciones que trabajen en tiempo real, siendo el objetivo actual buscar algoritmos y modelos cada vez mejores, que permitan encontrar un balance entre la eficacia del modelo y la capacidad de procesamiento necesaria para su funcionamiento.

Redes Neuronales Artificiales

Una red neuronal artificial es un procesador de información, de distribución altamente paralela, que contiene varias unidades sencillas de procesamiento denominadas neuronas. Son modelos imitan el funcionamiento del sistema nervioso, sus unidades básicas, las neuronas, son organizadas en capas las cuales son: capa de entrada también denominada capa sensorial que es la encargada de recibir la información, capa oculta e donde se produce el proceso de aprendizaje y va relacionando y adecuando los pesos sinápticos y capa de salida que proporciona el resultado de todo el procesamiento dentro de la red neuronal, un ejemplo se muestra en la Figura 2. Este tipo de modelos puede aproximar una gran cantidad de modelos predictivos, cuya relación será determinada durante el entrenamiento de la red (IBM, 2021).

Figura 2

Ejemplo de una red neuronal



Nota. Adaptada de *Estructura de una red neuronal*, de IBM, 2021, https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model

Características Principales y Elementos

Las redes neuronales artificiales cuentan con ciertas características y elementos fundamentales que permiten simular el comportamiento del sistema nervioso humano, por lo que su funcionamiento y estructura están basadas en las redes neuronales biológicas (Bautista Ramos, 2021).

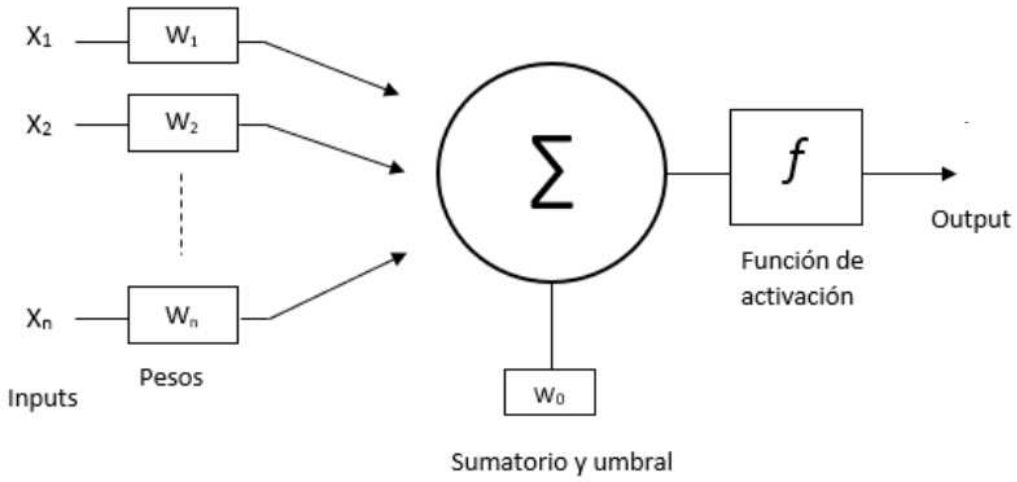
Las principales características se describen a continuación.

Neurona Artificial. Es una unidad básica de la red que recibe dos tipos de entradas: excitadoras e inhibidoras; estas entradas, representadas por valores numéricos, se multiplican por pesos sinápticos que indican una importancia relativa de cada una de ellas, se suman y si

el resultado de la suma supera un umbral, la neurona se activa, el umbral se define mediante una función de activación, la representación de esta estructura se puede ver en la Figura 3.

Figura 3

Representación de una neurona artificial con sus elementos



Nota. Adaptada de *Estructura de neurona biológica y artificial,* de S. Bautista, 2021, Universidad Carlos III de Madrid.

Función de red. Es una función que se forma a través de la combinación lineal de las entradas con los pesos sinápticos y de existir la suma de un sesgo como se muestra en la Ecuación 1.

$$Net_j = \sum_{i=1}^{n} W_{j,i}\, X_{i+\theta} \tag{1}$$

Donde W representa a los pesos entre la neurona i y la neurona j, Xi los valores de las entradas y θ el sesgo.

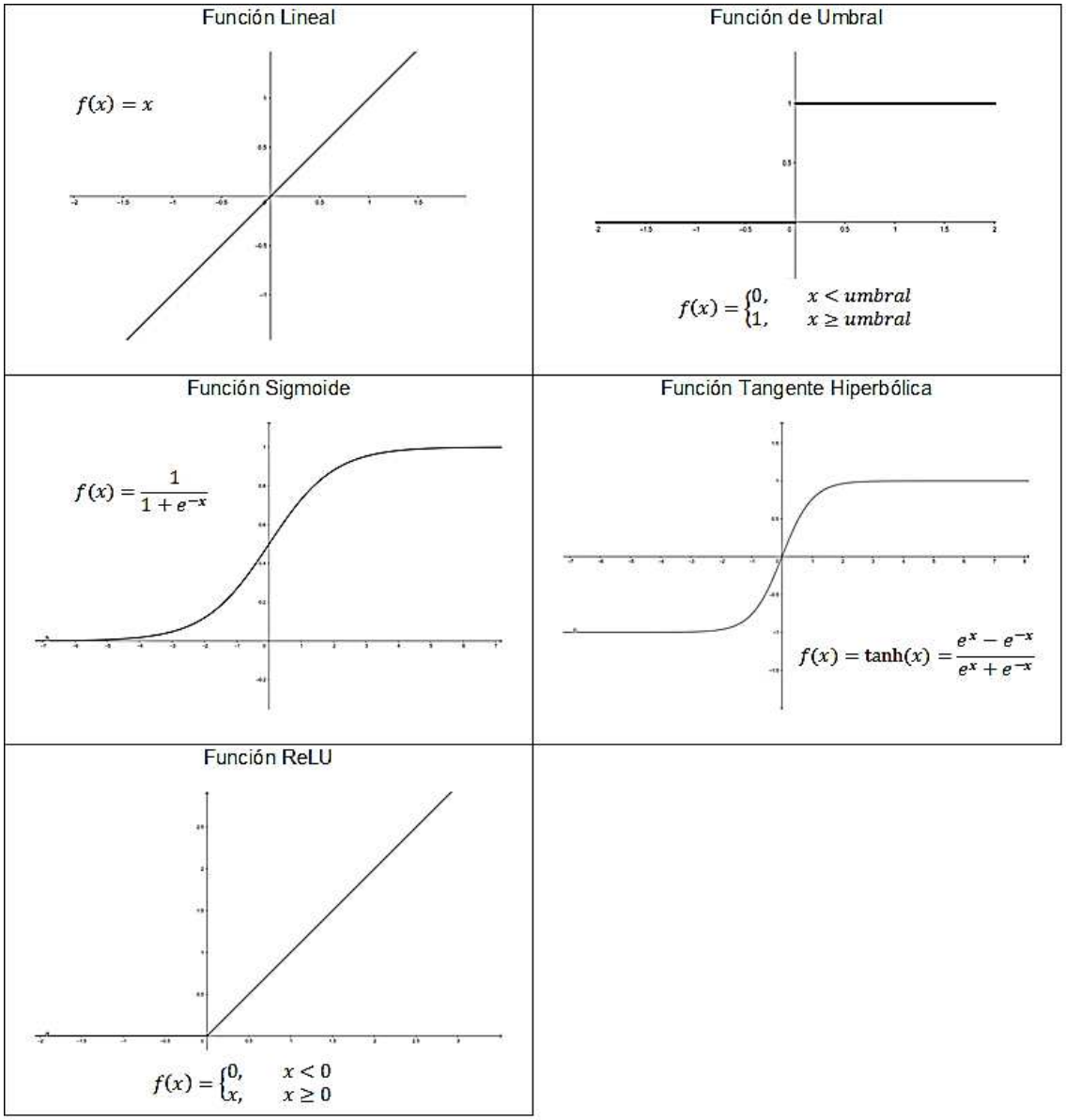
Función de Activación. Las funciones de activación transforman las entradas globales en un valor de activación, existen varios tipos de funciones como: lineales, de umbral y no lineales. Las funciones lineales generan una salida acorde con una regresión lineal, las funciones de umbral proporcionan una salida binaria como se muestra en la Ecuación 2.

$$f(Net) = \begin{cases} 0, & Net < umbral \\ 1, & Net \geq umbral \end{cases} \quad (2)$$

Por último, las funciones no lineales permiten crear relaciones complejas entre las entradas y las salidas, entre este tipo de funciones se pueden destacar: la función sigmoide cuya salida se encuentra en un rango de 0 a 1 y su representación gráfica toma la forma de una S, la función tangente hiperbólica es similar a la sigmoide sin embargo su rango se encuentra definido entre -1 y 1, y la función ReLU (Rectified Linear Unit) que está definida en un rango de 0 al infinito y tiene como característica que cuando su función y derivada son iguales a cero generan la desactivación completa de una neurona lo que puede ayudar en el proceso de descarte probabilístico de neuronas denominado “Dropout” (Bautista Ramos, 2021). A continuación, en la Figura 4 se muestran los tipos de funciones de activación antes mencionados.

Figura 4

Funciones de Activación y sus Ecuaciones

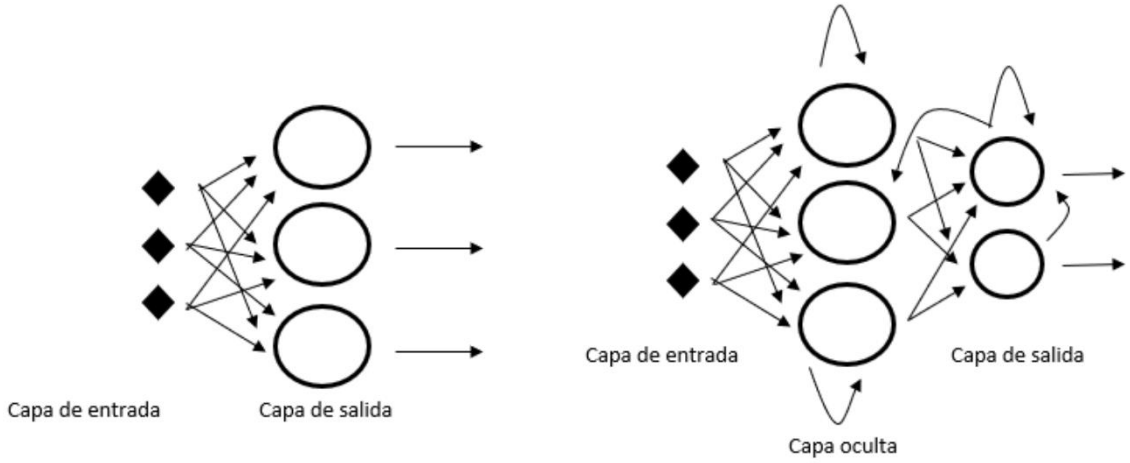


Arquitectura de una Red Neuronal Artificial. Determina la distribución de neuronas en las diferentes capas, cómo se distribuyen entre sí y cuál es su patrón de conexiones, pueden dividirse según el flujo de información en redes feedforward cuyo flujo de información va hacia adelante y redes recurrentes o realimentadas llamada redes feedback; considerando

su estructura en cambio existen redes monocapa o redes multicapa, todo esto se presenta en la Figura 5.

Figura 5

Representación de los Tipos de Redes Neuronales



Nota. Red neuronal monocapa feedforward y red neuronal multicapa feedback Reproducida de Redes feedforward monocapa y feedback multicapa de tres neuronas ocultas, de S.Bautista, 2021.

Redes feedforward. La información fluye únicamente de forma unidireccional desde la capa de entrada hacia la capa de salida.

Redes feedback. La información puede fluir entre neuronas de una misma capa, neuronas de distintos niveles, incluso puede fluir hacia la misma neurona, la función de estas redes en encontrar la activación de las neuronas de acuerdo con un período de tiempo.

Redes monocapa. Es una red neuronal simple en la que las entradas fluyen hacia una única capa de salida.

Redes multicapa. Es una red neuronal compuesta por varias capas intermedias denominadas capas ocultas en la cuales se realizan toda clase de cálculos.

Regla de Aprendizaje. Las redes neuronales son capaces de aprender por medio de entrenamiento, es conformada por sistemas de aprendizaje que de igual forma dependen del problema a resolver.

Entrenamiento. En principio la red en su fase de aprendizaje adapta los pesos sinápticos para que su respuesta sea correcta de acuerdo con los datos de entrada. Durante la fase de entrenamiento se tienen dos tipos de aprendizaje: aprendizaje supervisado y no supervisado

Aprendizaje Supervisado. Consiste en proveer a la red de parejas de estímulos y respuestas correctas, durante el entrenamiento la respuesta de la red neuronal a estos estímulos será comparada con la respuesta correcta, y dependiendo de esta comparación se ajustarán los pesos sinápticos de la red, la idea es que estos pesos vayan cambiando para que la respuesta que genera la red a los estímulos se acerque cada vez más a la respuesta correcta. Este proceso se realiza con todas las parejas de patrones y respuestas correctas, los pesos sinápticos cambian y se vuelve a realizar el proceso hasta que la red responda de forma adecuada terminando así la fase de entrenamiento ***(Gestal Pose, 2009)***.

Aprendizaje No Supervisado. En este tipo de aprendizaje no se provee a la red de la respuesta correcta, por lo que no existe comparación entre una respuesta correcta y la respuesta de la red neuronal, se espera que la red construya sus propias asociaciones y aprenda a distinguir ciertas características por sí misma ***(Gestal Pose, 2009)***.

Problemas en el entrenamiento. Durante el entrenamiento pueden encontrarse problemas como que la red no encuentre la exactitud adecuada o no sea capaz de ajustarse los suficiente a los datos presentados y se quede estancada independientemente del tiempo de entrenamiento, a esto se le denomina “underfitting”, en este caso es importante revisar que los datos sean significativos al problema a resolver, tanto en calidad como en cantidad, otra opción es cambiar el diseño de la red neuronal modificando las capas que conforman las mismas.

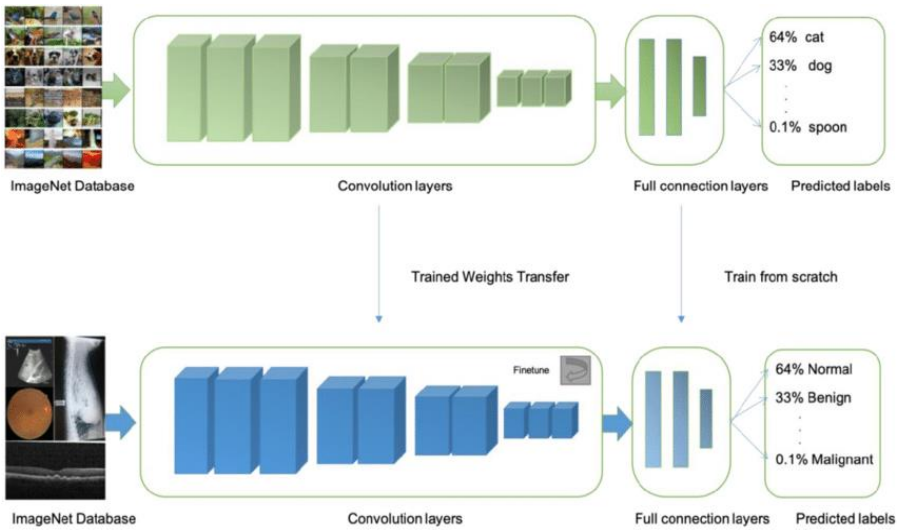
Existe otro problema muy común es el sobreentrenamiento de la red denominado “overfitting” que significa que la red ha aprendido muy bien los patrones y características de los datos con los que ha entrenado, pero no es capaz de generalizar y encontrar patrones similares en datos que no han sido procesados en el entrenamiento, esto puede deberse a que la red es demasiado compleja para el problema o que ha sido entrenada por demasiadas épocas. En el proceso de entrenamiento es muy común que se vaya minimizando el error conforme la red aprende, sin embargo, llegado a cierto punto el error crecerá nuevamente perdiendo capacidad de generalización y producirá un sobreentrenamiento (Gestal Pose, 2009).

Transfer Learning. Es una técnica que se basa en elegir y usar una red neuronal ya entrenada para una tarea y adaptarla para otra tarea similar, pero con un conjunto de datos diferente aprovechando el conocimiento ya adquirido con anterioridad. Así se puede aprovechar del entrenamiento, validación y optimización de modelos que han sido sometidos a grandes procesos y llevarlos a una aplicación propia.

En la Figura 6 se muestra un ejemplo de Transfer Learning de una red neuronal pre entrenada con la base de datos ImageNet y la aplicación del método Transfer Learning para entrenar la red en la clasificación de enfermedades mediante el uso de imágenes obtenidas para uso médico como tomografías, rayos x, ecografías, entre otras.

Figura 6

Representación de un Ejemplo de Transfer Learning



Nota. Reproducido de Illustrations of transfer learning: a neural network is pretrained on ImageNet and subsequently trained on retinal, OCT, X-ray images, B-scans for different disease Classifications, de Xu y otros, 2019, Theranostics.

Métodos de optimización. Son métodos para minimizar el error durante el entrenamiento para encontrar un modelo óptimo, entre los cuales se encuentran los siguientes.

SGD. Descenso estocástico de gradiente, aplica un descenso de gradiente o, dicho de otra manera, estimar numéricamente el valor mínimo del error tomando en cuenta un solo ejemplo de los datos de entrenamiento a la vez.

RMSprop. Su objetivo es minimizar el error mientras el método converge, ajusta el "Learning Rate" o ratio de aprendizaje de forma automática en cada cambio de los pesos sinápticos. El learning rate determina que tanto van a variar los pesos sinápticos de la red neuronal con el fin de aprender los patrones o características.

Adam. Es uno de los optimizadores más usados que combina las propiedades de los métodos SGD y RMSprop, generan una tasa de aprendizaje adaptativo, optimizando la memoria requerida en el entrenamiento.

Deep Learning

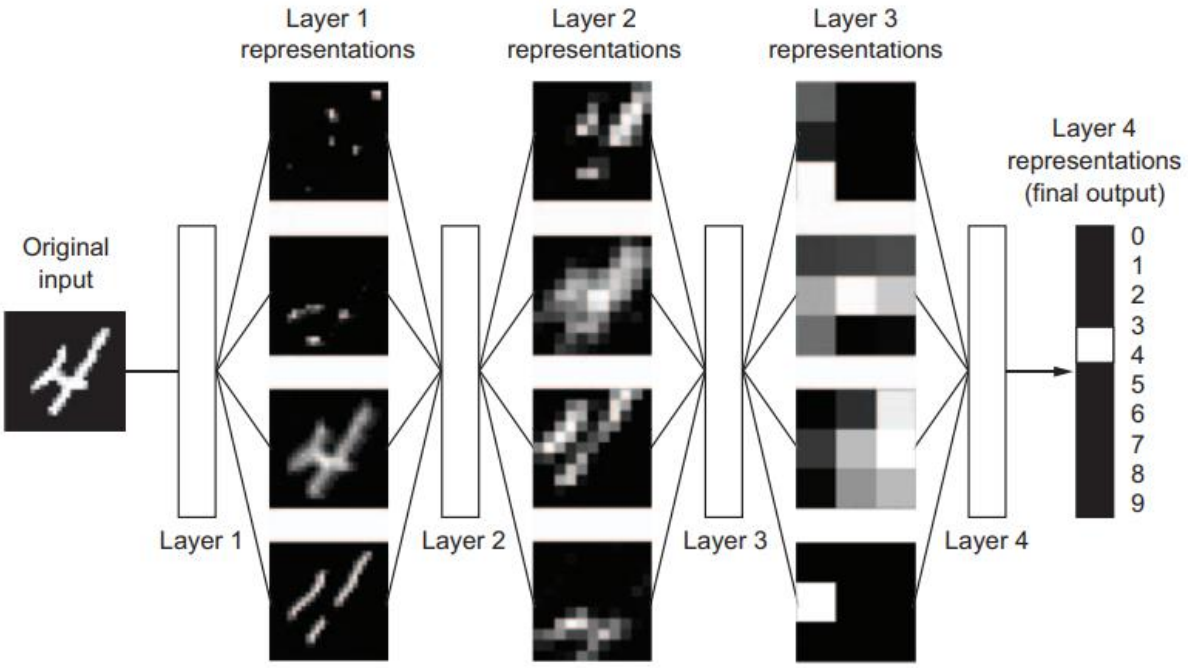
El Deep Learning o Aprendizaje Profundo es una rama de la inteligencia artificial y del aprendizaje automático que se centra en crear grandes modelos de redes neuronales multicapas hechas para la toma de decisiones en los que se involucran datos extensos y complejos. Estos modelos de inteligencia artificial tienen toda clase de aplicaciones relacionadas a la medicina, robótica, seguridad, economía, visión artificial, generación automática de texto, y muchas otras aplicaciones con las que probablemente el ser humano conviva a diario (Kelleher, 2019).

Redes Neuronales Convolucionales

Las redes neuronales convolucionales o CNN es un tipo de red que se deriva del perceptrón multicapa, su estructura y funcionamiento se relaciona mucho con la corteza visual de los seres vivos. Su aplicación en matrices bidimensionales la hace especialmente útil en el campo de la clasificación, segmentación y reconocimiento de imágenes, así como cualquier aplicación relacionada con visión artificial. En este tipo de redes se utilizan dos términos, Kernel que es la matriz de pesos sinápticos que se multiplica por la matriz de entrada con el objetivo de extraer las características relevantes de una imagen como bordes, colores, texturas, formas; por otro lado, el término Filtro se refiere al resultado de una concatenación de matrices kernel (Kelleher, 2019). Una representación de estas CNN se puede ver en la Figura 7.

Figura 7

Representación del Aprendizaje de una CNN

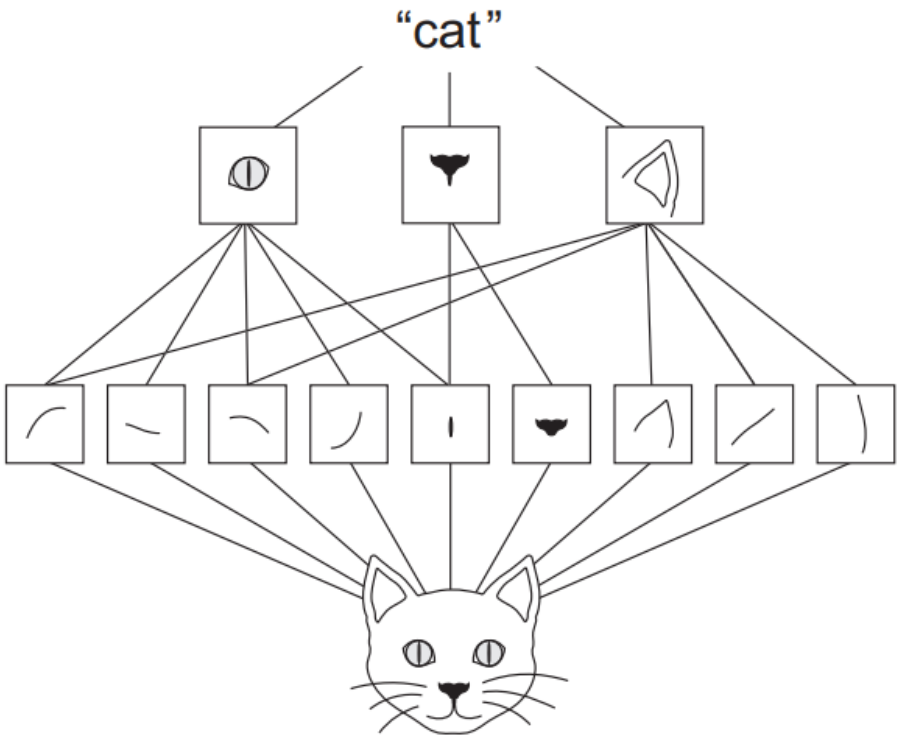


Nota. Reproducida de Deep representations learned by a digit-classification model, de F. Chollet, 2018, Manning Publications Co.

Las CNN tienen dos características importantes, la primera es que los patrones que aprenden son útiles independientemente de una translación, es decir si una CNN aprende un patrón que se encuentra en una posición de una imagen este mismo patrón puede ser reconocido incluso si se traslada a otro lugar dentro de la imagen; la segunda característica es que pueden reconocer las jerarquías espaciales de los patrones, por ejemplo si una capa es capaz de reconocer ciertos patrones, la siguiente capa será capaz de reconocer patrones relacionados a las características extraídas anteriormente en la primera capa (Chollet, 2018). Se puede ver el ejemplo conceptual en la Figura 8.

Figura 8

Representación Visual del Reconocimiento Jerárquico de Patrones



Nota. Reproducida de The visual world forms a spatial hierarchy of visual modules: hyperlocal edges combine into local objects such as eyes or ears, which combine into high-level concepts such as “cat”, de F. Chollet, 2018, Manning Publications Co.

VGG16

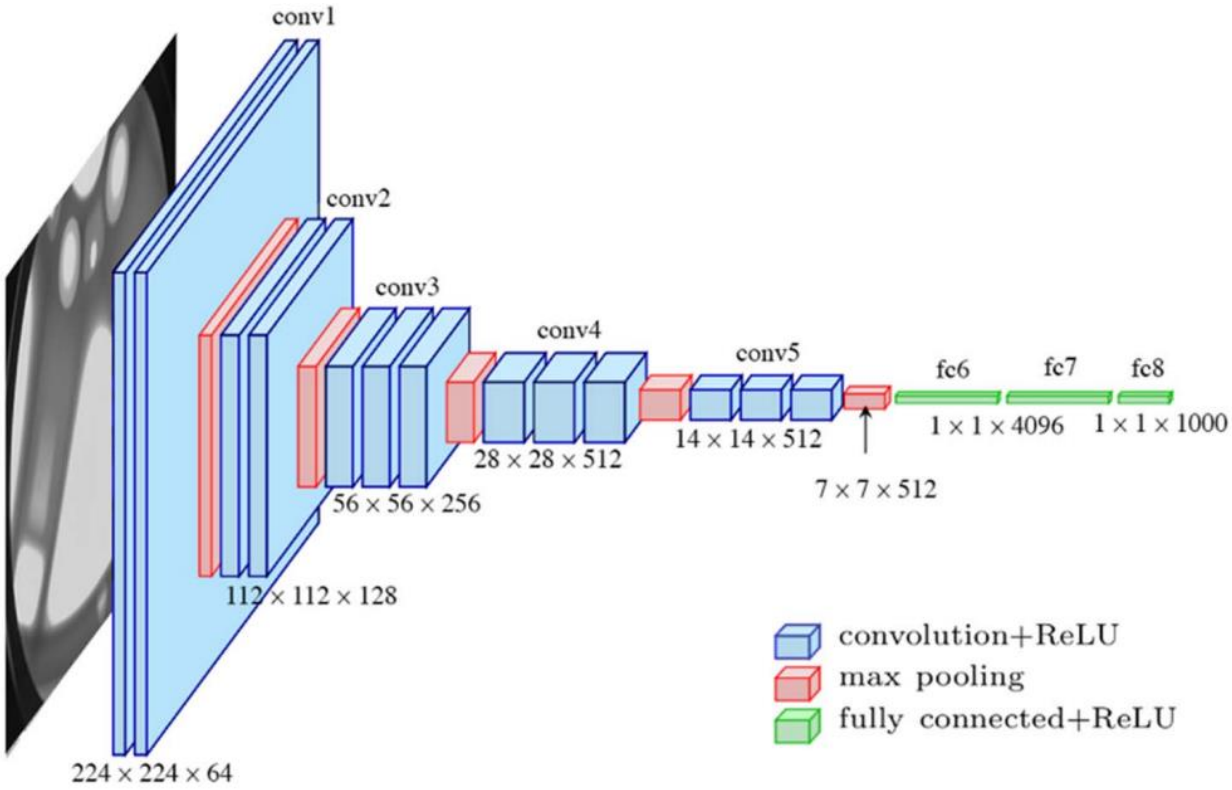
Es una CNN desarrollada por Karen Simonyan y Andrew Zisserman que ha logrado resultados de 96% a 97% de exactitud en la clasificación de imágenes de la base de datos ImageNet, que se conforma por casi un millón y medio de imágenes repartidas en 1000 clases diferentes.

VGG16 está conformada por 13 capas convolucionales y 3 capas densas, cuyo número total de capas determina su nombre, tiene una variante, la red VGG19 que tiene en su arquitectura mayor número de capas y como es obvio requiere de mayor cantidad de recursos

computacionales. Para el entrenamiento del modelo las imágenes utilizadas fueron cambiadas de resolución a imágenes RGB de 224x224 pixeles. La arquitectura de VGG16 se puede visualizar en la Figura 9 para mayor comprensión.

Figura 9

Representación Gráfica de la Arquitectura de VGG16



Nota. Reproducida de The standard VGG-16 network architecture, de Ferguson y otros, 2017, IEEE International Conference on Big Data (Big Data).

Este modelo está disponible en la API Keras, tanto su estructura como sus pesos sinápticos, por lo que sirve como base para cualquier aplicación relacionada con visión artificial.

Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes o RNN son capaces de procesar y obtener patrones o características de datos secuenciales, es por ello por lo que son muy utilizadas en aplicaciones de audio, video, procesamiento del lenguaje, entre otras aplicaciones. Este tipo de

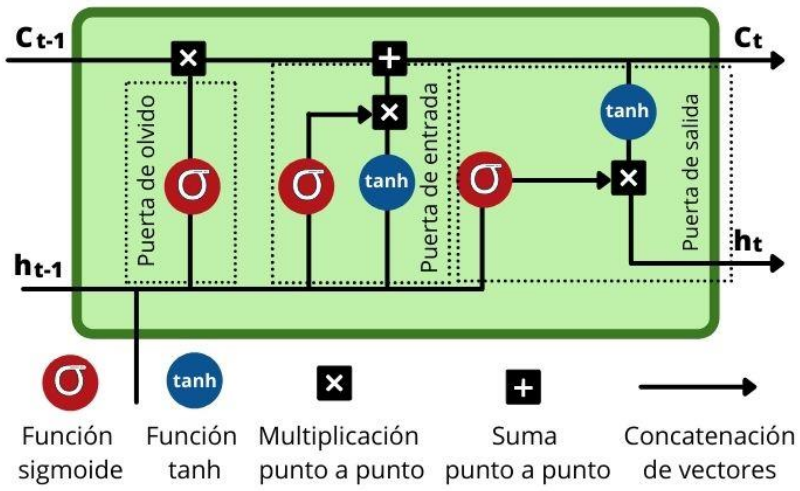
redes están compuestas de una especie de memoria interna que permite captar las dependencias secuenciales y temporales mediante la evaluación de un dato actual con un dato anterior. Un problema común de las RNN más sencillas es que no tienen la capacidad de aprender patrones que se extiendan mucho en el tiempo por lo que se creó un tipo de celda de memoria más compleja que ex capaz de extraer los patrones de secuencias de mayor longitud. Esta celda se denomina como Long-Short Term Memory o LSTM (Chollet, 2018).

LSTM

Son redes recurrentes que tienen la capacidad de aprender patrones de eventos pasados a largo plazo recordando las características temporales relevantes y olvidando las características que no son representativas. Esto es posible ya que las células LSTM están compuestas de bloques de memoria y tres compuertas de entrada, salida y olvido. La compuerta de entrada controla el flujo de información a la celda de memoria, la compuerta de salida controla el flujo de información de la celda de memoria a otros bloques LSTM y la compuerta del olvido que determina los patrones que no son necesarios o no representan los patrones de las secuencias de los datos eliminando dichos patrones innecesarios de la memoria (Ramírez Alcocer y otros, 2018). A continuación, en la Figura 10 se puede observar la estructura de una célula LSTM.

Figura 10

Elementos de la Célula LSTM



Nota. Reproducida de LSTM (Long-Short Term Memory), Cañadas, 2021.

Tensorflow

Tensorflow es una librería de código abierto que está diseñada para desarrollar e implementar modelos de aprendizaje automático, es capaz de ejecutarse en sistemas desde teléfonos móviles hasta máquinas y dispositivos computacionales como tarjetas GPU. bajo esta interfaz se pueden generar varios algoritmos ya sea de entrenamiento o inferencia para modelos de redes neuronales profundas. es muy utilizado tanto en el campo de la producción como en el campo de la investigación para aplicaciones como: reconocimiento de voz, visión por computadora, robótica, procesamiento del lenguaje, entre muchas otras.

Características básicas y funciones principales

Es capaz de realizar cálculos numéricos basados de matrices multidimensionales, también llamados “tensores”, estos cálculos pueden ser hechos a través de procesamiento distribuido usando CPU. GPU, múltiples máquinas o TPU siendo estos dos últimos grandes herramientas para ejecutar acciones rápidamente.

Tensorflow implementa algo denominado diferenciación automática que permite el cálculo de gradientes, generalmente de error o pérdida de un modelo respecto a sus pesos independientemente del número de tensores no escalares.

Tiene herramientas que sirven para la optimización del rendimiento de los modelos, es decir acelerar el entrenamiento o la inferencia de estos, de igual manera permite guardar un modelo cuando se termine de entrenar o crear puntos de control para guardar y restaurar el estado de un modelo permitiendo su uso posterior en cualquier sistema compatible.

Tensorflow 2 incluye la API Keras, que tiene funciones que facilitan esta creación de bucles de entrenamiento del modelo, permite establecer los parámetros de entrenamiento, elegir el optimizador, segmentar el conjunto de datos, cambiar la tasa de aprendizaje, realizar ajuste fino, graficar las métricas obtenidas, incluso guardar el historial de entrenamiento.

OpenCV

De acuerdo con la página oficial de OpenCV (s.f.) es una librería de código abierto creada para su uso en visión artificial y aprendizaje automático, que inicialmente fue desarrollada como un proyecto de Intel, es compatible con algunos lenguajes de programación como: C++, Phyton o Java. Utiliza una estructura modular que permite acceder a varias funcionalidades relacionadas con la captura y procesamiento de imagen, las cuales incluyen: captura y escritura de video e imagen, filtrado de imagen, cambio de resolución, estimaciones de movimiento de un vídeo, sustracción de fondo, algoritmos de seguimiento de objetos, estimación de pose, calibraciones de cámara, detección de objetos, de igual forma permite capturar y crear vídeos, entre muchas otras.

OpenCV utiliza una librería que sirve para realizar operaciones numéricas con matrices llamada Numpy, lo hace más fácil su integración con otras librerías que también requieren de cálculos a través de Numpy.

Características básicas y funciones principales

OpenCV es una herramienta eficiente que permite crear diversas aplicaciones de visión artificial permitiendo acceder a un catálogo de funciones accesibles y de uso sencillo, cuyas características principales son:

Espacio de nombres CV. Las clases y funciones de OpenCV se encuentran en un espacio específico de nombres “cv”.

Administración automática de memoria. Se utilizan destructores que eliminan la asignación de búferes de memoria cuando es necesario, tomando en cuenta la posibilidad de un intercambio de datos.

Asignación automática de los datos de salida. El tamaño y el tipo de los datos de salida se determinan a través de los datos de entrada, estas propiedades sirven para asignar o reasignar automáticamente el espacio de memoria necesarios para los parámetros de la función de salida.

Aritmética de saturación. Existen funciones como: conversión de color, contraste, nitidez, interpolación; qué pueden hacer que los valores de una imagen se salgan del rango disponible, por lo que se aplica esta aritmética de saturación para ajustar los valores de las matrices, que representan a una imagen, dentro del rango asignado.

Manejo de errores. Se hace uso de excepciones para señalar errores que pueden aparecer cuando a pesar de tener una entrada con un formato adecuado, el uso de un algoritmo resulta imposible por cualquier motivo.

Subprocesamiento múltiple y re-accesibilidad. Las funciones y métodos de diferentes instancias de clase pueden llamarse desde diferentes subprocesos.

Google Colaboratory y Paperspace Gradient

Google Colaboratory o Google Colab es un producto de Google Research que permite programar en lenguaje Python y ejecutarlo desde el navegador, dando acceso gratuito a

recursos informáticos adecuados para el trabajo con aprendizaje automático, ciencia de datos, educación entre otros.

Colab provee de recursos computacionales que pueden incluir una GPU o una TPU, pero su uso se ve limitado en tiempo de acceso debido a sus políticas de uso con el objetivo de ofrecer un entorno interactivo, mientras más recursos se utilicen menor tiempo de acceso gratuito se tendrá. Existe la opción de suscribirse a Cobab Pro o Colab Pro+ lo cual permite usar mejores recursos y por algo más de tiempo, sin embargo, no se garantiza un tiempo de acceso en específico. Y una tercera opción de pago que permite acceder a máquinas virtuales de Google Colab, gestionar los recursos de hardware requeridos y utilizar el entorno de forma persistente. A continuación, en la Tabla 1 se muestra de los recursos computacionales que se pueden encontrar en Colab.

Tabla 1

Recursos de hardware disponibles en Google Colaboratory

GPU	GPU RAM	CPUs	RAM
K80	12 GB	2 vCPU	13 GB
T4	16 GB	2 vCPU	13 GB hasta 25 GB
P100	16 GB	2 vCPU	13 GB hasta 25 GB
V100	16 GB	2 vCPU	hasta 52 GB RAM

Gradient es un entorno creado por Paperspace, ofrece recursos similares a Google Colaboratory con ciertas diferencias. En este entorno se puede elegir qué tiempo va a ser requerido, la inactividad no conlleva una expulsión del entorno, además se tiene una mayor variedad de GPUs a disposición como se muestra a continuación en la Tabla 2.

Tabla 2

Recursos de hardware disponibles en Paperspace Gradient

GPU	GPU RAM	CPUs	RAM
M4000	8 GB	8 vCPU	30 GB
P4000	8 GB	8 vCPU	30 GB
P5000	16 GB	8 vCPU	30 GB
RTX4000	8 GB	8 vCPU	30 GB
RTX5000	16 GB	8 vCPU	30 GB
A4000	16 GB	8 vCPU	45 GB
A5000	24 GB	8 vCPU	45 GB
A6000	48 GB	8 vCPU	45 GB

Métricas de Rendimiento

Las métricas de rendimiento son de utilidad para problemas de clasificación en donde se tiene el objetivo de discriminar distintos algoritmos de aprendizaje automático y aprendizaje profundo, para elegir el mejor dependiendo del objetivo de aplicación (Borja Robalino y otros, 2020). La base de estas métricas es la matriz de confusión como se muestra en la Figura 11.

Figura 11

Matriz de Confusión

		PREDICCIÓN	
		Positivos	Negativos
VALOR REAL	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

VP: cantidad de positivos que fueron clasificados correctamente.

VN: cantidad de negativos que fueron clasificados correctamente

FN: cantidad de positivos que fueron clasificados como negativos.

FP: cantidad de negativos que fueron clasificados como positivos.

Las métricas para la evaluación que se utilizarán para la evaluación del modelo propuesto se describen a continuación en la Tabla 3.

Tabla 3

Métricas de Rendimiento y su Descripción

Métrica	Fórmula	Descripción
Accuracy (Exactitud)	$\frac{VP + VN}{VP + VN + FP + FN}$	Proporción de clasificaciones predichas de manera correcta sobre el total de instancias.
Recall (Sensibilidad) o Tasa de Verdaderos Positivos	$\frac{VP}{VP + FN}$	Proporción de casos positivos bien clasificados.
Especificidad o Tasa de Verdaderos Negativos	$\frac{VN}{VN + FP}$	Proporción de casos negativos bien clasificados.
Probabilidad de Falsa Alarma	$1 - \frac{VN}{VN + FP}$	Proporción de casos positivos mal clasificados (error Tipo I).

Nota. Adaptado de Estandarización de Métricas de Rendimiento para Clasificadores Machine y Deep Learning, de Borja y otros, 2020, Revista Ibérica de Sistemas e Tecnologías de Información.

Capítulo III

Metodología

Con el fin de detectar el evento de asalto a peatones se propone desarrollar un clasificador de video, que está basado en el uso de redes neuronales de aprendizaje profundo, para lo cual se utiliza una red compuesta por un modelo VGG16 para cada uno de los N fotogramas, encargado de la obtención de sus características, junto con una red LSTM que contiene K neuronas que aprenderá patrones de las secuencias de fotogramas dando como resultado la identificación de una situación de “asalto a peatones” o “no asalto a peatones”.

Para el entrenamiento, validación y pruebas del modelo se utilizarán clips de video de duraciones de 5, 10 y 15 segundos, que contengan eventos de asaltos y situaciones normales en la calle, los cuales serán ordenados y clasificados.

En cuanto a la programación se la realizará en Google Colaboratory que permite manejar un entorno listo para la ejecución de código en lenguaje Python y puede realizar el procesamiento a través de GPU o TPU.

Preparación de los datos de entrenamiento

Los datos de entrenamiento son videos relacionados con asaltos a peatones capturados por cámaras de videovigilancia. Fueron tomados principalmente de la plataforma Youtube y de la base de datos UCF-Crime, preparados usando el editor de video llamado “Kdenlive” y organizados en carpetas.

Después de haber ordenado los elementos, se subió toda la base de datos a Google Drive. Con el fin de entrenar de mejor manera el modelo, los datos almacenados serán aumentados con técnicas de inversión horizontal de imagen, giro de imagen y añadiendo ruido.

Para esto se ha utilizado la librería “VidAug”, y se ha realizado un programa que hará cambios de forma aleatoria a cada video presente en la base de datos. El programa extrae una lista de los datos existentes en cada carpeta, añade ciertos cambios y almacena nuevos videos

generados en las mismas carpetas, manteniendo los videos originales. Un ejemplo de los videos generados se muestra en la Figura 12.

Figura 12

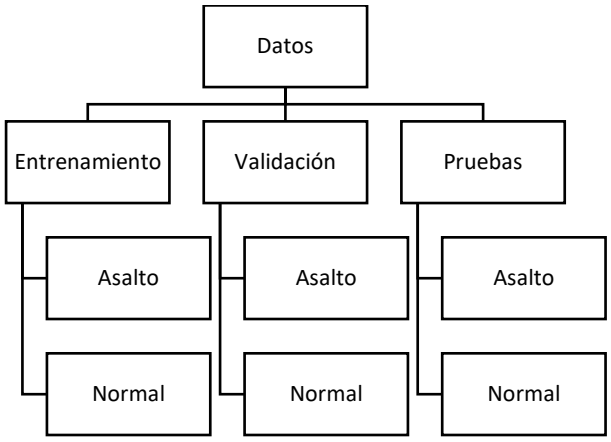
Videos Generados Con "VidAug"

Nota. Captura del video original y videos generados mediante "VidAug"

Existieron dos periodos de entrenamiento en los que se fueron probando distintos modelos, en el primer período se contaba con 65 videos relacionados con el evento "asalto a peatones" y 65 videos en los que se muestra una situación de "no asalto a peatones" una duración de 5 segundos y todos ordenados en carpetas separados para las etapas de entrenamiento, validación y pruebas como se muestra en la Figura 13 .

Figura 13

Diagrama Jerárquico de la Base de Datos del Primer Período de Entrenamiento

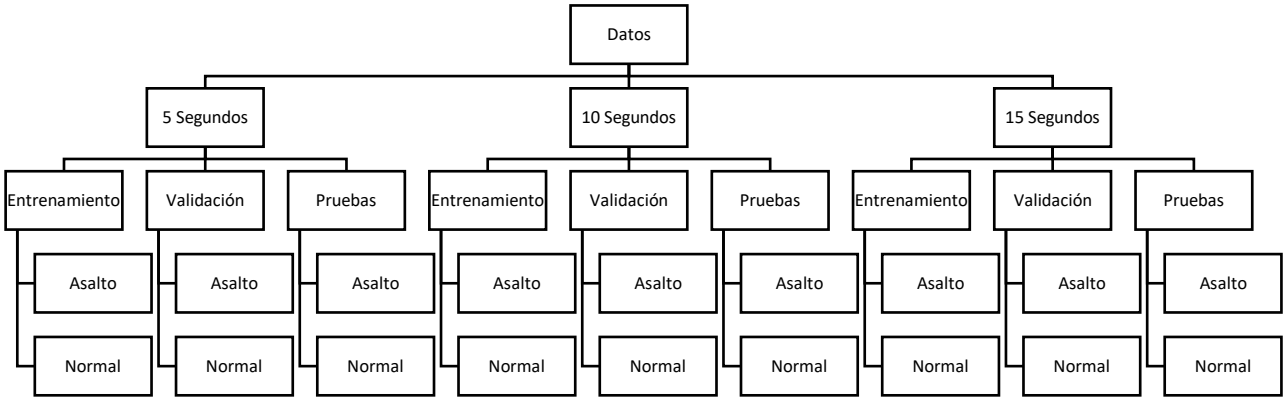


Para el segundo período se propuso probar con una base de datos con mayor cantidad de ejemplos para verificar si se podía mejorar los resultados obtenidos, en este período se contó con 120 videos relacionados con el evento “asalto a peatones” y 120 videos en los que se muestra una situación de “no asalto a peatones”. Esta vez se crearon clips de video, de 5, 10 y 15 segundos, tiempo suficiente para la visualización de un asalto o de una situación normal.

Estos clips de video se han clasificado en carpetas de acuerdo con el tiempo de duración, han sido separados en conjuntos de datos de entrenamiento, validación y pruebas, y además han sido ordenados por su categoría denominadas “Asalto” y “Normal” como se puede observar en la Figura 14.

Figura 14

Diagrama Jerárquico de la Base de Datos del Segundo Período de Entrenamiento



Siguiendo las recomendaciones, se reparte un 60% de los datos para entrenamiento, 20% para validación y 20% para pruebas.

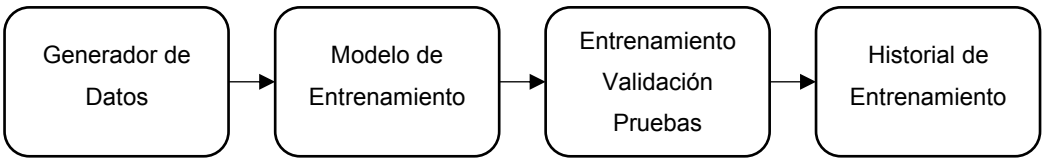
Diseño del modelo basado en redes neuronales convolucionales y recurrentes

Para crear un clasificador de video, útil para la detección de asaltos a peatones, se ha diseñado un programa para las fases de entrenamiento, validación y pruebas de modelos de redes neuronales.

El programa consta de cuatro etapas, para empezar, se crean generadores de datos para cada una de las fases, seguido de esto se establece y se crea el modelo de entrenamiento compuesto por el modelo VGG16 en conjunto con una red neuronal LSTM, luego se entrena, valida y prueba el modelo para finalmente guardar el historial de entrenamiento, así como se muestra en la Figura 15.

Figura 15

Diagrama de Bloques del Programa



Generadores de Datos

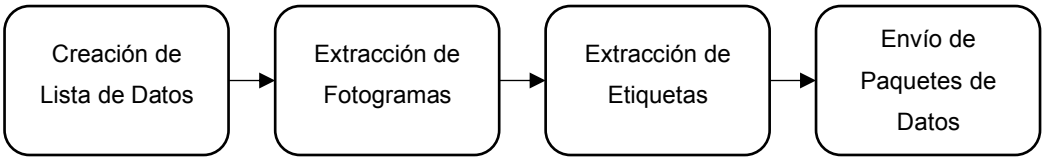
La base de datos está conformada por videos de los cuales se extraerán conjuntos de fotogramas obteniendo grandes matrices que los representan y que ocuparán gran espacio de memoria. Hay dos opciones para cargar los datos para el entrenamiento de la red, la primera es cargar todos los datos y realizar el proceso de entrenamiento, la segunda es crear generadores que permitan cargar los datos por paquetes. La desventaja de cargar todos los datos a la vez es que se puede llegar a saturar la memoria RAM si se trabaja con datos que requieran gran espacio de almacenamiento.

Los generadores dividen la base de datos en pequeños paquetes y son enviados conforme el programa lo necesita. Cada generador crea una lista de direcciones de los videos

presentes en una carpeta. De lista generada se van tomando pequeños conjuntos de direcciones, se extraen los fotogramas y las etiquetas de los videos correspondientes a dichas direcciones, estos datos son enviados y el proceso se repite hasta terminar la lista como se muestra de manera general en la Figura 16 y con detalles de programación en la Figura 17.

Figura 16

Diagrama de Bloques del Generador de Datos



Para la extracción de la lista de datos, tiene como entrada el directorio de los datos, como los datos están contenidos en dos carpetas, “Asalto” y “Normal”, se van extrayendo los directorios de los videos de estas carpetas y los guarda en una lista, cuando todos los archivos de la carpeta han sido listados, se reordena la lista de forma aleatoria y se devuelve la lista resultante, esto se representa en la Figura 18 y su programación se encuentra representada en Figura 21.

Figura 17

Diagrama de Flujo del Generador de Datos

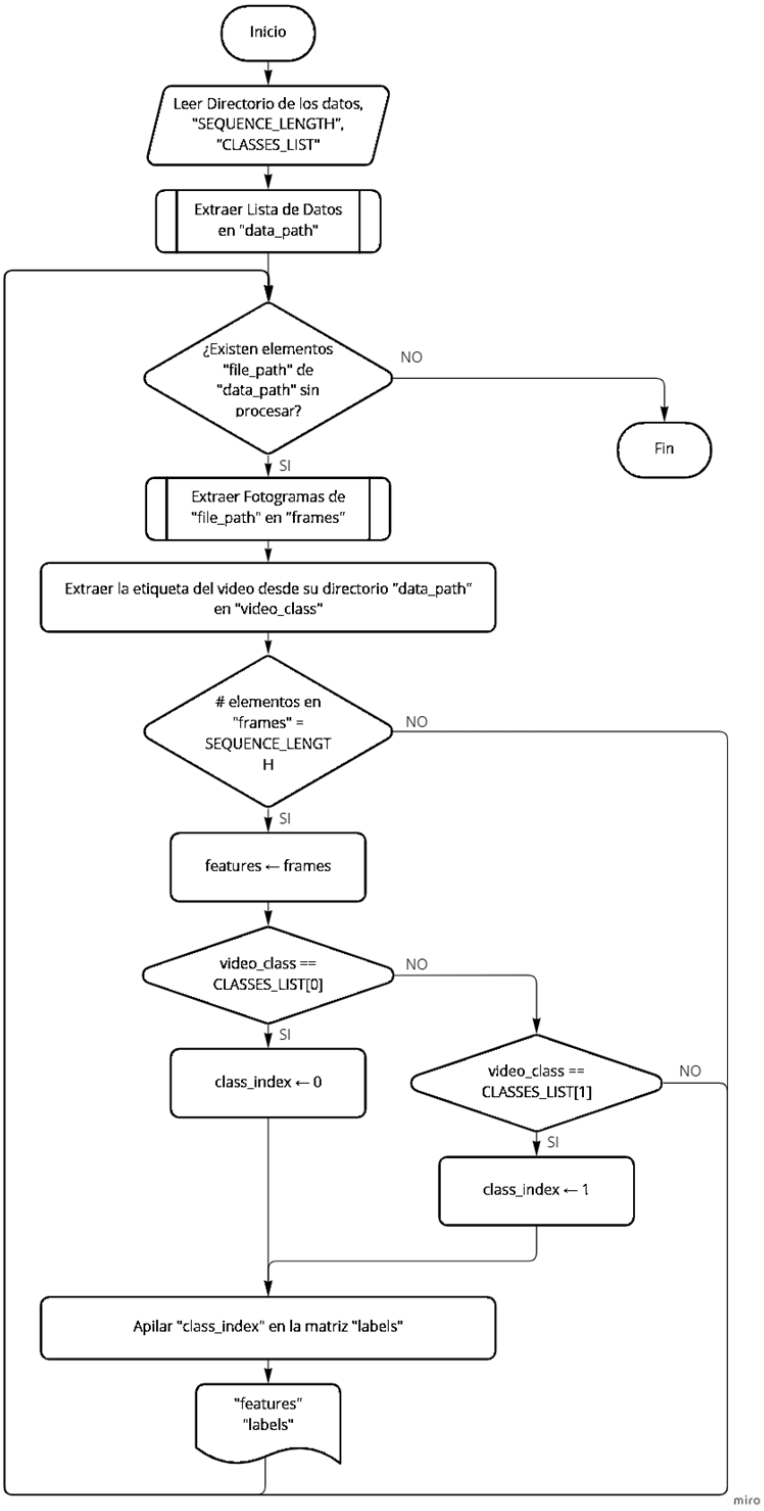
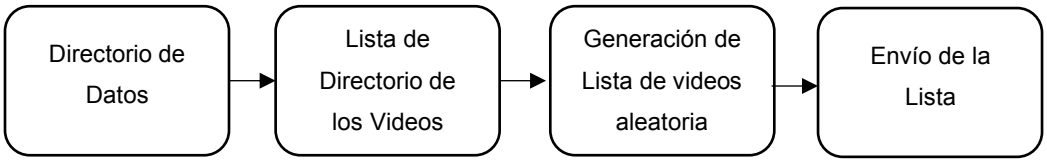


Figura 18

Diagrama de Bloques del Extractor de la Lista de Datos



Se van capturando los fotogramas en matrices haciendo una especie de muestreo del total del video como se muestra en la Figura 19, se cambia la resolución de cada fotograma a 224x224 pixeles, que es la resolución admitida por el modelo VGG16, se normaliza la matriz resultante de cada fotograma, y se guarda todo el conjunto de fotogramas extraídos en una matriz, así se muestra en la Figura 20.

Figura 19

Muestreo de Fotogramas

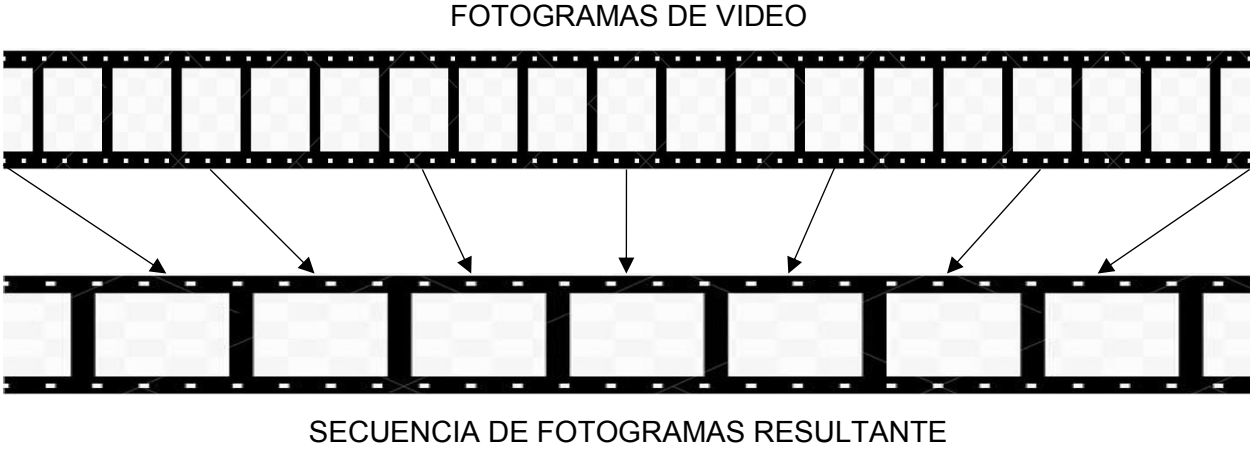


Figura 20

Diagrama *de Bloques del Extractor de Fotogramas*

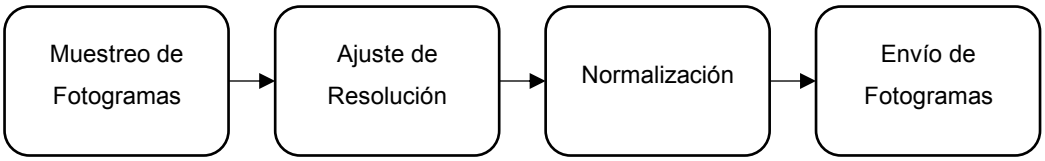
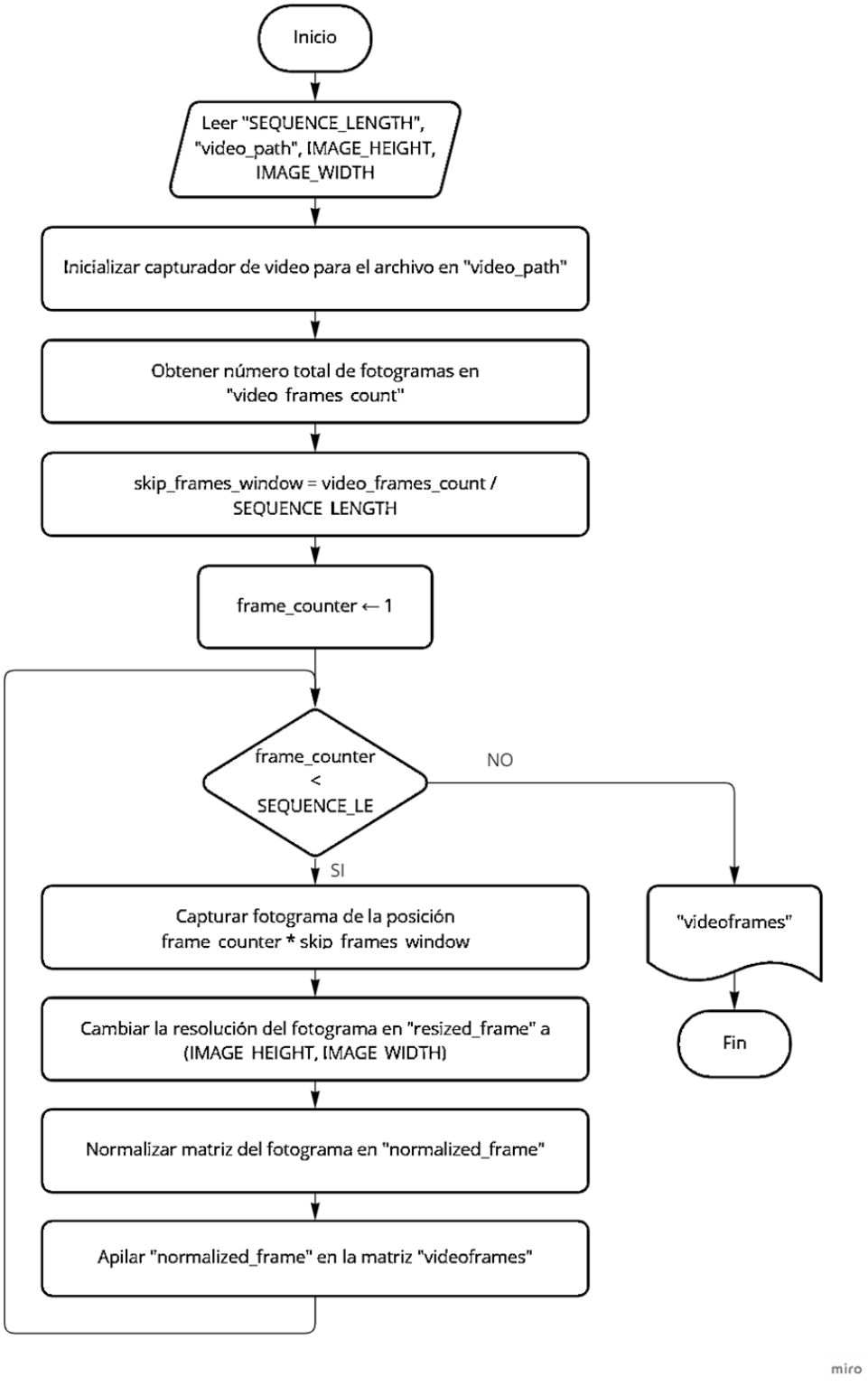


Figura 21

Diagrama de Flujo de Extractor de Fotogramas



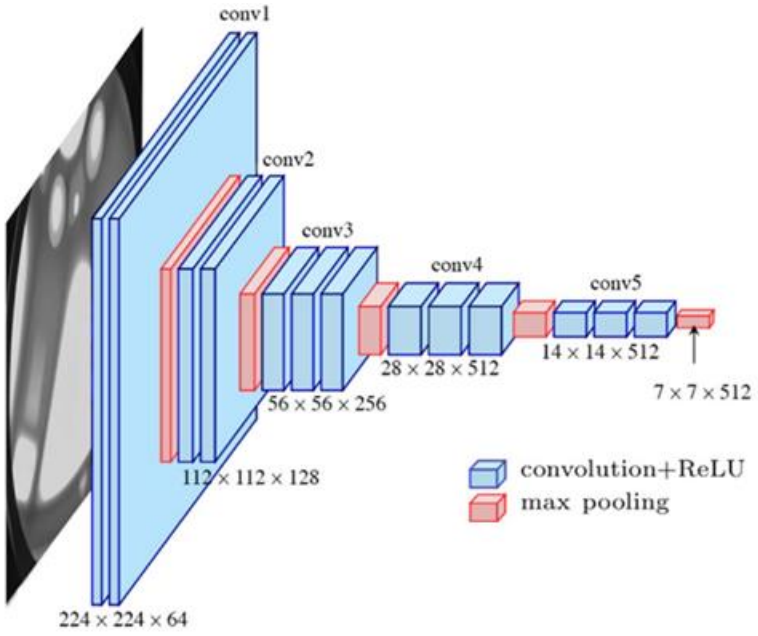
Modelo de Entrenamiento

La idea del modelo es extraer las características de una cantidad determinada de fotogramas de los videos, evaluar su relación a lo largo del tiempo y determinar si se muestra un evento de asalto.

Con el modelo VGG16, que ha sido entrenado para la clasificación de imágenes, y mediante la eliminación de la última capa se obtiene una red que es capaz de extraer las características de una imagen, al tener una secuencia de imágenes lo que se ha hecho es distribuir el modelo VGG16 para que haga el reconocimiento todos los fotogramas de la secuencia del videoclip, esto se logra mediante la clase “TimeDistribuited”, la estructura de VGG16 con su capa de clasificación eliminada se detalla en la Figura 22.

Figura 22

Representación Gráfica de la Arquitectura de VGG16 sin Etapa de Clasificación



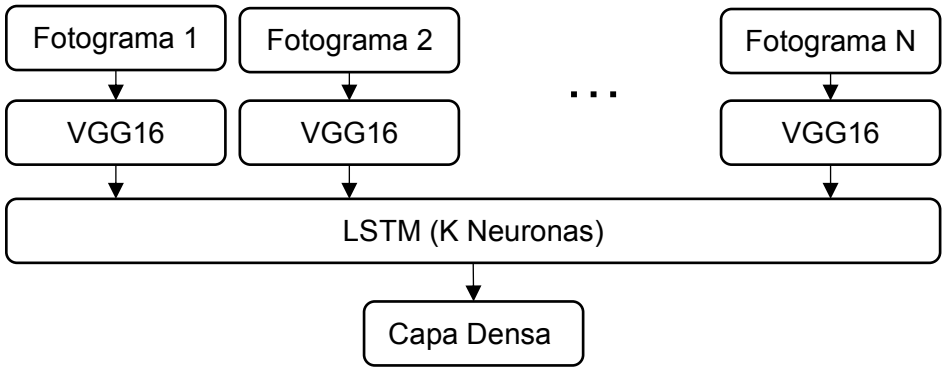
Nota. Adaptada de The standard VGG-16 network architecture, de Ferguson y otros, 2017, IEEE International Conference on Big Data (Big Data).

Una vez obtenidas las características de los fotogramas de la secuencia es necesario reconocer cómo cambian y se relacionan en el tiempo, por lo que se ocupa una red recurrente, en este caso se utiliza una capa de neuronas LSTM, útiles para reconocimiento de patrones en elementos secuenciales.

Finalmente se añade una capa densa con un número de neuronas igual al número de clases a evaluar, en este caso dos, un ejemplo de representación final del modelo se encuentra en la Figura 23.

Figura 23

Diagrama de representación del modelo propuesto.



Entrenamiento, validación y pruebas

Teniendo los datos listos y el modelo establecido se comienza el entrenamiento del modelo, para esto existen una serie de parámetros que permitirán optimizar el proceso dependiendo de los recursos disponibles. Se han tenido diversas consideraciones de acuerdo con la experiencia que se fue adquiriendo durante el proceso de entrenamiento y pruebas. El hardware fue cambiando con respecto a las consideraciones iniciales por diversos motivos que más adelante se detallarán, el software se fue modificando con el fin de optimizarlo y por último al no existir una guía metodológica específica para redes neuronales muchas veces los parámetros de entrenamiento se fueron ajustando a las condiciones que va presentando el entrenamiento haciendo una especie de juego de prueba y error.

Recursos de Hardware

Al inicio de la etapa de entrenamiento se utilizó una Laptop con las características detalladas en la Tabla 4, se hicieron pequeñas pruebas del programa con el fin de encontrar cualquier error de programación, o fallos de ejecución.

Tabla 4

Especificaciones de Hardware Utilizadas

Tipo de Hardware	Modelo
Procesador	Intel(R) Core(TM) i5-7300HQ de 2.50Ghz
Memoria RAM	8 GB
Memoria SSD	1 TB
Disco Duro	1 TB
Tarjeta Gráfica	NVIDIA GeForce GTX 1050
VRAM	4GB

Sin embargo, al momento de realizar un entrenamiento utilizando todos los parámetros, todos los elementos de base de datos, y el modelo completo, los recursos no fueron suficientes. Es por ello que se buscó una alternativa, en principio se utilizó la plataforma Google Colab, pero por las limitaciones de recursos, sobre todo por restricciones de tiempo, más adelante se tomó como mejor opción a Paperspace Gradient.

A continuación, en la se presenta una comparativa de los recursos de memoria RAM y GPU utilizados durante el entrenamiento tanto en Colab como Gradient, considerando que en Google Colab se tiene una tarjeta de video con 16 GB de VRAM y en Paperspace Gradient una tarjeta con 8GB de VRAM.

Tabla 5

Recursos de Memoria RAM y GPU Utilizados en Entrenamiento

Plataforma	Memoria RAM	GPU
Google Colab	4 GB	90%
Paperspace Gradient	5.4 GB	100%

Registro y Optimización del Entrenamiento

El entrenamiento consta de funcionalidades que permitirán registrar y guardar los mejores modelos obtenidos durante el proceso mediante “Callbacks” que también permiten configurar la parada del entrenamiento cuando alguno de sus parámetros no esté mejorando.

El parámetro que se ha elegido para controlar la parada del entrenamiento es la pérdida de validación, asegurándose que sea la mínima posible lo evitará caer en problemas de overfitting, una vez hayan pasado un número determinado de épocas sin que haya mejoras el entrenamiento se detiene y se recuperan los mejores resultados.

Para guardar los modelos generados, de igual forma se elige un parámetro. Hay modelos cuyo entrenamiento dura más tiempo del que se tiene disponible, ya sea en Google Colab o Paperspace Gradient, por lo que en ocasiones el entrenamiento fue dividido en dos tandas, en la primera se dio prioridad a la menor pérdida de validación por lo que es el parámetro que determinará cuándo se guarda un modelo. Después de la primera tanda se carga el modelo guardado y se continúa la segunda tanda de entrenamiento con ciertas modificaciones en los parámetros si es que se considera necesario.

La tasa de aprendizaje se ha establecido en un valor de 0.0001 para el inicio del entrenamiento, este valor se ha ido probando mediante la experimentación. Para mejorar el aprendizaje se eligió tener una tasa de aprendizaje variable que va disminuyendo mediante la multiplicación de un factor de 0.75 conforme la pérdida de validación no está mejorando, esto permite un ajuste cada vez más fino y evita que el error comience a elevarse.

Para el número de épocas se ha establecido como máximo 50, aunque no necesariamente se cumplirá debido a todas las condiciones establecidas anteriormente.

Historial de Entrenamiento

Una vez finalizado el entrenamiento se pueden guardar los valores de exactitud y pérdidas, también pueden ser graficados para un posterior análisis de resultados.

Al final del proceso se pueden realizar las pruebas con los datos asignados para esta fase, verificando los resultados obtenidos en el entrenamiento.

Capítulo IV

Resultados

Durante el período de entrenamiento se fueron variando parámetros de número de neuronas LSTM y longitud de la secuencia de imágenes evaluando cuáles serían los óptimos para una buena clasificación de videos.

El número de fotogramas, la resolución, el directorio de datos y el directorio de resultados se puede configurar en la sección del código mostrada en la Figura 24. Los datos al estar ordenados en carpetas con archivos de video de 5, 10 y 15 segundos de duración.

Figura 24

Código de Referencia Para Cambios en Resolución, Tamaño de Secuencia, y Directorio de Datos y Resultados

```
# Definir ancho y alto de frame
IMAGE_HEIGHT , IMAGE_WIDTH = 224, 224

# Definir el tamaño de las secuencias a entrenar
SEQUENCE_LENGTH = 60

DATASET_DIR = "/content/drive/MyDrive/Tesis/Clasificador/Datos/10 Segundos"
RESULTS_DIR = "/content/drive/MyDrive/Tesis/Clasificador/Resultados"
```

El número de neuronas LSTM se puede configurar en la sección de construcción del modelo como se muestra en la Figura 25, en donde se ha elegido utilizar para el caso 25 neuronas LSTM.

Figura 25

Código de Referencia Para Cambios en el Número de LSTM

```
model = Sequential()

model.add(TimeDistributed(modelTran, input_shape=(SEQUENCE_LENGTH, IMAGE_HEIGHT, IMAGE_WIDTH, 3))) #Modelo VGG16
model.add(TimeDistributed(Flatten()))
model.add(LSTM(25))
model.add(Dense(len(CLASSES_LIST), activation='softmax'))
```

Existen otros parámetros a configurar como el tamaño del lote o batch en los generadores de datos de entrenamiento, validación y pruebas, todo esto en la sección de Generación del conjunto de datos como se muestra en la Figura 26, donde se puede ver que se

eligió un batch de 6, este número no se mantuvo fijo pues la capacidad de procesamiento puede limitar este parámetro.

Figura 26

Código de Referencia Para Cambios en el Tamaño del Lote o Batch

```
#Generación del conjunto de datos
val_dataset = tf.data.Dataset.from_generator(val_generator,
                                             output_types= (tf.float64,tf.float32),
                                             output_shapes=(tf.TensorShape([SEQUENCE_LENGTH, IMAGE_HEIGHT, IMAGE_WIDTH, 3]),
                                                            tf.TensorShape([2]))).batch(6)
val_iterator = iter(val_dataset)
xv,yv = val_iterator.get_next()

print(val_dataset.element_spec)
```

Pruebas y experimentos

En este apartado se presentarán los mejores resultados obtenidos en dos períodos de entrenamiento, con un total de cuatro variantes del modelo, el primero correspondiente al primer período en el que se trabajó únicamente con 130 videos de 5 segundos, y los tres siguientes correspondientes a variantes del modelo en los que se empleó una mayor cantidad de datos con duraciones de 5, 10 y 15 segundos.

Primera Variante del Modelo

Esta variante del modelo fue configurada y entrenada con las siguientes características:

- Número de neuronas LSTM: 25
- Número de fotogramas: 30
- Duración de video: 5 segundos
- Número de videos: 130

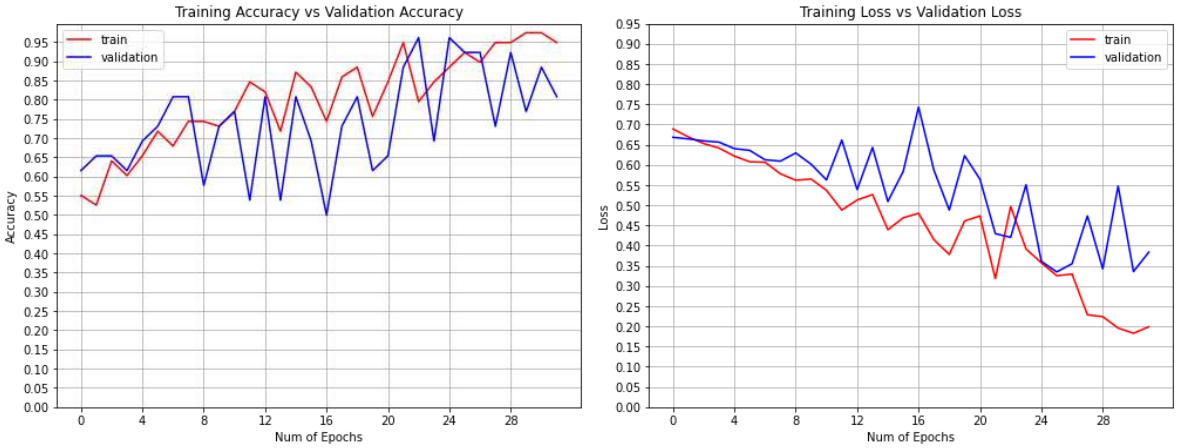
Los 130 videos fueron duplicados con la librería VidAug y para el entrenamiento la tasa de aprendizaje variable fue establecida en 0.0001 en el inicio del entrenamiento y cambió en la época 31 a 0.000075.

Una vez cumplido el entrenamiento se pudieron recopilar los datos de exactitud y pérdidas, que se representan en la Figura 27.

En los gráficos de la Figura 27 se puede observar que tanto los valores de exactitud como los de pérdidas oscilan conforme el entrenamiento avanza, lo que puede deberse a una tasa de aprendizaje muy alta, sin embargo, se puede notar que a pesar de esto, el modelo mejora su exactitud y reduce sus pérdidas conforme pasan las épocas de entrenamiento.

Figura 27

Gráficos de Exactitud y Pérdidas de la Primera Variante del Modelo



Se pueden destacar tres resultados los cuales han sido escogidos y ordenados de acuerdo con el número de épocas que fueron entrenados. Estos resultados han sido probados con los videos disponibles para la fase de pruebas, que evidentemente no han sido tomados en cuenta para el entrenamiento y validación, los resultados de esto se pueden apreciar en la Tabla 6.

Tabla 6

Resultados de Exactitud y Pérdidas de la Primera Variante del Modelo

Épocas de entrenamiento	Exactitud de Entrenamiento	Pérdidas de Entrenamiento	Exactitud de Validación	Pérdidas de Validación	Exactitud de Prueba	Pérdidas de Prueba
9	0.80	0.45	0.73	0.53	0.69	0.69
16	0.94	0.22	0.79	0.48	0.58	0.72
19	0.98	0.15	0.81	0.47	0.54	0.75

Los resultados reflejan que, si bien en cuanto a entrenamiento y validación los modelos responden de forma adecuada con una exactitud sobre el 81%, al momento de ser utilizados en las pruebas no se acercan al rendimiento teniendo apenas un 69% de exactitud como máximo, esto puede deberse a que los datos son insuficientes y también que los datos de prueba podrían no representar el problema de la misma manera que los datos de entrenamiento y validación.

Segunda Variante del Modelo

Esta variante del modelo fue configurada y entrenada con las siguientes características:

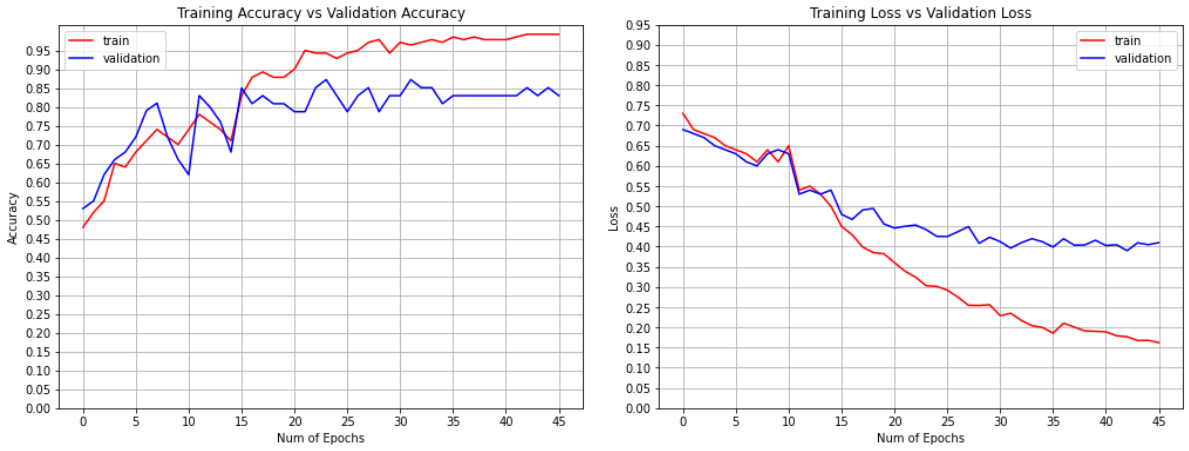
- Número de neuronas LSTM: 20
- Número de fotogramas: 60
- Duración de video: 10 segundos
- Número de videos: 233

La tasa de aprendizaje inicial fue establecida en 0.0001 y cambió en la época 16 a 0.000075, y luego, en la época 36, bajó a 0.0000563.

El historial de entrenamiento tanto de exactitud como de pérdidas se puede ver en la Figura 28. En este caso se puede notar que la red neuronal aprende de forma continua y que el cambio de tasa de aprendizaje ayuda a que el entrenamiento vaya haciendo un ajuste cada vez más fino.

Figura 28

Gráficos de Exactitud y Pérdidas de la Segunda Variante del Modelo



Se han elegido tres resultados que se han considerado los mejores, nuevamente han sido ordenados de acuerdo con el número de épocas que fueron entrenados y han sido probados con los videos disponibles que no se han tomado en cuenta para el entrenamiento y validación, los resultados de esto se pueden apreciar en la Tabla 7. Cabe recalcar que no se han elegido los resultados de las últimas épocas de entrenamiento ya que en cuanto a la exactitud y pérdidas de validación el modelo se ha visto estancado, es decir a partir de ahí el modelo simplemente ha comenzado a memorizar los datos de entrenamiento lo cual puede causar problemas de generalización.

Tabla 7

Resultados de Exactitud y Pérdidas de la Segunda Variante del Modelo

Épocas de entrenamiento	Exactitud de Entrenamiento	Pérdidas de Entrenamiento	Exactitud de Validación	Pérdidas de Validación	Exactitud de Prueba	Pérdidas de Prueba
16	0.94	0.30	0.87	0.44	0.74	0.55
25	0.93	0.30	0.83	0.43	0.74	0.55
32	0.96	0.23	0.87	0.40	0.76	0.52

En estos resultados se puede observar que la exactitud y las pérdidas presentes en las pruebas no se alejan demasiado de los valores obtenidos en entrenamiento y validación. Los

tres resultados son similares sin embargo el mejor de ellos sería el que fue entrenado durante 32 épocas.

Tercera Variante del Modelo

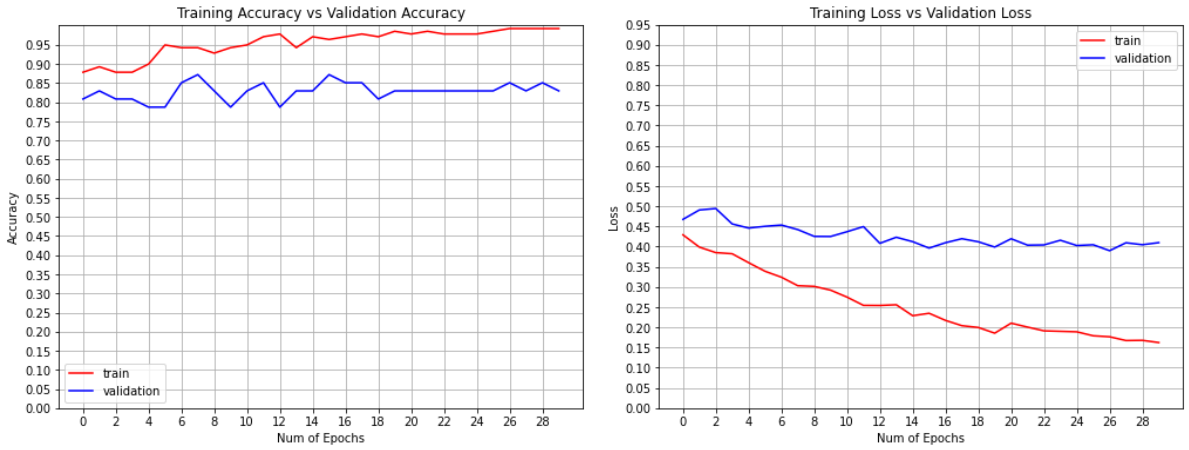
Esta variante del modelo fue configurada y entrenada con las siguientes características:

- Número de neuronas LSTM: 15
- Número de fotogramas: 40
- Duración de video: 5 segundos
- Número de videos: 240

La tasa de aprendizaje inicial fue establecida en 0.0001, que bajó a 0.000075 en la época 19, luego a 0.0000563 en la época 24 y finalmente a 0.0000422 en la época 29. Se puede ver el historial de entrenamiento de este modelo en la Figura 29.

Figura 29

Gráficos de Exactitud y Pérdidas de la Tercera Variante del Modelo



Una vez más se presentan los resultados del entrenamiento resumidos en la Tabla 8. Se han elegido tres resultados que se han considerado los mejores, una vez más han sido ordenados de acuerdo con el número de épocas que fueron entrenados y han sido probados con los videos disponibles que no se han tomado en cuenta para el entrenamiento y validación.

Tabla 8

Resultados de Exactitud y Pérdidas de la Tercera Variante del Modelo

Épocas de entrenamiento	Exactitud de Entrenamiento	Pérdidas de Entrenamiento	Exactitud de Validación	Pérdidas de Validación	Exactitud de Prueba	Pérdidas de Prueba
18	0.89	0.34	0.83	0.45	0.71	0.62
19	0.93	0.26	0.81	0.46	0.71	0.62
23	0.97	0.18	0.83	0.44	0.69	0.65

Cuarta Variante del Modelo

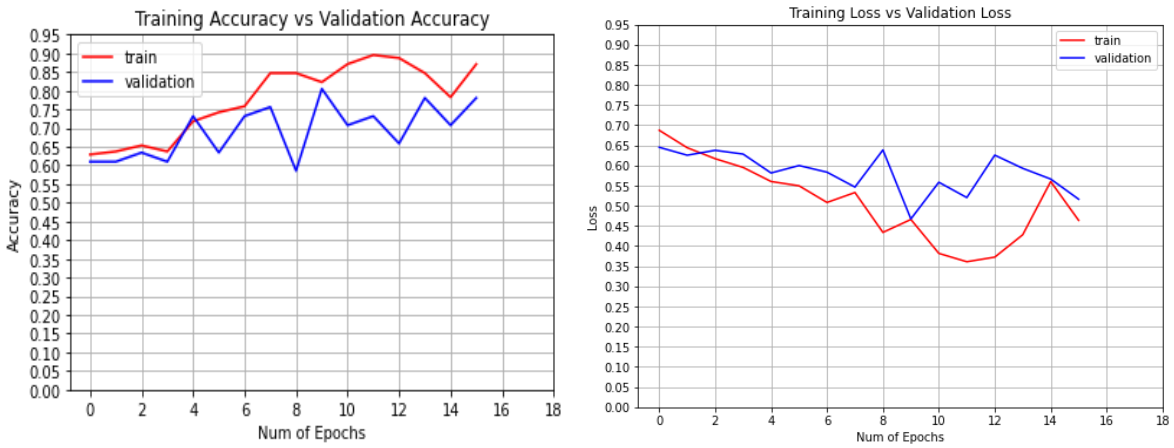
Esta variante del modelo fue configurada y entrenada con las siguientes características:

- Número de neuronas LSTM: 20
- Número de fotogramas: 90
- Duración de video: 15 segundos
- Número de Videos: 206

La tasa de aprendizaje establecida para todo el entrenamiento fue de 0.0001. Su historial de entrenamiento se ve representado en la Figura 30.

Figura 30

Gráficos de Exactitud y Pérdidas de la Cuarta Variante del Modelo



Finalmente, se han elegido tres resultados que se han considerado los mejores, una vez más han sido ordenados de acuerdo con el número de épocas que fueron entrenados y han sido probados con los videos disponibles que no se han tomado en cuenta para el entrenamiento y validación. Los resultados del entrenamiento se presentan en la Tabla 9.

Tabla 9

Resultados de Exactitud y Pérdidas de la Cuarta Variante del Modelo

Épocas de entrenamiento	Exactitud de Entrenamiento	Pérdidas de Entrenamiento	Exactitud de Validación	Pérdidas de Validación	Exactitud de Prueba	Pérdidas de Prueba
10	0.82	0.47	0.80	0.47	0.61	0.67
11	0.86	0.35	0.78	0.44	0.71	0.65
19	0.96	0.23	0.80	0.40	0.71	0.70

Para finalizar se exponen capturas del modelo en funcionamiento, para lo cual se han generado videos de prueba añadiendo la etiqueta generada ya sea de un evento de “asalto” o de una situación “normal” o de “no asalto”.

Figura 31

Capturas del funcionamiento del modelo en videos de prueba

Análisis de Resultados

Para este análisis se escogerá el mejor resultado de cada variante del modelo para evaluar su rendimiento y establecer realmente cuál es el mejor, se realizará la matriz de confusión de cada uno y se calcularán las métricas de precisión considerando la importancia de la detección del asalto.

Métricas de la Primera Variante

Se ha evaluado el modelo con 26 videos, los cuales están divididos en 13 videos con escenas de asalto y 13 videos con escenas normales (sin asalto), dando como resultado la matriz de confusión presentada en la Figura 32.

Figura 32

Matriz de Confusión de la Primera Variante

		PREDICCIÓN	
		Asalto Positivos	Normal Negativos
VALOR REAL	Asalto Positivos	8	5
	Normal Negativos	3	10

Una vez obtenida la matriz de confusión se calculan las métricas que definirán el rendimiento del modelo, el cálculo de estas métricas se resume en la Tabla 10**Tabla 13**.

Tabla 10

Métricas de Rendimiento de la Primera Variante

Métrica	Valor
Exactitud	0.6923
Recall	0.6154
Especificidad	0.7692
Falsa Alarma	0.2308

Métricas de la Segunda Variante

Se ha evaluado el modelo con 46 videos, los cuales están divididos en 22 videos con escenas de asalto y 24 videos con escenas normales (sin asalto), dando como resultado la matriz de confusión presentada en la Figura 33.

Figura 33

Matriz de Confusión de la Segunda Variante

		PREDICCIÓN	
		Asalto Positivos	Normal Negativos
VALOR REAL	Asalto Positivos	15	7
	Normal Negativos	4	20

Una vez obtenida la matriz de confusión se calculan las métricas que definirán el rendimiento del modelo, el cálculo de estas métricas se resume en la Tabla 11.

Tabla 11 *Métricas de Rendimiento de la Segunda Variante*

Métrica	Valor
Exactitud	0.7609
Recall	0.6818
Especificidad	0.8333
Falsa Alarma	0.1667

Métricas de la Tercera Variante

Se ha evaluado el modelo con 48 videos, los cuales están divididos en 24 videos con escenas de asalto y 24 videos con escenas normales (sin asalto), dando como resultado la matriz de confusión presentada en la Figura 34.

Figura 34

Matriz de Confusión de la Tercera Variante

		PREDICCIÓN	
		Asalto Positivos	Normal Negativos
VALOR REAL	Asalto Positivos	17	7
	Normal Negativos	7	17

Una vez obtenida la matriz de confusión se calculan las métricas que definirán el rendimiento del modelo, el cálculo de estas métricas se resume en la Tabla 12.

Tabla 12

Métricas de Rendimiento de la Tercera Variante

Métrica	Valor
Exactitud	0.7083
Recall	0.7083
Especificidad	0.7083
Falsa Alarma	0.2917

Métricas de la Cuarta Variante

Se ha evaluado el modelo con 41 videos, los cuales están divididos en 18 videos con escenas de asalto y 23 con escenas normales (sin asalto), dando como resultado la matriz de confusión presentada en la Figura 35.

Figura 35

Matriz de Confusión de la Cuarta Variante

		PREDICCIÓN	
		Asalto Positivos	Normal Negativos
VALOR REAL	Asalto Positivos	14	4
	Normal Negativos	8	15

Una vez obtenida la matriz de confusión se calculan las métricas que definirán el rendimiento del modelo, el cálculo de estas métricas se resume en la Tabla 13.

Tabla 13

Métricas de Rendimiento de la Cuarta Variante

Métrica	Valor
Exactitud	0.7073
Recall	0.7778
Especificidad	0.6522
Falsa Alarma	0.3478

Una vez presentados los resultados se destaca la segunda variante del modelo, debido a su exactitud del 76.09%, y su probabilidad de falsa alarma de 16.67%, también destaca la cuarta variante por su recall del 77.78% que representa la cantidad de asaltos que reconocería sin embargo tiene una exactitud del 70.73% y una probabilidad de falsa alarma de 34.78%.

Por los resultados también se puede concluir que los videos de 10 y 15 segundos de duración son las mejores opciones para el entrenamiento del clasificador, ya que la primera y tercera variante, que fueron entrenadas con videos de 5 segundos, fueron las de peores resultados.

El uso de una mayor cantidad de datos ayudó a que el modelo entrene de mejor forma y se logren mejores resultados de exactitud sobre todo en la etapa de pruebas, la cual es muy importante de cara a evaluar el modelo.

Ullah (2021) presenta una tabla comparativa de la exactitud de su modelo, con otros modelos conocidos para clasificación de video, con la base de datos UCF-Crime que forma parte de la base de datos utilizada, lo que permite situar de mejor manera el modelo propuesto en el presente proyecto lo que se puede ver en la Tabla 14.

Tabla 14

Comparativa de modelos utilizados en clasificación de video

Modelo0	Exactitud
VGG-16	72.66
VGG-19	71.66
FlowNet	71.33
DEARESt	76.66
Residual LSTM	78.43
VGG16-LSTM	76.09

Nota. Adaptado de Performance of the proposed model with state-of-the-art techniques and the best results are represented in bold, de Ullah y otros, 2021, Sensors.

Se puede observar que el modelo presentado se acerca a los resultados del modelo Residual LSTM y DEARESt, lo que motiva a mejorar el modelo aplicando otras técnicas tanto para el procesamiento y preparación de datos como para el entrenamiento validación y pruebas.

Capítulo V

Conclusiones, Recomendaciones y Trabajos Futuros

Conclusiones

Se desarrolló un clasificador de video mediante un modelo de redes neuronales de aprendizaje profundo VGG16-LSTM que tiene una exactitud del 76.09% , como una opción válida para la detección de ciertos eventos como el asalto a peatones.

Se ha generado una base de datos representativa de los eventos a tratar con alrededor de 679 clips de video de 5, 10 y 15 segundos de duración, representando escenas con asalto y escenas normales sin asalto, para el entrenamiento, validación y pruebas del modelo del clasificador.

El entrenamiento del modelo clasificador obtuvo mejores resultados con el uso de clips de video de 10 y 15 segundos de duración, sin embargo por la alta demanda de recursos computacionales que se requiere para su procesamiento, los clips de 10 segundos serían los más adecuados.

La combinación de modelos de redes VGG16 y LSTM logra un mejor desempeño en la clasificación de video que las de modelo simple VGG16, VGG19 y FlowNet..

Recomendaciones

Incrementar la cantidad de datos representativos del problema a resolver, especialmente los clips de video de 10 y 15 segundos de duración, la calidad y cantidad de los datos puede ser crucial para el éxito en el desarrollo de soluciones basadas en clasificación de video.

Hacer uso de herramientas computacionales en la nube como Google Colaboratory o Gradient de PaperSpace pueden ser de gran ayuda, ya que permiten un mayor alcance de las soluciones de software por su capacidad de procesamiento y hardware comparado con un computador de escritorio, especialmente por el uso de GPU.

Trabajos Futuros

El modelo de clasificador de vídeo binario propuesto funciona con videos en tiempo diferido, y se puede mejorar su desempeño incrementando los datos de entrenamiento y variando la forma de procesamiento de los datos antes de ser ingresados al clasificador, por tanto en un futuro se podrían implementar otros algoritmos para clasificadores multiclase de video en tiempo real.

Bibliografía

Bautista Ramos, S. (2021). *Estudio de la longevidad aplicando Redes Neuronales Artificiales.* Universidad Carlos II de Madrid.

Borja Robalino, R., Monleon Getino, A., & Rodellar Benedé, J. (2020). Estandarización de Métricas de Rendimiento para Clasificadores Machine y Deep Learning. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 172-184.

Cañadas, R. (22 de Noviembre de 2021). *Redes Neuronales Recurrentes*. Retrieved 25 de 07 de 2022, from Abdatum: https://abdatum.com/tecnologia/redes-neuronales-recurrentes

Chollet, F. (2018). *Deep Learning with Python.* Manning Publications Co.

Comisión Interamericana de Derechos Humanos. (2009). *Informe Sobre Seguridad Ciudadana y Derechos Humanos.*

Dynatec. (07 de Julio de 2021). *Computer Visión: la visión artificial y sus aplicaciones.* https://dynatec.es/2021/07/24/computer-vision-la-vision-artificial-y-sus-aplicaciones/

ECU 911. (2016). *Cámaras de Videovigilancia*. https://www.ecu911.gob.ec/camaras-de-videovigilancia/

ECU 911. (2021, Junio 2). *Sistema Integrado de Seguridad ECU 911*. https://www.ecu911.gob.ec/camaras-de-videovigilancia/

Ferguson, M., Ak, R., Lee, Y.-T., & Law, K. (2017). *Automatic Localization of Casting Defects with Convolutional Neural Networks.* IEEE International Conference on Big Data (Big Data).

Fernández Carrobles, M. M., Deniz, O., & Maroto, F. (2019). Gun and Knife Detection Based on Faster R-CNN for Video Surveillance. In *Pattern Recognition and Image Analysis* (pp. 404-452). Springer International Publishing.

Fiscalía General del Estado. (2021). *Estadísticas de Robos.* https://www.fiscalia.gob.ec/estadisticas-de-robos/

Fiscalía General del Estado. (2021). *Estadísticas de Robos*. Retrieved 24 de Marzo de 2022, from https://www.fiscalia.gob.ec/estadisticas-de-robos/

García Carretero, D. M., & Parada Medina, R. (2020). *Detección de eventos anómalos en un entorno industrial mediante del uso de técnicas de Federated -learning.* Cataluña: Universitat Oberta de Catalunya.

Gestal Pose, M. (2009). *Introducción a las Redes de Neuronas Artificiales.* Departamento de Tecnologías de la Información y las Comunicaciones de la Universidad de Coruña.

IBM. (17 de 08 de 2021). *Redes Neuronales*. Retrieved 14 de Junio de 2022, from https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=nodes-neural-networks

Jasso López, L. C. (2020). Seguridad ciudadana y tecnología: uso, planeación y regulación de la videovigilancia en Latinoamérica. *Dikê, Revista de Investigación en Derecho, Criminología y Consultoría Jurídica, 14*(27), 5-27.

Jasso López, L. C. (2020). Seguridad ciudadana y tecnología: uso, planeación y regulación de la videovigilancia en Latinoamérica. *Revista de investigación en Derecho, Criminología y Consultoría Jurídica, 14*(27), 5-27.

Kelleher, J. D. (2019). *Deep LearningThe MIT Press Essential Knowledge series.* MIT Press.

Lin, W., Gao, J., Wang, Q., & Li, X. (2021). Learning to detect anomaly events in crowd scenes from synthetic data. *Neurocomputing*(436), 248-259.

Ministerio del Interior. (2019). *Plan Específico de Seguridad Pública y Ciudadana.*

OpenCV team. (s.f.). *OpenCV-Python Tutorials*. Retrieved 15 de 06 de 2022, from https://docs.opencv.org/4.x/d9/df8/tutorial_root.html

Oxford Languages. (s.f.). *Lexico: Diccionario de inglés y español*. Retrieved 05 de Junio de 2022, from https://www.lexico.com/es

Paez Murillo, C. A., Peón Escalante, I. E., & Ramírez Pedraza, Y. (2018). Contexto de la seguridad ciudadana en América Latina y el Caribe: revisión de literatura (2007- 2017).

Revista Científica General José María Córdova, 16(24), 83-106. https://doi.org/10.21830/19006586.360

Pérez Esquivel, A. (2021). Desafíos de la videovigilancia automatizada. *Derecho y Ciencias Sociales*(24), 100-122.

Ramírez Alcocer, U. M., Tello Leal, E., & Ríos Alvarado, A. B. (2018). Modelo Basado en Redes Neuronales Recurrentes LSTM para la Predicción de la Siguiente Actividad en Procesos de Negocio. *Pistas Educativas, 40*(130).

Real Academia Española. (2021). *Diccionario de la lengua española*. Retrieved 2022 de Junio de 05, from https://dle.rae.es

Sreenuu, G., & Saleem, M. A. (2019). Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *J Big Data, 6*(48).

Ullah, W., Ullah, A., Hussain, T., Khan, Z., & Baik, S. (2021). Videos, An Efficient Anomaly Recognition Framework Using anAttention Residual LSTM in Surveillance. *Sensors, 21*(8).

Wang, L., Li, W., Li, W., & Van Gool, L. (2018). Appearance-and-Relation Networks for Video Classification. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1430-1439.

Waqas, S., Chen, C., & Mubarak, S. (2018). Real-world Anomaly Detection in Surveillance Videos. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Wu, D., Zheng, S.-J., Zhang, X.-P., Yuan, C.-A., Cheng, F., Zhao, Y., Lin, Y.-J., Zhao, Z.-Q., Jiang, Y.-L., & Huang, D.-S. (2019). Deep Learning-based Methods for Person Re-Identification: A Comprehensive Review. *Neurocomputing, 337*, 354-371.

Xu, J., Xue, K., & Zhang, K. (2019). Current status and future trends of clinical diagnoses viaimage-based deep learning. *Theranostics, 9*(25).

Apéndices