UNIVERSIDAD
NACIONAL
DE COLOMBIA

Implementar un sistema de reconocimiento e identificación de rostros sobre secuencias de video mediante un modelo de Redes Neuronales Convolucionales y Transfer Learning

Fabio Andres Roa Garcia

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2021

Implementar un sistema de reconocimiento e identificación de rostros sobre secuencias de video mediante un modelo de Redes Neuronales Convolucionales y Transfer Learning

Fabio Andres Roa Garcia

Trabajo de grado presentado como requisito parcial para optar al título de:

Magíster en Ingeniería de Sistemas y Computación

Director:

Ph.D Luis Fernando Niño Vásquez

Línea de Investigación:

Sistemas Inteligentes

Grupo de Investigación:

Laboratorio de investigación en sistemas inteligentes - **LISI**

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2021

La tecnología es importante, pero lo único que realmente importa es qué hacemos con ella.

Muhammad Yunus

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

_____

Fabio Andres Roa Garcia

Fecha 20/08/2021

Agradecimientos

Agradezco a Stiven Alejandro, Deivid Nikolas y Grace Maria, mis hermosos hijos, ya que son ellos quienes me motivan para seguir superándome y para alcanzar nuevas metas. A mi esposa por estar a mi lado y permitirse soñar junto a mí un mejor futuro. A mis padres por todo el apoyo y la motivación. Adicionalmente un agradecimiento a la Universidad Nacional de Colombia, especialmente al profesor Luis Fernando Niño y al grupo LISI por guiarme en el desarrollo de este proyecto.

Resumen

Implementar un sistema de reconocimiento e identificación de rostros sobre secuencias de video mediante un modelo de Redes Neuronales Convolucionales y Transfer Learning

Hoy en día gracias a la innovación tecnológica se ha logrado obtener un aumento significativo en la producción de contenidos multimedia a través de dispositivos como celulares Tablet y computadoras.

Este aumento de contenido multimedia en su mayor parte está en formato de video e implica una necesidad de encontrar información útil sobre este tipo de formato, pero el problema resulta ser una tarea tediosa ya que no se puede analizar información útil sobre los vídeos sin que resulte en un uso excesivo de recursos y largos tiempos de ejecución.

Afortunadamente en el campo de la biometría y análisis de imágenes se han dado avances importantes en los últimos años, de esta manera, se han formalizado técnicas de reconocimiento facial mediante el uso de redes neuronales convolucionales apoyándose por algoritmos de transfer learning y clasificación.

Estas técnicas en conjunto, se pueden aplicar al análisis de video, realizando una serie de pasos adicionales para optimizar los tiempos procesamiento y la precisión del modelo.

El propósito de este trabajo es utilizar el modelo ResNet-34 junto con transfer Learning para el reconocimiento e identificación de rostros sobre secuencias de video.

Palabras clave: Redes Neuronales Convolucionales, Aprendizaje profundo, Reconocimiento facial, Transferencia de aprendizaje, Aprendizaje residual profundo, k vecinos más próximos, OpenCV, Dlib.

Abstract

Implement a face recognition and identification system on video sequences through a model of Convolutional Neural Networks and Transfer Learning

Nowadays, thanks to technological innovation, it has been possible to obtain a significant increase in the production of multimedia content through devices such as tablet cell phones and computers.

This increase in multimedia content for the most part is in video format and implies a need to find useful information about this type of format, but the resulting problem will be a tedious task since it is not possible to analyze useful information about the vídeos without it being in excessive use of resources and long execution times.

Fortunately, in the field of biometrics and image analysis, there have been important advances in recent years, in this way, facial recognition techniques have been formalized through the use of convolutional neural networks supported by transfer learning and classification algorithms.

Together, these techniques can be applied to video analysis, performing a series of additional steps to optimize processing times and model accuracy.

The purpose of this work is to use the ResNet-34 model and Transfer Learning for face recognition and identification on video footage.

Keywords: CNN, Deep Learning, Face Recognition, Transfer Learning, Deep Residual Learning, KNN, OpenCV, Dlib.

Este Trabajo Final de maestría fue calificado en diciembre de 2021 por el siguiente evaluador:

Jean Pierre Charalambos Hernández
Profesor Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia

Contenido

Pág.

Lista de figuras

Pág.

Lista de tablas

Pág.

Siglas

Abreviatura	Término
IA	Artificial Intelligence
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
CCTV	Circuito Cerrado de Televisión
DL	Deep Learning
VGG	Visual Geometry Group
ML	Machine Learining
CUDA	Compute Unified Device Architecture

1. Introducción

Hoy en día la innovación tecnológica ha permitido un aumento significativo en la producción de contenido multimedia, como imágenes, audio, texto y video, estos contenidos son producidos y compartidos diariamente gracias a las cámaras de video incorporadas en dispositivos como celulares, tablets y computadores [1].

El formato de video se ha proliferado rápidamente por internet y al mismo tiempo surge la necesidad de encontrar información útil sobre el contenido multimedia, por ejemplo realizar un filtro basado en texto permite recuperar un conjunto de información de acuerdo al texto ingresado, pero buscar información útil en contenido de vídeo resulta ser una tarea tediosa.

Para el análisis de vídeo se pueden utilizar diferentes técnicas, que van desde el análisis manual realizado por un operador humano, el análisis de aprendizaje transductivo y el análisis basado en técnicas de inteligencia artificial [2].

Esta última permite analizar la información del video por ejemplo para ubicar e identificar rostros humanos, la implementación de inteligencia artificial para la búsqueda de rostros humanos sobre video resulta pertinente toda vez que favorece el análisis óptimo sobre el contenido, siendo esto de gran utilidad para diferentes problemas del mundo real.

En este trabajo nos enfocamos en el reconocimiento e identificación de rostros sobre secuencias de video utilizando el modelo de redes neuronales convolucionales ResNet-34 y transfer learnig, específicamente se buscó implementar un marco adecuado con alta precisión (accuracy) en la identificación de personas y que a la vez responda de forma rápida (latency), mediante el uso de herramientas como dlib, OpenCV y Face_Recognition.

Este trabajo está organizado de la siguiente manera:

En la Seccion 2, se presentan los objetivos que se esperan alcanzar en el desarrollo de las actividades integradas del trabajo.

En la Sección 3, se puede encontrar la Justificación y motivación para el desarrollo del trabajo, así como la revisión de los conceptos relacionados al mismo.

En la Sección 4, se realiza una introducción a la metodología implementada y una explicación breve de las fases que la componen.

En la sección 5, se describe de forma conceptual los elementos necesarios para el desarrollo del proyecto.

En la sección 6, se presenta el paso a paso utilizado para la recolección de los datos, así como el proceso de depuración y alistamiento de los mismos.

En la sección 7, se relacionan los ambientes o entornos de trabajo utilizados para la ejecución del modelo y las características de hardware asociadas a cada ambiente.

En la sección 8, se expone como se llevó a cabo el proceso de entrenamiento del modelo en cada uno de los ambientes de trabajo propuestos.

En la sección 9, se realiza la evaluación del modelo con un dataset de imágenes y con un dataset de video clips.

En la sección 10, se presentan los resultados obtenidos en el desarrollo del trabajo y finalmente en la sección 11 se presentan las conclusiones y recomendaciones como apoyo a trabajos posteriores.

2. Objetivos

2.1.1 Objetivo General

Implementar un sistema de reconocimiento e identificación de rostros sobre secuencias de video mediante el uso de un modelo de Redes Neuronales Convolucionales y Transfer Learning .

2.1.2 Objetivos Específicos

- OE1: Seleccionar un conjunto de datos de vídeos que contengan rostros humanos

- OE2: Seleccionar un modelo de Redes Neuronales Convolucionales pre-entrenadas para reconocimiento facial.

- OE3: Realizar un proceso de transferencia de aprendizaje *(transfer learning)* del modelo de Redes Neuronales Convolucionales para el reconocimiento facial en vídeos.

- OE4: Validar la calidad y rendimiento del modelo implementado.

3. Marco teórico

El reconocimiento e identificación de sujetos basados en técnicas biométricas de reconocimiento facial son esenciales para la industria y representan una parte integral en la automatización de datos de imagen o video en información procesable [3]. Sin embargo, se presentan muchos desafíos en cuanto al desempeño de las técnicas utilizadas hasta el momento, primordialmente porque su implementación no contempla aspectos de escenarios del mundo real [4] [5] en donde las técnicas tradicionales realizan la identificación de rostros basados en imágenes estáticas[6], pasando por alto que la mayor cuota de contenido digital está siendo aportado a partir de contenido multimedia especialmente del video [7], a través de sistemas de video vigilancia [8], cámaras web y cámaras de dispositivos móviles [9]. Por ende, es necesario evaluar la posibilidad de enfocar los esfuerzos en construir un sistema robusto, rápido y eficiente [10] que permita el reconocimiento e identificación de personas en secuencias de video.

Con el fin de ilustrar la necesidad de un modelo como el propuesto en el párrafo anterior, se hace necesario un rápido recorrido sobre los aspectos más importantes del desarrollo de las técnicas de reconocimiento facial:

La era de la globalización ha traído consigo una serie de ventajas que hacen más fácil algunas tareas que los humanos no podían hacer o hacer de manera deficiente. En las últimas décadas los esfuerzos han sido enfocados a crear máquinas y aplicaciones que puedan apoyar las labores rutinarias de las personas agilizando procesos empresariales y científicos en disciplinas como medicina, artes, clima, minería, ambiente y agricultura, entre otras [11].

Estos esfuerzos han surgido del deseo del ser humano de crear modelos evolucionados que tengan características “inteligentes” y que permitan la ejecución de tareas de forma automatizada. Lo que conlleva a la generación de numerosos trabajos que se materializaron en máquinas con la capacidad de desarrollar cálculos matemáticos a gran velocidad, procesar lenguaje natural, jugar, reconocer objetos, componer música y muchas otras tareas que inicialmente solo eran ejecutadas por los humanos. De esta forma, nace el dominio de la Inteligencia Artificial (AI), término acuñado por John McCarthy en 1956 y que representa el estudio de la disciplina del conocimiento [12] .

Una de las áreas de interés dentro de la AI es la rama denominada visión artificial, cuyo principal objetivo es emular la visión humana dotando a un sistema informático con capacidades de análisis de video e imágenes en tiempo real, que permitan la identificación del contexto, siendo capaz de realizar la detección de objetos y rostros humanos.

Estas aplicaciones tienen mucho potencial siendo empleada entre otras cosas en áreas como seguridad, medicina y transporte [13][14]. Justamente en el sector de

la seguridad, en el campo de la Biometría, también se ha contado con importantes avances que permiten la identificación de personas a través de sus características intrínsecas que pueden ser características de conducta o físicas [15] y que gracias al gran potencial de la técnica de reconocimiento facial se puede realizar la identificación de individuos a partir de sus características faciales.

Pero, construir un sistema de visión artificial no es una tarea sencilla, ya que requiere de un alto procesamiento informático y el uso de diferentes técnicas y modelos que se ajusten al contexto específico, como es mencionado por Kim [16] en su trabajo de reconocimiento de objetos en movimiento usando recursos limitados. Además, se debe garantizar que los sistemas de reconocimiento facial sean confiables y respondan adecuadamente a parámetros de calidad guiados por las condiciones de iluminación [17], o como lo describe Nguyen-Meidine [18] en su trabajo de comparación de los detectores de rostro y cabeza basados en CNN para aplicaciones de videovigilancia en tiempo real, el rostro humano presenta grandes desafíos para el reconocimiento por computadora, ya que la cara presenta una gran cantidad de características como son la orientación, pose, oclusión parcial, la expresión facial, presencia de bello, de anteojos, color de piel, dimensión y fondos desordenados.

Ahora bien, el campo de visión artificial cuenta con varias técnicas para la solución de problemas de reconocimientos de rostros: por un lado los métodos clásicos en los que destacan las técnicas basadas en la apariencia [20], donde el objetivo de los algoritmos es clasificar las diferentes caras en un nuevo subespacio vectorial, de estas técnicas sobresalen:

- Análisis de componentes principales (PCA)
- Transformada discreta del coseno (DCT)
- Proyecciones de conservación de localidades (LPP)
- Análisis discriminante lineal (LDA)

Estos métodos clásicos son muy utilizados en la solución de múltiples problemas, pero presentan un menor rendimiento comparados con los métodos basados en aprendizaje profundo [10], siendo estos últimos más valorados principalmente por la diferencia en rendimiento y calidad de la respuesta, aunque pueden tener un mayor costo computacional requiriendo un hardware más robusto con altos requisitos de CPU y GPU.

El uso de Redes Neuronales Convolucionales (CNN, por su sigla en inglés para Convolutional Neural Networks) ha logrado un gran éxito en el procesamiento y clasificación de imágenes y reconocimiento de patrones en objetos. Por ejemplo, se hace referencia a la comparación con métodos tradicionales, como PCA, LDA y los Histogramas de Patrones Binarios Locales (LBPH), mencionados en el trabajo de Bolívar Chacua [19] y al trabajo de Jisoo Park [20] en el que realiza la comparación entre 5 métodos convencionales como son simple thresholding, adaptive thresholding, background subtraction, K-means clustering y las CNN. En

los dos trabajos mencionados se encuentran resultados superiores en la detección de objetos mediante el uso de las redes neuronales convolucionales.

Aunque el surgimiento de las redes neuronales convolucionales profundas ha llevado a grandes mejoras en el rendimiento de varios conjuntos de datos de reconocimiento y verificación de rostros, el problema de reconocimiento facial aún no está resuelto [21] ya que cuenta con adversidades como son:

1. Vulnerabilidades en el sistema biométrico [22][23], donde Galbally presenta una serie de amenazas potenciales y realiza una clara explicación en las diferencias que existen entre comprometer una clave o token a comprometer una huella dactilar o rasgos faciales, ya que una clave puede ser cambiada frecuentemente pero no pasa lo mismo con la huella dactilar.

2. Robo de identidad: Este representa la vulnerabilidad a la que se ve expuesta el sistema cuando un atacante presenta frente a un dispositivo con cámara la imagen impresa de un rostro, logrando la identificación y autorización por parte del sistema [24], dejando en evidencia un problema del reconocimiento facial en sistemas de seguridad en tiempo real.

3. Entrenamiento: Los algoritmos de aprendizaje de redes neuronales tienen como inconveniente el alto costo computacional generado en el proceso de entrenamiento, básicamente la red neuronal debe aprender un nuevo modelo desde cero cada vez que las categorías de los objetos difieren: por ejemplo factores externos como iluminación, fondo, tipo de sensor, ángulo de visión implican un nuevo proceso de entrenamiento que requiere una gran cantidad de datos [25][26].

Para mitigar el desgaste y tiempo de entrenamiento, surge una técnica innovadora denominada Transfer Learning, que busca usar el conocimiento existente del modelo para ayudar al proceso de aprendizaje, ahorrando tiempo y reduciendo el costo del entrenamiento desde cero [27].

3.1 Justificación y Motivación

Las aplicaciones desarrolladas para reconocimiento facial se pueden clasificar en dos grupos: análisis de imágenes y análisis de video, cada una de ellas cuenta con sus propios desafíos técnicos en donde el criterio para su uso depende del contexto donde van a ser aplicadas. De esta forma y de acuerdo a la naturaleza propia del análisis biométrico mediante el reconocimiento facial, se clasifica el trabajo que se propone en el grupo de análisis de video, ya que está enfocado en buscar y encontrar en las secuencias de video los rostros de personas.

A continuación, se presentan algunos problemas comunes en el uso de reconocimiento facial basado en video, con el fin de identificar las diferentes soluciones propuestas en la revisión de literatura, de tal forma que la apropiación de dichas soluciones sirva como base para el desarrollo del objetivo general de este trabajo.

Uno de los problemas clave del reconocimiento facial en el análisis de secuencias de video es extraer las características discriminatorias de video espacio-temporal [28], por esto Zhang propone el uso de un método híbrido en donde se emplean una CNN espacial que procesa imágenes faciales estáticas y una CNN temporal que procesa imágenes de flujo óptico, con el fin de aprender por separado las características de espacio-tiempo.

En la mayoría de sistemas basados en cámaras, la detección y localización de personas supone un reto importante, debido al movimiento que puede realizar la persona, la cámara o ambas, además de factores como la distancia entre la persona y la cámara que puede generar desenfoque, la variación de la luz y la detección de otros objetos [29]. En este trabajo Herrmann propone un sistema de dos etapas que incluya un método de generación de propuestas (MSER) Maximally Stable Extremal Regions , por sus siglas en Inglés y un clasificador basado en CNN para verificar si las propuestas detectadas realmente son personas.

Factores como la alta dimensionalidad, no rigidez, variación multiescala, influencia de la iluminación y ángulo de las expresiones faciales dificultan el proceso de obtención de imágenes o vídeos. En consecuencia, los algoritmos de reconocimiento facial presentan baja precisión en el reconocimiento, problemas de sobreajuste, explosión de gradiente e inicialización de parámetros [30]. Por esto, Zhiwen Liu propone un algoritmo basado en la inicialización adaptativa de parámetros mediante CNN y el método de red de memoria a largo-corto plazo (LSTM, del inglés long–short-term memory network).

La identificación de rostros en películas o vídeos representa un gran desafío, debido a que la apariencia de los personajes puede variar drásticamente entre escenas, influyendo también los diferentes estilos cinematográficos, dificultando así el aprendizaje de una representación facial universal entre vídeos [31], Zhanpeng Zhang, en ausencia de anotaciones faciales precisas en el video objetivo, propone un ciclo de retroalimentación en el que la representación profunda proporciona características robustas para el agrupamiento de rostros.

Paralelamente, Enrique G. presentó una solución basada en la implementación del algoritmo MSSRC, Mean Sequence Sparse Representation-based Classification , para el problema que representa la falta de vídeos anotados y el costoso procesamiento requerido en el reconocimiento facial sobre secuencias de video [32].

Además de los desafíos técnicos, también se debe tener en cuenta la privacidad de los datos de las personas que aparecen en imágenes y vídeos [33]. El buen manejo y la protección de la información relacionada con estos sistemas está regulada por el marco legal de cada país, para este caso concreto la ley Colombiana en el literal c del artículo 3 de la ley 1581 de 2012 determina que la tarea de monitoreo y observación a través de sistemas de vigilancia implica la recolección de datos personales [34] y por ende, dicta una serie de consideraciones para garantizar el tratamiento de los datos personales haciendo uso de las mejores prácticas nacionales e internacionales en esta materia.

La captura de video sobre individuos presenta un escenario en el cual se puede solicitar la protección de datos evocando el derecho a la privacidad [35] y por ende anonimizar los rostros presentes en el recurso multimedia [36], allí se propone la identificación y ofuscación de rostros mediante el uso de una arquitectura Faster R-CNN, la cual hace que el costo computacional sea más eficiente.

Finalmente, habiendo realizado un análisis de la situación actual y teniendo en cuenta aspectos como el marco legal, la dificultad que se presenta al obtener grandes cantidades de datos para entrenar un modelo y la fuerte tendencia de analizar contenidos sobre video, se opta por implementar un sistema de reconocimiento e identificación de rostros sobre secuencias de video haciendo uso de modelos de redes neuronales convolucionales y Transfer Learning.

3.2 Revisión de Conceptos

A continuación se describe brevemente los diferentes conceptos, métodos, librerías y tecnologías exploradas en el presente trabajo:

3.2.1 Inteligencia Artificial (AI):

De acuerdo con Turin [37] la AI es el proceso en el cual una máquina aprende a realizar una tarea mostrando un comportamiento inteligente a partir de los datos o de su entorno.

El concepto de AI se ha vuelto muy popular en los últimos años debido al aumento significativo de los datos y de la capacidad de procesamiento de los mismos, allí donde anteriormente se tenía un problema de rendimiento y tiempos de respuestas absurdamente largos, hoy en día se cuenta con hardware robusto que aumenta la capacidad de procesamiento permitiendo la implementación de modelos de AI mucho más complejos y eficientes.

3.2.2 Red Neuronal Artificial (Artificial Neural Network)

La red neuronal artificial (ANN), es una herramienta potente que permite realizar aprendizaje automático dentro de una gama amplia de escenarios, en los que se

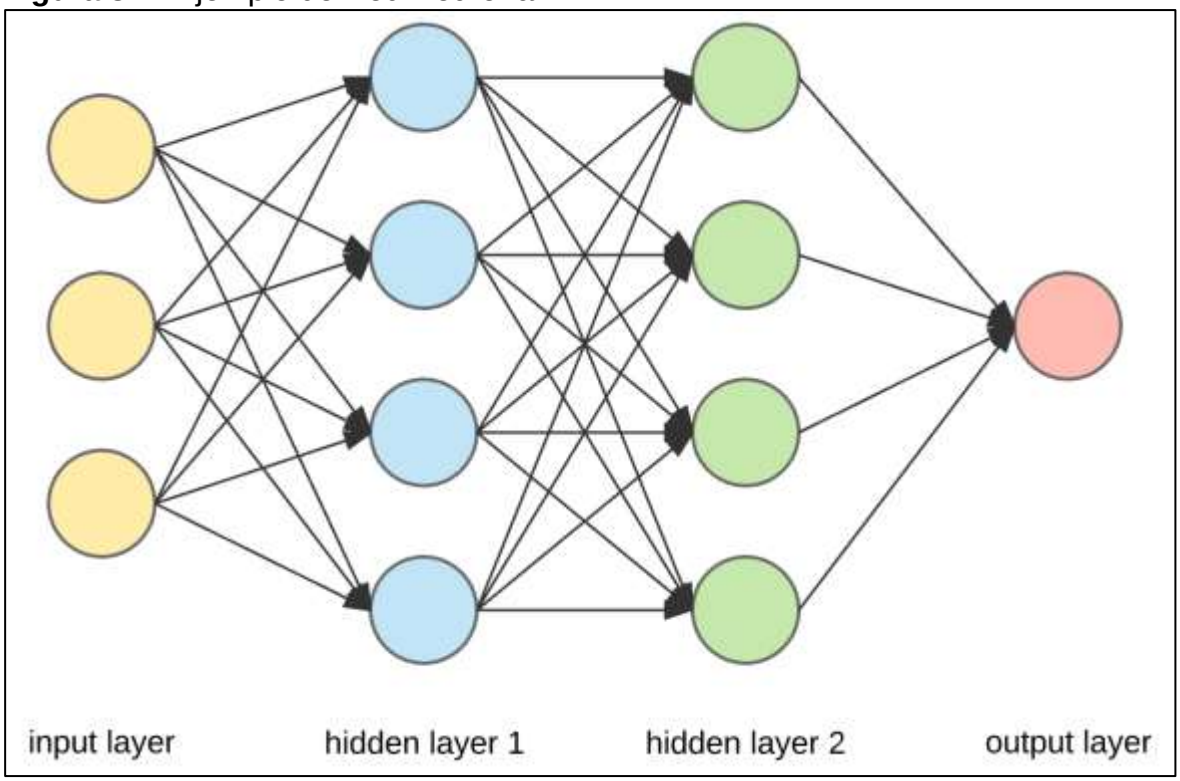
incluyen visión por computador, procesamiento de lenguaje natural, Juegos de estrategia, Salud, Marketing etc. Algunas de sus ideas nacen inicialmente inspiradas en el estudio del cerebro humano [38] y de las reglas de aprendizaje supervisado.

La red neuronal está formada por un conjunto de células elementales Figura 3-1 denominadas perceptrón simple: Un perceptrón es la representación matemática inspirada en una neurona biológica[39] y al igual que esta cuenta con un conjunto de entradas llamadas dendritas y un canal de salida llamo Axón.

La unión de varios perceptrones da lugar a una red neuronal que puede tener variedad de capas permitiendo así especializar la red en una labor específica.

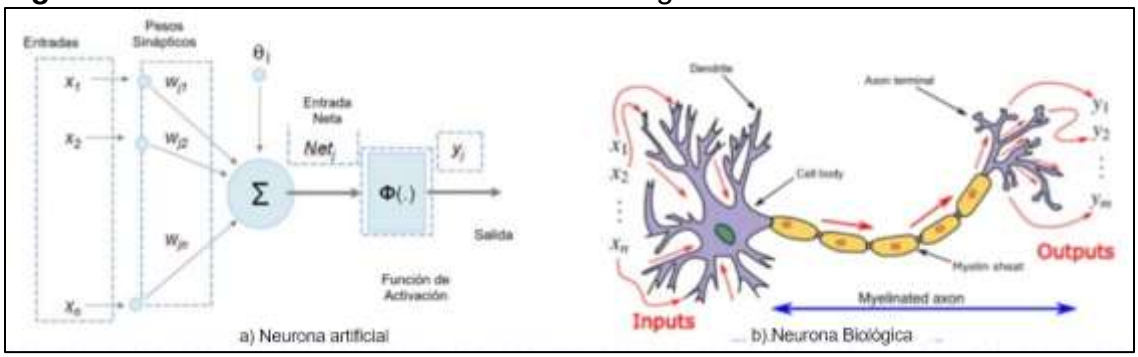
De manera general se puede decir que una ANN representa un conjunto de algoritmos de optimización que intentan modelar matemáticamente el proceso que aprenden utilizando una función matemática no lineal que transforma grupos de variables de entrada en un conjunto de salida.

Figura 3-1: Ejemplo de Red Neuronal



Tomada de [38]

Figura 3-2 Neurona artificial vs Neurona Biológica



Tomada de https://www.researchgate.net/figure/Figura-7-Neurona-biologica-versus-artificial-Una-neurona-artificial-es-una-unidad-de_fig6_262746657

3.2.3 Aprendizaje Profundo (Deep Learning)

El aprendizaje profundo (Deep Learning) es un tipo de aprendizaje orientado a hacer abstracciones de alto nivel: aprender cosas complejas.

Goodfellow y Joshua [40] mencionan que el verdadero problema de la AI es resolver problemas que para los humanos son sencillos de hacer, pero difíciles de describir formalmente, como el reconocimiento del habla, la identificación de imágenes o hacer predicciones ante una situación. Una forma de desarrollar este problema es en el proceso de aprendizaje, donde una computadora debe tener la capacidad de entender el mundo como una jerarquía de conceptos relacionados entre sí, de esta forma construir conceptos complejos a partir de otros más simples, al recopilar el conocimiento de la experiencia, se evita que un humano deba especificarlo él mismo a la computadora. Si se expresara gráficamente la jerarquía de conceptos, esta se observaría en varias capas que generarían un "gráfico profundo", de ahí el nombre.

El Deep learning es una de las bases de la inteligencia artificial (AI) y el interés actual en el Deep learning se debe en parte al auge que tiene ahora la inteligencia artificial.

A continuación se presenta un conjunto de novedades que están integrando avances en el aprendizaje profundo:

- Mejoras algorítmicas han elevado el desempeño de los métodos del aprendizaje profundo.
- Nuevos métodos de aprendizaje basado en máquina han mejorado la precisión de los modelos.
- Se han desarrollado nuevas clases de redes neurales que encajan bien en aplicaciones como la traducción de texto y la clasificación de imágenes.

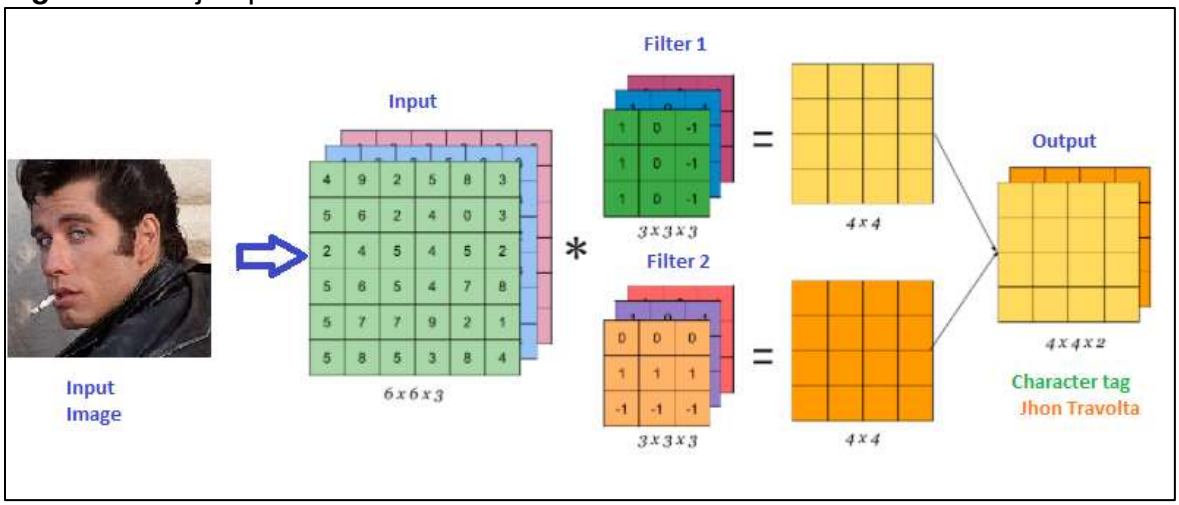
- Se tienen más datos disponibles para construir redes neurales con muchas capas profundas, incluyendo datos de streaming de la IOT, datos textuales de medios sociales, notas de médicos y transcripciones de investigaciones.
- Los adelantos en procesamiento en la nube y unidades de procesamiento gráfico han puesto a nuestra disposición una cantidad increíble de poder de cómputo.

3.2.4 Redes Neuronales Convolucionales (Convolutional Neural Network)

Son muchas las referencias que tratan de hacer un acercamiento a las Redes Neuronales Convolucionales (CNN). Buscando un lenguaje fácil de entender, son un tipo de arquitectura artificial basadas en el funcionamiento de la corteza visual humana, estos algoritmos reconocen patrones a partir de imágenes, principalmente, aquellas características que en principio se componen de líneas y posteriormente en conjunto aprenden a diferenciar elementos más complejos, por ejemplo: figuras geométricas, objetos específicos, animales, personas, lugares, entre muchos otros. Cabe resaltar que a diferencia de los humanos, los ordenadores procesan las imágenes en forma de tensores de números, los cuales, para el caso de imágenes a escala de grises (blanco y negro), albergan valores que van de 0 hasta 255 según la intensidad de luz, cuando se requiere trabajar con este tipo de imágenes, a veces se requiere hacerle transformaciones y adaptarlas para que los resultados sean óptimos. Sin embargo, para imágenes a color existen tres canales RGB: Red (rojo), Green (verde), y Blue (azul), los cuales varían dependiendo de la intensidad de la luz, el color de fondo, la perspectiva y el ángulo desde el que se haya obtenido la imagen dada [41]

Este tipo de redes convolucionales abstraen características y patrones en las imágenes a partir de operaciones matemáticas como kernels que funcionan a modo de filtro. Este último funciona como una matriz para hacer transformaciones en la imagen y así poder obtener contornos a un mayor nivel de detalle.

Figura 3-3: Ejemplo de CNN

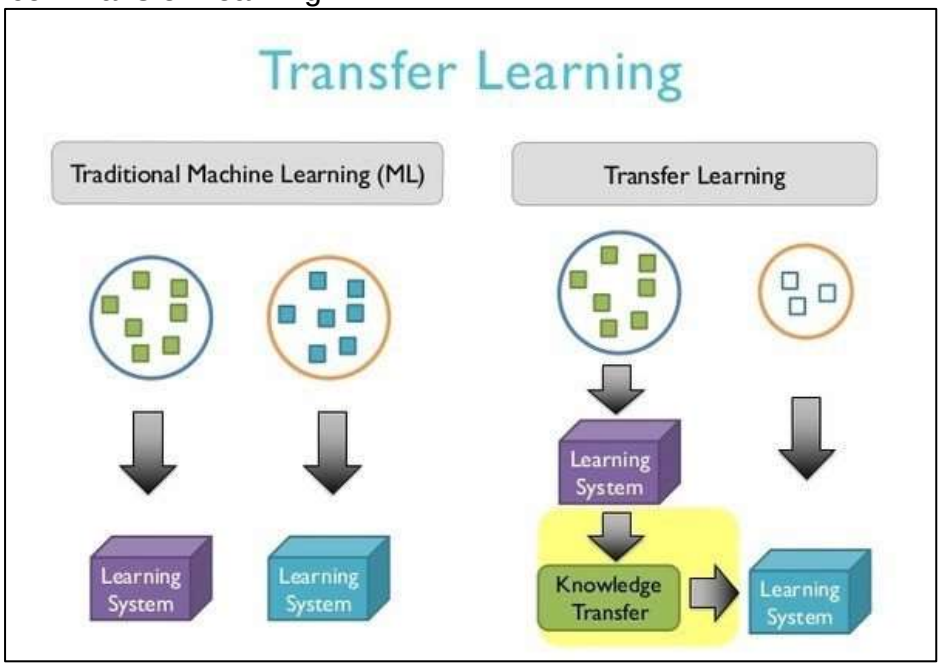


Como se observa en la figura 3-3, inicialmente tenemos una imagen como input que se traduce una a una matriz con los valores de acuerdo a los colores que la componen, posteriormente tenemos una convolución definida como la multiplicación de matrices entre el kernel y la imagen, en el cual el kernel recorre la imagen y obtiene un nuevo tensor. Otro elemento importante es el Padding el cual consiste en añadir un borde de ceros en el tensor. Por último se tiene la salida de la red, que para el ejemplo se corresponde a la etiqueta del actor Jhon Travolta.

3.2.5 Transferencia de Aprendizaje (Transfer Learning)

El transfer learning se manifiesta como una de las técnicas más importantes del Deep learning para el aprendizaje automático en AI. Consiste en aprovechar una gran cantidad de información relacionada con una resolución de un problema y utilizarla sobre otro distinto, teniendo en cuenta que éste debe compartir características similares tal y como se ilustra en la Figura 3-4. Dicho de otra forma, modificar patrones ya entrenados (o redes neuronales) para reconocer ciertas características y así poder reconocer otras similares sin realizar el entrenamiento desde cero todas las veces.

Figura 3-4: Diferencia entre el proceso tradicional de entrenamiento en ML y el proceso con Transfer Learning



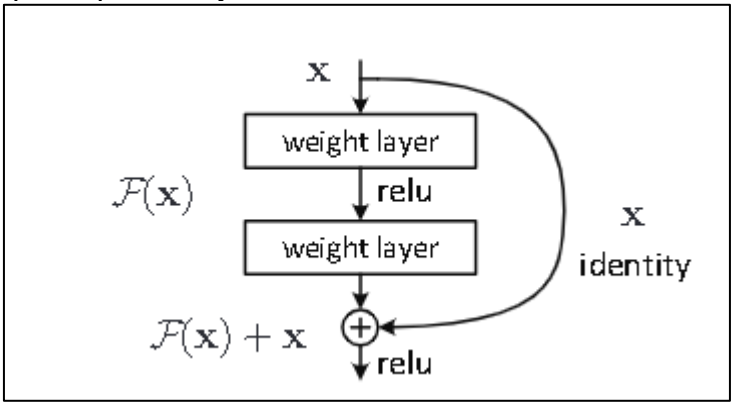
Tomada de https://ichi.pro/es/transfiere-y-aprenderas-descripcion-general-del-aprendizaje-por-transferencia-177126688571069

El transfer Learning aparece como un recurso economizador de tiempo de gigantescas magnitudes de cálculo. Con los datos obtenidos de una tarea similar, pero más genérica, evita unas costosas aproximaciones a un punto de partida óptimo.

3.2.6 Deep Residual Learning

Hasta el año 2015, Las redes neuronales profundas presentaban como problema la dificultad para entrenarlas a medida que aumenta el número de capas, generando un aumento en la variable de error de entrenamiento(degradación de la red) y generando también el problema del desvanecimiento del gradiente. En su trabajo Deep Residual Learning Kaiming He y Xiangyu Zhang [42], propone una arquitectura denominada ResNet (Red Residual) que resuelve el problema de profundidad de la red, implementando una conexión de salto (Función identidad) donde la salida de una capa se alimenta a la capa más profunda en la red. en la siguiente figura se puede apreciar el principio de la función Identidad :

Figura 3-5: Bloque -Aprendizaje Residual



Tomada de [42]

En la Figura 3-5 se ilustra el flujo de una red neuronal convolucional plana (VGG), con una conexión adicional llamada identidad que permite agregar en medio más capas sin que esto degrade el comportamiento de la red.

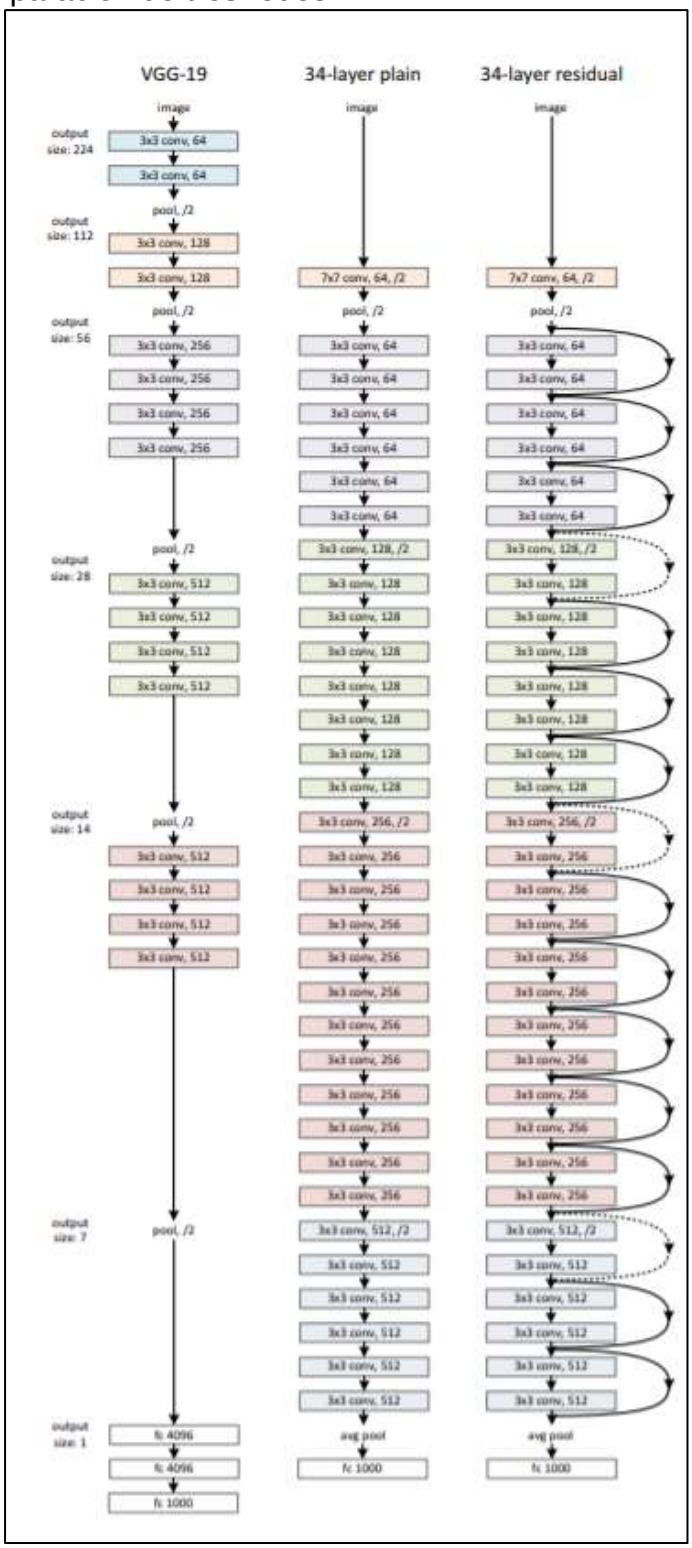
Con este aporte se logró un avance importante en el desarrollo del Deep Learning toda vez que:

- Se resolvió el problema de degradación de las redes con el aumento de capas.

- Permite la construcción de redes más especializadas (muy profundas)
- Resuelve el problema del desvanecimiento del gradiente

A continuación se presenta un ejemplo de la arquitectura de una red plana VGG de 19 capas, una red plana de 34 capas y la misma red de 34 capas con la función identidad (ResNet),

Figura 3-6 : Comparación de tres redes



Tomada de [42]

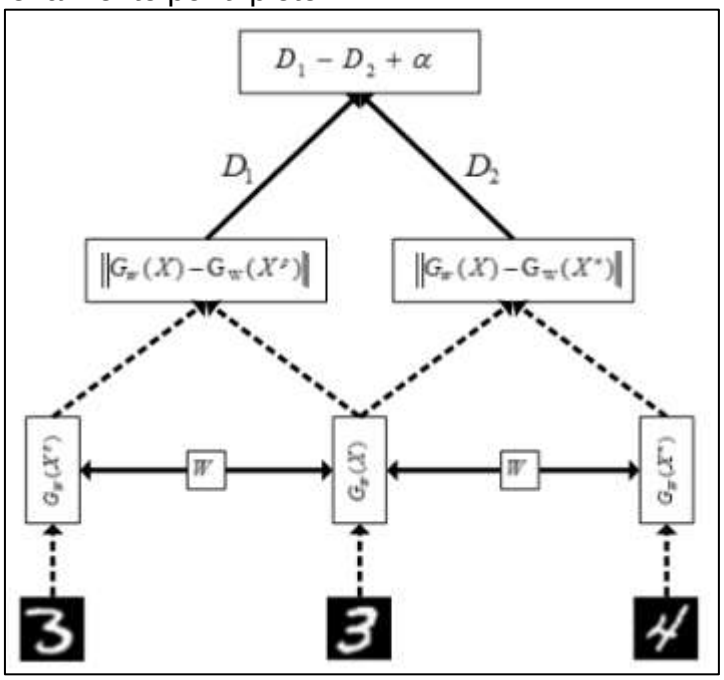
3.2.7 Aprendizaje Métrico profundo (Deep metric learning)

Tiene como objetivo medir la similitud entre muestras manteniendo una métrica de distancia óptima para la tarea de aprendizaje, a diferencia del Deep Learning donde se entrena una red que acepta una imagen de entrada y genera una clasificación / etiqueta para esa imagen, el Deep Metric Learning genera una sola etiqueta con un vector de características de valor real [43].

En este sentido Deep Metric Learning utiliza redes neuronales convolucionales para aprender automáticamente las características discriminatorias de las imágenes pasadas como input, de esta manera calcular la métrica correspondiente y devolver un vector de características únicas para dicha imagen.

Deep Metric Learning implementa un paso de entrenamiento por tripletes que consta de 3 imágenes en las cuales dos de ellas son de la misma clase y una tercera de una clase diferente. Esta técnica utiliza una métrica de distancia con el uso del espacio Euclidiano para comparar los objetos en el proceso de reconocimiento de patrones, de esta forma construye un vector de características basados en las imágenes de la misma clase.

Figura 3-7: Entrenamiento por triplete



Tomada de [43]

3.2.8 OpenCV (Open Computer Vision)

Biblioteca Open Source de visión artificial desarrollada por Intel. Desde su aparición en 1999, ha sido implementada en miles de aplicaciones y hoy en día es reconocida como la biblioteca más popular de visión artificial.

A continuación se mencionan las características más relevantes que han llevado a su popularidad:

- Open source, bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación.
- Multiplataforma, para sistemas operativos Android, Windows, Mac OS X, GNU/Linux, con diversas arquitecturas de hardware x86,x64., ARM (celulares y Raspberry Pi) .
- Documentada y explicada, donde la organización se preocupa por mantener actualizada la documentación para los desarrolladores, ejemplos de implementación de funciones, tutoriales de dominio público, fomentación de sitios de formación.

3.2.9 Librería Dlib

Marco de trabajo escrito en C++ que permite desarrollar aplicaciones de ML y análisis de datos con el fin de solucionar problemas del mundo real.

Dlib es fácilmente integrable con Python y otros lenguajes de programación lo que la convierte en una librería preferida en proyectos de inteligencia artificial.

En este proyecto se usa dlib ya que esta cuenta con una implementación de Deep metric learning, que es usado para construir nuestro vector de características faciales.

Dlib ya cuenta con el modelo ResNet-34 entrenado con más de 3 millones de imágenes del dataset LFW (Labeled Faces in the Wild), de esta forma permite tomar el modelo entrenado y pasarle pocos datos de entrenamiento implementando así el mecanismo de Transfer Learning.

Es de aclarar que según datos del autor de la librería el modelo Resnet-34 comparado con otros modelos tradicionales alcanza un accuracy del 99.38%, lo que la convierte en uno de los modelos con mayor precisión.

3.2.10 Framework Face Recognition

Uno de los campos en donde se ha implementado CNN en investigaciones es el Face Recognition (Reconocimiento facial), el cual consiste en un método para

identificar o verificar la identidad de un individuo usando su rostro. Estos sistemas se pueden utilizar para identificar personas en imágenes (fotos), vídeos o tiempo real.

El Framework Face Recognition envuelve la librería dlib permitiendo su uso de una forma más amigable desde el lenguaje Python.

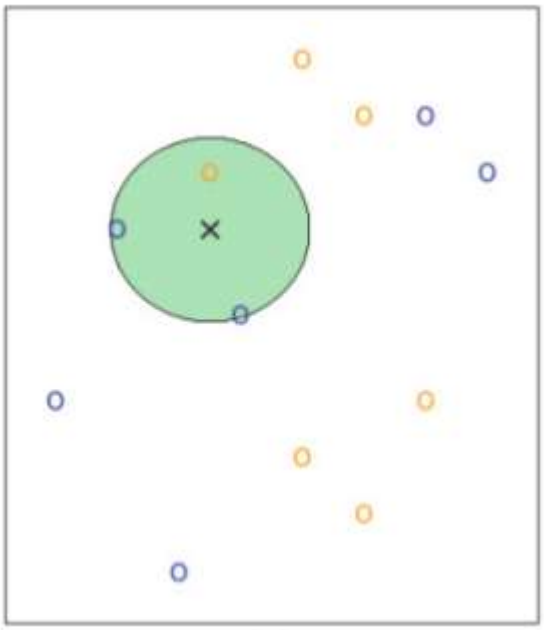
Este framework cuenta con una serie de algoritmos que permiten hacer la tarea de reconocimiento e identificación facial mucho más rápida y precisa.

3.2.11 Vecinos más cercanos (KNN)

Algoritmo de clasificación utilizado en métodos de aprendizaje automático que funciona identificando los vecinos más cercanos de una muestra determinada [44].

Si se proporciona un número entero positivo K y una muestra x0, KNN clasifica calculando la distancia por ejemplo Euclidiana entre las clases de K puntos más cercanos a la muestra x0, tal como se muestra en la siguiente Figura.

Figura 3-8: Observación de los vecinos más cercanos a x



Tomada de [44]

3.2.12 Tipos de licencia

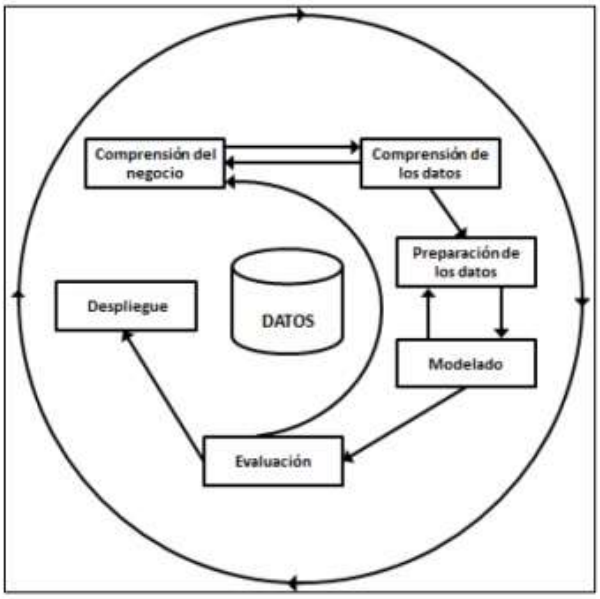
Cuando se habla de software de código abierto (Open Source), se hace referencia a que el código fuente y otros derechos son exclusivos para quienes poseen los derechos de autor, los cuales son publicados bajo una licencia de código abierto o forman parte del dominio público. Por otro lado cuando se habla de archivos como audio, video o texto con una licencia de dominio público, se hace referencia a una licencia de tipo Creative Commons (CC), la cual otorga un permiso al público para usar y modificar estos archivos bajo términos y condiciones de elección.

4. Metodología

Para el desarrollo del presente trabajo se adoptó la metodología (CRISP-DM) Cross-Industry Standard Process for Data Mining por sus siglas en inglés [45], debido a su alta reputación al ser utilizada en muchos estudios y su gran crecimiento como un estándar en la industria, logrando unificar procesos de desarrollo de software, Data Mining y desarrollo de aplicaciones de AI [46].

CRISP-DM está compuesta por 6 fases como se muestra en la Figura 4-1, que permiten cumplir con el desarrollo de los objetivos propuestos, presentando un enfoque sencillo al tiempo de permitir la gestión y dirección del proyecto.

Figura 4-1: Fases de la metodología CRISP-DM



Tomada de [45]

A continuación se realiza una descripción de las fases metodológicas:

Fase 1: Comprensión del negocio
Esta fase se enfoca en comprender los objetivos del proyecto, definir los requisitos y convertirlos en la definición formal del problema.

Fase 2: Comprensión de los Datos:

Esta fase se centra en la recopilación de datos en bruto, teniendo como propósito la calidad de los mismos y la detección de subconjuntos de datos interesantes para la realización del proyecto.

Fase 3: Preparación de los Datos:
En esta fase se cubren todas las actividades relacionadas con la construcción del conjunto de datos final, estas actividades incluyen: limpieza, transformación, discretización, reducción e ingeniería de características.

Fase 4: Modelado:
En esta fase se seleccionan y aplican los diferentes algoritmos y técnicas de modelado como son CNN y *Transfer Learning* .

Esta fase puede ser cíclica dependiendo de las técnicas seleccionadas, si esto es así, la fase retorna a la fase anterior de preparación de datos y continúa iterativamente, hasta que el conjunto de datos sea consecuente con los modelos aplicados.

Fase 5: Evaluación:
Esta fase se enfoca en la evaluación y validación de los modelos construidos, con el fin de medir la calidad y rendimiento de acuerdo a los requerimientos y objetivos del proyecto.

Fase 6: Despliegue:
En esta fase se implementa el producto final en una aplicación del mundo real junto con los entregables asociados a las fases anteriores, así como el informe final que consolide la especificación técnica, desarrollo del proyecto y los resultados obtenidos.

5. Comprensión del negocio

Para llevar a cabo el reconocimiento e identificación facial, se plantea como solución el uso de redes neuronales convolucionales aplicada sobre secuencias de video.

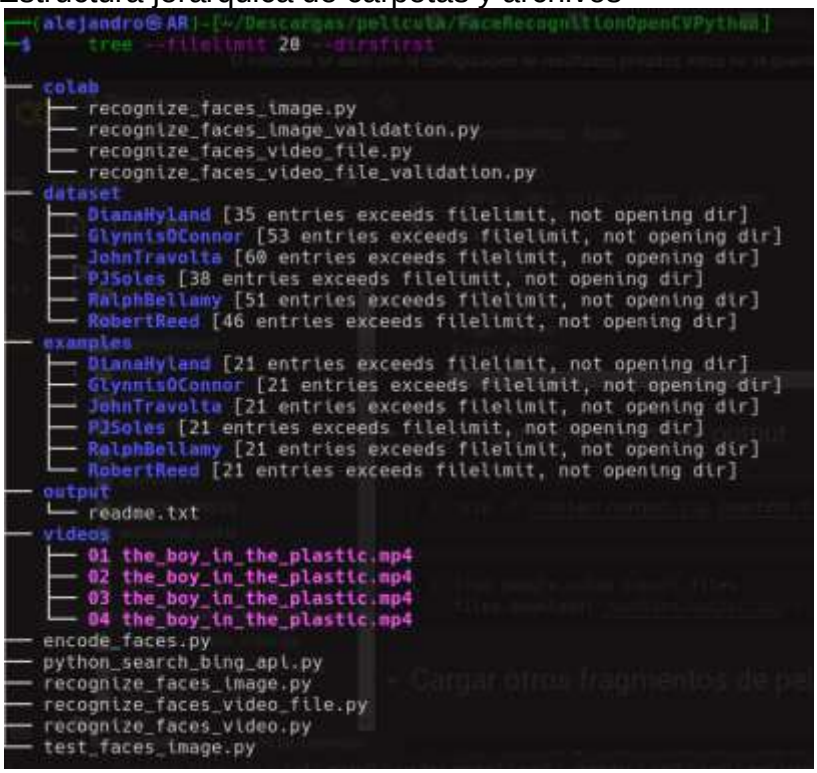
Para el desarrollo del proyecto y la ejecución del modelo se hace necesarios los siguientes insumos:

a) Un video libre de derechos de autor
b) Un modelo de red neuronal pre-entrenada
c) Un dataset de imágenes para entrenar
d) Un dataset de imágenes para validar
e) Un dataset de vídeos para procesar

Con el fin de organizar los diferentes recursos como imágenes, videos, archivos de código Python etc., se propone una estructura jerárquica de archivos y carpetas tal y como se muestra en la Figura 5-1.

De igual forma esta estructura se puede encontrar en el repositorio de GitHub relacionado en el apartado de anexos.

Figura 5-1: Estructura jerárquica de carpetas y archivos



A continuación detallaremos la estructura de carpetas y archivos:

- **Dataset/** : Carpeta que contiene las imágenes de los rostros de entrenamiento organizadas en subcarpetas nombradas de acuerdo a cada uno de los actores seleccionados
- **Examples/** : Carpeta que contiene las imágenes de los rostros de validación organizadas en subcarpetas nombradas de acuerdo a cada uno de los actores seleccionados
- **Output/ :** Carpeta donde se almacena la salida de las imágenes y vídeos procesados por el modelo.
- **Vídeos/ :** Carpeta que contiene los videoclips que se quieren procesar.
- **Colab/ :** Carpeta que contiene script de Python y permiten ejecutar las tareas de entrenamiento y validación en Google Colab
- **Python_search_bing_api.py** : Script de apoyo para la construcción del dataset
- **Encode_faces.py:** script que permite identificar el vector de 128d características de los rostros de los actores.
- **recognize_faces_image.py:** script que permite la detección e identificación de rostros sobre imágenes estáticas.
- **recognize_faces_video_file.py:** script que permite la detección e identificación de rostros sobre archivos de video.

6. Preparación de los datos

A continuación se describen los pasos que se llevaron a cabo en la recolección de datos necesarios (dataset), utilizados en las fases de entrenamiento y validación:

6.1 Video para la investigación

Como el propósito del trabajo es analizar secuencias de video, se optó por tomar como insumo una película de acceso libre que tuviera las siguientes características:

- Libre de derechos de autor para ejercicios académicos.
- Enriquecida con personajes y diferentes escenarios.
- Calidad de video aceptable.
- Contener actores conocidos para compararlos posteriormente con un set de datos de imágenes.

Para obtener una película que cumpliera con esas características se procedió a validar la página web de Internet Archive (https://archive.org/) una biblioteca digital gestionada por una organización sin ánimo de lucro dedicada a la preservación de archivos (capturas de sitios web, recursos multimedia, software). Entre su sección de video esta Feauture Films, largometrajes de dominio público.

Realizando una exploración en la página Internet Archive se validó las características que inicialmente se manifestaron, en especial que el video contara con la presencia de actores conocidos, como resultado se seleccionó **The boy in the plastic bumbble** una película con licencia de tipo Cretive Commons, estadounidense de 1976 inspirada en las vidas de David Vetter y Ted DeVita, quienes carecían de sistemas inmunológicos efectivos. Sin entrar al detalle de la sinopsis (ya que es irrelevante para investigación), dentro del reparto están actores como Jhon Travolta (reconocido mundialmente), Glynnis O'Connor, Robert Reed, Diana Hyland.

6.2 Conjunto de Imágenes para entrenamiento y validación

Con el propósito de obtener un conjunto de imágenes de cada uno de los actores presentes en la película, se optó por realizar la descarga directamente de internet, teniendo en cuenta que dichas imágenes cuenten con licencia para uso libre y que el mecanismo de descarga sea automático ya que el conjunto de datos supera 250 imágenes y descargarlas de forma manual podría resultar engorroso.

Teniendo en cuenta el reparto principal de la película se seleccionaron los siguientes actores:

a) Diana Hyland,
b) Glynnis O'Connor,
c) John Travolta,
d) PJ Soles
e) Ralph Bellamy.
f) Robert Reed,

Figura 6-1: Actores seleccionados para el análisis del video

6.2.1 API de búsqueda en la web de Bing

Microsoft proporciona un API de búsqueda en la web de Bing que tiene la capacidad de combinar miles de millones de páginas web para encontrar recursos como imágenes, vídeos y noticias.

El API de búsqueda de imágenes de Bing permite que cualquier usuario con una suscripción puede encontrar resultados de imágenes que incluyen miniaturas, URL de imágenes completas, publicación de información de sitio web, metadatos de imágenes y más. Adicionalmente se da la posibilidad de realizar clasificación sobre los resultados y filtrados que simplifican la búsqueda de resultados específicos.

Dado que esta API permite descargar cantidad de imágenes de los personajes de la película seleccionada, se procedió a crear una cuenta como desarrollador en el sitio https://www.microsoft.com/en-us/bing/apis/bing-web-search-api, una vez realizada la suscripción se procedió a obtener un api Key valido para ser utilizado en un cliente y poder así acceder a la búsqueda y descarga de imágenes.

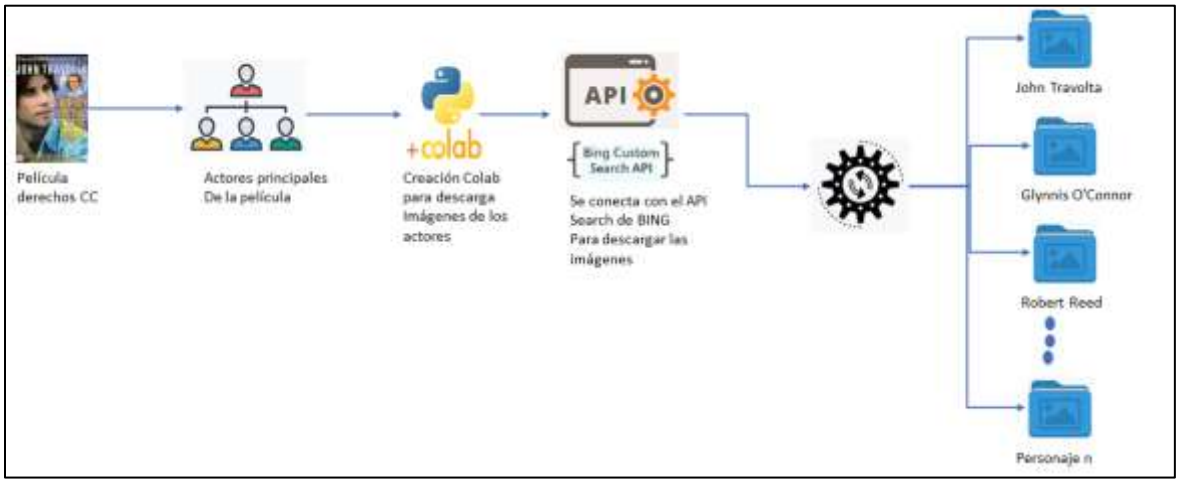
6.2.2 Desarrollo de script en Python para realizar la descarga

Usando este API de búsqueda, con el objetivo de recolectar imágenes de los actores principales de la película de forma iterativa y con propiedades muy similares en las imágenes, se procedió a realizar un script en Python capaz de consumir los servicios del API de Bing.

Este script recibe como parámetro un texto cualquiera, que será enviado a Bing para su respectiva búsqueda, posteriormente el proceso crea un folder con el nombre del personaje y finalmente descarga allí alrededor de 100 imágenes.

Como simplificación de lo descrito en los puntos anteriores, en la Figura 6-2 se encuentra el proceso que se llevó a cabo para la preparación de los datos (imágenes de los actores principales de la película “The boy in the plastic bumbble”).

Figura 6-2: Descripción del proceso que hace el script de descarga



6.2.3 Limpieza y transformación de datos.

Teniendo el conjunto de datos ya descargados y recordando que se descargó 100 imágenes por cada personaje, se procedió a revisar que dichas imágenes fueran aptas para el proceso de entrenamiento, de esta manera se inicia el proceso de limpieza y transformación de los datos.

A continuación se listan las características que se deben cumplir en cada imagen para ser catalogada como apta para el proceso de entrenamiento.

- Buena calidad de la imagen
- Se visualizará la cara del personaje sin ningún tipo de distractor u objeto como lentes, o en el caso de los hombres una barba significativa.
- Edad del personaje en un rango similar o no tan lejano de cuando filmaron la película, por ejemplo, si se descargaron imágenes de los personajes cuando eran niños, estás se descartaron.
- Que la imagen del personaje no se encuentre repetida.
- El personaje debe estar solo en la imagen y no acompañado de otra persona.

- El rostro del personaje que se está descargando debe estar completo, y no recortado.
- Que la imagen del personaje no sea una caricatura o dibujo a mano.

Después de ejecutar el script de Python, realizamos una visualización de las 100 imágenes descargadas por actor, posteriormente con los criterios descritos anteriormente, se empiezan a eliminar las imágenes que se consideran que no cumplen para el modelo propuesto. Cabe recalcar que se estableció que nuestro modelo requiere un mínimo de 30 imágenes por actor, después de realizar la depuración, la cantidad de imágenes de cada actor oscila entre las 30 y 85 imágenes.

En la Figura 6-3, se muestra a modo de ejemplo el proceso de limpieza del conjunto de datos, en este caso específicamente del actor John Travolta,

Figura 6-3: Proceso de depuración de imágenes

Una vez terminado el proceso de depuración, obtenemos como resultado el siguiente conjunto de datos de entrenamiento:

Tabla 6-1: Conjunto de datos de entrenamiento por actor

Actor	Número de imágenes para entrenamiento
Diana Hyland	35
Glynnis O'Connor	53
John Travolta	60
PJSoles	38
Ralph Bellamy	51
Robert Reed	43

Figura 6-4 : Ejemplo del conjunto de Imágenes de entrenamiento

Para el conjunto de datos de validación, se siguen los mismos pasos de descargar y depuración obteniendo como resultado 5 imágenes de validación para cada actor, tal y como se ve en la Tabla 6-2:

Tabla 6-2: Conjunto de imágenes de validación por actor

Actor	**Número de imágenes para entrenamiento**
Diana Hyland	5
Glynnis O'Connor	5
John Travolta	5
PJSoles	5
Ralph Bellamy	5
Robert Reed	5

7. Preparación de Entorno de trabajo

Para realizar la parte práctica del presente trabajo, se requiere aprovisionar un sistema operativo con unas configuraciones de Hardware mínimas así como la instalación de las librerías Python necesarias.

Adicionalmente se realizó la configuración en un ambiente local y en un ambiente en la nube, con el fin de medir el desempeño del modelo tanto en el entrenamiento como en la ejecución final.

A continuación se describen las características de Hardware en los dos ambientes.

7.1 Ambiente Local

En el ambiente local se hizo uso de un dispositivo Nvidia Jetson Nano, que es un dispositivo de hardware al estilo de Raspberry Pi pero con una GPU incorporada específicamente para la ejecución de modelos de aprendizaje profundo de manera eficiente.

La placa Nvidia Jetson Nano es una computadora con sistema operativo Linux y una GPU compatible con la aceleración CUDA[1] de envidia.

En la Tabla 7-1 se puede apreciar las características de la placa Jetson Nano.

Tabla 7-1: Características de la placa Jetson Nano

Sistema Operativo	CPU	GPU	RAM	Disco
Ubuntu Linux	CPU Quad-core ARM A57 @ 1.43 GHz.	GPU 128-core Maxwell.	Memoria 2 GB 64-bit LPDDR4 25.6 GB/s.	Memoria MicroSD de 64GB

7.2 Ambiente en la nube

Para el ambiente en la nube se eligió la aplicación web de Google Colab, ya que es un entorno preconfigurado con todos los elementos necesarios para el entrenamiento y validación de modelos de ML.

[1] Plataforma de computación paralela que se encarga de procesar los datos de entrada y salida de la GPU

En la Tabla 7-2 se puede apreciar las características del entorno de ejecución en Google Colab:

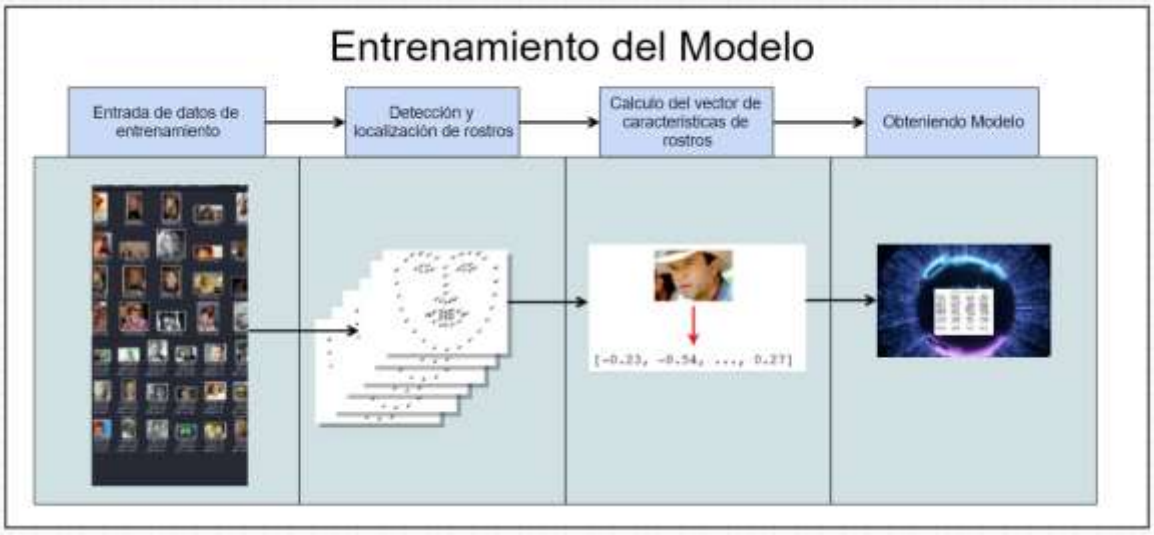
Tabla 7-2: Características entorno en la nube Google Colab

Sistema Operativo	**CPU**	**GPU**	**RAM**	**Disco**
Ubuntu Linux	Intel Xeon CPU 2.20 GHz	Nvidia Tesla T4	13 GB	Capacidad de almacenamiento de acuerdo a configuración con Google Drive

8. Entrenamiento del modelo

Para el entrenamiento del modelo en el ambiente local y el ambiente en la nube, se hace uso de los scripts de apoyo creados en lenguaje de programación Python.

Figura 8-1: Flujo del entrenamiento del modelo de reconocimiento e identificación facial



En esta sección se describe el flujo empleado para el entrenamiento del modelo de acuerdo al diagrama de la Figura 8-1.

1. **Entrada de datos de entrenamiento**: En este paso, se da como Insumo la carpeta dataset/ que contiene todas las imágenes de rostros de los actores seleccionados.
2. **Detección y localización de rostros:** En este paso se hace uso de las librerías OpenCV, Dlib y Face_Recognition, como insumo se proporciona el listado total de imágenes y como resultado se recibe un vector con las posiciones de los rostros en cada imagen.
3. **Cálculo del vector de características de rostros:** En este paso se hace uso del modelo Resnet-34 para crear el vector de 128-d características de un rostro.

   Recordemos que la red fue entrenada con más de 3 millones de rostros y hace parte de las herramientas que provee la librería Dlib y su soporte para el aprendizaje métrico profundo.

   Tomando como base el entrenamiento previo de la red, la utilizamos para generar los vectores de características para cada una de las imágenes de

los actores, para ello nos apoyamos en el script denominado encode_faces.py

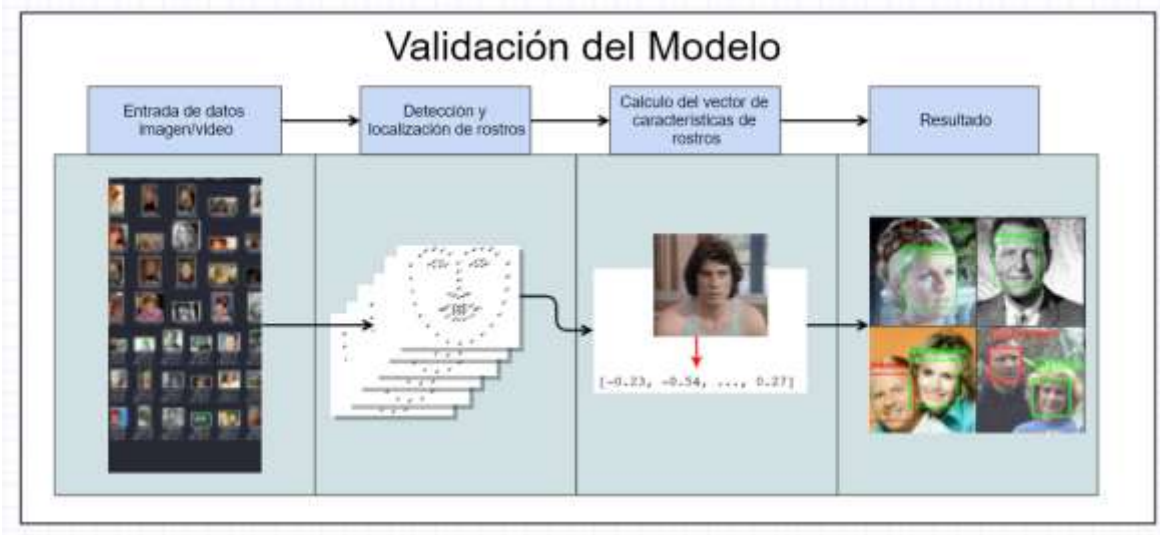
Figura 8-2: El Framework Face_Recognition genera un vector de características numéricas con un valor real de 128-d por rostro

4. **Modelo**: El modelo resultante del proceso anterior, es el archivo serializado encodings.pickle que contiene los vectores de características faciales de cada uno de los actores.

9. Evaluación del modelo

Para la fase de evaluación o validación, se procede con la ejecución del modelo tanto en el ambiente local como en el ambiente en la nube, en ambos casos se hace uso de los script de apoyo creados en lenguaje de programación Python, proporcionando los dataset de prueba y verificando que el modelo clasifique correctamente los datos ingresados.

Figura 9-1: Flujo de la validación del modelo de Reconocimiento e Identificación Facial



A continuación se describe el flujo empleado para la validación del modelo de acuerdo al diagrama de la Figura 9-1.

1. **Entrada de datos** : En este paso, se proporciona como parámetro la ruta de la carpeta examples/ que contiene 5 imágenes por cada actor. Cabe recordar que estas imágenes son diferentes a las imágenes utilizadas para la fase de entrenamiento; a continuación se muestra en la Figura 9-2 las imágenes utilizadas:

Figura 9-2: Imágenes utilizadas para la validación del modelo (5 por cada actor)

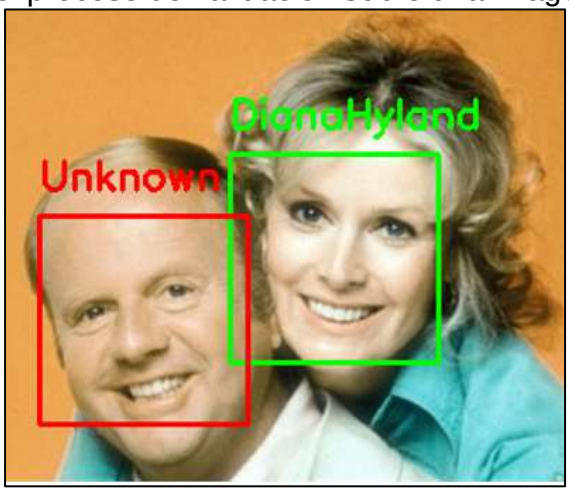
2. **Detección y localización de rostros**: En este paso se hace uso de las librerías OpenCV, Dlib y Face_Recognition, como parámetro se proporciona el listado total de imágenes y como resultado se recibe un vector con las posiciones de los rostros en cada imagen.

3. **Cálculo del vector de características de rostros**: Utilizamos el modelo ResNet-34 para generar los vectores de características de cada una de las imágenes de los actores, para ello nos apoyamos en el script denominado encode_faces.py

4. **Resultado**: En este punto el objetivo es encontrar una persona de nuestra base de datos de personas conocidas (archivo encodings.pickle) que tenga la menor distancia a la imagen de entrada, para ello se hace uso del algoritmo de clasificación KNN, obteniendo como resultado el nombre del actor si fue encontrado o por el contrario la etiqueta “desconocido” para indicar que no está en nuestra base de datos.

   Una vez tengamos la etiqueta de identificación correspondiente y la posición de los rostros, se procede a dibujar un rectángulo de color verde o rojo si hubo coincidencia o no, sobre los rostros detectados en la imagen, el resultado final se puede apreciar en la Figura 9-3.

Figura 9-3: Resultado del proceso de validación sobre una imagen con dos rostros



En la Figura 9-4 se muestran los resultados para todas las imágenes de validación por cada actor:

Figura 9-4: Resultados de todas las imágenes de validación

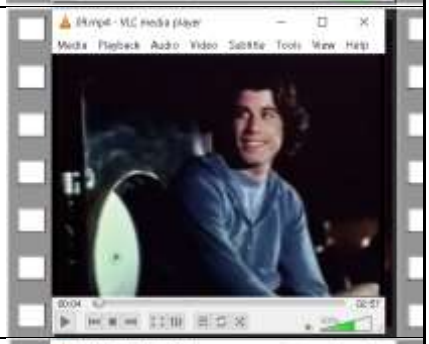
9.1.1 Experimentos sobre Vídeos

Como el objetivo general de este trabajo es Reconocimiento e identificación de rostros sobre secuencia de vídeos, se procede a extraer 13 video clips de la película **The boy in the plastic bumbble**, con el fin de poder realizar un análisis sobre una muestra más pequeña que permita evaluar el comportamiento del modelo tanto en precisión como en rendimiento, a continuación se presenta en la Tabla 9-1 la relación de cada video clip:

Tabla 9-1: Descripción de cada video Clip

Nombre	Duración (hh-mm-ss)	Observación	Video
01.mp4	00:02:22	En este fragmento, se presenta a los actores de perfil y con accesorios como gorros y sombreros	

02.mp4	00:00:33	En este fragmento se presenta una cantidad de rostros desconocidos	
03.mp4	00:01:11	En este fragmento se incorpora el rostro de un bebe de perfil y variación en las condiciones de iluminación	
04.mp4	00:01:20	En este fragmento se incorpora el rostro de un niño de 5 años aproximadamente, rostros desenfocados por la cámara y de mala calidad	
05.mp4	00:01:55	En este fragmento se presenta al personaje principal usando un gorro	
06.mp4	00:01:07	En este fragmento se presenta una escena en un recinto oscuro, en donde los rostros de las personas presentan sombras	

07.mp4	00:02:01	En este Fragmento se presenta un grupo de personas que no fueron parte del grupo de entrenamiento, así como una escena del actor principal usando gafas y nariz falsa	
08.mp4	00:00:39	En este Fragmento se presenta a dos actores dialogando, bajo unas condiciones totalmente ideales	
09.mp4	00:02:57	En este fragmento hay poca iluminación y escenas con personas desconocidas	
10.mp4	00:00:50	En este fragmento se presenta al personaje principal dentro de un traje que solo permite ver el rostro	
11.mp4	00:01:30	En este fragmento también se muestra al personaje utilizando un traje y casco	

12.mp4	00:02:04	en este fragmento hay poca iluminación	
13.mp4	00:01:08	En este fragmento hay poca iluminación y el personaje además está usando el traje	

En esta sección se describe el flujo empleado para la validación del modelo con entradas de video.

1. **Entrada de datos** : En este paso, se da como Insumo la carpeta vídeos/ que contiene los 13 video clips mencionados en la tabla anterior.

2. **Detección y localización de rostros**: En este paso se hace uso de las librerías OpenCV, Dlib y Face_Recognition, como insumo se proporciona el listado total de vídeos y como resultado se recibe un vector con las posiciones de los rostros tomado de los frames de cada video clip.

3. **Cálculo del vector de características de rostros:** Tomando como base el entrenamiento previo de la red, la utilizamos para generar los vectores de características para cada una de las imágenes de los actores, para ello nos apoyamos en el script denominado encode_faces.py

4. **Resultado**: En este punto el objetivo es encontrar la etiqueta del actor con mayor coincidencia (KNN), una vez tengamos la etiqueta de identificación correspondiente y la posición de los rostros, se procede a dibujar un rectángulo de color verde (si hubo coincidencias) o rojo(si no hubo coincidencias), sobre los rostros detectados en la secuencia de video, el resultado final es generado en la carpeta output/, donde queda guardado el video con la identificación procesada.

10. Resultados

10.1 Conjunto de imágenes de validación

A continuación se presentan los resultados obtenidos en la etapa de validación del modelo con el conjunto de imágenes de los actores en cada uno de los ambientes propuestos

En este caso, se contó con un total de 41 imágenes, se identifica cuantas de ellas dieron resultados correctos (identifica bien al actor con su nombre),incorrectos (reconoce un rostro pero lo clasifica mal) y desconocido(identifica un rostro y lo clasifica correctamente como desconocido), con estos valores procedemos a realizar el cálculo del accurancy.

Tabla 10-1: Resultados de la precisión del modelo, obtenidos en el ambiente local para la etapa de validación con un dataset de imágenes

Jetson Nano	Correctas	Incorrectas	Desconocido	Total	Accurancy
John Travolta,	6	0	0	6	100.00%
Glynnis O'Connor,	5	0	0	5	100.00%
Robert Reed,	5	1	0	6	83.33%
Diana Hyland,	8	0	0	8	100.00%
Ralph Bellamy.	5	0	0	5	100.00%
Desconocidos	6	5	0	11	54.55%

Tabla 10-2: Resultados de la precisión del modelo, obtenidos en el ambiente de nube para la etapa de validación con un dataset de imágenes

Colab	Correctas	Incorrectas	Desconocido	Total	Accurancy
John Travolta,	6	0	0	6	100.00%
Glynnis O'Connor,	5	0	0	5	100.00%
Robert Reed,	5	1	0	6	83.33%
Diana Hyland,	8	0	0	8	100.00%
Ralph Bellamy.	5	0	0	5	100.00%
Desconocidos	6	5	0	11	54.55%

En este caso se identifica que el resultado es el mismo en los dos ambientes por lo que se puede concluir que el modelo final y la precisión del mismo no es afectado por las características de los ambientes utilizados.

Sin embargo los tiempos de procesamiento en las etapas de entrenamiento y validación si presentan cambios significativos como se ve en la Tabla 10-3:

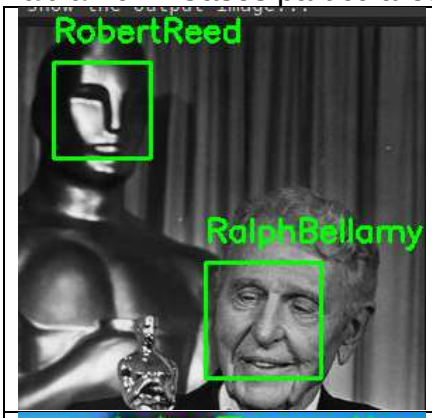
Tabla 10-3: Resultados de los tiempos de ejecución en los ambientes para la etapa de validación con un dataset de imágenes

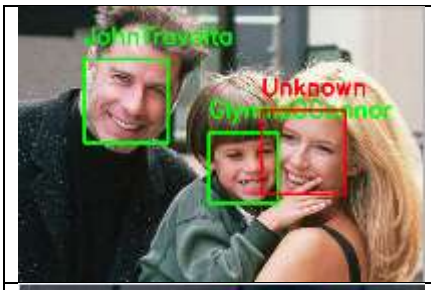
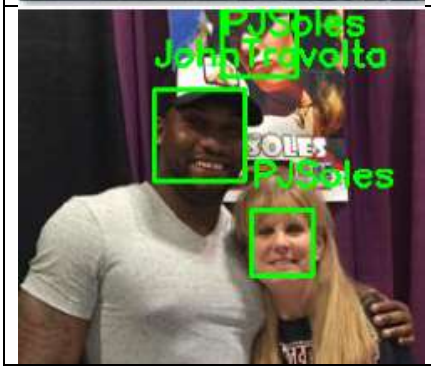
	Etapa	Tiempo de ejecución (minutos)
Jetson nano	Entrenamiento	11.1
Jetson nano	Validación	3
Colab	Entrenamiento	3.2
Colab	Validación	0.9

Lo anterior deja ver que a mayor características de hardware es mucho mejor el desempeño del modelo.

Adicionalmente se presentaron algunos casos particulares que vale la pena mencionar:

Tabla 10-4: Casos particulares observados en la etapa de validación de imágenes

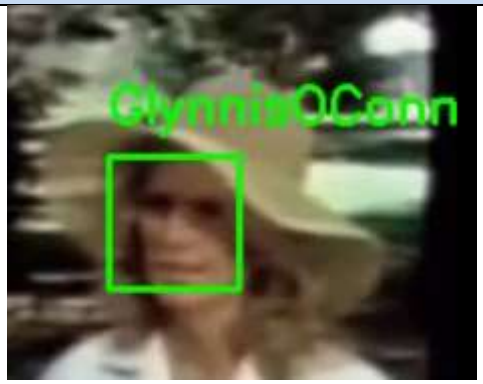
	En este caso se puede apreciar que el modelo reconoció un rostro humano sobre una estatuilla y a su vez lo clasifico como Rober Reed, indicando claramente un error en la clasificación ya que el modelo encuentra que el vector de características faciales de dicho actor es muy cercano al vector de características de la estatuilla.
	En este caso la imagen presentada no corresponde a John Travolta pero se cree que la ofuscación y poca calidad de la imagen contribuyeron para que el modelo clasificara mal el rostro.

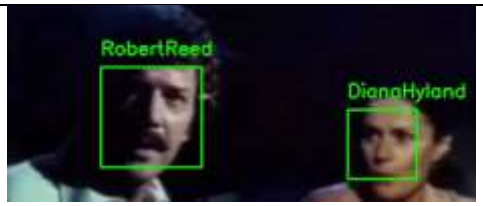
	En este caso se observa que el modelo clasifico el rostro del niño como un rostro conocido.
	En este caso se observa que la red identifico dos veces al mismo actor en dos rostros diferentes, mostrando un error en la clasificación e identificación.
	Igual que en las imágenes anteriores el modelo encuentra que el rostro identificado corresponde a una persona conocida como John Travolta.

10.2 Conjunto de video clips de validación

En el caso de la validación de vídeos no es tan sencillo realizar un conteo de los frames analizados por el modelo, por lo que se recurre a validar si el modelo clasifica correctamente un frame en particular, partiendo de un caso de prueba con un resultado esperado, tal como se ve en la Tabla 10-5.

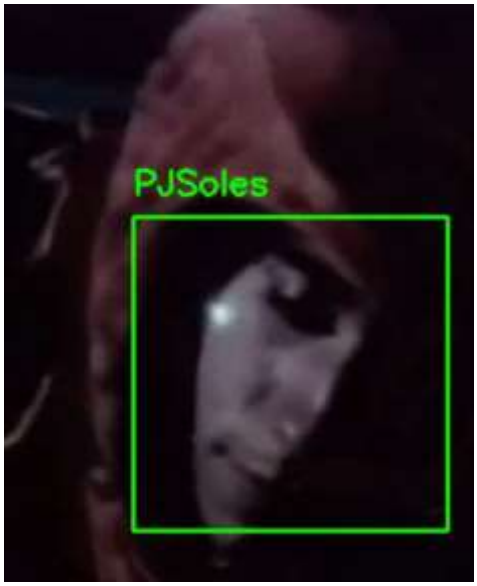
Tabla 10-5: Resultados del análisis de video clips, a partir de casos de prueba

Nombre	Resultado esperado	Frame del video	Observación	Resultado
01.mp4	Reconocimiento de rostros con accesorios			✓
02.mp4	Reconocimiento de rostros desconocidos			✓
03.mp4	Reconocimiento del rostro de un Bebe y clasificación como desconocido		En este frame se esperaba que el modelo pudiera identificar un rostro y lo clasificara como desconocido, sin embargo el modelo falló en esta tarea. Se asume que puede ser por factores de calidad en la imagen y sombras presentes en la parte de los ojos	✗

04.mp4	Reconocimiento del rostro de un Niño y clasificación como desconocido			✓
05.mp4	Reconocimiento de rostros con accesorios			✓
06.mp4	Reconocimiento con baja iluminación			✓
07.mp4	Reconocimiento con gafas y nariz falsa			✓

08.mp4	Reconocimiento de actores conocidos			✓
09.mp4	Reconocimiento con baja iluminación			✓
10.mp4	Reconocimiento de rostro sobre un traje y casco		En este frame el modelo fue capaz de identificar un rostro humano, sin embargo lo clasificó de forma incorrecta debido a la baja calidad de la imagen	✗

11.mp4	Reconocimiento de rostro sobre un traja y casco		En este frame el modelo fue capaz de identificar un rostro humano, sin embargo lo clasificó de forma incorrecta debido a la baja calidad de la imagen	✗
12.mp4	Reconocimiento con baja iluminación			✓

13.mp4	Reconocimiento con baja iluminación y casco		En este frame el modelo fue capaz de identificar un rostro humano, sin embargo lo clasificó de forma incorrecta debido a la calidad de la imagen y poca iluminación	✗

Adicionalmente en la Tabla 10-6, se relaciona el respectivo link de Youtube para cada uno de los video clips resultantes del proceso de validación del modelo.

Tabla 10-6: Relación de video clips con link de YouTube

Video	Link YouTube
01.mp4	https://youtu.be/YpVI0X1upCs
02.mp4	https://youtu.be/6t6IBwH-kcs
03.mp4	https://youtu.be/Gjot8QODjio
04.mp4	https://youtu.be/AKzn2DoJiNk
05.mp4	https://youtu.be/1P1J-8aDZOQ
06.mp4	https://youtu.be/9KB7p3F-Uaw
07.mp4	https://youtu.be/zr4QprS0Uhg
08.mp4	https://youtu.be/RK4niSF-iXI
09.mp4	https://youtu.be/-y8spdHlyDA
10.mp4	https://youtu.be/nmAqIjwbfD0
11.mp4	https://youtu.be/XtA1ZPCY5YE
12.mp4	https://youtu.be/_o0mg0M-oMA
13.mp4	https://youtu.be/v8iG6aV2Zgg

A continuación se presenta los resultados obtenidos en las 13 pruebas sobre video en términos de precisión del modelo (Accurancy):

Tabla 10-7: Resultados de la precisión del modelo obtenidos en los dos ambientes, para la etapa de validación con un dataset de vídeos

	Pruebas correctas	**Pruebas Incorrectas**	**Total**	**Accurancy**
Análisis sobre video clips	9	4	13	69.23%

Adicionalmente se presentaron algunos casos particulares en el análisis de las pruebas que vale la pena tener en cuenta:

Figura 10-1: Casos particulares observados en la etapa de validación de video

En la Figura 10-1, se aprecia como en escenarios donde se presentan factores de oclusión, poca iluminación y baja calidad de la imagen, el modelo no fue capaz de reconocer rostros humanos en algunas imágenes y en otras realizó mal el proceso de identificación.

Finalmente se presentan los resultados del tiempo que le tomo al modelo procesar el total de los video clips en los dos ambientes propuestos:

Tabla 10-8: Resultados de los tiempos de ejecución en los ambientes, para la etapa de validación con un dataset de vídeos

	Etapa validación	**Tiempo (hh:mm:ss)**
Jetson nano	13 video clips	03:40:10
Colab	13 video clips	01:50:50

Se aprecia que el tiempo de ejecución en el entorno de nube es muy bajo en relación con la ejecución sobre la placa Jetson Nano, esto se debe a que la placa cuenta con tan solo 2 GB de RAM lo que impacta drásticamente en los frames por segundo que puede analizar.

11. Conclusiones y recomendaciones

11.1 Conclusiones

En este trabajo se aplicó el modelo de reconocimiento facial ResNet-34 sobre unos fragmentos extraídos de una película, en donde se puede utilizar la información de los rostros encontrados en un frame específico para calcular el vector de 128-d características faciales y mediante el uso de KNN buscar en el modelo entrenado con características faciales de 5 actores de las películas para clasificar la imagen.

- Se aprecia que el modelo entrenado no varía en relación a las características de hardware de los ambientes propuestos.

- Se corrobora que efectivamente la relación de características de hardware como Memoria RAM, GPU, Soporte CUDA hacen la diferencia en los tiempos de ejecución del modelo.

- Se aprecia que aspectos como iluminación, oclusión, ángulo y calidad de la imagen influyen negativamente en el comportamiento del modelo, por lo cual sigue siendo un problema a ser resuelto en los modelos de reconocimiento facial.

- Se logra evaluar la calidad de los videos e imágenes resultantes del proceso de validación y ejecución, evidenciando que estos no presentan modificaciones adicionales a los rectángulos de identificación de rostro.

- Se logra determinar que el modelo generado, se puede reutilizar para el entrenamiento de otros modelos ya que aplica el concepto de transfer learning.

Finalmente, se cumplió exitosamente el objetivo general, toda vez que se implementó un sistema de reconocimiento e identificación de rostros sobre secuencias de video mediante el uso de un modelo de Redes Neuronales Convolucionales y Transfer Learning.

11.2 Recomendaciones

- Se recomienda realizar la prueba con un video que tenga mejor calidad ya que el video objeto del estudio es de la década de los años 70.

- Se recomienda realizar la prueba localmente con dispositivos que cuenten con una GPU con soporte CUDA y tengan una capacidad de memoria RAM superior a 8 GB.

- Con el fin de identificar otro tipo de rostros de personas como niños y adultos mayores, se recomienda realizar el entrenamiento del modelo con este perfil de rostros.

- Se recomienda realizar pruebas con video clips diferentes a películas, por ejemplo videos grabados por sistemas como CCTV, para validar el modelo en un entorno real.

Finalmente si se considera pertinente, se puede complementar el uso del modelo con el uso de una aplicación web o desktop que permita la carga de archivos de imagen y vídeos de tal manera que se pueda mostrar los resultados de la ejecución del modelo de una forma amigable para el usuario final.

Anexo: Tabla de productos generados

A continuación se relacionan los hipervínculos de los diferentes productos generados en este trabajo.

Nro.	Producto	Hipervínculo
1	Notebook Google colab para el entrenamiento y validación del modelo	https://colab.research.google.com/drive/1GjtMw4AMwIdjZgQfCYTBfm8xLaod6ESP?usp=sharing
2	Repositorio GitHub, FaceRecognitionOpenCVPython	https://github.com/senseiRoa/FaceRecognitionOpenCVPython
3	Lista de reproducción en YouTube, con los videos analizados por el modelo	https://www.youtube.com/watch?v=YpVI0X1upCs&list=PL8c9nGV_opEav6K88r0cpgMwwn8z0PjUZ&index=1

Bibliografía

[1] M. Liu and Z. Liu, “Deep Reinforcement Learning Visual-Text Attention for Multimodal Video Classification,” in *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications - {MULEA} ’19*, pp. 13–21.

[2] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10. pp. 1345–1359, Oct-2010.

[3] X. Ran, H. Chen, Z. Liu, and J. Chen, “Delivering Deep Learning to Mobile Devices via Offloading,” in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network - {VR}/{AR} Network ’17*, pp. 42–47.

[4] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” vol. 4, no. 11, p. e00938, 2018.

[5] G. Szirtes, D. Szolgay, Á. Utasi, D. Takács, I. Petrás, and G. Fodor, “Facing reality: an industrial view on large scale use of facial expression analysis,” in *Proceedings of the 2013 on Emotion recognition in the wild challenge and workshop - {EmotiW} ’13*, pp. 1–8.

[6] G. Levi and T. Hassner, “Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns,” in *Proceedings of the 2015 {ACM} on International Conference on Multimodal Interaction - {ICMI} ’15*, pp. 503–510.

[7] R. Ewerth, M. Mühling, and B. Freisleben, “Robust Video Content Analysis via Transductive Learning,” vol. 3, no. 3, pp. 1–26.

[8] M. Parchami, S. Bashbaghi, and E. Granger, “{CNNs} with cross-correlation matching for face recognition in video surveillance using a single training sample per person,” in *2017 14th {IEEE} International Conference on Advanced Video and Signal Based Surveillance ({AVSS})*, pp. 1–6.

[9] H. Khan, A. Atwater, and U. Hengartner, “Itus: an implicit authentication framework for android,” in *Proceedings of the 20th annual international conference on Mobile computing and networking - {MobiCom} ’14*, pp. 507–518.

[10] L. N. Huynh, Y. Lee, and R. K. Balan, “DeepMon: Mobile GPU-based Deep Learning Framework for Continuous Vision Applications,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 82–95.

[11] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, “Big data analytics: Computational intelligence techniques and application areas,” *Technol. Forecast. Soc. Change*, vol. 153, p. 119253, 2020.

[12] U. Schmidt-Erfurth, A. Sadeghipour, B. S. Gerendas, S. M. Waldstein, and H. Bogunović, “Artificial intelligence in retina,” vol. 67, pp. 1–29.

[13] M. Mittal *et al.*, “An efficient edge detection approach to provide better edge connectivity for image analysis,” *IEEE Access*, vol. 7, pp. 33240–33255, 2019.

[14] D. Sirohi, N. Kumar, and P. S. Rana, “Convolutional neural networks for 5G-enabled Intelligent Transportation System : A systematic review,” vol. 153, pp. 459–498.

[15] A. Kumar, A. Kaur, and M. Kumar, “Face detection techniques: a review,” *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 927–948, 2019.

[16] K. S. Gautam and S. K. Thangavel, “Video analytics-based intelligent surveillance system for smart buildings,” *Soft Comput.*, vol. 23, no. 8, pp. 2813–2837, 2019.

[17] J. Yu, K. Sun, F. Gao, and S. Zhu, “Face biometric quality assessment via light CNN,” vol. 107, pp. 25–32.

[18] L. T. Nguyen-Meidine, E. Granger, M. Kiran, and L.-A. Blais-Morin, “A comparison of {CNN}-based face and head detectors for real-time video surveillance applications,” in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications ({IPTA})*, pp. 1–7.

[19] B. Chacua *et al.*, “People Identification through Facial Recognition using Deep Learning,” in *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2019, pp.

1–6.
[20] J. Park, J. Chen, Y. K. Cho, D. Y. Kang, and B. J. Son, “CNN-based person detection using infrared images for night-time intrusion warning systems,” *Sensors (Switzerland)*, vol. 20, no. 1, 2020.
[21] A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa, “The Do’s and Don’ts for CNN-Based Face Verification,” in *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 2545–2554.
[22] J. Galbally, “A new Foe in biometrics: A narrative review of side-channel attacks,” vol. 96, p. 101902.
[23] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, “Latent Backdoor Attacks on Deep Neural Networks,” in *Proceedings of the 2019 {ACM} {SIGSAC} Conference on Computer and Communications Security*, pp. 2041–2055.
[24] Y. Akbulut, A. Sengur, U. Budak, and S. Ekici, “Deep learning based face liveness detection in vídeos,” in *2017 International Artificial Intelligence and Data Processing Symposium ({IDAP})*, pp. 1–4.
[25] J. Zhang, W. Li, P. Ogunbona, and D. Xu, “Recent Advances in Transfer Learning for Cross-Dataset Visual Recognition: A Problem-Oriented Perspective,” vol. 52, no. 1, pp. 1–38.
[26] C. X. Lu *et al.*, “Autonomous Learning for Face Recognition in the Wild via Ambient Wireless Cues,” in *The World Wide Web Conference on - {WWW} ’19*, pp. 1175–1186.
[27] J. C. Hung, K.-C. Lin, and N.-X. Lai, “Recognizing learning emotion based on convolutional neural networks and transfer learning,” vol. 84, p. 105724.
[28] S. Zhang, X. Pan, Y. Cui, X. Zhao, and L. Liu, “Learning Affective Video Features for Facial Expression Recognition via Hybrid Deep Learning,” *IEEE Access*, vol. 7, pp. 32297–32304, 2019.
[29] C. Herrmann, T. Müller, D. Willersinn, and J. Beyerer, “Real-time person detection in low-resolution thermal infrared imagery with MSER and CNNs,” p. 99870I.
[30] F. An and Z. Liu, “Facial expression recognition algorithm based on parameter adaptive initialization of CNN and LSTM,” vol. 36, no. 3, pp. 483–498.
[31] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “Joint Face Representation Adaptation and Clustering in Vídeos,” in *Computer Vision – {ECCV} 2016*, vol. 9907, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, pp. 236–251.
[32] E. G. Ortiz, A. Wright, and M. Shah, “Face recognition in movie trailers via mean sequence sparse representation-based classification,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3531–3538.
[33] “Privacy Protection for Life-log Video.” [Online]. Available: https://www.researchgate.net/publication/4249807_Privacy_Protection_for_Life-log_Video. [Accessed: 13-Jun-2021].
[34] SUPERINTENDENDENCIA DE INDUSTRIA Y COMERCIO, “Proteccion de datos personales en sistemas de videovigilancia,” 2016.
[35] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, “Recurrent Neural Networks for Emotion Recognition in Video,” in *Proceedings of the 2015 {ACM} on International Conference on Multimodal Interaction - {ICMI} ’15*, pp. 467–474.
[36] E. Flouty, O. Zisimopoulos, and D. Stoyanov, “FaceOff: Anonymizing Vídeos in the Operating Rooms,” in *{OR} 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, vol. 11041, D. Stoyanov, Z. Taylor, D. Sarikaya, J. McLeod, M. A. González Ballester, N. C. F. Codella, A. Martel, L. Maier-Hein, A. Malpani, M. A. Zenati, S. De Ribaupierre, L. Xiongbiao, T. Collins, T. Reichl, K. Drechsler, M. Erdt, M. G. Linguraru, C. Oyarzun Laura, R. Shekhar, S. Wesarg, M. E. Celebi, K. Dana, and A. Halpern, Eds. Springer International Publishing, pp. 30–38.
[37] A. Turing, “Maquinaria computacional e Inteligencia Alan Turing, 1950,” 1950.
[38] G. R. Yang and X. J. Wang, “Artificial Neural Networks for Neuroscientists: A Primer,” *Neuron*, vol. 107, no. 6, pp. 1048–1070, Sep. 2020.
[39] J. Singh and R. Banerjee, “A Study on Single and Multi-layer Perceptron Neural Network,”

in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019.
[40] I. G. and Y. B. and A. Courville, *Deep Learning*. 2016.
[41] E. Stevens, L. Antiga, and T. Viehmann, “Deep Learning with PyTorch.”
[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.”
[43] M. KAYA and H. Ş. BİLGE, “Deep Metric Learning: A Survey,” *Symmetry 2019, Vol. 11, Page 1066*, vol. 11, no. 9, p. 1066, Aug. 2019.
[44] B. R. Vasconcellos, M. Rudek, and M. de Souza, “A Machine Learning Method for Vehicle Classification by Inductive Waveform Analysis,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 13928–13932, Jan. 2020.