UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Electrónica y Redes de Comunicación

Trabajo de Grado Previo a la Obtención del Título de

Ingeniero en Electrónica y Redes de Comunicación

TEMA:

Diseño de un sistema prototipo de reconocimiento facial para la identificación de personas en la Facultad de Ingeniería en Ciencias Aplicadas (FICA) de la Universidad Técnica del Norte utilizando técnicas de Inteligencia Artificial.

Autor: Chacua Criollo Bolívar Eduardo.

Director: PhD. Iván Danilo García Santillán.

Ibarra - Ecuador

2019

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO	
Cédula de identidad	100384576-3
Apellidos y nombres	Chacua Criollo Bolívar Eduardo
Dirección	Ibarra, Cdla. La Victoria. Hugo Guzmán Lara & Eduardo Garzón Fonseca
E-mail	bechacua@utn.edu.ec
Teléfono móvil	0993666988
DATOS DE LA OBRA	
Título	DISEÑO DE UN SISTEMA PROTOTIPO DE RECONOCIMIENTO FACIAL PARA LA IDENTIFICACIÓN DE PERSONAS EN LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS (FICA) DE LA UNIVERSIDAD TÉCNICA DEL NORTE UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL.
Autor	Bolívar Eduardo Chacua Criollo
Fecha	17/9/2019
Programa	Pregrado
Titulo	Ingeniero en Electrónica y Redes de Comunicación
Director	PhD. Iván Danilo García Santillán

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 17 días del mes de Septiembre de 2019

EL AUTOR:

Bolívar Eduardo Chacua Criollo

CI:100384576-3

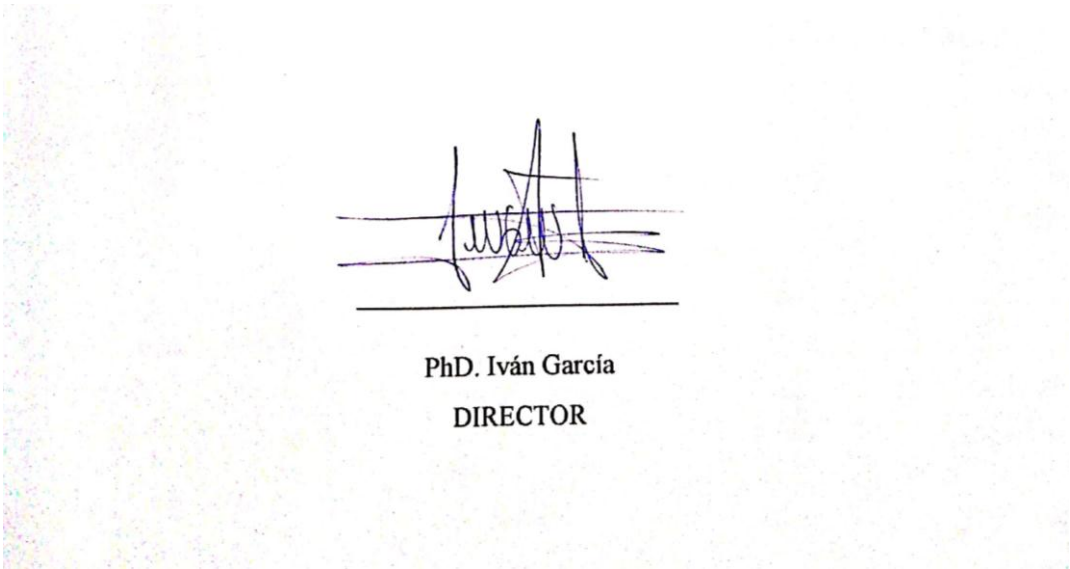
UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

PhD. IVÁN GARCÍA SANTILLÁN, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de titulación "DISEÑO DE UN SISTEMA PROTOTIPO DE RECONOCIMIENTO FACIAL PARA LA IDENTIFICACIÓN DE PERSONAS EN LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS (FICA) DE LA UNIVERSIDAD TÉCNICA DEL NORTE UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL", fue realizado en su totalidad por el Sr. Bolívar Eduardo Chacua Criollo, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor a la verdad.



PhD. Iván García
DIRECTOR

DEDICATORIA

Este trabajo se lo dedico a las personas más importantes en mi vida, esto va por ustedes:

A mis padres Segundo y Patricia, por forjar a la persona que soy hoy en día, por ser mis mejores amigos y apoyarme incondicionalmente en todo momento, por ser un ejemplo de superación para mí y mis hermanos, además de demostrarme que en la vida se puede lograr cualquier objetivo con mucho esfuerzo y valentía.

A mis hermanos Patricio y David, por enseñarme que cada día es un regalo y que no importa cuán difícil se torne nuestra vida, es preciso seguir adelante a pesar de las circunstancias. David, admiro infinitamente la tenacidad con la que luchas cada día, la capacidad de inspirar y reconfortar mi alma, la de nuestra familia, y la de muchas personas. Eres un ejemplo a seguir!!!. Te quiero mucho querido hermano.

Bolívar Eduardo Chacua

AGRADECIMIENTOS

Agradezco infinitamente:

Al creador del universo, por otorgarme el placer de vivir y ser feliz en este plano astral.

A mis padres, por darme la vida, por brindarme la oportunidad de formarme profesionalmente y hacer posible muchos de mis anhelos.

A mis amigos más cercanos, por acompañarme y alentarme fervientemente en esta breve travesía por la academia universitaria de la UTN.

Al Ing. Iván García PhD, por ser mi guía en esta ardua investigación y contribuirme con sus altos conocimientos en la elaboración de este trabajo de titulación, además de encaminarme en el apasionante campo de la inteligencia artificial.

A mis asesores Ing. Paul Rosero MsC y Ing. Luis Suárez MsC, por sus sugerencias para el desarrollo de este trabajo.

Y, finalmente, gracias a ti estimado lector, espero que disfrutes leyendo esta humilde tesis siquiera una pequeña parte de lo que yo he disfrutado escribiéndola.

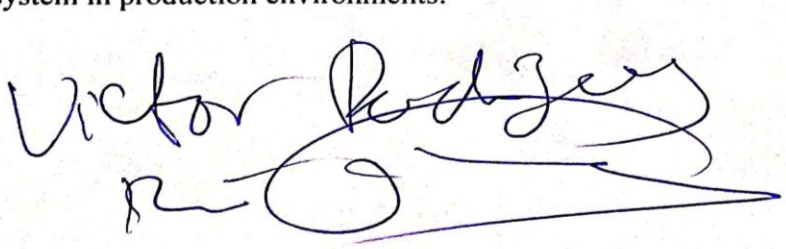
Bolívar Eduardo Chacua

RESUMEN

En la actualidad, los sistemas de monitorización de circuito cerrado de televisión (CCTV), control de acceso, y muchas otras aplicaciones relacionadas con la seguridad, incorporan técnicas de reconocimiento facial. Esta herramienta disruptiva, se diferencia de otras técnicas biométricas, ya que los rostros pueden ser reconocidos a distancia. Así, estas aplicaciones pueden incorporase en diferentes instituciones con la finalidad de restringir el acceso a personas no autorizadas/desconocidas, evitando daños y pérdidas al bien público y privado. El objetivo del presente trabajo fue identificar personas en entornos controlados y no controlados dentro del edificio universitario de la FICA, que ha sufrido problemas de inseguridad en varias ocasiones. Al ser un tema abierto de estudio y de ardua investigación en el área de la Inteligencia Artificial (IA), en este documento se presenta el diseño completo de un sistema de reconocimiento facial combinando una arquitectura de Red Neuronal Convolucional (CNN) y el poder de clasificación del algoritmo de Máquinas de Vector Soporte (SVM), implementadas bajo tecnología de procesamiento paralelo (CUDA) a través de una unidad de procesamiento grafico (GPU). Todo el procedimiento de desarrollo e implementación se describe a detalle, donde se inicia con el entrenamiento de la CNN usando el conjunto de datos VGGFace2, para el aprendizaje y generalización de incrustaciones faciales profundamente discriminativas de tamaño de 512 bytes por rostro mediante la supervisión conjunta de las señales de pérdida de softmax y la pérdida central. Consecuentemente se emplea SVM como clasificador en varios experimentos con diferentes cantidades de clases, para finalmente mostrar la eficiencia del enfoque en los entornos mencionados en tiempo real, empleando una muestra de individuos para el entrenamiento del sistema, logrando resultados bastante aceptables. Finalmente, el sistema propuesto establece un punto de partida para el desarrollo de un sistema más robusto en entornos de producción.

ABSTRACT

Currently, Closed Circuit Television (CCTV) monitoring systems, access control, and other security-related applications incorporate facial recognition techniques. This disruptive tool differs from other biometric techniques, as faces can be recognized remotely. Thus, these applications can be incorporated in different institutions with the purpose of restricting access to unauthorized/unknown persons, avoiding damages and losses to the public and private property. The objective of this work was to identify people in controlled and uncontrolled environments within the FICA faculty, which has suffered insecurity problems on several occasions. Being an open topic of study and arduous research in the area of Artificial Intelligence (AI), this research presents the complete design of a facial recognition system combining a Convolutional Neural Network architecture (CNN) and the reliability of Support Vector Machines algorithm (SVM), implemented under parallel processing technology (CUDA) through a graphics processing unit (GPU). The development and implementation procedure is described in detail, beginning with CNN training using the VGGFace2 dataset, for the learning and generalization of discriminatory facial embeddings of 512 bytes per face size by joint supervision of softmax loss and center loss signals. Consequently, SVM is used as a classifier in several experiments with different amounts of classes, to finally show the efficiency of the approach in the chosen environments in real time, using a sample of individuals for the training of the system, achieving quite acceptable results. Finally, the proposed system establishes a starting point for the development of a more robust system in production environments.



ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

1. CAPÍTULO I. Antecedentes

El capítulo de antecedentes busca dar una breve introducción al lector debido a que le permite conocer las razones que impulsaron al desarrollo de este proyecto. A continuación, se presenta la problemática, los objetivos tanto general como específicos planteados, el alcance y la justificación que fundamenta este proyecto.

1.1. Tema

Diseño de un sistema prototipo de reconocimiento facial para la identificación de personas en la Facultad de Ingeniería en Ciencias Aplicadas (FICA) de la Universidad Técnica del Norte utilizando técnicas de Inteligencia Artificial.

1.2. Problema

En la actualidad la tecnología cubre muchas áreas de la vida cotidiana, uno de ellos es el área de la seguridad. La necesidad de incrementar la seguridad se deja sentir en todo el mundo, no solo por compañías privadas sino también por los gobiernos y las instituciones públicas. Debido a esto, últimamente los sistemas de video vigilancia inteligente se han convertido en una importante área de investigación gracias a su aplicación en el sector de la seguridad. La video vigilancia es, de hecho, una tecnología clave para la lucha contra el terrorismo y el crimen, además es de gran ayuda en la seguridad pública. Tratando de responder a estas necesidades de seguridad, la comunidad científica se centra en detectar, seguir e identificar a las personas, así como identificar su comportamiento (Alahi, Vandergheynst, Bierlaire, & Kunt, 2010).

Llevar un registro de control de ingreso de personas mediante video vigilancia inteligente surge debido al aumento de cámaras instaladas en diferentes edificios, así como a la necesidad de mayor seguridad y a la aceptación cada vez mayor de la sociedad hacia este tipo de sistemas. De

hecho, actualmente se puede considerar como una poderosa tecnología para el control de la seguridad. La principal diferencia entre el sistema tradicional y el sistema de video vigilancia inteligente radica en el análisis automático de la escena, dicho análisis automático puede comprender diferentes tareas importantes para la seguridad en general, algunas de estas tareas son: detección, seguimiento, e identificación de personas (García & Martínez, 2013). La identificación de personas es una de las principales tareas de un sistema de video vigilancia inteligente, para ello, normalmente se utilizan características biométricas, las cuales son aquellos rasgos fisiológicos que hacen únicos a los seres humanos como la cara, la huella dactilar, la voz, la retina, etc (Jain, Dass, & Nandakumar, 2004).

El reconocimiento facial, en las últimas décadas, se ha convertido en un tema de investigación multidisciplinar, involucrando a investigadores de muchas áreas, de la informática, matemáticas y neurociencias, por su naturaleza poco intrusiva (solo es necesario una fotografía del sujeto). Este tipo de sistemas siguen siendo estudiados a pesar del uso de otros métodos muy fiables de identificación, como el análisis de huellas dactilares, pero estos últimos requieren de la colaboración del sujeto a reconocer. Así pues, el reconocimiento facial tiene aplicaciones muy diversas, pero sobre todo relacionado con la seguridad y el control de sujetos (Álvarez López, 2016).

La inseguridad es un problema de carácter urgente que se suscita en toda organización dentro de la sociedad, el cual puede ser abordado desde diferentes perspectivas dentro del campo de la inteligencia artificial. En este sentido, el uso de un sistema que lleve el seguimiento o registro del ingreso de personas a cualquier entidad de una manera exhaustiva utilizando tecnologías de reconocimiento facial de alto nivel mediante sistemas de video vigilancia es de suma importancia

ya que brindan un efecto positivo en la seguridad de las personas y los bienes materiales que existen en ella.

La FICA no es la excepción ante posibles eventos de robo que se puedan suscitar en el futuro y puedan atentar con la seguridad de las personas y sus bienes materiales. El problema radica en que el sistema de video vigilancia que se encuentra disponible en sus instalaciones no es suficiente para llevar un control de seguridad eficiente y/o confiable para identificar automáticamente a las personas que ingresan a la facultad y que contribuya con el control de seguridad del personal docente y administrativo, así como al equipamiento e instalaciones de la facultad.

Además, cabe recalcar que el equipamiento electrónico que posee la FICA es de mayor valor que el que poseen otras facultades en la UTN, por lo que se requiere de especial atención en temas de seguridad. Por las razones expuestas es necesario agregar un nivel de seguridad ante posibles robos y/o daños a las instalaciones, llevando un registro automático de las personas pertenecientes a la institución y cuáles no lo son. De momento, la FICA solamente posee cámaras de video vigilancia las cuales ofrecen video de los sucesos a lo largo del día, lo que resulta difícil y tedioso al momento de descifrar la identidad de una persona con la observación directa del video. Estas cámaras deben ser aprovechadas de mejor manera para explotar el potencial de la tecnología actual en auge como lo es la inteligencia artificial y en especial énfasis en el campo del reconocimiento facial.

1.3. Objetivos

1.3.1. Objetivo general.

Diseñar un sistema prototipo de reconocimiento facial para la identificación de personas en la FICA de la Universidad Técnica del Norte utilizando técnicas de Inteligencia Artificial.

1.3.2. Objetivos Específicos.

- Elaborar un marco teórico respecto a las técnicas de reconocimiento facial y sistemas de video vigilancia inteligente que sirva como sustento técnico en la investigación.
- Construir una base de datos de entrenamiento con las características faciales de los estudiantes y docentes de la FICA.
- Implementar algoritmos de reconocimiento facial basadas en recolección y procesamiento de imágenes por computadora de acuerdo a condiciones de iluminación y posicionamiento facial que permitan el adecuado reconocimiento de individuos en un sistema prototipo en la FICA.
- Validar y documentar el funcionamiento del sistema prototipo de reconocimiento facial en el ambiente operacional de video vigilancia de la FICA.

1.4. Alcance

Para seleccionar los mejores componentes de la propuesta se determinará el software y hardware de desarrollo del prototipo que tome en cuenta la operatividad y mejor capacidad de adaptación a los requerimientos y prestaciones de la propuesta mediante la utilización de estándar IEEE 29148.

El diseño del sistema prototipo consta de un módulo de adquisición de datos que en el caso práctico de esta propuesta es una cámara de alta definición (HD) la cual se encargará de capturar una secuencia de fotogramas de video en tiempo cuasi-real para su posterior tratamiento. La importancia de la calidad de la imagen que brinde la cámara determina la precisión con la cual el sistema pueda acertar la compatibilidad o similitud en la posterior comparación de rostros con la base de datos mediante el uso de algoritmos de reconocimiento facial. Una de las principales ventajas del reconocimiento facial, es que se trata de un método no intrusivo, es decir, los datos pueden ser adquiridos incluso sin que el sujeto se percate de ello. Además, el aspecto facial es el método más utilizado de manera natural por los seres humanos para reconocerse unos a otros.

El módulo de tratamiento y comparación de fotogramas consta de la implementación de algoritmos de reconocimiento facial que son un conjunto de técnicas de visión artificial que aportan muchas ventajas en la tarea de reconocimiento, además del empleo de bases de datos de imágenes de entrenamiento; estos aspectos serán implementadas bajo una plataforma de lenguaje de programación que puede ser Visual Studio, Matlab o Python solo por mencionar algunas. El procedimiento de la selección de la técnica de reconocimiento facial se basará en el análisis de imágenes de los rostros de los individuos mediante métodos de procesamiento y extracción de características propias, considerando su comportamiento ante variantes en ambientes controlados y no controlados. La selección de los algoritmos dependerá de igual manera de la disposición y documentación que se encuentren en la actualidad de los algoritmos de reconocimiento facial en los diferentes lenguajes de programación de cada plataforma de desarrollo y su factibilidad de empleo en conjunto con la base de datos para su posterior implementación en el sistema final. Además, los métodos propuestos siguen en una progresiva investigación, desarrollo e

implementación en proyectos de visión por computadora. Adicional a lo anterior se considerará la integración de algoritmos de reconocimiento facial incluso con el uso de lentes en los individuos.

Un aspecto muy importante a destacar trata acerca de la robustez de la base datos y de la rapidez que se procesan las imágenes de todas las iteraciones en el procesamiento y comparación de los rostros. Por lo tanto, se usará una plataforma de base de datos de uso local que ofrezca estabilidad y pueda tener escalabilidad en cuanto a la capacidad del sistema, tales como MySQL, Microsoft SQL Server, Oracle, solo por mencionar algunas. La selección de la base de datos se realizará de acuerdo al costo de emplearlas, ya que existen bases de datos de libre uso y distribución, y otras que agregan un costo por la licencia de software propietario, además del tamaño de datos que se van a gestionar, tiempos de respuesta ante consultas, y la posible escalabilidad según el incremento de número de personas en la facultad. Para fines de este proyecto se utilizará en la fase de evaluación de pruebas una base de datos de libre distribución instalada localmente sobre la computadora que gestiona el sistema, así se observará su rendimiento y se propondrá seguir con su uso o se considerará migrar a una con mejores prestaciones.

Todo actuando en sinergia (Figura 1) podrá ofrecer un sistema de registro y vigilancia al acceso dentro de la facultad, implementado sobre un ordenador.

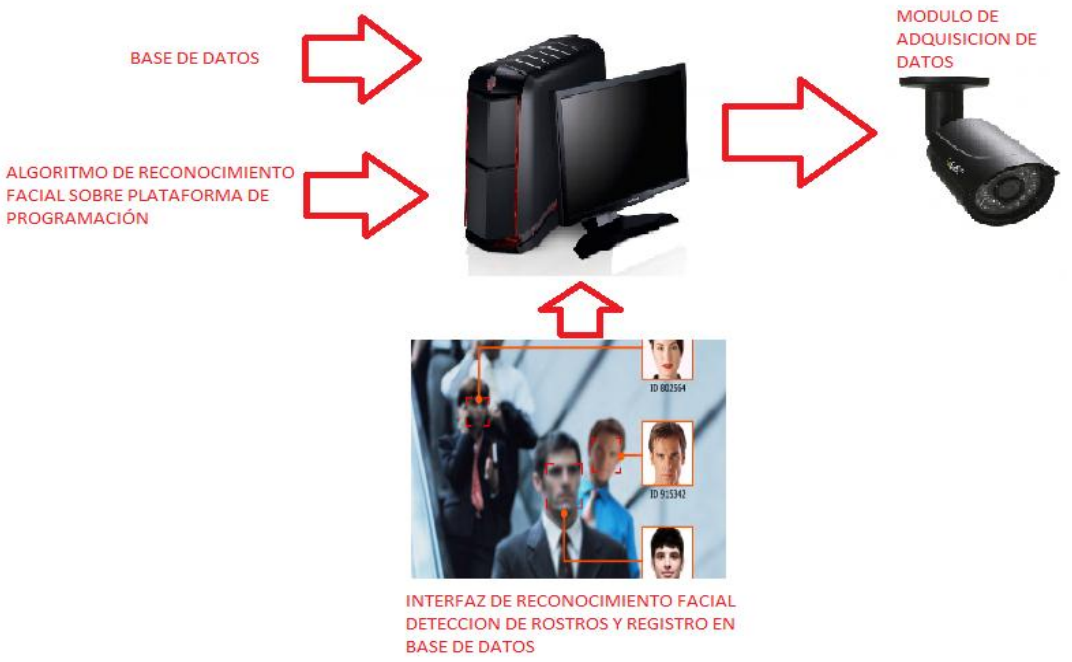


Figura 1. Esquema Topológico del sistema de reconocimiento facial.

Fuente: Autoría

El sistema en una fase de inicio se probará en un ambiente de pruebas controlado pudiendo ser un aula de clase donde aspectos como la iluminación del lugar, postura y expresión del individuo son de cierto modo controlables, difiriendo de un ambiente no controlado en donde existe mucha variación de dichas condiciones, en el que se hace especial énfasis en la iluminación del lugar de pruebas sin dejar de lado los demás aspectos. También se realizarán pruebas en ambientes no controlados pudiendo ser el caso de la entrada principal o las escaleras de acceso al primer piso de la FICA. Cabe recalcar que dichos aspectos afectan en gran manera la tarea de reconocimiento o verificación aceptable por parte del sistema con cualquier técnica de reconocimiento facial disponible en la actualidad, ya que los sistemas que poseen dichas tecnologías no son del todo eficientes en ese sentido y aún se investigan soluciones en el ámbito de la visión por computadora, por lo que se buscará la manera más eficiente posible de controlarlos y corregirlos para así obtener un adecuado rendimiento del sistema.

Además, se iniciará la fase de evaluación del sistema con un set de entrenamiento con características faciales de 25 personas para realizar la tarea de reconocimiento y obtener información relativa al rendimiento del sistema para correcciones posteriores del mismo. Paulatinamente se incrementará la cifra de individuos a reconocer hasta la posibilidad de cubrir la cifra total de individuos pertenecientes a la FICA.

De esta manera el sistema prototipo final dispondrá de un módulo de reportes el cual producirá informes diarios en relación a las personas que ingresan a la FICA en el transcurso del día, identificando a las personas existentes en la base de datos y a las que no se encuentran en ella.

Este reporte será presentado en forma de un documento diario que contenga el nombre, la fecha y hora de detección del individuo y una captura del rostro en el momento de la detección de ser posible. Específicamente se almacenará en un documento y en una lista de la base de datos la hora de reconocimiento, y el nombre del individuo reconocido. En otra lista se colocarán las personas que no han sido detectadas con la hora de detección y la captura del rostro. El reporte se realizará de esa manera para prescindir de un ordenador de muy altas características de procesamiento ya que utilizar algoritmos de visión artificial en tiempo real supone medianamente una considerable carga computacional juntamente con el uso de una base de datos. El acoplamiento de la base de datos local hacia la base de datos institucional de la UTN queda para un trabajo futuro en el cual el sistema de video vigilancia inteligente propuesto se considere desplegar alrededor de todo el campus universitario.

El sistema en su fase final será instalado en una computadora desde la cual se realizarán todas las pruebas y validaciones de reconocimiento facial, además actualmente la cámara del módulo de adquisición de datos ya se encuentra posicionada como parte del sistema de video vigilancia lo cual facilitará realizar las posteriores pruebas de funcionamiento. Sin embargo, para

la fase de evaluación se utilizará una cámara adicional con casi la misma calidad de imagen en relación a la cámara del sistema de video vigilancia a utilizar. El rendimiento podría ser medido de acuerdo a los tiempos de procesamiento de la imagen para la detección, seguimiento, y reconocimiento facial, conjuntamente con los tiempos de respuesta de las consultas de verificación en la base de datos. Los falsos positivos y negativos en el momento de la detección también son maneras de evaluar el sistema para su posterior disminución de la tasa de error del sistema.

Con el sistema prototipo final se realizarán pruebas de funcionamiento del sistema en un lugar estratégico de las instalaciones de la FICA donde los efectos negativos de los cambios de luz, postura de individuos, y entre otros aspectos tengan un mínimo impacto en el sistema en la tarea de reconocimiento y donde se concentre la mayor cantidad de afluencia de individuos utilizando una de las cámaras del sistema de video vigilancia disponible o una de adquisición propia.

1.5. Justificación

La implementación de los sistemas de reconocimiento facial está en un surgimiento vertiginoso que viene de la mano de tecnología como la inteligencia artificial en unas de sus aplicaciones de enfoque a la visión artificial, para poder reconocer a través de cámaras a los ciudadanos; muchas aplicaciones se dan en este ámbito que tiene un abanico extenso de soluciones a problemas comunes de la sociedad teniendo impacto en el sector industrial, científico o de seguridad, y entre otros. Este proyecto busca brindar una solución adecuada en beneficio de la seguridad y del control del registro de las personas que acceden a la FICA de manera concurrida. De esta manera se incursiona en el uso de nuevas tecnologías, es así que el impacto no solamente se observará en la seguridad de la FICA sino también abrirá una brecha de estudio para la

generación de nuevo conocimiento a los estudiantes y posteriores innovaciones en el área de estudio de la inteligencia artificial en aplicaciones de visión por computador.

El proyecto supondrá una mejora de la seguridad en la FICA debido a que contribuirá a la identificación de personas conocidas y desconocidas a lo largo de los sucesos del día para de alguna manera evitar posibles hurtos en las instalaciones y el ingreso de personas ajenas a la comunidad académica. Los principales beneficiarios del sistema prototipo se clasifican en beneficiarios directos e indirectos dentro de los cuales se identifican a continuación: Los beneficiarios directos constan de un aproximado de 2328 estudiantes, y 120 docentes de la FICA. Además, los beneficiarios indirectos engloban al resto de la comunidad de la UTN (estudiantes, docentes, personal administrativo, autoridades, etc).

Finalmente, el sistema prototipo establece un punto de partida para el despliegue de video vigilancia inteligente a lo largo del campus universitario para la creación de nuevas aplicaciones y nuevos trabajos de grado.

2. CAPÍTULO II. Revisión Bibliográfica

En el presente capítulo se presenta la recopilación bibliográfica de la investigación realizada para el desarrollo de este proyecto, se da a conocer un breve resumen acerca de los antecedentes del reconocimiento facial, así como también de los rasgos faciales comúnmente empleados y las aplicaciones que se benefician en la actualidad de esta tecnología. Además, se abordan detalladamente los pilares fundamentales de la Inteligencia Artificial los cuales son el Aprendizaje Automático y el Aprendizaje Profundo, que posteriormente sirven de sustento teórico y práctico para el desarrollo del proyecto. Se indican también los principios básicos de una aplicación de Visión Artificial, algunos someros conceptos de la representación de las imágenes en los computadores y las principales librerías de desarrollo de Inteligencia Artificial. Finalmente se da un breve resumen de los principales métodos de detección facial y reconocimiento facial usados en la academia y la industria, haciendo un especial enfoque en el empleo de redes neuronales convolucionales (CNN´s) además de los conjuntos de datos usados en la fase de entrenamiento.

2.1. Antecedentes del reconocimiento facial

El rostro es el rasgo biométrico más popular que se usa para reconocer visualmente a los humanos y al ser un rasgo muy característico de fácil acceso, somos capaces de discernir entre diferentes personas sólo con la información del mismo. Se ha utilizado ampliamente para autenticar la identidad de las personas a través de permisos de conducir, pasaportes, tarjetas de identificación y otros tokens de identidad relacionados. Desde los comienzos de la visión artificial, el reconocimiento facial ha sido estudiado debido a su importancia práctica e interés teórico de científicos cognitivos. Sin embargo, el diseño de técnicas de verificación de caras que sean robustas y efectivas ha sido una tarea muy desafiante. Aunque los algoritmos iniciales de

reconocimiento facial utilizaban modelos simples que utilizaban fotometría y geometría, los enfoques modernos implementan modelos matemáticos y técnicas computacionales complejas. Todos los desafíos principales en el dominio de la verificación facial se pueden atribuir al problema de la variabilidad dentro de la escena, que se traduce en cambios en las características asociadas con las imágenes faciales obtenidas de la misma persona. Estas variables incluyen iluminación, postura, distancia de la cámara, expresiones de la cara, estilo de barba, uso de lentes o gafas de sol, tipo de peinado, variabilidad del tono de piel debido a los diferentes cosméticos, arrugas debido al envejecimiento y dirección de la fuente de luz (Sadhya, Gautam, & Singh, 2017).

Sin duda alguna, los problemas de identificación biométrica constituyen una de las grandes áreas de expansión de la percepción artificial en los últimos años. En cierto sentido este hecho no es más que el reflejo de una mayor preocupación, tanto en el ámbito público como en el privado, por las cuestiones relacionadas con la seguridad. Hasta la fecha, son muchos los tipos de muestras biométricas que han sido estudiadas y utilizadas: las huellas dactilares, el iris, la firma, la voz, la palma de la mano, las orejas, las venas de la retina o de la mano, el ADN, la forma de caminar, la dinámica en el uso del teclado (escribir una frase), el rostro humano, etc. Cada tipo tiene sus propias ventajas e inconvenientes, de los cuales se deduce su ámbito de aplicación específico. Por ejemplo, el reconocimiento mediante ADN es ideal en aplicaciones de análisis forense, las huellas dactilares o las caras pueden usarse en el control de accesos a un edificio, mientras que en una aplicación bancaria es preferible la firma, por mencionar algunas aplicaciones útiles en la actualidad (García Mateos, 2007).

De forma general, la fiabilidad de los sistemas de reconocimiento biométrico puede cuantificarse en función de una serie de factores y criterios que se mencionan a continuación:

- La mayor o menor intrusividad (invasivo), desde el punto de vista del usuario.
- La precisión esperada para ese tipo de medidas.
- El coste económico de implantación en todos los niveles.
- El esfuerzo, esto es la mayor o menor facilidad de obtención de las muestras.

En la Figura 2 se muestra un análisis comparativo de estos criterios de la compañía Zephyr Biometrics para algunos de los tipos de biométricas con mayor presencia, no solamente en los ámbitos puramente de investigación, sino también en el mundo comercial. Cada tipo de biométrica se pondera con 4 factores en una escala relativa. El centro del diagrama significa un valor bajo para ese factor

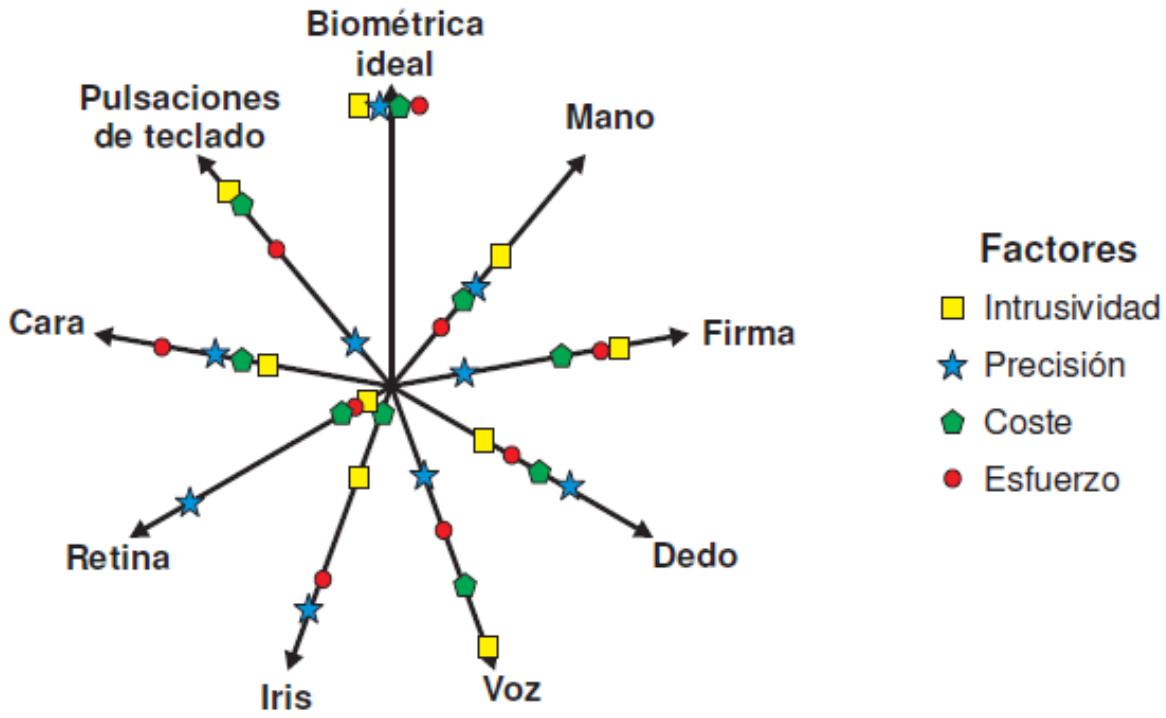


Figura 2. Análisis de la fiabilidad de diversos tipos de sistemas biométricos.

Fuente: Adaptado de (García Mateos, 2007)

Es un hecho que ninguna de las alternativas se aproxima en todos sus parámetros al sistema biométrico ideal. No obstante, hay que destacar que el reconocimiento facial presenta un buen

compromiso entre los diferentes criterios. De la misma manera, si nos fijamos en datos objetivos como cuotas de mercado, según el International Biometric Group (IBG), para el 2006-2010 la tecnología de reconocimiento facial se encontraba en segundo lugar con un 19% de cuota, justo por detrás de las basadas en huellas dactilares con un 43% de cuota como se muestra en la Figura 3.

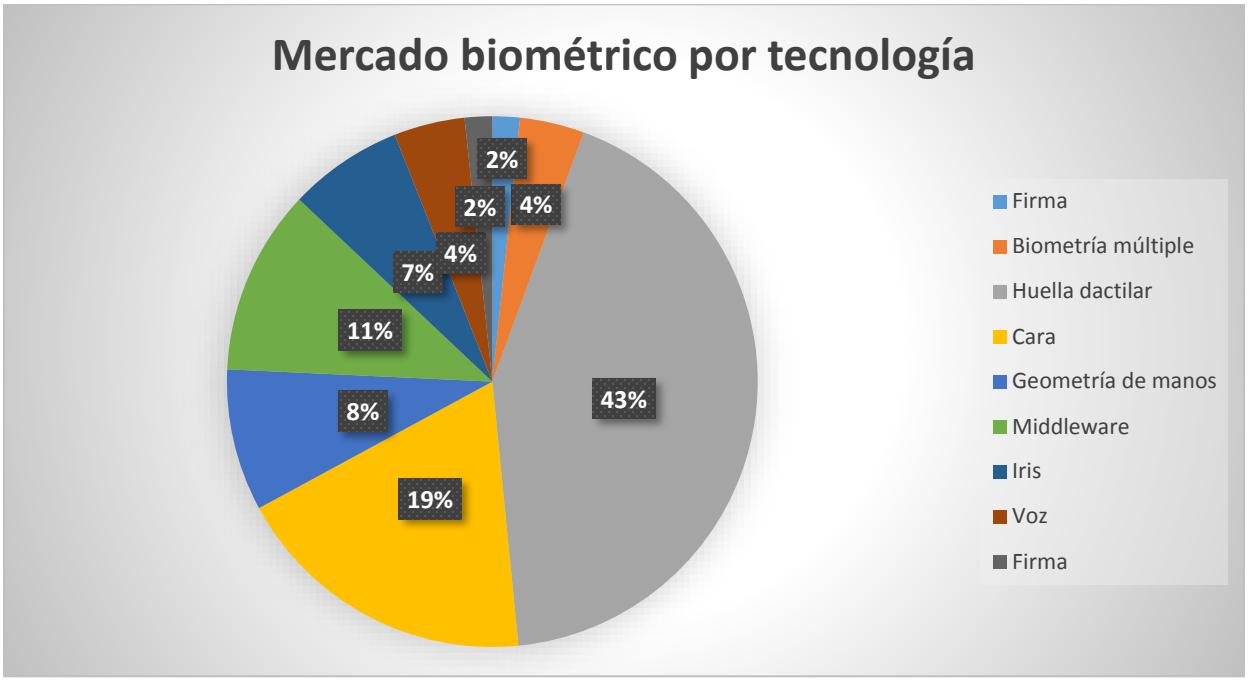


Figura 3. Tasa porcentual del mercado biométrico por tecnología.

Fuente: Adaptado de (Jung, 2006)

También es preciso mencionar el campo de aplicación al que se destinan las tecnologías de reconocimiento facial como se visualiza en la Figura 4. En el campo de la identificación de criminales e identificación de civiles se logra apreciar un especial interés, con una cuota de 26% y 34% respectivamente, denotando la preocupación de la sociedad hacia el ámbito de la seguridad.

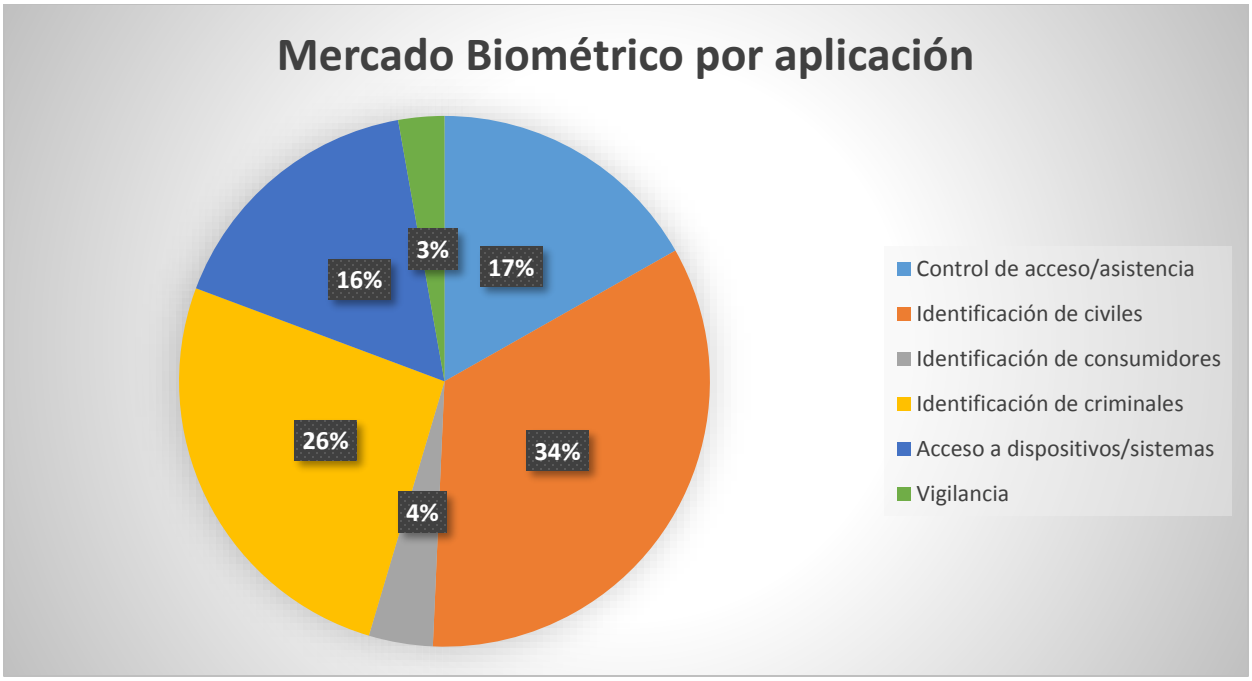


Figura 4. Tasa porcentual del mercado biométrico por aplicación.

Fuente: Adaptado de (Jung, 2006)

De acuerdo a estos análisis, es claro el lugar destacado que ocupa la investigación en reconocimiento automático de caras humanas, tanto en el número de publicaciones, como de grupos de trabajo y congresos específicos sobre el tema. En la actualidad, este campo es uno de los más maduros y con rápido crecimiento en el ámbito de la visión artificial. De hecho, compañías como Google y Facebook ya utilizan esta tecnología con una precisión bastante aceptable. El servicio de banca virtual que ofrece el banco del Pacifico en Ecuador es otro ejemplo de su uso.

2.1.1. Breve historia del reconocimiento facial.

El reconocimiento facial automatizado es un concepto de gran interés desde hace muchos años. El comienzo de las investigaciones en esta técnica se remonta a los años 60, cuando *W. W. Bledsoe* y su equipo de investigación desarrollaron los primeros sistemas de reconocimiento. Estos sistemas eran semiautomáticos, ya que requerían un administrador para localizar rasgos característicos en las fotografías.

En 1988, *L. Sirobich y M. Kirby* aplicaron análisis de componentes principales (PCA, por sus siglas en inglés), una técnica estándar del álgebra lineal, al problema del reconocimiento facial. Esto fue considerado como un hito al mostrar que eran requeridos menos de 100 componentes para cifrar acertadamente la imagen de una cara convenientemente alineada y normalizada. Un año más tarde, *T. Kohonen* definió esta técnica de reconocimiento basada en la caracterización de la cara por la extracción de los autovectores de la matriz de covarianza como *Eigenfaces*.

En 1991 *Turk & Pentland* demostraron que el error residual de codificar las *Eigenfaces* se podía utilizar para detectar caras en las imágenes, un descubrimiento que permitió sistemas automatizados de reconocimiento facial en tiempo real. Si bien la aproximación era un tanto forzada, creó un interés significativo en posteriores desarrollos de estos sistemas.

La tecnología capturó la atención del público a partir de la reacción de los medios a una prueba de implementación en el Super Bowl de la NFL en 2001, la cual capturó imágenes de vigilancia y las comparó con una base de datos de fotos digitales. Esta demostración inició un debate sobre cómo usar la tecnología para satisfacer necesidades nacionales, mientras se tomaban en consideración las preocupaciones sociales y de privacidad del público.

En 1991 *Cheng et al.* introduce el método de análisis discriminante lineal (LDA). Este método trata de encontrar un sub-espacio lineal que maximice la separación de dos clases de patrones (caras). Más tarde en 1997, *Belheumeur* introdujo el método *fisherfaces* para el reconocimiento facial. Este método es una combinación de métodos PCA y LDA. El método PCA se usa para resolver problemas singulares al reducir las dimensiones antes de ser usado para realizar el proceso LDA. Sin embargo, la debilidad de este método es que el proceso de reducción de dimensión de PCA causa alguna pérdida de información discriminante útil en el proceso de LDA.

En 1996 *Ojala* presenta un método no paramétrico llamado histograma de patrones binarios locales (LBPH), que resume las estructuras locales de las imágenes de manera eficiente al comparar cada pixel con sus píxeles cercanos. Sus propiedades más importantes son su tolerancia con respecto a los cambios de iluminación y su simplicidad computacional. Ha sido ampliamente explotado en muchas aplicaciones, por ejemplo, análisis de imágenes faciales, recuperación de imágenes y videos, modelado de entornos, inspección visual, análisis de movimientos, análisis biomédico y de imágenes aéreas, sensores remotos, etc.

Durante algunos años los métodos mencionados fueron utilizados con un éxito promedio en tareas de reconocimiento facial. Sin embargo, las constantes investigaciones en este campo han dado lugar al establecimiento de nuevas técnicas con mejorías impresionantes. En la nueva era de la visión artificial se están explotando conceptos y/o modelos de aprendizaje profundo o Deep Learning con un enfoque especial a redes neuronales convolucionales (CNNs). Muchas investigaciones se han dado en torno a este método que fue introducido en 2012 por *Krizhevsky, Sutskever & Hinton* para la clasificación de imágenes de gran resolución mediante una arquitectura de red neuronal medianamente compleja. Así, una CNN de acuerdo a su capacidad de profundidad y amplitud puede realizar suposiciones sólidas, y en su mayoría correctas, consiguiendo altos niveles de verificación facial. Sin embargo, su alto costo computacional hace que requieran de un hardware robusto con altos requerimientos de CPU & GPU[1].

[1] GPU: Unidad de Procesamiento Gráfico con enfoque práctico hacia el área del aprendizaje profundo, disponible en la línea de tarjetas de video de NVIDIA con tecnología de procesamiento paralelo (CUDA).

2.1.2. Rasgos faciales importantes en el reconocimiento facial.

La cara alberga un conjunto de rasgos bien definidos que proporcionan una gran cantidad de información sobre un sujeto. Esta información permite identificar a un sujeto de entre una multitud a simple vista por un humano, lo cual no sucede de la misma manera con un sistema de reconocimiento facial, ya que los computadores carecen de la capacidad de reconocer fácilmente patrones y/o características, puesto que necesitan de enfoques matemáticos bastante complejos para el análisis de la identidad de un sujeto. En el estado actual del reconocimiento facial existen métodos: (i) holísticos, (ii) basados en características, e (iii) híbridos que han sido ampliamente utilizados en la industria y de los cuales se hablará más a detalle en la sección 2.8.2.

Para el caso de los Métodos holísticos son aquellos que utilizan toda la región de la cara. Los Métodos basados en características buscan marcadores específicos como ojos, nariz, boca para luego usar un marcador estructural, pero requieren mayor precisión al momento de extraer los marcadores. Los Métodos híbridos son los más similares a la percepción humana, estos combinan los métodos holísticos y basados en características para reconocer un rostro de manera más precisa (Jimenez Encalada, 2015).

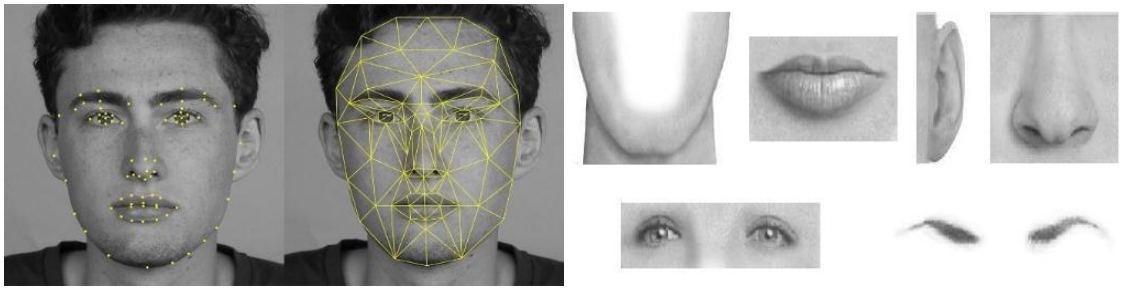


Figura 5. Rasgos faciales.

Fuente: Autoría

Particularmente los rasgos más significativos que componen el rostro humano se pueden apreciar en la Figura 5, y son los siguientes:

- Orejas: Habitualmente la variabilidad que presentan entre individuos es eminentemente geométrica, siendo el tamaño la característica que mejor las define. Las orejas al estar situadas en los laterales de la cara, pueden estar ocluidas por el pelo, generando variaciones no deseadas. Es por ello que en muchos sistemas de reconocimiento la región de la cara que se extraen y las excluyen para evitar esta variabilidad.
- Cejas: Compuestas por vello situado en la parte superior de la cara justo encima de los ojos, ofrecen diferentes características a tomar en cuenta como son el grosor, la forma, el espesor y el color del vello. Su localización puede estar modificada por la expresión, aunque, por lo general, no existe mucha variación del resto de características frente a diferentes gestos.
- Ojos: dada su complejidad, son quizá unos de los rasgos más discriminativos de la cara. Situados en la mitad superior de la cara, están compuestos por pestañas, párpados y el globo ocular que a su vez se diferencia en córnea, iris y pupila. Ofrecen gran variabilidad entre sujetos puesto que su geometría es diferente para cada uno. Aunque se considere un rasgo muy importante, los sistemas de reconocimiento facial no lo usan, ya que requieren mayor coste computacional.
- Nariz: La nariz está situada aproximadamente en el centro de la cara. Su forma varía en gran medida entre los usuarios y la misma no suele ser afectada en los cambios de expresión. Los dos orificios nasales suelen ser un buen punto característico cuando se miden distancias.

- Boca: situada en la parte inferior de la cara, es otro rasgo característico que facilita información del individuo. Como característica particular, debido a la gran flexibilidad y diversidad de movimientos que puede realizar este rasgo, es posible encontrar gran variabilidad en un mismo sujeto dependiendo de si está sonriendo, si tiene la boca abierta, está sacando la lengua, etc. Los labios son el componente que siempre está visible y que suelen definir el aspecto de la boca.
- Barba: situado en la zona del mentón de la parte inferior de la cara, es un rasgo con bastante variabilidad en los hombres, ya que el bello facial crece en la barbilla, el cuello, y los pómulos de distinta forma. Este rasgo ofrece características prominentemente únicas y diferenciables para cada individuo.

Sin embargo, la confiabilidad del sistema muy a menudo no solo depende de los rasgos característicos, sino de elementos artificiales comunes que suelen contribuir a la pérdida de fiabilidad como lo son las gorras, bufandas, lentes y gafas de sol, siendo los lentes y las gafas los que mayor oclusión generan.

2.1.3. Aplicaciones de los sistemas de reconocimiento facial.

Una de las razones por las que el reconocimiento facial ha atraído tanta atención en la investigación y desarrollo sostenido en los últimos 30 años es su gran potencial en numerosas aplicaciones gubernamentales y comerciales. Es así que una gran cantidad de aplicaciones prácticas pueden beneficiarse de esta tecnología, a continuación, en la Tabla 1 se mencionan algunos escenarios de uso.

Tabla 1. Escenarios de aplicación del reconocimiento facial

Categoría	Escenarios de aplicación
Identificación facial	Licencias de conducir, programas de derechos, inmigración, identificación nacional, pasaportes, registro de votantes, registro de asistencia social, banca virtual.
Control de acceso	Control de cruce de fronteras, acceso a instalaciones, acceso a vehículos, quiosco inteligente y cajeros automáticos, acceso a computadoras, acceso a programas informáticos, acceso a la red informática, acceso a programas en línea, acceso a transacciones en línea, acceso a aprendizaje a distancia, acceso a exámenes en línea, acceso a bases de datos en línea.
Seguridad	Alerta terrorista, sistemas seguros de embarque en vuelo, escaneo de audiencias en el estadio, seguridad informática, seguridad de aplicaciones informáticas, seguridad de bases de datos, cifrado de archivos, seguridad de intranet, seguridad de Internet, registros médicos, terminales comerciales seguros.
Vigilancia	Video vigilancia avanzada, vigilancia de plantas nucleares, vigilancia de parques, vigilancia de vecindarios, vigilancia de la red eléctrica, control de CCTV, control de portal.
Tarjetas inteligentes	Seguridad de valor almacenado, autenticación de usuario.
Cumplimiento de la ley	Detención de crímenes y alerta de sospechosos, reconocimiento de ladrones de tiendas, rastreo e investigación de sospechosos, verificación de antecedentes sospechosos, análisis posterior al evento, fraude de asistencia social.
Bases de datos de caras	Indización y recuperación de caras, etiquetado automático de caras, clasificación de caras.
Gestión multimedia	Búsqueda basada en caras, segmentación y resumen de videos basados en caras, detección de eventos.
Interacción ordenador humano (HCI)	Juegos interactivos, computación proactiva, realidad aumentada.
Aplicaciones Móviles y redes sociales	Desbloqueo móvil y de aplicaciones, identificación y etiquetamiento automático de rostros en redes sociales.

Fuente: Adaptado de (Stan Z & Anil K, 2011)

Como se puede apreciar en la Tabla 1 existen muchos escenarios de aplicación en constante aparición, por lo que en la actualidad la cantidad de estos sistemas y empresas comerciales han aumentado considerablemente, debido a la mejora adicional de las tecnologías de reconocimiento facial y la asequibilidad aumentada.

2.2. Inteligencia Artificial (IA)

La inteligencia artificial tiene por objeto el estudio del comportamiento inteligente en las máquinas. A su vez, el comportamiento inteligente supone percibir, razonar, aprender, comunicarse y actuar en entornos complejos. Una de las metas a largo plazo de la IA es el desarrollo de máquinas que puedan hacer todas estas cosas igual, o quizá incluso mejor, que los humanos. Otra meta de la IA es llegar a comprender este tipo de comportamiento, sea en las máquinas, en los humanos o en otros animales. Por tanto, la IA persigue al mismo tiempo metas científicas y metas de ingeniería. Otra definición de Marvin Minsky menciona que, la inteligencia artificial es "el estudio de como programar computadoras que posean la facultad de hacer aquello que la mente humana puede realizar", o en un sentido amplio: *la Inteligencia Artificial es una ciencia orientada al diseño y construcción de máquinas que implementan tareas propias de humanos dotados de inteligencia* (Pajares Martinsanz & Santos Peñas, 2006).

2.2.1. Visión artificial.

La visión por computador o visión artificial es un área del campo de la inteligencia artificial, y comprenden un conjunto de técnicas u algoritmos de procesamiento y extracción de características de una imagen o video que permiten que un computador o dispositivo electrónico tenga la capacidad de interpretar el significado de distintas imágenes o escenarios con distintas características, a través de la obtención de datos desde un dispositivo óptico como una cámara fija

o de video en una representación digital. Es así que un sistema de visión artificial requiere de dos elementos fundamentales, el primero es el hardware encargado de la percepción de las imágenes y el segundo el software encargado del procesamiento de la información.

La Visión Artificial se encarga de construir nuevos y más sofisticados algoritmos capaces de obtener información de bajo nivel visual, además es muy eficaz en tareas visuales repetitivas y tediosas para el hombre. Debido al progreso significativo en el campo de la visión artificial y la tecnología de sensores visuales, estas técnicas se utilizan hoy en una amplia variedad de aplicaciones del mundo real, como la interacción inteligente entre personas y computadores, la robótica, aplicaciones multimedia, entre otros (Platero Dueñas, 2009).

En su contraparte, la visión humana es capaz de ver y entender el mundo 3D que lo rodea fácilmente. Sin embargo, recuperar y comprender la estructura 3D del mundo a partir de imágenes bidimensionales 2D capturadas por cámaras es una tarea difícil para los computadores. Por ejemplo, dado un conjunto suficientemente grande de imágenes de un objeto capturado desde una variedad de vistas (Figura 6), los algoritmos de visión artificial pueden reconstruir un modelo de superficie 3D denso y preciso del objeto usando correspondencias densas en múltiples vistas. No obstante, a pesar de todos estos avances, la comprensión de las imágenes al mismo nivel que los humanos sigue siendo un desafío.

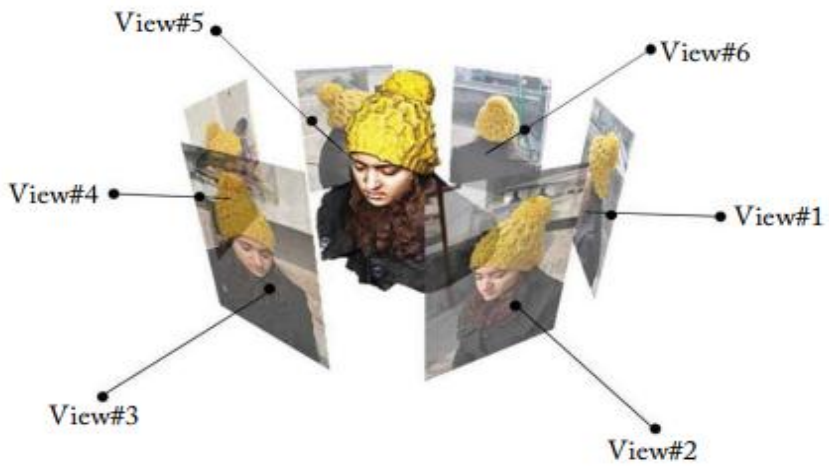


Figura 6. Dado un conjunto de imágenes 2D de un objeto (por ejemplo, el cuerpo humano superior), se puede reconstruir un modelo 3D denso del objeto utilizando algoritmos de visión artificial.

Fuente: Adaptado de (Khan, Rahmani, Shah, & Bennamoun, 2018)

En el trabajo de Platero se indica que la visión humana, respecto a la artificial, ofrece las siguientes ventajas:

Tabla 2. Ventajas de la visión humana vs. artificial

Sistema Humano	**Sistema artificial**
Mejor reconocimiento de objetos.	Mejor midiendo magnitudes físicas.
Mejor adaptación a situaciones imprevistas.	Mejor para la realización de tareas rutinarias.
Utilización de conocimiento previo.	Mejor en tareas de procesamiento de imágenes digitales de bajo nivel.
Mejor en tareas de procesamiento de imágenes visuales de alto nivel.	

Fuente: Adaptado de (Platero Dueñas, 2009)

2.2.2. Aplicaciones de la Visión Artificial.

Hoy en día numerosas aplicaciones potenciales se encuentran relacionadas a la visión artificial, comprendiendo una amplia variedad de sistemas en diferentes disciplinas, en la Tabla 3 se citan de manera breve dichas aplicaciones:

Tabla 3. Aplicaciones de la Visión Artificial

Área de producción	Aplicación
Control de calidad	Inspección de productos (papel, aluminio, acero, etc.). Identificación de piezas. Etiquetados (fechas de caducidad, etc.). Inspección de circuitos impresos. Control de calidad de los alimentos (naranjas, etc.).
Robótica	Control de soldaduras. Guiado de robots (vehículos no tripulados). Robots con capacidades avanzadas.
Biomedicina	Análisis de imágenes de microscopia (virus, células, proteínas). Resonancias magnéticas, tomografías, genoma humano.
Astronomía	Exploración del espacio.
Reconocimiento de caracteres	Control de cheques, inspección de textos.
Control de trafico	Matrículas de coches. Tráfico viario.
Meteorología	Predicción del tiempo.
Agricultura	Interpretación de fotografías aéreas. Control de plantaciones.
Militares	Seguimiento de objetivos. Vigilancia por satélites.
Vigilancia y seguridad	Conteo de personas. Rastreo de personas y vehículos. Sistemas de control de acceso. Seguimiento de trayectorias.

Fuente: Adaptado de (Platero Dueñas, 2009) & (Morante Cendrero, 2012)

En cada una de estas áreas de producción la visión artificial se ha convertido en una parte omnipresente en la industria y fuera de ella.

2.2.3. Configuración y etapas básicas de una aplicación de visión artificial.

Para el desarrollo de una aplicación de visión artificial de cualquier tipo existen dos pilares fundamentales del sistema físico que son:

> El sistema de formación de las imágenes y el sistema de procesamiento de éstas. En el primer apartado estaría constituido por el subsistema de iluminación, de captación de la imagen y de adquisición de la señal en el computador. Una vez introducida la señal en el computador, ésta es procesada mediante los algoritmos para transformar en información de alto nivel. La cual puede ser utilizada para su representación visual, para actuar en el planificador de un robot o ser fuente de datos para un autómata programable. En definitiva, múltiples periféricos pueden ser receptores de esta información y vincularse con el sistema de procesamiento de las imágenes. (Platero Dueñas, 2009, pág. 25)

Dicha configuración consta de subsistemas que se encargan de realizar diferentes procesos que actúan en sinergia para entregar un análisis de la escena capturada, estas tienen las siguientes reflexiones introductorias:

- Subsistema de iluminación: conjunto de artefactos que producen radiación electromagnética para que incidan sobre los objetos a visualizar. Se puede citar algunos elementos como lámparas, pantallas fotográficas, filtros de luz, láseres.
- Subsistema de captación: son los transductores que convierten la radiación reflejada luminosa en señales eléctricas. Fundamentalmente se habla de las cámaras CCD, no sólo en el espectro visible, sino que van desde la radiación gamma hasta la radiofrecuencia o microondas, dando paso a sensores de ultrasonidos, sonar, radar, telescopía.
- Subsistema de adquisición: las señales eléctricas procedentes de las cámaras forman la señal de vídeo. Hay una tendencia creciente a que su naturaleza sea de tipo digital, pero todavía existen muchas señales de vídeo de carácter analógico

(CCIR, PAL, RS170, NTSC). Para ser tratadas hay que muestrearlas y cuantificarlas. Ambas tareas son realizadas por las tarjetas de adquisición.

- Subsistema de procesamiento: Suele ser un computador o un clúster de computadores, dependiendo de las necesidades de los algoritmos de Visión Artificial. Parten de una representación digital de las imágenes y procesan esta información hasta alcanzar otro tipo de información de más alto nivel. La transformación dependerá de la algoritmia.
- Subsistemas de periféricos: conjunto de elementos receptores de la información de alto nivel. Puede ser un monitor de altas prestaciones gráficas, un automatismo, una impresora sacando las características.

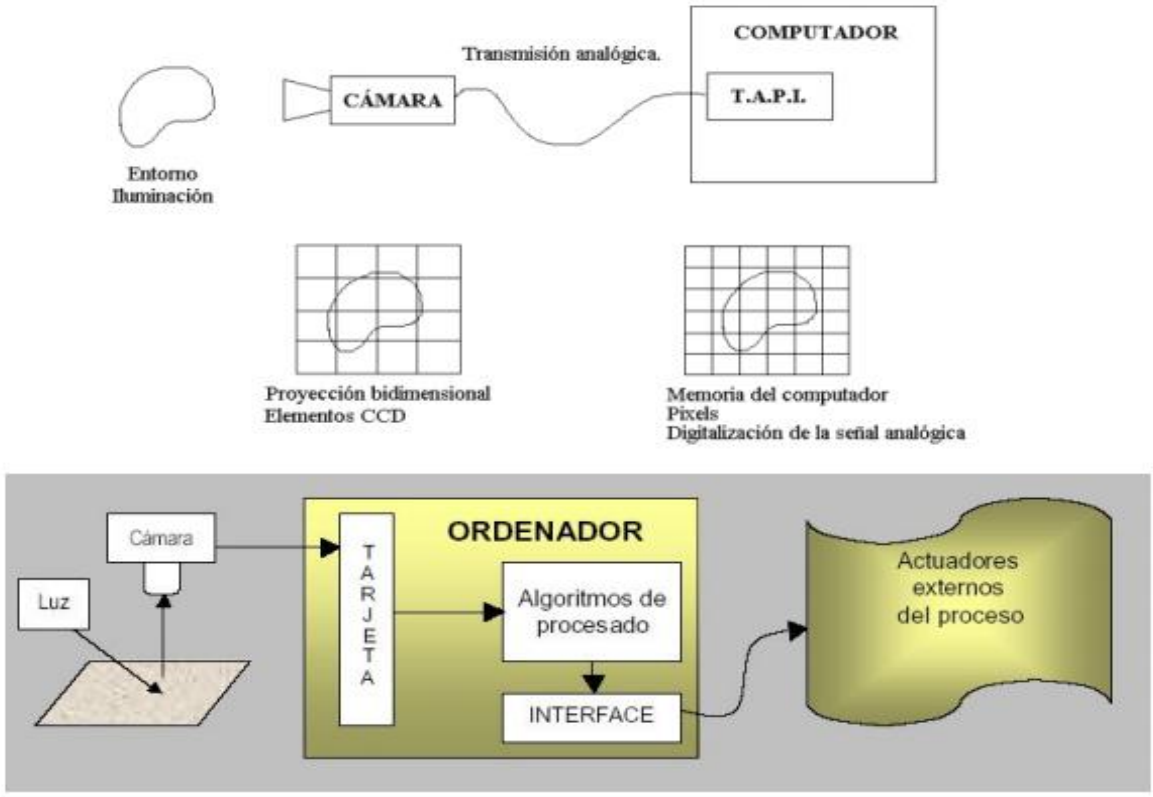


Figura 7. Subsistemas físicos de un equipo de visión artificial.

Fuente: Adaptado de (Platero Dueñas, 2009)

En la Figura 7 se muestran todos los subsistemas interconectados, representando un típico sistema de visión artificial. Sin embargo, cada aplicación de visión artificial tiene sus particularidades y se estructura de una serie de etapas las cuales engloban a cada subsistema. En la Figura 8 se muestra un esquema simplificado general para la puesta en marcha de una aplicación.

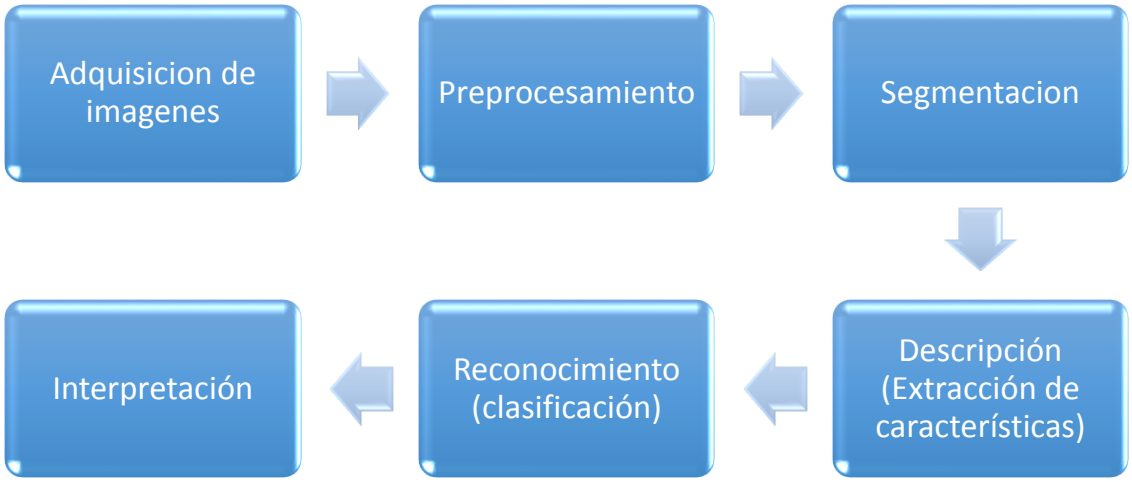


Figura 8. Etapas de un sistema de Visión Artificial.

Fuente: Adaptado de (Platero Dueñas, 2009) & (García-Santillán & Caranqui, 2014)

La primera etapa es la construcción del sistema de formación de las imágenes, y el objetivo de esta etapa es realzar las características visuales de los objetos (formas, texturas, colores, sombras) encontrados en las imágenes. Del buen diseño de esta etapa dependerá el éxito de la aplicación (Sobrado Malpartida, 2003).

Una vez adquirida la imagen se pasará a la segunda etapa de preprocesado. El objetivo es mejorar la calidad informativa de la imagen adquirida, mejorando la relación señal-ruido (SNR), regularizando la imagen, mejorando el contraste, realzando características de la imagen, entre otras (Platero Dueñas, 2009).

La tercera etapa de segmentación se encarga de evaluar si cada pixel de la imagen pertenece o no al objeto de interés en función de los valores RGB o HSV, esta etapa genera una imagen

binaria, representados por 0 a los que no pertenecen (fondo) y por uno a los píxeles que si pertenecen al objeto de interés (Sobrado Malpartida, 2003).

En la cuarta etapa se pasa a la extracción de características mediante operaciones de tipo morfológico o empleando características basadas en la textura, forma o color lo que permite representar las características relevantes que posee un objeto en específico (Platero Dueñas, 2009).

En la quinta etapa se emplean una variedad de técnicas de clasificación, como redes neuronales, sistemas expertos, lógica borrosa, clasificadores estadísticos, entre otros, para la clasificar los objetos con características comunes del espacio 3D proporcionándoles una etiqueta de alto nivel (Platero Dueñas, 2009).

En la última etapa de interpretación se visualizan los resultados de todo el proceso previo, aplicado a cualquier tarea de reconocimiento visual para la que fueron diseñados.

2.2.4. Disciplinas relacionadas con la visión artificial.

En el trabajo de Platero se menciona que, la visión artificial al ser un proceso complejo suele ser de tipo multidisciplinar y se encuentra dividida en varias etapas o procesos mencionados en la anterior sección. En cada una de ellas, las imágenes y la cantidad de información se va refinando hasta lograr el reconocimiento del objeto buscado. Es preciso destacar que la naturaleza del proyecto hace que se incida en una disciplina más que en otra. Generalmente, se requiere el manejo de las siguientes técnicas:

- Fotografía y Óptica: crear el ambiente de iluminación adecuada en la adquisición de las imágenes, muchas veces requiere del uso de técnicas profesionales de fotografía y vídeo. La selección de la óptica y de la cámara, el uso de filtros y

polarizadores, las técnicas de iluminación con pantallas y la elección de los tipos de focos son algunas habilidades que se pueden mencionar.

- Procesamiento Digital de Imágenes (*Image Processing*): hace referencia a los algoritmos de computación que convierte la imagen digital adquirida en otra de mayor relevancia.
- Reconocimiento de Patrones (*Pattern Recognition*): disciplina, dentro de la Inteligencia Artificial, dedicada a la clasificación de las señales y a la búsqueda de patrones existentes dentro de éstas. Se encuentran incluidas las técnicas de clasificadores estadísticos, Redes Neuronales, Sistemas Expertos, Lógica Borrosa.
- Computación Gráfica (*Computer Graphics*): presenta el problema inverso de la Visión Artificial. Si en Visión se desea extraer las características físicas de las imágenes, la Computación Gráfica se dedica a la generación de los modelos geométricos. Cada vez más, la Visión Artificial emplea la Computación Gráfica para representar las conclusiones extraídas del análisis de las imágenes adquiridas.

2.2.5. Limitaciones de la visión artificial.

En el trabajo de Fernández García (2016) & Caballero Barriga (2017) se indica que la visión artificial pretende reproducir artificialmente el sentido de la visión humana en un ordenador, pero al igual que posee capacidades también se presentan limitaciones que se mencionan a continuación:

- Cambios de iluminación: con los distintos tonos de luz y los cambios bruscos de iluminación se crean sombras o reflejos que impiden tener una imagen clara para ser procesada por el computador.

- Cambios de escala: los cambios de tamaños en las imágenes impiden que se realice el seguimiento adecuado de los objetos, con lo que se incrementa el tiempo de procesamiento y se emplean más recursos del hardware.
- Deformación de los objetos: una imagen deformada impide que se realice el seguimiento y extracción de las características de los objetos de interés.
- Oclusión: ocurre cuando un objeto que se desea procesar se encuentra detrás de otra, con lo que la única información que obtenemos es la que se encuentra visible, generando pérdida de datos sobre la imagen, afectando cualquier tarea de visión artificial.

2.3. Conocimientos generales sobre imágenes

A continuación, se realiza una breve revisión de los tipos de imágenes que pueden ser manipuladas y/o procesadas por los computadores para distintos fines u aplicaciones dentro del campo de la visión artificial.

2.3.1. Representación de las imágenes en los computadores.

Las imágenes para ser procesadas en el computador han sido adquiridas a través de la cámara de vídeo y puestas en su memoria empleando tarjetas de adquisición de vídeo. Esta señal es de carácter bidimensional y es expresada como una función $f(x, y)$ de dos dimensiones que representa la intensidad de luz de la imagen en el punto (x, y), que ha sido discretizada en los dos ejes y cuantizada en brillo, estas imágenes se almacenan en matrices, un pixel por celda. La Figura 9 muestra la representación de una imagen en la lógica computacional.

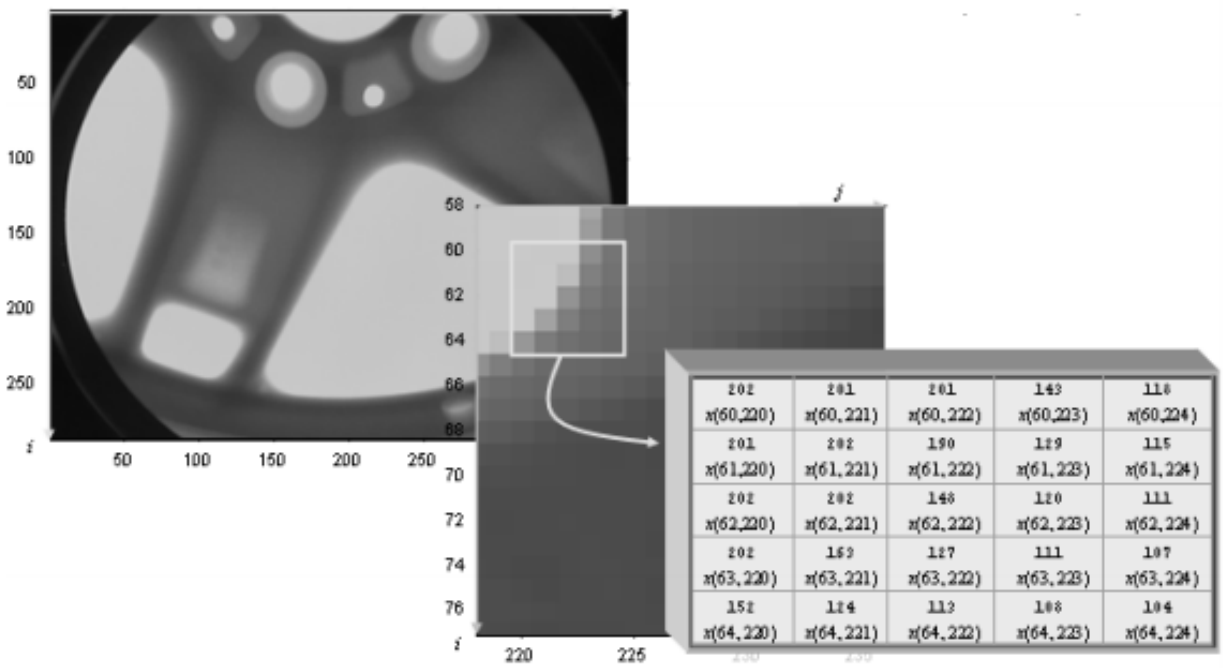


Figura 9. Organización matricial uniforme de una imagen digital.

Fuente: Adaptado de (Li & Dong, 2014)

2.3.1.1. Clasificación de imágenes.

En el trabajo de García Santillán (2008) se indica que dependiendo de los valores que pueda tomar cada uno de los píxeles de la imagen podemos distinguir entre 4 tipos de imágenes (Figura 10):

- Imágenes en color (RGB)(a): Estás imágenes no sólo están formadas por una matriz de píxeles, sino por tres, sus tres componentes básicas. Así pues, cada píxel de una imagen en color está a su vez formado por tres píxeles, un valor para cada uno de estos píxeles que pertenecerán a los tres componentes de color básicos (rojo, verde y azul)

- Imágenes indexadas (b): En estas imágenes se reduce los colores de la imagen a un máximo de 256 y consiste de una matriz de datos X y un mapa de color, los pixeles en la imagen son enteros, los cuales apuntan (índices) a los valores de color almacenados en el mapa de color.
- Imágenes en escala de grises (o de intensidad) (c): Existen 256 valores que el píxel puede tomar, siendo cada uno de estos valores un nivel de gris diferente, por lo que el píxel puede estar entre [0, 255].
- Imágenes binarias (d): En estas imágenes el píxel sólo puede tomar valor de 0 o 1, siendo 0 negro y 1 blanco.
- Imágenes en movimiento (o vídeo): Estás imágenes pueden ser de los 3 tipos anteriores, simplemente existe una variable más, t, que representa la variación entre imágenes, para obtener un vídeo.

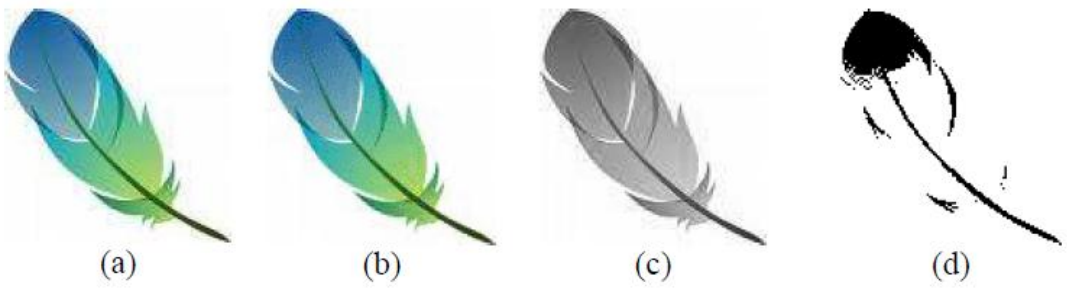


Figura 10. Tipos de imágenes digitales.

Fuente: Adaptado de (García Santillán, 2008)

2.4. Biblioteca de visión artificial

OpenCV es una librería software de visión artificial y sistemas de aprendizaje automático diseñada para conseguir una eficiencia computacional elevada y enfocada en aplicaciones de tiempo real desarrollada nativamente en C/C++ y bajo una licencia BSD (*Berkeley Software Distribution*). Esta licencia permite el uso y modificación del código tanto para fines comerciales como académicos. Actualmente dispone de interfaces de desarrollo o API (*Application Program Interface*) para C++, C, Python, Java y MATLAB siendo compatible con Linux, Mac OS X y Windows, así como también con dispositivos móviles Android y iOS. La librería ofrece más de 2500 algoritmos que han sido optimizados llegando a abarcar un conjunto completo de algoritmos en el campo de la visión artificial y del aprendizaje automático (Martínez Guerrero, 2018).

2.5. Aprendizaje Automático (Machine Learning)

El aprendizaje automático es un tipo de inteligencia artificial (IA) que permite a las computadoras aprender de los datos sin ser programados explícitamente como ocurre en la programación tradicional. En otras palabras, el objetivo del aprendizaje automático es diseñar métodos que realicen el aprendizaje de forma automática utilizando observaciones del mundo real (llamadas "datos de entrenamiento"), sin la definición explícita de reglas o lógica por parte de los humanos ("entrenador" / "supervisor"). En ese sentido, el aprendizaje automático puede considerarse como programación por muestras de datos. En resumen, el aprendizaje automático consiste en aprender a mejorar el futuro en función de lo que se experimentó en el pasado (Khan, Rahmani, Shah, & Bennamoun, 2018).

Se ha propuesto un conjunto diverso de algoritmos de aprendizaje automático para cubrir la gran variedad de datos y tipos de problemas. Dichos métodos de aprendizaje se pueden dividir principalmente en tres enfoques principales, tales como: aprendizaje supervisado, aprendizaje semi-supervisado, y aprendizaje no supervisado (Shai & Shai, 2014). Para el desarrollo de este proyecto se utilizará en su totalidad algoritmos y/o conceptos de aprendizaje supervisado, ya que se proporcionará una vasta cantidad de datos con sus respectivas etiquetas (respuestas) para entrenar una función o modelo.

2.5.1. Aprendizaje supervisado.

En el aprendizaje supervisado, se proporcionan ejemplos de la forma (x_i, y_i) y se asume una función de aprendizaje o mapeo f tal que, $f(x_i) = y_i$. El objetivo es encontrar la función f, de tal forma que dicha función capture los "patrones generales" presentes en los datos de entrenamiento y se pueda aplicar para predecir valores y, a partir de diversos valores de muestra de entrada x. Generalmente cada x_i es una descripción de algún objeto, situación o evento y cada y_i es un descriptor simple. Existen diferentes formas de función de mapeo $f(,)$, incluyendo regresión lineal univariable y multivariable, árboles de decisión, bosques de decisión aleatoria (RDF), regresión logística (LR), máquinas de vector soporte (SVM), redes neuronales artificiales (ANN), máquinas de kernel, y clasificadores bayesianos. También se ha propuesto una amplia gama de algoritmos de aprendizaje para estimar estos diferentes tipos de mapeos (Villegas Quezada, 2005).

2.5.2. Aprendizaje no supervisado.

El aprendizaje no supervisado es donde uno solo tendría datos de entrada x y ninguna variable de salida correspondiente *y*. Se llama aprendizaje no supervisado porque (a diferencia del

aprendizaje supervisado) no hay resultados de verdad y no hay supervisor. El objetivo del aprendizaje no supervisado es modelar la estructura / distribución subyacente de los datos para descubrir una estructura interesante en los datos. El método de aprendizaje no supervisado más común es el enfoque de agrupamiento, como el agrupamiento jerárquico, el agrupamiento k-means, los modelos de mezcla gaussianos (GMM), los mapas autoorganizados (SOM) y los modelos ocultos de Markov (HMM) (Khan, Rahmani, Shah, & Bennamoun, 2018).

2.5.3. Aprendizaje semi-supervisado.

Los métodos de aprendizaje semi-supervisados se ubican entre el aprendizaje supervisado y el no supervisado. Estos métodos de aprendizaje se utilizan cuando hay una gran cantidad de datos de entrada disponibles y solo algunos de los datos están etiquetados. Un buen ejemplo es un archivo fotográfico en el que solo algunas de las imágenes están etiquetadas (por ejemplo, perro, gato, persona) y la mayoría no están etiquetadas (Khan, Rahmani, Shah, & Bennamoun, 2018).

2.5.4. Aprendizaje por refuerzo.

El aprendizaje por refuerzo, RL (*Reinforcement Learning*), consiste en mapear situaciones a acciones maximizando un escalar denominado señal de refuerzo o recompensa. Es una técnica de aprendizaje basado en prueba y error, que estima una función de valor de acción. El aprendizaje por refuerzo se utiliza cuando no existe una información detallada sobre la salida deseada. En este tipo de aprendizaje no se indican cuáles son las acciones correctas. La idea es que el sistema explore el entorno y observe el resultado de las acciones sobre algún índice de resultados que permita obtener información para su aprendizaje. Los elementos que forman un sistema de aprendizaje por refuerzo son un agente, el entorno, una política, una función de valor, y, opcionalmente, un modelo del entorno (Mnih, y otros, 2013).

2.6. Aprendizaje Profundo (Deep Learning)

El aprendizaje profundo es una nueva área de investigación del aprendizaje automático, que se ha introducido con el objetivo de solventar muchos de los retos de la inteligencia artificial. Básicamente los algoritmos de aprendizaje profundo pueden aprender características, niveles de representación, y tareas directamente de imágenes, texto y sonido, eliminando la necesidad de la selección manual de características, mediante modelos discriminativos profundos (por ejemplo, redes neuronales profundas o DNN, redes neuronales recurrentes o RNN, redes neuronales convolucionales o CNN[2], etc.) y modelos generativos / no supervisados (por ejemplo, máquinas de Boltzmann restringidas o RBM, redes de creencias profundas o DBN, máquinas de Boltzmann profundas (DBMs), codificadores automáticos regularizados, etc.) (Li & Dong, 2014).

De hecho, se consideran dos aspectos clave: (1) modelos que consisten de múltiples capas o etapas de procesamiento de información no lineal; y (2) métodos para el aprendizaje supervisado o no supervisado de la representación de características en capas sucesivamente más altas y abstractas. El aprendizaje profundo se encuentra en las intersecciones entre las áreas de investigación de redes neuronales, inteligencia artificial, modelado gráfico, optimización, reconocimiento de patrones, procesamiento de señales, visión artificial (Goodfellow, Bengio, & Courville, 2016). La Figura 11 muestra la relación entre algunos de ellos.

[2] CNN: Convolutional Neural Networks (Redes Neuronales Convolucionales), son un tipo de redes neuronales usadas para el campo de la Visión Artificial.

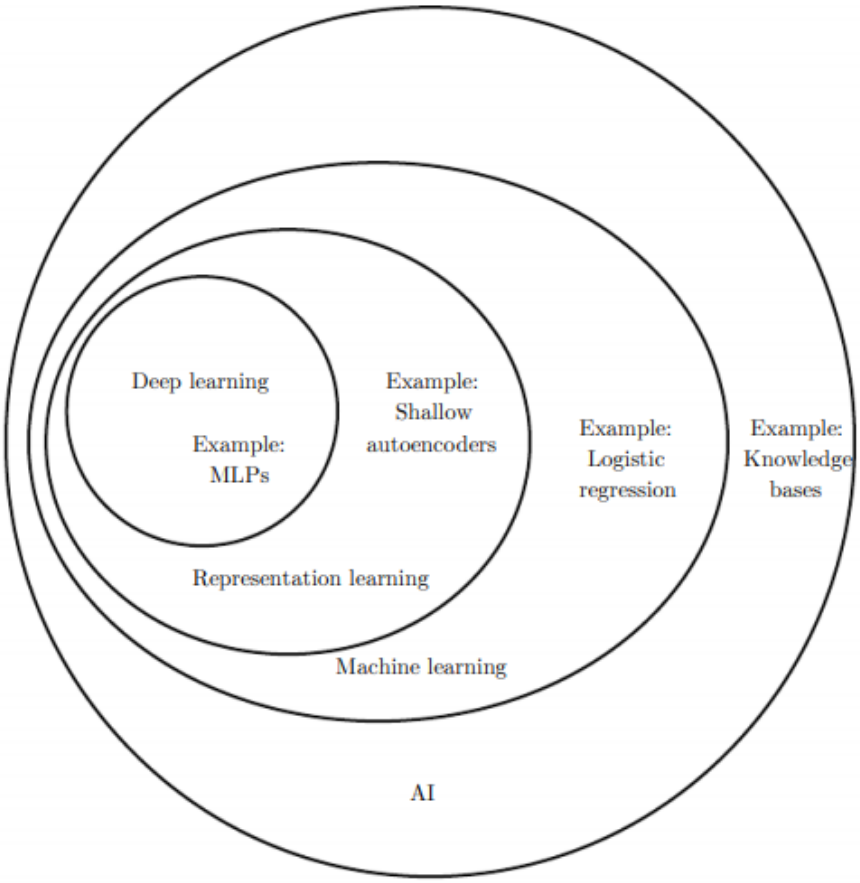


Figura 11. Relación entre el aprendizaje profundo y algunos campos de IA.

Fuente: Adaptado de (Goodfellow, Bengio, & Courville, 2016)

Tres razones importantes para la popularidad del aprendizaje profundo hoy en día son el aumento drástico de la capacidad de procesamiento de chips (por ejemplo, GPGPU[3]), el tamaño significativamente mayor de los datos utilizados para el entrenamiento y los avances recientes en aprendizaje automático e Investigación de procesamiento de señales/información. Estos avances han permitido a los métodos de aprendizaje profundo explotar con eficacia funciones no lineales compuestas y complejas, para aprender representaciones de funciones jerárquicas y distribuidas, y para hacer un uso efectivo de los datos etiquetados y no etiquetados.

[3] GPGPU: Unidades de Procesamiento Gráfico de Propósito General

Hay tres ventajas clave que ofrece el aprendizaje profundo:

- Simplicidad: en lugar de ajustes específicos del problema y detectores de características personalizados, las redes profundas ofrecen bloques arquitectónicos básicos, capas de red, que se repiten varias veces para generar grandes redes.
- Escalabilidad: los modelos de aprendizaje profundo son fácilmente escalables a grandes conjuntos de datos. Otros métodos competitivos, por ejemplo, las máquinas del kernel, encuentran serios problemas computacionales si los conjuntos de datos son enormes.
- Transferencia de dominio: un modelo aprendido en una tarea es aplicable a otras tareas relacionadas y las características aprendidas son lo suficientemente generales para trabajar en una variedad de tareas que pueden tener escasos datos disponibles.

Debido al tremendo éxito en el aprendizaje de estas redes neuronales profundas, las técnicas de aprendizaje profundo son actualmente modernas para la detección, segmentación, clasificación y reconocimiento (es decir, identificación y verificación) de objetos en imágenes. Los investigadores ahora están trabajando para aplicar estos éxitos en el reconocimiento de patrones a tareas más complejas como los diagnósticos médicos y la traducción automática de idiomas (Khan, Rahmani, Shah, & Bennamoun, 2018).

En el trabajo de Li & Dong (2014) se menciona que dependiendo de cómo se diseñen las arquitecturas y técnicas, por ejemplo, síntesis/generación o reconocimiento/clasificación, se puede clasificar en términos generales la mayor parte del trabajo en esta área en dos clases principales: redes profundas para el aprendizaje supervisado, y no supervisado. En la siguiente sección se explican más a detalle.

2.6.1. Redes profundas para aprendizaje supervisado.

El objetivo de este tipo de redes es proporcionar directamente poder discriminativo para propósitos de clasificación de patrones, a menudo caracterizando las distribuciones posteriores de clases condicionadas a los datos visibles. Los datos de la etiqueta de destino siempre están disponibles en forma directa o indirecta para dicho aprendizaje supervisado. También se les llama redes profundas discriminativas.

2.6.1.1. Redes Neuronales.

Las redes neuronales están inspiradas en el trabajo de la corteza cerebral de los mamíferos y de manera técnica pueden definirse como un conjunto de elementos de procesamiento de la información altamente interconectados, que son capaces de aprender con la información que se les alimenta. La principal característica de esta nueva tecnología de redes neuronales es que puede aplicarse a un gran número de problemas que pueden ir desde problemas complejos reales a modelos teóricos sofisticados como por ejemplo reconocimiento de imágenes, reconocimiento de voz, análisis y filtrado de señales, clasificación, discriminación, análisis financiero, predicción dinámica, etc. (Pérez & Santín, 2007).

Las redes neuronales se pueden agrupar en dos categorías genéricas según la forma en que se propaga la información en la red, a continuación, se detallan cada una:

- Redes Feed-forward: El flujo de información en una red Feed-forward ocurre solo en una dirección. Si la red se considera un gráfico con neuronas como sus nodos, las conexiones entre los nodos son tales que no hay ciclos en el gráfico. Algunos ejemplos incluyen Perceptrón Multi Capa (MLP) y CNN´s. La Figura 12 muestra la arquitectura de una red MLP básica.

- Redes Feed-back: Como su nombre lo indica, las redes Feed-back tienen conexiones que forman ciclos dirigidos. Esta arquitectura les permite operar y generar secuencias de tamaños arbitrarios. Además, estas redes muestran capacidad de memorización y pueden almacenar información y secuencias de relaciones en su memoria interna. Los ejemplos de tales arquitecturas incluyen redes neuronales recurrentes (RNN) y las unidades de memoria a largo plazo (LSTM).

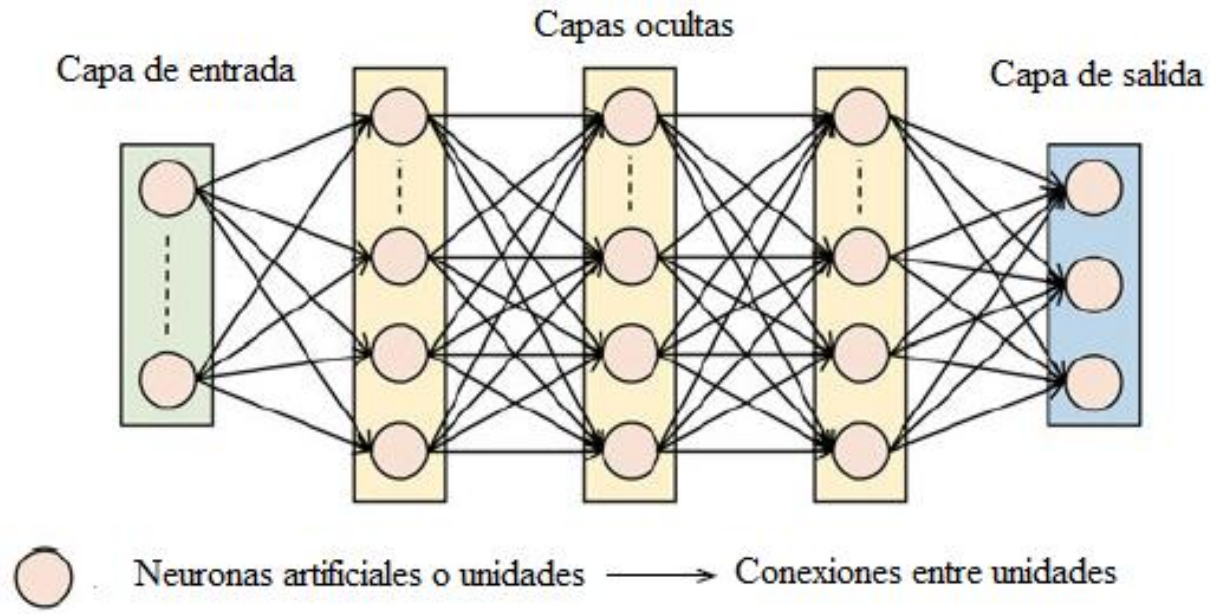


Figura 12. Ejemplo de una red neuronal Feed-forward.

Fuente: Adaptado de (Khan, Rahmani, Shah, & Bennamoun, 2018)

De forma resumida, la entrada se alimenta a través de una capa de entrada que representa un vector de datos de entrada y la capa final es la capa de salida que hace predicciones o conjunto de posibles etiquetas. Las capas intermedias u ocultas realizan el procesamiento a través de los nodos que se encuentran interconectados, e implementan una función de activación que, dada una entrada, decide si el nodo se activará o no.

2.6.1.2. Redes Neuronales Convolucionales (CNN´s).

Las CNN´s son una de las categorías más populares de redes neuronales, especialmente para datos de alta dimensión (por ejemplo, imágenes y videos). Las CNN´s funcionan de manera muy similar a las redes neuronales estándar. Sin embargo, una diferencia clave es que cada unidad en una capa CNN es un filtro bidimensional que está convuelto con la entrada de esa capa. Las CNN´s preparan las neuronas en forma tridimensional (ancho, largo y profundidad de capa). Además, este formato abarca las dimensiones de una imagen en formato RGB o HSV, la Figura 13 muestra la operación de una CNN, haciendo uso de diferentes tipos de capas que conforman la configuración de la arquitectura, tales como: capa convolucional, capa pooling, capa fully-connected, y funciones de activación ReLU (Rectified Linear Unit).

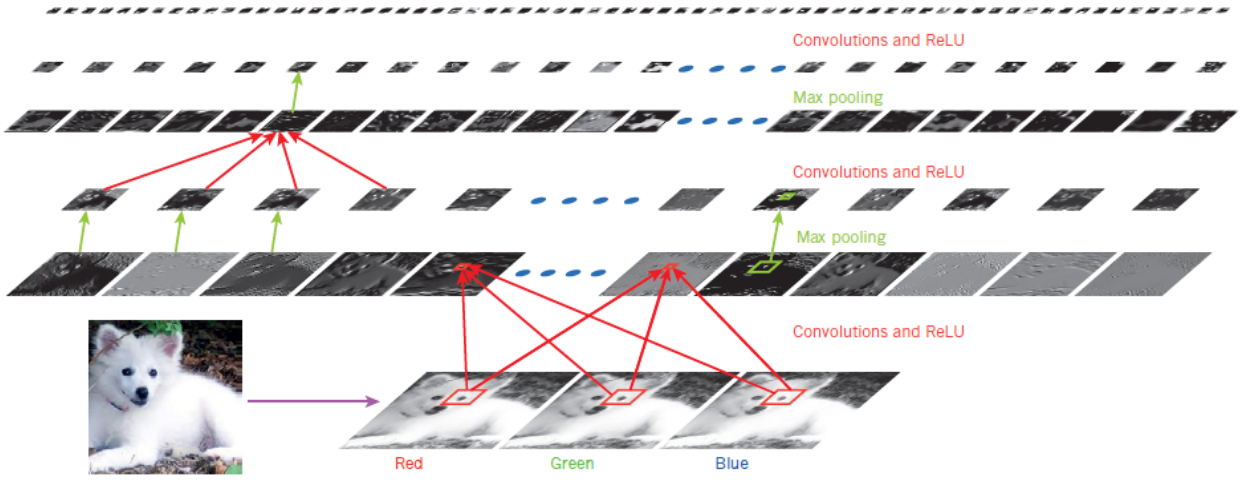


Figura 13. Aplicación de una CNN sobre una imagen.

Fuente: Adaptado de (LeCun, Bengio, & Hinton, 2015)

2.6.1.2.1. *Capa Convolucional.*

En esta capa es donde se encuentra la idea principal de este tipo de red neuronal. Consiste en la realización de operaciones de productos y sumas entre la capa de partida y los *K* filtros a aplicar que generan un mapa de características, estas operaciones se pueden observar en la Figura

14. Dichas características corresponden a cada posible ubicación del filtro en la imagen original. El mismo filtro sirve para extraer la misma característica en cualquier parte de la entrada reduciendo así el número de conexiones y generando una nueva imagen mucho más discriminativa (Martínez Guerrero, 2018).

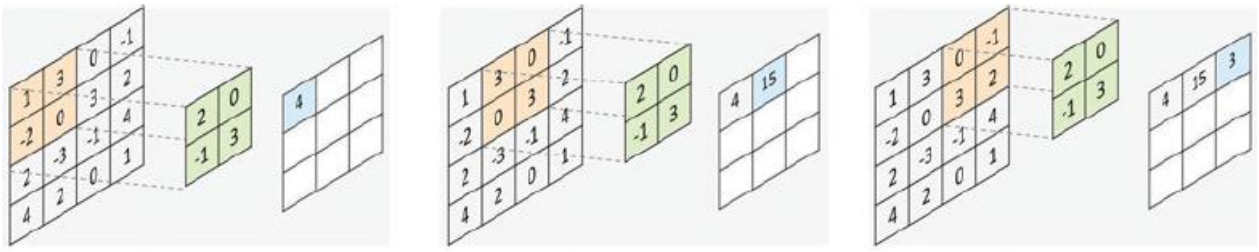


Figura 14. Operación de convolución de una capa de tamaño 4x4 con un filtro 2x2.

Fuente: Adaptado de (Khan, Rahmani, Shah, & Bennamoun, 2018)

2.6.1.2.2. *Capa Pooling.*

La capa pooling opera independientemente en cada nivel de profundidad y la forma de realizar la reducción es mediante la extracción de estadísticas como el promedio o el máximo de una región fija del mapa de características, además es muy común introducirlas entre capas convolucionales. Habitualmente la reducción se realiza con la operación max o average pooling con una ventana de NxN dimensiones consiguiendo así una reducción en el volumen de los datos de entrada (anchura y altura). En la Figura 15 se muestra esta operación.

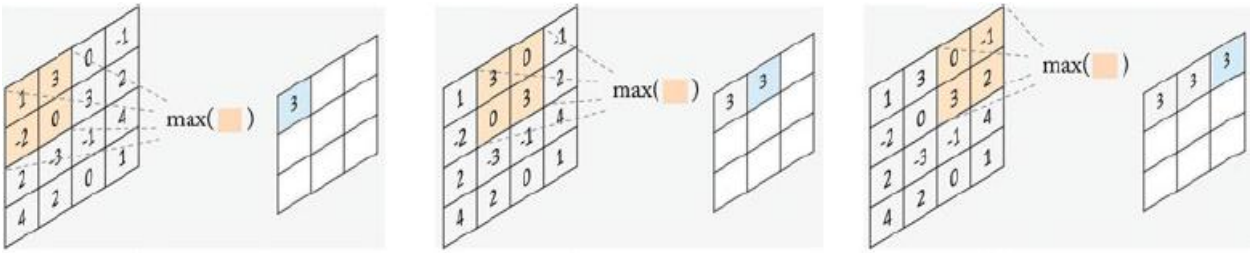


Figura 15. Operación de max-pooling con una región pooling 2x2.

Fuente: Adaptado de (Khan, Rahmani, Shah, & Bennamoun, 2018)

2.6.1.2.3. *Capa Fully Connected.*

Este tipo de capa son las mismas que se utilizan en una red neuronal convencional, creando conexiones todos con todos. Habitualmente suele ser utilizada después de una capa convolucional o pooling, por lo que nos encontramos con que requiere reorganizar los elementos de la matriz tridimensional en un vector. En estas capas existe un parámetro de gran utilidad, el Dropout. Este parámetro de configuración determina con que probabilidad va a existir la conexión entre dos neuronas. Este parámetro permite reducir el sobreentrenamiento de una red (Martínez Guerrero, 2018).

2.6.2. Redes profundas para el aprendizaje no supervisado o generativo.

Están diseñadas para capturar la correlación de alto orden de los datos observados o visibles con fines de análisis de patrones o síntesis cuando no hay información disponible sobre las etiquetas de la clase objetivo. El aprendizaje de representaciones o características no supervisadas en la literatura se refiere a esta categoría de las redes profundas. Cuando se utiliza en el modo generativo, también puede tener la intención de caracterizar las distribuciones estadísticas conjuntas de los datos visibles y sus clases asociadas cuando estén disponibles y se traten como parte de los datos visibles. En este último caso, el uso de la regla de Bayes puede convertir este tipo de redes generativas en discriminativas para el aprendizaje.

2.7. Lenguaje de programación interpretado: Python

Python es un lenguaje de programación interpretado, multiparadigma, de alto nivel con un tipado dinámico fuerte, dotado de una gestión automática de los recursos, de un alto grado de introspección y de un sistema de gestión de excepciones. Es libre y gratuito, funciona sobre todas las plataformas, apareció en 1990 y posee varias implementaciones, entre ellas CPython, Jython,

IronPython y PyPy. Su licencia es la “Python Software Foundation License”. Es relativamente cercana a la licencia BSD y compatible con la licencia GPL. Su sintaxis es minimalista, explicita, clara, sencilla y lo suficientemente cercana al lenguaje natural como para permitir que un algoritmo se comprenda tras su primera lectura. Una de las ventajas de este lenguaje es que la elaboración de una reflexión, de un algoritmo compuesto por palabras, se declina de forma prácticamente natural (Chazallet, 2016).

Python es comúnmente aplicado en muchos dominios de desarrollo. Por ejemplo, se pueden encontrar herramientas que permiten usar Python para proyectos de inteligencia artificial, tales como: TensorFlow, Scikit-Learn, PyBrain, y entre otras distribuciones (Lutz, 2013). Por las razones expuestas y como se verá en secciones posteriores, el proyecto será desarrollado en su totalidad en este lenguaje.

2.7.1. Bibliotecas de desarrollo de aplicaciones de inteligencia artificial.

En este apartado se detallan de manera breve las bibliotecas de inteligencia artificial empleadas en el desarrollo e implementación de las soluciones de aprendizaje automático y aprendizaje profundo involucradas en el proyecto.

2.7.1.1. TensorFlow.

TensorFlow fue desarrollado originalmente por el equipo de Google Brain y es una biblioteca de código abierto para aprendizaje automático, capaz de construir y entrenar redes neuronales muy grandes eficientemente, para detectar y descifrar patrones, además de correlaciones análogos al aprendizaje y razonamiento usados por los humanos. TensorFlow se escribe con una API de Python sobre un motor C / C ++ para cálculos numéricos utilizando gráficos de flujo de datos. Los nodos del gráfico representan operaciones matemáticas y los bordes

representan tensores. TensorFlow admite múltiples backends (bases de datos), CPU o GPU en plataformas de escritorio, servidor o móviles. Tiene enlaces bien soportados a Python y C ++. TensorFlow también tiene herramientas para apoyar el aprendizaje por refuerzo (Aurélien , 2017).

2.7.1.2. Scikit-Learn.

Scikit-Learn es una biblioteca de código abierto diseñada para el lenguaje de programación Python muy fácil de usar. Tiene implementado algoritmos de aprendizaje automático de manera eficiente por lo que es un buen punto de partida para implementar algoritmos, herramientas de evaluación y selección de modelos. Además, está construido sobre NumPy, SciPy y MatplotLib (Aurélien , 2017) (Buitink, y otros, 2013).

2.8. Clasificación de métodos de detección y reconocimiento facial

En este apartado se tratará de manera resumida los métodos de vanguardia en tareas de detección y reconocimiento facial hasta la fecha, para así lograr obtener una comprensión más profunda de este campo de aplicación de la visión artificial y determinar el mejor método a emplear en el proyecto.

2.8.1. Detección facial.

La detección facial es un campo de mucho interés en tareas de visión artificial, por lo que muchos enfoques se han dado para resolver esta tarea hasta el momento. Sin embargo, en esta sección se mencionará solo dos de los más importantes hasta la actualidad, el método de Viola & Jones (Viola & Jones, 2001) & (Viola & Jones, 2004) y las CNN´s en cascada (Zhang, Zhang, Li, & Qiao, 2015). La naturaleza de este proyecto hace imprescindible decidir el mejor método que se adecue al entorno en el que se desplegará la aplicación. Es por eso que en las secciones posteriores se expondrán las razones por las cuales usar CNN´s es la mejor opción en nuestro caso. El

algoritmo de Viola & Jones ha sido utilizado exhaustivamente en muchas aplicaciones, pero a través de extensas investigaciones se indica que este tipo de detector puede degradarse significativamente en aplicaciones del mundo real con variaciones visuales más grandes de rostros humanos, incluso con características y clasificadores más avanzados, lo cual no sucede con enfoques de CNN´s (Zhang, y otros, 2017).

2.8.1.1. Viola & Jones.

Viola & Jones (2001) es el primer método de detección de rostros en tiempo real desarrollado por estudiantes de la Universidad de Cambridge, el resultado de esta investigación es el método de clasificación en cascada que puede ser entrenado para identificar muchos tipos de objetos rígidos, sin embargo, el principal enfoque es para la detección facial. Este detector se fundamenta en tres conceptos clave: el primero de ellos es conocido como "Imagen Integral", que permite que las características tipo *"Haar"* utilizadas por este detector se puedan computar muy rápido. El segundo es un algoritmo de aprendizaje automático, *"Adaboost"*, que selecciona sólo las características importantes de todo el conjunto. El tercer concepto es la creación de una estructura en "cascada", la combinación de clasificadores complejos, que rechaza el fondo de la imagen de entrada pasando más tiempo de cálculo en las áreas que puedan contener el objeto de interés (Viola & Jones, 2004).

Específicamente, el método de clasificación en cascada es un algoritmo de detección de rostros basado en características tipo *"Haar"*, donde la suma del valor de píxel de la región blanca menos la suma del valor de píxel de la región negra puede entregarnos el valor característico del rectángulo de la característica, que se utiliza como base de la detección facial (Figura 16).

Figura 16. Rectángulos de características tipo Haar.

Fuente: Adaptado de (Meng, Shengbing, Yi, & Meng, 2014)

La imagen integral se utiliza para calcular el valor característico de los rectángulos de la característica y se puede calcular a partir de una imagen mediante unas pocas operaciones por píxel. Cada cálculo podría simplificarse como un tiempo fijo de operaciones de suma o resta. La Ecuación 1 y la Ecuación 2 muestran cómo obtener la integral (x, y) del punto $ii(x, y)$ y el valor de píxel de la región rectangular S. $i(x', y')$ es el valor de píxel de la imagen del punto (x', y') , (x_A, y_A), (x_B, y_B), (x_C, y_C) y (x_D, y_D) son las coordenadas de la esquina superior izquierda, superior derecha, inferior izquierda y derecha de la región rectangular, respectivamente.

$$\boldsymbol{ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')}$$

Ecuación 1. Ecuación de la Imagen Integral.

Fuente: Adaptado de (Viola & Jones, 2004)

$$\boldsymbol{S = ii(x_D, y_D) - ii(x_C, y_C) - ii(x_B, y_B) + ii(x_A, y_A)}$$

Ecuación 2. Ecuación del valor de pixel S.

Fuente: Adaptado de (Meng, Shengbing, Yi, & Meng, 2014)

Al aprender del conjunto de entrenamiento dado y del rectángulo de características, podemos generar una función clasificadora $f = (X, \theta)$, donde X es el rectángulo de características, θ es el valor de umbral. El valor de retorno de la función f está determinado por la relación de X y θ. El algoritmo de AdaBoost se utiliza para seleccionar los rectángulos que cubren las

características faciales. Esos rectángulos de características forman un clasificador débil y un grupo de clasificadores débiles crea un clasificador fuerte. Después de concatenar el clasificador se obtiene un clasificador en cascada que puede juzgar si la ventana de escaneo contiene información facial (Meng, Shengbing, Yi, & Meng, 2014). Los resultados de detección facial mediante este enfoque se muestran en la Figura 17.

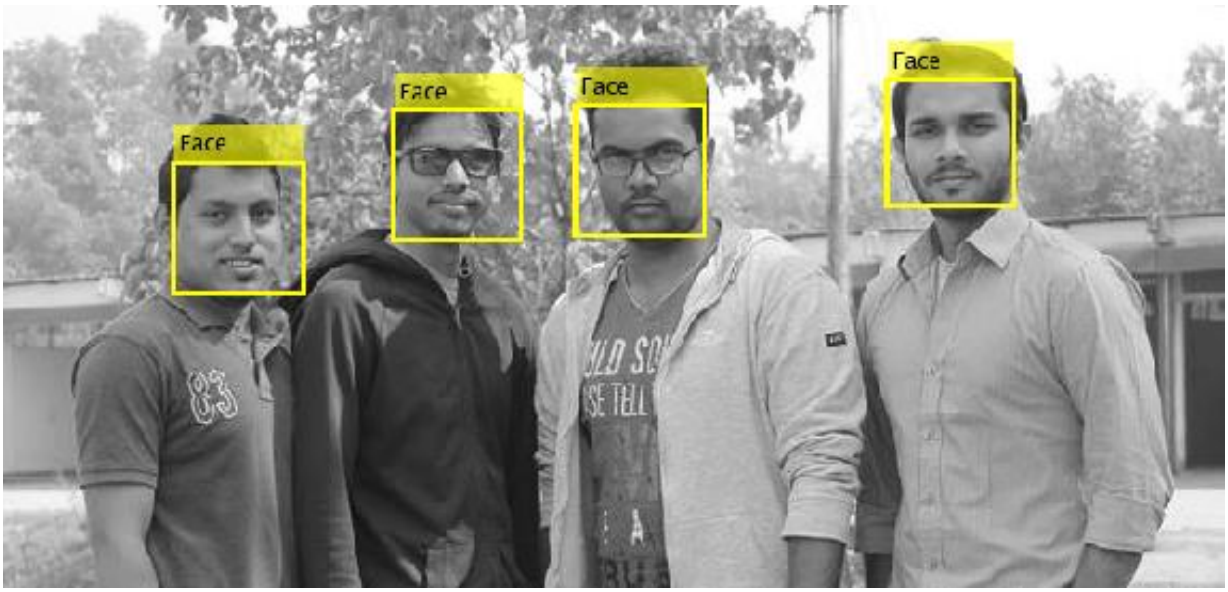


Figura 17. Resultados del detector Viola & Jones.

Fuente: Adaptado de (Atiqur, 2015)

2.8.1.2. CNN´s en cascada.

Las CNN´s han logrado progresos notables en una variedad de tareas de visión artificial, como la clasificación de imágenes, detección y reconocimiento facial. Este enfoque de aprendizaje profundo logra un rendimiento impresionante en tareas de detección facial en aplicaciones del mundo real en un entorno sin restricciones, donde se enfrentan diferentes desafíos como las variaciones de pose, iluminación y oclusiones (Zhang, y otros, 2017).

En la investigación de Zhang (2015) sobre detección conjunta y alineación de rostros usando CNN´s se propone una arquitectura en cascada con tres etapas de redes convolucionales profundas cuidadosamente diseñadas para predecir la ubicación de la cara y la localización de puntos de referencia. En la primera etapa, se producen ventanas faciales candidatas rápidamente a través de una CNN superficial llamada red de propuesta (P-Net). Luego, refina las ventanas rechazando una gran cantidad de ventanas faciales que no son caras a través de una CNN más compleja llamada red de refinamiento (R-Net). Finalmente, utiliza una CNN más poderosa llamada red de salida (O-Net), para refinar el resultado nuevamente, identificar regiones faciales con más supervisión y generar cinco posiciones de puntos faciales.

2.8.1.2.1. *Arquitectura CNN.*

En la Figura 18 se puede visualizar la+ distribución secuencial de las tres etapas de CNN´s mencionadas anteriormente. Esta arquitectura está conformada por capas convolucionales con imágenes de entrada de entrenamiento de dimensiones $12x12x3$ con filtro de $3x3$ y una profundidad de 10 capas, luego se definen capas pooling en su variación de operación max pooling con un tamaño de ventana de $2x2$. La razón de usar el tamaño de filtro de $3x3$ en las redes P-Net, R-Net, y O-Net es para reducir la capacidad de computación y aumentar la profundidad para obtener un mejor rendimiento. Se aplica PReLU (*Parametric-Rectified-Linear-Unit*) como función de activación de no linealidad después de las capas de convolución y de conexión completa. Al final de la arquitectura en cascada se obtiene una salida que determina si la CNN ha detectado uno o varios rostros mediante el entrenamiento de la red usando un enfoque de clasificación binaria (rostro/no rostro), luego se determinan 4 puntos de regresión de cuadro delimitador con más probabilidad de que se encuentre uno o varios rostros, finalmente se entregan 5 puntos de referencia faciales para cada cuadro (Zhang, Zhang, Li, & Qiao, 2015).

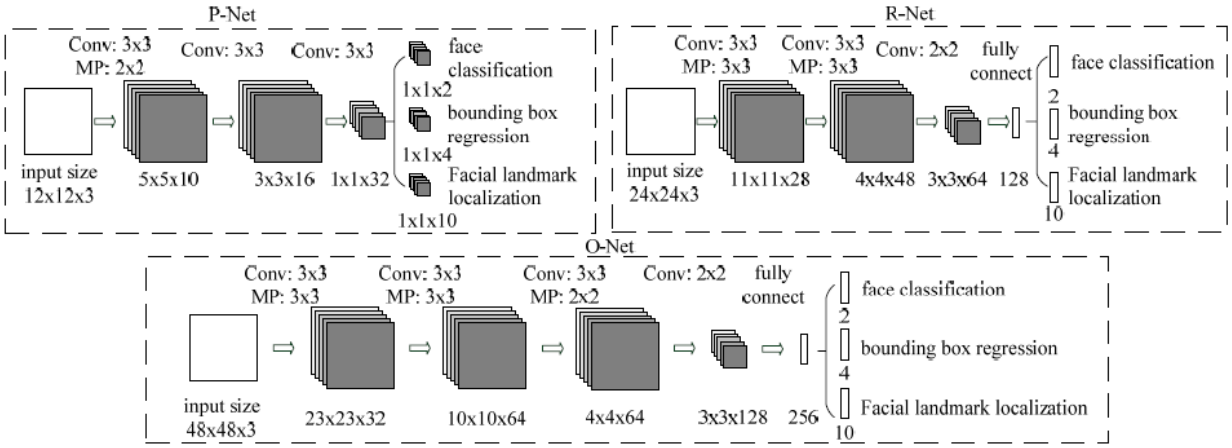


Figura 18. Arquitectura CNN: P-Net, R-Net y O-Net.

Fuente: Adaptado de (Zhang, Zhang, Li, & Qiao, 2015)

2.8.1.2.2. *Entrenamiento.*

Se provechan tres tareas para entrenar los detectores CNN: clasificación rostro/no rostro, regresión de cuadro delimitador y localización de puntos faciales. Su entrenamiento es llevado a cabo por los siguientes métodos.

1) Clasificación de la cara: el objetivo de aprendizaje está formulado como un problema de clasificación de dos clases. Para cada muestra x_i, se emplea la pérdida de entropía cruzada dada por la Ecuación 3:

$$L_i^{det} = -(y_i^{det} log(p_i) + (1 - y_i^{det})(1 - log(p_i)))$$

Ecuación 3. Función de pérdida de entropía cruzada (Cross Entropy)

Fuente: Adaptado de (Zhang, Zhang, Li, & Qiao, 2015) & (Zhang, y otros, 2017)

Donde p_i es la probabilidad producida por la red que indica que la muestra x_i es una cara. La notación $y_i^{det} \in \{0,1\}$ denota la etiqueta de *verdad fundamental (ground thruth).*

2) Regresión del cuadro delimitador: Para cada ventana candidata, predecimos el desplazamiento entre ella y la verdad del terreno más cercana (es decir, los cuadros delimitadores a la izquierda, arriba, alto y ancho). El objetivo de aprendizaje se formula como un problema de regresión, y empleamos la pérdida euclidiana para cada muestra x_i, tal y como se muestra en la Ecuación 4:

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

Ecuación 4. Función de pérdida Euclidiana de cuadro delimitador.

Fuente: Adaptado de (Zhang, Zhang, Li, & Qiao, 2015)

Donde $\hat{y}_i^{box}$ es el objetivo de regresión obtenido de la red y y_i^{box} es la coordenada de la verdad del terreno. Hay cuatro coordenadas, que incluyen la parte superior izquierda, la altura y el ancho, y por lo tanto $y_i^{box} \in \mathbb{R}^4$.

3) Localización de puntos faciales: similar a la tarea de regresión de cuadro delimitador, la detección de puntos faciales se formula como un problema de regresión y minimizamos la pérdida euclidiana, tal y como se muestra en la Ecuación 5:

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

Ecuación 5. Función de pérdida Euclidiana de puntos de referencia faciales.

Fuente: Adaptado de (Zhang, Zhang, Li, & Qiao, 2015)

Donde $\hat{y}_i^{landmark}$ son las coordenadas del punto facial obtenidas de la red y $y_i^{landmark}$ es la coordenada de la verdad del terreno para la muestra i-ésima. Hay cinco puntos de referencia faciales, que incluyen: el ojo izquierdo, el ojo derecho, la nariz, la esquina de la boca izquierda y la esquina de la boca derecha, y así $y_i^{landmark} \in \mathbb{R}^{10}$.

Finalmente, hay que mencionar que todo el proceso de aprendizaje se realiza empleando el algoritmo de optimización de descenso de gradiente estocástico (SGD) (Zhang, Zhang, Li, & Qiao, 2015). Además, dada la complejidad de las CNN´s y la cantidad de ejemplos de entrenamiento se emplea unidades de procesamiento grafico (GPU) para disminuir el tiempo de entrenamiento (Nvidia High Performance Computing, 2019) (Nvidia Developer, 2019). En la Figura 19 se puede visualizar la efectividad de este detector de rostros con un alto índice de aciertos y un porcentaje de falsos positivos casi nulo en escenarios no controlados con iluminación variable.

Figura 19. Resultados del detector en cascada CNN.

Fuente: Adaptado de (Zhang, Zhang, Li, & Qiao, 2015)

2.8.2. Reconocimiento Facial.

El reconocimiento facial es sin duda el campo de interés más significativo de la visión artificial desde hace más de treinta años, con un número de artículos que duplica al de la detección de rostros y multiplica por diez al de seguimiento (García Mateos, 2007). Es por esto que es una labor difícil sintetizar con precisión el ingente número de métodos y/o publicaciones en la comunidad científica, y pretender cubrir de forma exhaustiva las investigaciones existentes se sale claramente del objetivo de este proyecto. Sin embargo, se realizará un repaso somero de los trabajos más relevantes hasta la fecha y con más impacto, tales como: *Eigenfaces (basado en PCA), Fisherfaces (basado en LDA), Local Binary Patterns Histograms (LBPH), Elastic Bunch Graph Matching (EBGM) y Hidden Markov Models (HMM).*

Finalmente, se ahondará la investigación al enfoque más prometedor y con mejores resultados en la literatura, el uso de CNN´s para la generación de un modelo de incrustaciones faciales. En la Figura 20 se distinguen resumidamente las categorías de técnicas de reconocimiento facial.

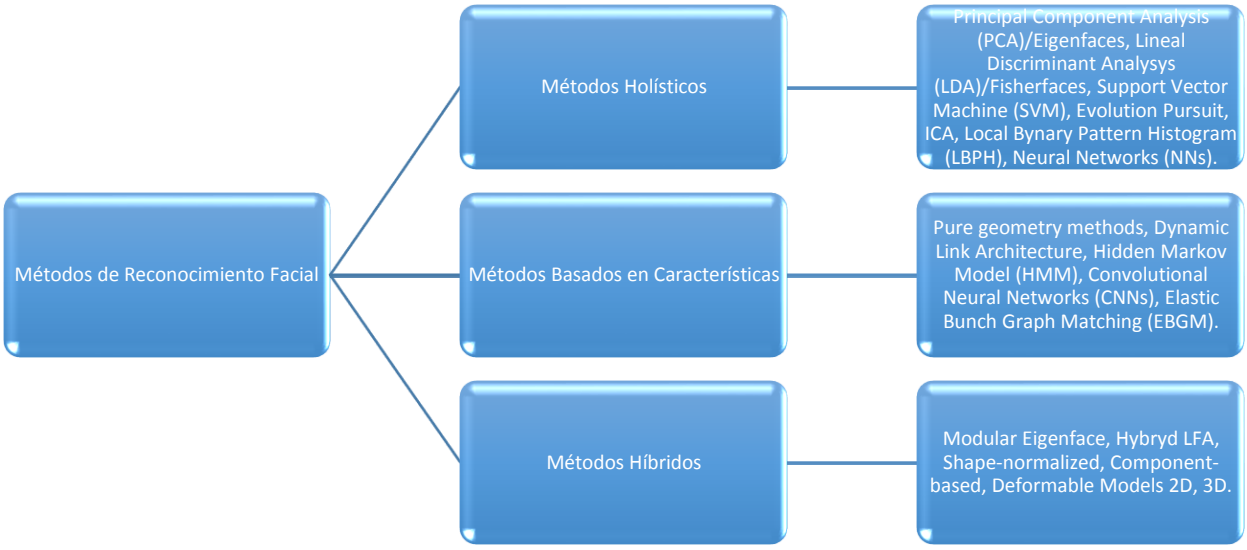


Figura 20. Clasificación de métodos de reconocimiento facial.

Fuente: Adaptado de (García Mateos, 2007) & (Alvear Puertas, 2016)

Tal y como se mencionó en la sección 2.1.2, cada categoría aborda la tarea de reconocimiento en función de cómo las imágenes se traten de forma global o por partes. Es así que los métodos holísticos, basados en características e híbridos contienen cada uno un abanico de técnicas diferentes.

- Métodos Holísticos. La cara se representa como un todo, sin distinguir explícitamente partes dentro de la misma.
- Métodos basados en Características. El reconocimiento se realiza en base a características extraídas de la cara o de partes de la misma (de los componentes faciales como boca, ojos, nariz, etc.).
- Métodos Híbridos. Incorporan al mismo tiempo información global de la cara y propiedades asociadas a los elementos faciales.

2.8.2.1. Eigenfaces (PCA).

A finales de 1980, se comprobó la capacidad de obtener buenas reconstrucciones de las caras en espacios de reducida dimensionalidad, usando *análisis de componentes principales* (*PCA*). El método planteado es bastante elemental. A partir de las imágenes de la galería, se obtienen los *autovectores* o *eigenfaces* de la matriz de covarianzas que denominan las *autocaras* (Figura 21), descartando los de menor *autovalor* asociado. Los ejemplos de la galería se representan como puntos en ese autoespacio, dados por la proyección en la base de autocaras. Dada una imagen nueva, se proyecta igualmente en el espacio de caras. La clasificación se realiza por simple distancia euclidiana mínima en el autoespacio y de esa manera se obtiene la cara positivamente. Sin embargo, esta técnica es débil ante cambios de iluminación (García Mateos, 2007).

Figura 21. Eigenfaces de un conjunto de imágenes.

Fuente: Adaptado de (Domínguez Pavón , 2017)

2.8.2.2.Fisherfaces (LDA).

Esta técnica considera las imágenes de entrenamiento de un mismo individuo como clases (Figura 22), por lo tanto, existen el mismo número de clases que personas. Una vez definida las clases se procede a calcular dos matrices: la matriz de dispersión entre clases y la matriz de dispersión dentro de clases. Una vez calculada estas matrices se obtiene una matriz de proyección donde cada columna será la base del nuevo sub-espacio, denominada *Fisherfaces*. Con esta técnica

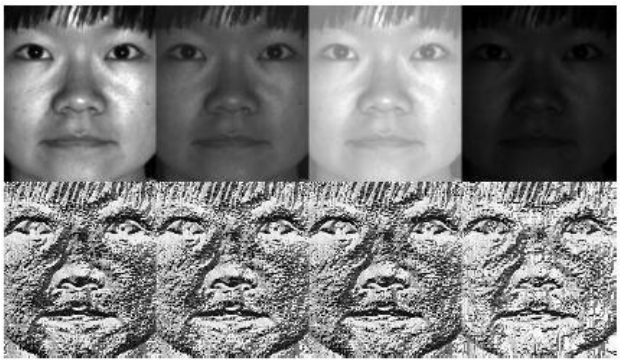
se consigue mayor robustez frente a cambios de iluminación, pero resulta computacionalmente más costosa (Espinoza Olguín & Jorquera Guillen, 2015).

Figura 22. Fisherfaces de un conjunto de imágenes.

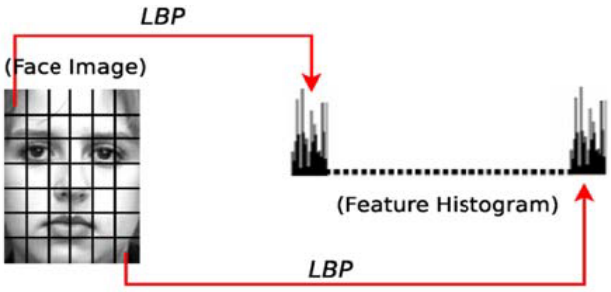
Fuente: Adaptado de (Domínguez Pavón , 2017)

2.8.2.3. Local Binary Patterns (LBP).

El algoritmo *LBP* es conocido como un buen descriptor de texturas a nivel local, utilizado en muchas aplicaciones de tratamiento de imágenes y reconocimiento de patrones. LBP etiqueta cada píxel de la imagen de acuerdo a los valores de sus píxeles vecinos. Para esto, se define un grado de vecindad y se les da el valor de 1 o 0 a estos píxeles según su nivel de intensidad sea mayor o menor que el valor del píxel central. A continuación, se recorren los vecinos y se genera una etiqueta binaria para el píxel central. Este proceso se repite sucesivamente para todos los píxeles de la imagen (Espinoza Olguín & Jorquera Guillen, 2015). En un enfoque para el reconocimiento facial, se trabaja con la imagen del rostro dividida en regiones (sub-imágenes). Con el operador LBP se codifica cada píxel de la sub-imagen y se recogen en un histograma regional. Posteriormente, se concatenan todos los histogramas regionales en un solo histograma global, para obtener una representación de la cara (Figura 23).



(a)



(b)

Figura 23. Definición de LBP. a) LBP sobre imágenes con diferente intensidad, b) Descripción del rostro mediante un histograma de características LBP.

Fuente: Adaptado de (Huang, Shan, Ardebilian, Wang, & Chen, 2011)

2.8.2.4. Elastic Bunch Graph Matching (EBGM).

EBGM es una técnica que fue diseñada para realizar tareas de reconocimiento facial utilizando solamente algunos puntos de interés y no la cara en su totalidad, aprovechando la estructura topológica similar (simétrica) que estas presentan. La implementación de este algoritmo requiere el uso de las *wavelets de Gabor*, que son filtros paso banda que permiten alcanzar la resolución conjunta de información máxima en los espacios bidimensionales espacial y frecuencial (Domínguez Pavón , 2017). Básicamente este algoritmo sigue los siguientes pasos:

- Primeramente, el algoritmo trata la extracción de las características locales. Para ello se define una estructura de grafo sobre la cara (Figura 24), cuyos nodos son puntos de interés que se puedan localizar fácilmente y que posean la misma estructura en todos los rostros, además, estos puntos deben mantener la simetría del rostro.
- Posteriormente, cada nodo del grafo es caracterizado utilizando un banco de filtros de Gabor de diferentes frecuencias y orientaciones. En cada nodo, se calcula la respuesta de todos los filtros, a lo que se le da el nombre de Jet. Finalmente, para una imagen nueva se busca en la base de datos el conjunto de jets que sean más similares.

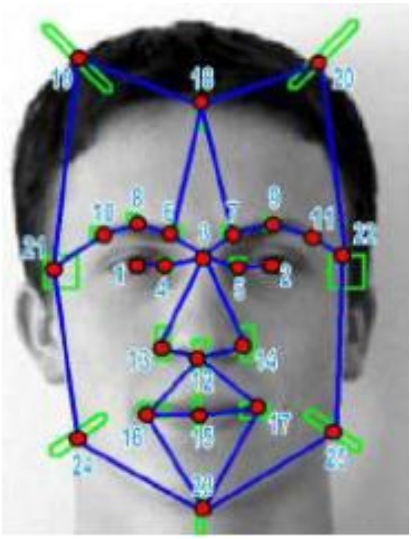


Figura 24. Rostro representado como grafo mediante EBGM.

Fuente: Adaptado de (Domínguez Pavón , 2017)

2.8.2.5. Hidden Markov Models (HMM).

Los denominados *HMM's* también han sido usados con éxito para el reconocimiento facial. Estas técnicas presentan robustez frente a cambios de iluminación, expresión y orientación,

otorgando así una ventaja frente a los métodos holísticos. Las técnicas basadas en HMM utilizan regiones horizontales de píxeles (Figura 25) que albergan a la frente, ojos, nariz, boca y barbilla sin obtener la posición exacta de cada rasgo. Cada una de estas regiones es asignada a un estado del HMM para el reconocimiento. Este método reduce significativamente la complejidad computacional.

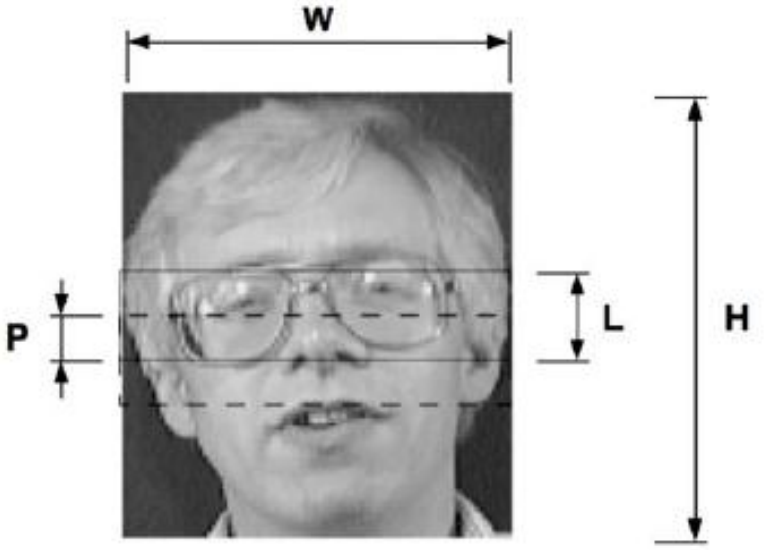


Figura 25. Parámetros y extracción de bloques de imagen facial con HMM.

Fuente: Adaptado de (Nefian & Hayes, 1998)

2.8.2.6.CNN´s.

A pesar de los importantes avances recientes en el campo del reconocimiento facial, la implementación de la verificación facial y el reconocimiento de manera eficiente a escala presenta serios desafíos para los enfoques actuales. A continuación, se presenta un enfoque de reconocimiento de rostros profundo usando una arquitectura CNN denominada *Resnet*. Dicha arquitectura es entrenada bajo la supervisión conjunta de las señales de *pérdida Softmax y pérdida Central* para el aprendizaje de características profundamente discriminativas.

2.8.2.6.1. *Arquitectura Resnet.*

La arquitectura de convolución profunda Inception se introdujo como GoogLeNet en sus inicios en el año 2012. A partir de este diseño, en el año 2015 surgió el modelo Inception Resnet bajo el principio de conexiones residuales. La característica destacada de esta arquitectura de red residual es la identidad que omite las conexiones en los bloques residuales, lo que permite capacitar fácilmente arquitecturas CNN muy profundas y obtener una mejor precisión. Debido a que existen diferentes variaciones del modelo Inception Resnet nos enfocamos en el modelo Inception Resnet V1 (Figura 26), el cual se compone de 3 principales módulos de cuadricula de tamaño $35x35$, $17x17$ y $8x8$, denominados bloques Inception-A, Inception B e Inception-C respectivamente, cada bloque de la red conlleva operaciones de capa de convolución, capa pooling, función de activación, entre otras (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017).

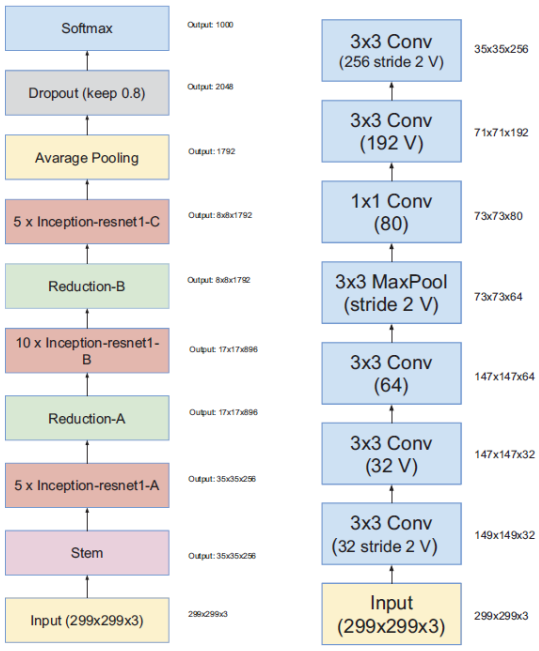


Figura 26. Arquitectura Inception Resnet V1.

Fuente: Adaptado de (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017)

2.8.2.6.2. *Entrenamiento de Inception Resnet V1*.

Para el entrenamiento de la CNN propuesta de manera supervisada se emplea el algoritmo de optimización de descenso de gradiente estocástico (SGD). Además, la formulación matemática de las señales de pérdida mencionadas anteriormente se muestran en la ecuación 6:

$$L = L_S + \lambda L_C = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \left\| x_i - c_{y_i} \right\|_2^2$$

Ecuación 6. Funciones de pérdida Softmax y Central.

Fuente: Adaptado de (Wen, Zhang, Li, & Qiao, 2016)

De la ecuación 6 podemos destacar que existen 2 partes: la primera de ellas es la definición de pérdida Softmax, donde x_i denota la i-ésima característica profunda, que pertenece a la clase y_i, W_j denota la j-ésima columna de los pesos de la red y b es el termino de sesgo o unidad de bias. La segunda parte corresponde a la definición de la pérdida Central, donde c_{y_i} denota el $y_i(i-ésimo)$ centro de clase de característica profunda.

Cabe recalcar algunos aspectos de la necesidad de supervisión conjunta. Si solo usamos la pérdida de Softmax como señal de supervisión, las características resultantes que se aprendieron en profundidad contendrían grandes variaciones dentro de la clase. Por otro lado, si solo supervisamos las CNN por la pérdida Central, las funciones y los centros profundamente aprendidos se degradarán a ceros (en este punto, la pérdida Central es muy pequeña). El simple uso de cualquiera de ellos no puede lograr un aprendizaje discriminativo. Por lo tanto, como se mencionó es necesario combinarlos para supervisar conjuntamente la CNN. Finalmente, se toman las características profundamente aprendidas y se utilizan para las tareas de identificación y verificación mediante la comparación de las distancias entre vecinos más cercanos mediante clasificadores y la comparación de umbrales (Wen, Zhang, Li, & Qiao, 2016). La efectividad de

este método se puede apreciar en la Figura 27, donde la CNN se ha entrenado, dado un conjunto de datos de entrenamiento (conjunto de prueba), luego se realiza un test de identificación en un conjunto diferente con millones de distractores (identidades) y se establece la identidad de mayor correspondencia o cercanía.

Figura 27. Algunos ejemplos de imágenes de rostros, incluidos el conjunto de pruebas y la galería consisten de al menos una imagen correcta de entre millones.

Fuente: Adaptado de (Wen, Zhang, Li, & Qiao, 2016)

2.9. Bases de datos de entrenamiento faciales

Las CNN´s han tomado por asalto la comunidad de visión por computadora, mejorando significativamente el estado del arte en muchas aplicaciones. Uno de los ingredientes más importantes para el éxito de tales métodos es la disponibilidad de grandes cantidades de datos de entrenamiento. El desafío de reconocimiento visual a gran escala de ImageNet (ILSVRC) iniciado en el 2010 fue fundamental para proporcionar estos datos categorizados para la tarea general de clasificación de imágenes. Más recientemente, los investigadores han puesto a disposición conjuntos de datos para la segmentación, clasificación de escenas y segmentación de imágenes

(Parkhi, Vedaldi, & Zisserman, 2015). En los últimos años los conjuntos de datos faciales empezaron a surgir y también mejoraron las tasas de acierto en la detección y el reconocimiento facial usando CNN´s, por lo que no hay duda alguna de que un conjunto de datos de entrenamiento robusto mejora casi cualquier tarea de visión por computadora. Es por eso que esta sección se aborda algunas de las bases de datos de entrenamiento más importantes orientadas a la detección y reconocimiento de rostros.

2.9.1. LFW.

Labeled Faces in the Wild (*LFW*) fue lanzado en un esfuerzo por estimular la investigación en el reconocimiento facial, específicamente para el problema de la verificación facial con imágenes de rostros sin restricciones. LFW es una base de datos de fotografías de rostros que contiene más de 13,000 imágenes de rostros recolectadas de la web pertenecientes a 5000 individuos. Cada rostro ha sido etiquetado con el nombre de la persona fotografiada, además algunas personas tienen dos o más fotos distintas dentro del conjunto de datos. La única restricción en estas caras es que fueron detectadas por el detector facial Viola-Jones (Learned Miller, Huang, RoyChowdhury, Li, & Hua, 2016).

2.9.2. VGGFace2.

Visual Geometry Group Face 2 en sus siglas es un nuevo conjunto de datos a gran escala. El conjunto de datos contiene 3,31 millones de imágenes de rostros de 9131 sujetos, con un promedio de 362.6 imágenes para cada sujeto. Las imágenes se descargan de la búsqueda de imágenes de Google y tienen grandes variaciones en la postura, la edad, la iluminación, el origen étnico y la profesión (por ejemplo, actores, atletas, políticos). El conjunto de datos se recopiló con tres objetivos en mente: (1) tener tanto un gran número de identidades como un gran número de

imágenes para cada identidad; (2) para cubrir una gran variedad de posturas, edades y etnias; y (3) minimizar el ruido de la etiqueta. De hecho, este conjunto de datos es el más extenso en la actualidad y es utilizado en muchas investigaciones sobre arquitecturas CNN con la finalidad de construir sistemas de reconocimiento facial eficientes (Cao, Shen, Xie, Parkhi, & Zisserman, 2018).

2.9.3. CASIA-WebFace.

CASIA-WebFace es un conjunto de datos a gran escala con un tamaño de 500.000 imágenes, que incluye aproximadamente las identidades de 10,000 sujetos con una cantidad de imágenes por sujeto variable de 2, 46 y 804. Sus contribuciones adicionales se resumen a continuación: (1) propusieron un canal semiautomático para construir conjuntos de datos de rostros a gran escala desde Internet; (2) capacitaron a una CNN profunda de alto rendimiento para el reconocimiento facial en ambientes no controlados con un rendimiento impresionante (Yi, Lei, Liao, & Li, 2014).

2.10. Metodología de desarrollo de software

El desarrollo de un nuevo proyecto implica esfuerzo físico e intelectual sostenido, con diversas dificultades. Por lo que es necesario la implementación de una metodología que permita guiar a los desarrolladores para el cumplimiento del proyecto; por lo que se define a una metodología como un modo sistemático de realizar, gestionar y administrar un proyecto con altas posibilidades de éxito. A continuación, se indican los principales modelos de desarrollo.

2.10.1. Modelo en V.

El modelo en V fue desarrollado por Alan Davis (año), y permite un trabajo secuencial en fases estrechamente conectadas para el desarrollo de cualquier proyecto con su debida

retroalimentación y documentación adecuada. Este modelo es una variación del modelo en cascada y es útil para proyectos que necesitan alta confiabilidad. En la Figura 35 se muestra el esquema del modelo en V:

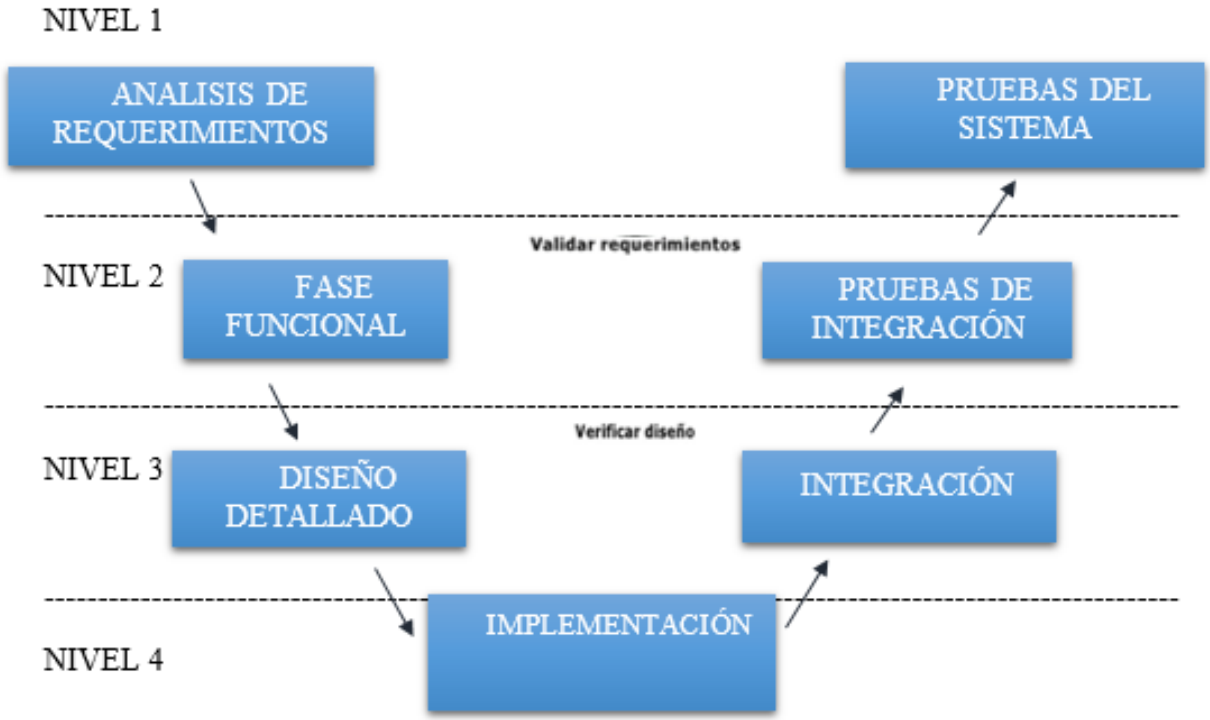


Figura 28. Etapas del modelo en V.

Fuente: Adaptado de (Flores, 2016) & (Casallas & Yie, 2016)

Las funciones que conlleva cada nivel del modelo en V se detallan a continuación:

NIVEL 1.- Orientado al cliente. Se compone del análisis de requisitos y especificaciones.

NIVEL 2.- Se dedica a las características funcionales del sistema propuesto.

NIVEL 3.- Define los componentes de hardware y software del sistema final.

NIVEL 4.- Es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

2.10.2. Modelo Lineal.

Es el modelo más simple de las distintas metodologías, posee las etapas del proyecto divididas y secuenciadas para ser desarrolladas independientemente, ya que no existe retroalimentación es recomendable usar esta metodología en proyectos pequeños donde no se requiere retroalimentación de alguna etapa. En la Figura 36 se muestra el ciclo de vida de este modelo.

Figura 29. Etapas del modelo lineal.

Fuente: Adaptado de (Pérez Montero, 2014)

2.10.3. Modelo en cascada.

El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo del proyecto se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada. En la Figura 37 se muestra las etapas del modelo en cascada mencionadas.

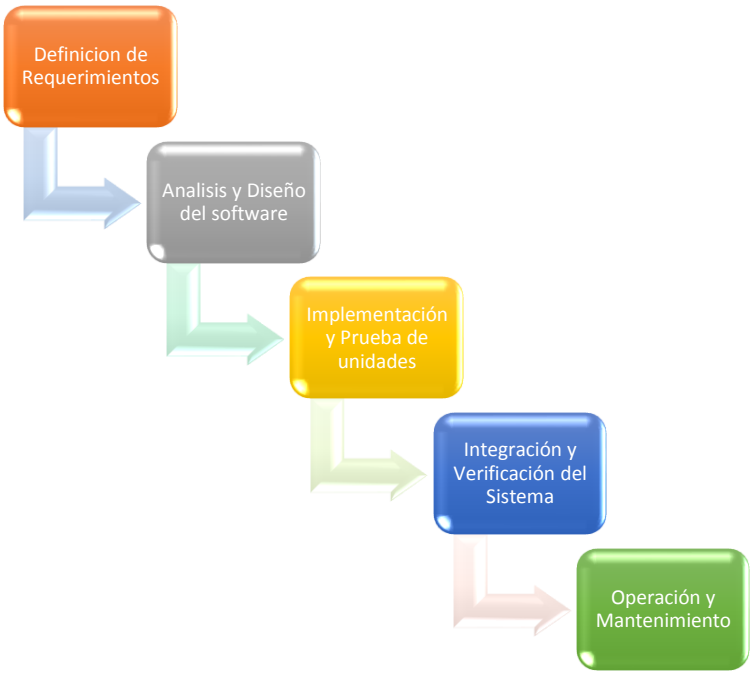


Figura 30. Etapas del modelo en cascada.

Fuente: Adaptado de (Casallas & Yie, 2016)

3. CAPÍTULO III. Desarrollo Experimental

En el presente capítulo se indica el desarrollo experimental del proyecto utilizando la metodología de desarrollo de software basado en el "*Modelo en Cascada*".

3.1. Metodología

Para el desarrollo de esta investigación, se siguió una metodología de desarrollo denominado Modelo en Cascada, el cual cumple en gran manera con el proceso de creación del proyecto, análisis, diseño, codificación, integración, pruebas y mantenimiento del software. De esta manera, se facilita la gestión del desarrollo del proyecto ordenando rigurosamente las etapas del ciclo del software y llevando a cabo sus objetivos fase por fase para poder continuar con la siguiente etapa. Es así que no existe relación o retroalimentación entre etapas en este modelo. Lo cual es perfectamente aplicable a proyectos de pequeña escala. Además, este proyecto, el cual tiene el objetivo de crear un sistema prototipo a través del uso de técnicas de inteligencia artificial de alto nivel, se encuentra orientado en su mayoría al desarrollo de software. Por lo que las fases de análisis, diseño y desarrollo del mismo podrían variar al momento de las pruebas de funcionamiento. Finalmente se concluye en que al ser un proyecto de investigación bastante amplio dentro del campo de la Inteligencia Artificial el impacto debe enfocarse mucho más en la elección de un método eficiente que logre solventar el objetivo del proyecto a través de la revisión de la literatura existente, mas no ofrecer una solución definitiva comercial. En la Figura 31 se muestra el modelo secuencial del Modelo en Cascada. A continuación, se especifica a detalle cada una de sus fases.

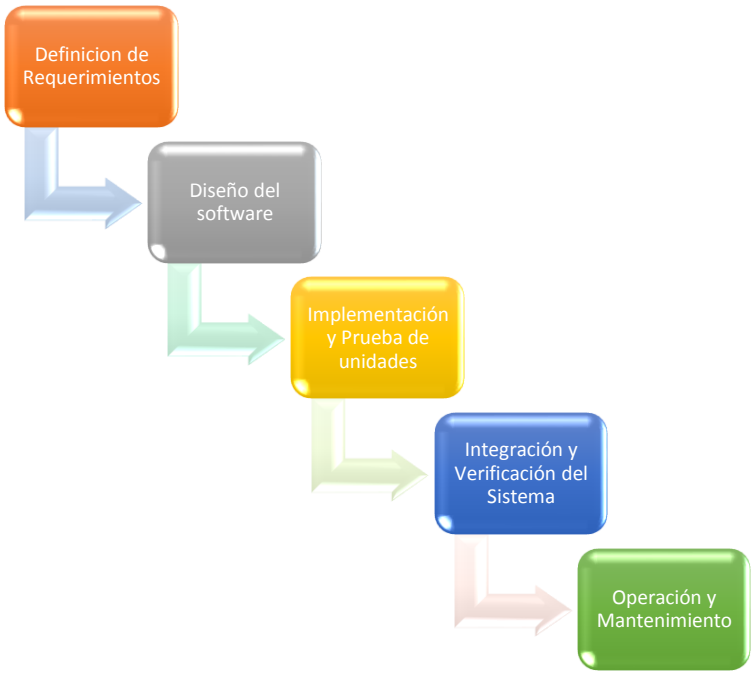


Figura 31. Modelo lineal en cascada.

Fuente: Adaptado de (Casallas & Yie, 2016)

En la etapa de definición de requerimientos del software se reúnen todos los requisitos que debe cumplir el software bajo las necesidades de funcionamiento del cliente. Esta fase es una de las más importantes y definen el éxito del proyecto a desarrollar ya que delimitan la funcionalidad del sistema propuesto.

En la etapa de diseño se traducen los requisitos en una representación de software en donde se enfocan las tareas de estructura de datos, arquitectura del software, representaciones de interfaz y el detalle procedimental o algoritmo.

En la etapa de implementación y pruebas de unidades se emplean las herramientas y soluciones elegidas en el diseño para que sea comprensible por el computador y así obtener un programa ejecutable, esta etapa va a depender estrechamente de lo detallado en la fase del diseño.

La etapa de integración y verificación del sistema está centrada en los procesos lógicos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores mediante pruebas pertinentes.

Finalmente, la etapa de operación y mantenimiento del software conlleva modificaciones o creación de funciones adicionales en el software a lo largo de su vida útil. Esta etapa no ha sido considerada aún en este proyecto.

3.2. Introducción al desarrollo del proyecto

Esta sección presenta los lineamientos principales que rigen el desarrollo del sistema prototipo de reconocimiento facial; los aspectos a cubrir son los siguientes: propósito, beneficiarios y objetivos del sistema.

3.2.1. Propósito del sistema.

El objetivo fundamental en el que se sustenta gran parte de este proyecto es el de diseñar una solución adecuada al problema del reconocimiento facial en entornos no controlados, con el uso de técnicas de vanguardia en el área de la Inteligencia Artificial y la Visión por Computador. Se pretende realizar la identificación automática de las personas que ingresan a las instalaciones de la FICA a lo largo del día y noche, permitiendo así de esa manera diferenciar a personas conocidas y desconocidas dentro de la institución con fines de seguridad. Además, de generar reportes de registro de acceso diario. Para lograr dicho objetivo es imprescindible tomar una muestra parcial de la población total de las personas pertenecientes a la FICA para demostrar el funcionamiento del sistema en la fase de pruebas en un entorno controlado y no controlado. Para la fase de pruebas en un entorno controlado se ha determinado 25 sujetos de prueba, mientras que para la fase de pruebas en un entorno no controlado sin restricciones el tamaño de la muestra se

encuentra definida en la sección 4.1 (Identificación de la población) del Cap. 4. La causa de la selección de una muestra aleatoria de la población se debe en gran medida a las limitaciones de hardware encontradas en el momento de la implementación unificada de las diferentes técnicas de inteligencia artificial seleccionadas para la identificación facial sobre un gran número de personas en tiempo real. Con dichas consideraciones se determinan futuras mejoras en cuanto a la precisión del sistema a medida que este escale en relación a los beneficiarios directos del sistema.

3.2.2. Beneficiarios.

Los beneficiarios del sistema se clasifican en beneficiarios directos e indirectos. Los beneficiarios directos constan de la población de estudiantes, docentes y personal administrativo de la FICA y los beneficiarios indirectos comprenden al resto de la comunidad de la UTN (estudiantes, docentes, personal administrativo, y autoridades de otras facultades).

A continuación, se detallan las características de cada tipo de usuario del sistema:

- Los usuarios directos resultan ser una parte importante del proyecto ya que son los sujetos de prueba en la tarea de reconocimiento facial, los cuales se caracterizan por ser miembros activos de la facultad y están registrados facialmente en el sistema bajo capacitación previa. Además, como usuarios directos se consideran a los sujetos que administran el sistema de cámaras de seguridad CCTV[4] de la universidad, o también los sujetos a cargo de los laboratorios de la FICA, los cuales tendrán acceso al video capturado a través del sistema CCTV en tiempo real por

[4] CCTV: El Circuito Cerrado de Televisión o CCTV es una tecnología de video vigilancia diseñada para supervisar una diversidad de ambientes y/o actividades; está conformada por cámaras análogas/digitales, monitores de video, entre otros dispositivos.

una de las cámaras principales. Dentro del sistema se generarán diariamente los registros de los sujetos identificados como conocidos y desconocidos.

- Los usuarios indirectos del sistema se conforman en general de todos los sujetos que ingresan al edificio de la FICA y han sido detectados e identificados por el sistema como desconocidos y se caracterizan por no estar registrados en el sistema ya sea porque no pertenecen a la facultad, o al resto de la comunidad de la UTN.

En un principio se consideró como beneficiarios directos a toda la comunidad de la FICA, pero debido a la alta cantidad de personas (mayor a 2000 personas) que se deberían registrar en el sistema surgen limitaciones en cuanto a la disponibilidad de tiempo de los sujetos de prueba para efectuar la recolección del conjunto de datos de entrenamiento y las correspondientes pruebas de desempeño de las metodologías implementadas bajo el sistema, además, la ingeniería del hardware tiende a requerir muchos más recursos de cómputo paralelo (GPU´s), lo cual se deriva en altos requerimientos económicos, por lo que se debería realizar inversiones en componentes adicionales. Hay que mencionar que el sistema es escalable, sin embargo, hay que tomar en cuenta los factores mencionados.

3.2.3. Objetivos del sistema.

A lo largo de la investigación desarrollada en base a la literatura existente (artículos científicos, libros, revistas) e implementación de diversas técnicas de inteligencia artificial con especial enfoque en el área de la visión artificial, es conveniente mencionar los resultados o conclusiones finales que permiten desarrollar los criterios finales en los que se consolida el sistema:

- Identificar a las personas que acceden a las instalaciones de la FICA para indicar información de clasificación como sujetos conocidos y desconocidos.
- Implementar algoritmos y/o metodologías de visión artificial diseñadas por la comunidad científica en los últimos años a través del aprendizaje profundo.
- Combinar operaciones de preprocesamiento de la imagen con la finalidad de minimizar factores negativos inherentes al entorno/lugar (como ruido, oclusión, variación de luminosidad), beneficiando a la eficiencia del sistema.
- Ensamblar una estación de trabajo que permita el uso de la potencia de cómputo paralelo de los módulos CPU y GPU para la capacitación de modelos de aprendizaje profundo y automático en tiempos reducidos con miras a la ejecución de la aplicación en tiempo real.
- Presentar la información de identificación facial por parte del sistema mediante reportes.

3.3. Requerimientos del sistema.

Para el análisis de los requerimientos del sistema se tomó como referencia el estándar ISO/ IEC / IEEE 29148: 2011 (ISO/IEC/IEEE, 2011) el mismo que contiene directrices para el proceso relacionado a la ingeniería de requisitos, específicamente ha sido desarrollado para ser implementado en los sistemas y productos de software y servicios a lo largo del ciclo de vida, ya que permite definir cada función que se requiere en el proyecto, las restricciones necesarias, y especificar los requisitos y funciones del sistema.

Las tablas que se muestran a continuación se han diseñado teniendo en cuenta las consideraciones del estándar, éstas contienen los requerimientos iniciales más relevantes del sistema, requerimientos de arquitectura y los requerimientos de stakeholders. El objetivo es

presentar de una manera concisa dicha información la cual permitirá realizar la selección de software, hardware y algunos aspectos específicos para el diseño del sistema. La Tabla 4 muestra los acrónimos empleados para referirse de forma abreviada a cada requerimiento

Tabla 4. Definición de acrónimos.

Acrónimo	Descripción
SySR	Requerimientos iniciales del Sistema
SRSH	Requerimientos de Arquitectura
StSR	Requerimientos de Stakeholders

Fuente: Autoría

El diseño propuesto para cada tabla incluye una columna donde se identifica el número de requerimiento, una columna destinada a la descripción detallada del requerimiento, la siguiente columna está destinada a indicar la prioridad del requerimiento la cual se subdivide en Alta, Media y Baja, esta valoración se puede visualizar en la Tabla 5 y es de suma importancia para la selección del software y hardware. Se incluye también una columna de relación que se utilizará en el caso de que un requerimiento sea totalmente dependiente de otro.

Tabla 5. Prioridad de los Requerimientos del sistema.

Prioridad	Descripción
Alta	Es un requerimiento crítico que debe incluirse durante el desarrollo del sistema. Si no se implementa puede afectar la funcionalidad.
Media	El no incluir este tipo de requerimiento puede afectar la decisión final del sistema, sin embargo, se puede omitir este requerimiento en condiciones de fuerza mayor.
Baja	Si no se incluye este requerimiento no se espera un impacto significante en la decisión final del sistema.

Fuente: Adaptado de (Pérez García, 2013)

3.3.1. Requerimientos iniciales del sistema.

En los requerimientos iniciales del sistema (SySR) se definen los limites funcionales del sistema en términos de comportamiento y propiedades del proyecto, que constituyen la descripción de los requerimientos de interfaces, de performance, de modos y estados, y requerimientos físicos. A continuación, se describen en la Tabla 6 los requerimientos iniciales del sistema (SySR).

Tabla 6. Requerimientos Iniciales del Sistema.

SySR					
REQUERIMIENTOS INICIALES DEL SISTEMA					
#	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
	REQUERIMIENTO DE INTERFAZ				
SySR1	El sistema deberá ejecutar la detección y reconocimiento de rostros en tiempo real.	X			
SySR2	El registro de identidad debe guardarse en tiempo real.	X			
SySR3	El sistema debe interactuar con una cámara del sistema CCTV a través de una conexión cableada tipo ethernet.	X			
SySR4	El sistema debe interactuar con una GPU(Unidad de Procesamiento Gráfico).	X			
SySR5	El sistema requiere conexión a la red eléctrica.	X			
	REQUERIMIENTO DE PERFORMANCE				
SySR6	Reconocimiento facial para la identificación y registro de personas conocidas	X			
SySR7	Reconocimiento facial para la identificación y registro de personas desconocidas.	X			
SySR8	Detección y reconocimiento facial en el entorno no controlado de la FICA.	X			
SySR9	Obtención de video en tiempo real de la cámara de seguridad del sistema CCTV.	X			
	REQUERIMIENTO DE MODOS/ESTADOS				
SySR10	El sistema debe permanecer activo y procesando el video obtenido de la cámara.	X			
	REQUERIMIENTO FISICOS				
SySR11	El sistema debe estar situado correctamente en un lugar donde no interfiera con la actividades de las personas y esté conectado a la red del sistema de seguridad directamente.	X			
SySR12	La cámara del sistema de seguridad debe estar colocada en un lugar estratégico donde se enfoque directamente al ingreso de las personas y donde aspectos como la oclusión, luminosidad, y ruido en la imagen sean reducidos.	X			
SySR13	La cámara del sistema de seguridad debe estar situada a una altura apropiada a la estatura promedio de los estudiantes de la FICA donde se pueda capturar el rostro de las personas de manera adecuada para realizar una correcta identificación facial.	X			

Fuente: Autoría

3.3.2. Requerimientos de arquitectura.

En los requerimientos de arquitectura (SRSH) se definen los requerimientos de hardware, software y el sistema eléctrico. A continuación, las directrices descritas en los requerimientos de Arquitectura se muestran en la Tabla 7 y son necesarias para la selección del hardware y software a emplear en el proyecto.

Tabla 7. Requerimientos de Arquitectura.

SRSH					
REQUERIMIENTOS DE ARQUITECTURA					
#	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
	REQUERIMIENTO DE DISEÑO				
SRSH1	La cámara debe estar empotrada sobre uno de los muros de la edificación enfocando correctamente al lugar donde la obtención de rostros se realizará.	X			SRSH2
SRSH2	El cable de conexión tipo ethernet debe ser extendido de manera correcta por el sistema de cableado estructurado de la FICA.		X		SRSH1
SRSH3	El servidor que aloja el sistema debe estar situado en un lugar espacioso y con ventilación suficiente.		X		
	REQUERIMIENTO DE HARDWARE				
SRSH4	El sistema requiere una unidad central de procesamiento (CPU) que permita el tratamiento de imágenes en tiempo real.		X		
SRSH5	El sistema requiere una unidad de procesamiento grafico (GPU) que permita el tratamiento de imágenes en tiempo real.	X			
SRSH6	Se requiere una fuente de poder mayor a 500 Watts para el abastecimiento de energía necesario por parte de la todos los componentes del servidor (placa madre, RAM, CPU, GPU, disipadores, etc.).	X			
SRSH7	Se requiere una fuente de poder con certificación 80 Plus	X			
SRSH8	Se requiere una placa madre con conexión vía puerto Ethernet para la recepción del flujo de video proveniente de la cámara IP.	X			
SRSH9	El sistema requiere una GPU que permita el entrenamiento de modelos de aprendizaje automático y profundo.	X			
SRSH10	El sistema requiere una GPU con tecnología de procesamiento de cómputo paralelo CUDA[5].	X			

[5] CUDA: Es una arquitectura de cálculo paralelo de NVIDIA que aprovecha la gran potencia de las GPU´s para proporcionar un incremento extraordinario del rendimiento del sistema encontrando innumerables aplicaciones prácticas en campos como el procesamiento de vídeo e imágenes, la biología y la química computacional, la simulación de la dinámica de fluidos, el análisis sísmico o el trazado de rayos (RTX), entre otras.

SRSH11	Se requiere de una cámara de alta resolución para la efectividad de las tareas de identificación facial (1080p o 720p).	X	
SRSH12	Las imágenes de la cámara deben ser procesadas con rapidez en la GPU.	X	
SRSH13	Se requiere disco duro de almacenamiento de gran capacidad para el almacenamiento de los registros.		X
SRSH14	El procesador del servidor debe ser compatible con la arquitectura de cualquier sistema operativo.	X	
SRSH15	El procesador debe ser de última generación.		X
SRSH16	El procesador debe ofrecer una potencia promedio de procesamiento de 2.4 Ghz o mayor.	X	
SRSH17	El procesador debe tener una tarjeta de gráficos integrados sobre el mismo chip		X
SRSH18	Se requiere una cámara con una conexión vía puerto Ethernet y que ofrezca transmisión a través del protocolo IP.	X	
	REQUERIMIENTO DE SOFTWARE		
SRSH19	Se requiere de un sistema operativo y lenguaje de programación de código abierto.	X	
SRSH20	Se requiere compatibilidad con la librería OpenCV y la cámara.	X	
SRSH21	Se requiere que el software permita ejecutar el código de visión artificial en tiempo real en el servidor.	X	
SRSH22	Se requiere de un sistema operativo que ejecute con rapidez los hilos de procesamiento del sistema.	X	
SRSH23	Se requiere software que permita usar de manera dinámica los recursos de la GPU en la capacitación de modelos de aprendizaje automático y profundo.	X	
SRSH24	Se requiere de un software de base de datos no relacional que permita obtener un bajo impacto o costo en el rendimiento del sistema.	X	
SRSH25	Se requiere de un software o kit de herramientas de diseño de interfaces de usuario (GUI) para la visualización de resultados.		X
SRSH26	Se requiere compatibilidad de software con bibliotecas de aprendizaje automático y profundo.	X	
SRSH27	Se requiere compatibilidad de software con tecnología de cómputo paralelo CUDA.	X	
	REQUERIMIENTO ELECTRICOS		
SRSH28	El sistema debe estar conectado a la red eléctrica de forma permanente.	X	

Fuente: Autoría

3.3.3. Requerimientos de stakeholders.

Los requerimientos de stakeholders comprenden a un grupo o individuo que tiene un interés directo en el resultado obtenido por el desarrollo del proyecto. La definición de los requerimientos de stakeholders (StSR) tiene como finalidad identificar los requisitos de los interesados por el sistema. Específicamente se analizan un conjunto de requerimientos operacionales y de usuario que tienen que ver con la interacción directa de los usuarios involucrados con el sistema. La Tabla 8 muestra los requisitos empleados en los requerimientos de stakeholders.

Tabla 8. Lista de Stakeholders del sistema.

Lista de Stakeholders
1. Estudiantes y docentes de la FICA
2. Personal Administrativo de la FICA
3. PhD. Iván García (Director del trabajo de titulación)
4. MsC. Paúl Rosero (CoDirector del trabajo de titulación)
5. MsC. Luis Suárez (CoDirector del trabajo de titulación)
6. Bolívar Chacua (Desarrollador del proyecto)

Fuente: Autoría

En la Tabla 9 se listan a los implicados o stakeholders que se toman en cuenta para el desarrollo de este proyecto:

Tabla 9. Requerimientos de Stakeholders.

	StSR				
	REQUERIMIENTOS DE STAKEHOLDERS				
#	REQUERIMIENTOS DE USO	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
	REQUERIMIENTOS OPERACIONALES				
StSR1	El sistema debe implementarse en las instalaciones de la FICA	X			
StSR2	Adquisición de datos de entrenamiento (imágenes del rostro) para construcción del clasificador de aprendizaje automático	X			
StSR3	Para el entrenamiento de los usuarios se debe poseer una base de datos de rostros de todos los individuos ordenada de manera correcta en carpetas	X			
	REQUERIMIENTO DE USUARIOS				
StSR4	Los usuarios directos del sistema pueden manipular las opciones del sistema		X		
StSR5	Los usuarios indirectos del sistema deben mirar hacia la cámara al menos una vez	X			
StSR6	Para la obtención de muestras de los rostros de los usuarios, se debe capturar fotografías de casi todos los ángulos o poses posibles de cada individuo	X			

Fuente: Autoría

3.3.4. Selección de Hardware y Software.

Para la selección de los componentes de hardware y software se realiza una tabla comparativa de especificaciones según los atributos de los requerimientos de Stakeholders, Sistemas y de Arquitectura, se evalúa un componente y mediante dicha tabla se obtiene una valoración de los atributos correspondientes (StRS, SySR, SRSH) y al final se elige al componente de mayor puntuación. Definiendo que un valor de puntuación de "1" cumple con el requerimiento y de una puntuación de "0" cuando no cumple con el requerimiento.

3.3.4.1. Hardware.

La selección del Hardware se realiza de acuerdo a los requerimientos de hardware establecidos en la Tabla 7 sobre Requerimientos de Arquitectura. Específicamente se seleccionará la unidad central de procesamiento (CPU), la unidad de procesamiento grafico (GPU), una fuente de poder adecuada, y la cámara a emplearse para el análisis de las imágenes por visión artificial,

- ***Procesador (CPU).***

Para la elección del procesador se eligieron 4 opciones que se adaptan a las necesidades del proyecto. La Tabla 10 muestra la valoración de cada requerimiento para la elección del procesador (CPU).

Tabla 10. Elección de CPU.

HARDWARE						**VALORACION TOTAL**
	SRSH4	**SRSH13**	**SRSH14**	**SRSH15**	**SRSH16**	
AMD/Ryzen 7 1800X	1	1	1	1	0	4
AMD/ Ryzen 7 1700X	1	1	1	1	0	4
INTEL/Core i7-7700K	1	1	1	1	1	5
INTEL/Core i7-8700	1	1	1	1	1	5
1 Cumple **0 No cumple**						

Elección: En la selección del procesador según la tabla de requerimientos de arquitectura, se concluye que es óptimo el uso de procesadores Intel Core i7 7700K/8700, ya que cumple con todos los requerimientos de una arquitectura robusta para el despliegue de aplicaciones de inteligencia artificial, lo verdaderamente destacable de estos procesadores es la integración de un chip gráfico dentro de la misma oblea de silicio, lo que brinda una ventaja, ya que la CPU podrá hacer uso de dicho chip para operaciones gráficas del sistema operativo, alivianando la carga de procesos en la GPU. Esto significa que, en lugar de desperdiciar las capacidades de la GPU en procesos gráficos del entorno del sistema operativo (GUI), estas pueden ser destinadas al chip grafico de la CPU. Asimismo, parcialmente se usa una cantidad de núcleos físicos y lógicos del CPU para la ejecución de operaciones de procesamiento paralelo.

Fuente: Autoría

A continuación, en la Tabla 11 se muestran las características técnicas principales del procesador INTEL/Core i7-8700, las especificaciones completas se pueden obtener en el siguiente enlace: https://intel.ly/2DEwhck

Tabla 11. Especificaciones técnicas del CPU.

ESPECIFICACIONES	Propiedades
Cantidad de núcleos	6
Cantidad de subprocesos	12
Frecuencia básica del procesador	3.20 GHz
Frecuencia turbo máxima	4.60 GHz
Caché	12 MB SmartCache
Gráficos incorporados	Gráficos UHD Intel® 630

Fuente: Adaptado de (Intel Corporation, 2017)

- ***Tarjeta de video (GPU).***

Para la elección del procesador se eligieron 2 opciones que se adaptan a las necesidades del proyecto. La Tabla 12 muestra la valoración de cada requerimiento para la elección de la tarjeta de video (GPU).

Tabla 12. Elección de GPU.

HARDWARE					VALORACION TOTAL
	SRSH5	**SRSH8**	**SRSH9**	**SRSH11**	
NVIDIA GEFORCE GTX 1080	1	1	1	1	4
AMD Radeon RX 480	1	0	0	1	2

1 Cumple
0 No cumple

Elección: En la selección de la tarjeta de video según la tabla de requerimientos de arquitectura, se concluye que es óptimo el uso de un módulo NVIDIA GEFORCE GTX 1080, ya que cumple con todos los requerimientos de arquitectura enfocados al diseño e implementación de algoritmos de inteligencia artificial para el procesamiento de imágenes mediante tecnología CUDA, la cual se encarga de realizar cálculos paralelos entre CPU y GPU.

Fuente: Autoría

A continuación, en la Tabla 13 se muestran las características técnicas principales de la tarjeta de video, las especificaciones completas se pueden obtener en el siguiente enlace: https://bit.ly/2iMcmNK

Tabla 13. Especificaciones técnicas del GPU.

ESPECIFICACIONES	Propiedades
Cantidad de núcleos CUDA	2560
Reloj base	1708 MHz
Velocidad de memoria	10 Gbps
Memoria de video dedicada	8 GB GDDR5X
Ancho de interfaz de memoria	256 bits
Ancho de banda de memoria	320 GB/s

Fuente: Adaptado de (NVIDIA Corporation, 2016)

- ***Fuente de poder.***

Para la elección de la fuente de poder del servidor se eligieron 3 opciones que se adaptan a las necesidades del proyecto. La Tabla 14 muestra la valoración de cada requerimiento para la elección de la fuente de poder más óptima.

Tabla 14. Elección de la fuente de poder.

HARDWARE			VALORACION TOTAL
	SRSH5	**SRSH9**	
CORSAIR CX750M	1	1	2
COOLER MASTER GX750W	1	1	2
ALTEK/QUASAD 750W	1	0	1
1 Cumple **0 No cumple**			

Elección: En la selección de la fuente de poder según la tabla de requerimientos de arquitectura, se concluye que es óptimo el uso de cualquiera de las fuentes de poder en sus variantes Corsair o Cooler Master de 750W, ya que poseen certificación 80 Plus, lo cual significa que usan un 20% menos energía y entregan estabilidad en la operación de los componentes conectados, brindando una larga vida útil a los componentes del servidor.

Fuente: Autoría

A continuación, en la Tabla 15 se muestran las características técnicas principales de la fuente de poder, las especificaciones completas se pueden obtener en el siguiente enlace: https://bit.ly/2y348XE

Tabla 15. Especificaciones técnicas de la fuente de poder.

ESPECIFICACIONES	Propiedades
Máxima potencia	750 Watts
Ventilador	120 mm
Conectores PCI-Express	2 (6+2 Pines)
Conectores SATA	8
Entrada de Voltaje	100 -240 VCA
Entrada de corriente	12A – 6A
Eficiencia 80 Plus	Bronce

Fuente: Adaptado de (Corsair Components, 2016)

- ***Cámara.***

Para la elección de la cámara se eligieron 3 opciones que se adaptan a las necesidades del proyecto. La Tabla 16 muestra la valoración de cada requerimiento para la elección de la cámara IP.

Tabla 16. Elección de la cámara IP.

HARDWARE			VALORACION TOTAL
	SRSH11	**SRSH18**	
HIKVISION CAMERA IP DS-2CD2142FWD-I(W)(S)	1	1	2
DAHUA IPC HDW2100	1	1	2
WEBCAM CAMERA	1	0	1
1 Cumple **0 No cumple**			
Elección: En la selección de la cámara IP según la tabla de requerimientos de arquitectura, se concluye que es óptimo el uso de una cámara IP marca: HIKVISION o DAHUA, ya que ofrecen trasmisión de datos vía protocolo IP y video con resolución 1080p y 720p.			

Fuente: Autoría

A continuación, en la Tabla 17 se muestran las características técnicas principales de la cámara seleccionada como primera opción, las especificaciones completas se pueden obtener en el siguiente enlace: https://bit.ly/2SAH77S

Tabla 17. Especificaciones técnicas de la cámara IP.

ESPECIFICACIONES	Propiedades
Sensor de imagen	1/3" Progressive Scan CMOS
Máxima resolución de imagen	2688 × 1520
Resoluciones disponibles	50 Hz: 20 fps (2688 × 1520), 25 fps (1920 × 1080), 25 fps (1280 × 720) 60 Hz: 20 fps (2688 × 1520), 30 fps (1920 × 1080), 30 fps (1280 × 720)
Interfaz de comunicación	Interfaz Ethernet RJ45 10M / 100M
Fuente de alimentación	12 VDC ± 25%, PoE (802.3af Class3)
Compresión de video	H.264 / MJPEG / H.264 +

Fuente: Adaptado de (HIKVISION Digital Technology Co., 2019)

3.3.4.2. Software.

Una vez seleccionado el hardware se procede a la selección del software, el cual se realiza en base a los requerimientos de software establecidos en la Tabla 7 sobre Requerimientos de Arquitectura. En esta sección se consideran algunas alternativas en cuanto a la codificación del sistema sobre una plataforma de programación adecuada y bien documentada. Debido a que la naturaleza de este proyecto se centra en capacitación e implementación de modelos de aprendizaje automático y profundo casi en su totalidad, es crucial seleccionar con el mayor cuidado posible, ya que, al ser una investigación en constante avance dentro del campo de la inteligencia artificial algunas plataformas conllevan mayor dificultad de implementación que en otras y son compatibles con tecnología de procesamiento paralelo CUDA. Además, debe tener compatibilidad con la biblioteca de visión artificial OpenCV.

- ***Software de programación.***

Para la selección del software de programación se seleccionaron 4 opciones, las cuales se adaptan parcialmente a los requerimientos del proyecto para su codificación. La Tabla 18 muestra la valoración de cada requerimiento para la elección del software de programación.

Tabla 18. Elección del software de programación.

SOFTWARE							**VALORACION TOTAL**
	SRSH 19	**SRSH 20**	**SRSH 21**	**SRSH 23**	**SRSH 26**	**SRSH 27**	
Python	1	1	1	1	1	1	6
Matlab	0	1	1	1	1	1	5
Visual Studio	0	1	1	1	1	1	5
Java	1	1	1	1	0	1	5
1 Cumple **0 No cumple**							

Elección: En la selección del software idóneo según los requerimientos del software especificados en la tabla de requerimientos de arquitectura la mayor valoración fue obtenida por Python. Este software de programación es altamente ideal para el diseño de aplicaciones de inteligencia artificial, ya que tiene compatibilidad con las más relevantes bibliotecas de aprendizaje automático y profundo (Tensorflow/Scikit-Learn) utilizadas en este proyecto.

Fuente: Autoría

3.4. Diseño del sistema

Una vez definidos los requerimientos y componentes que regirán el sistema, se procede a definir las directrices del diseño del sistema, tomando en cuenta los criterios en las etapas de análisis del proyecto y requerimientos del sistema, que permitirán el desarrollo e implementación del sistema prototipo de reconocimiento facial.

Además, se muestra la disposición de funciones del sistema mediante diagrama de bloques y diagrama de flujo, los cuales brindaran una guía ordenada a través de todos los procesos a efectuar en la codificación y posterior capacitación de modelos de visión artificial enfocadas a las tareas de detección y reconocimiento facial, para su ejecución en tiempo real.

3.4.1. Diagrama de bloques del sistema.

En esta sección se presentará a través de diagramas de bloques dispuestos en 3 etapas el proceso de capacitación y ejecución del sistema. Cada etapa engloba varios subprocesos afines a la función específica de cada bloque. Las primeras 2 etapas constan de una secuencia de bloques de estructura bastante similar, ya que denotan el proceso de entrenamiento de los modelos de visión artificial para la verificación facial; estos parten desde la adquisición de datos, procesamiento de información, construcción del modelo de aprendizaje profundo para extracción de características faciales profundas y el modelo de aprendizaje automático para clasificación de dichas características profundas. La tercera etapa gira en torno al uso de dichos modelos capacitados para la extracción de características faciales en tiempo real y comparación con la base de datos de rostros a través del modelo de clasificación entrenado previamente.

3.4.1.1. Diagrama de bloques general del sistema.

En la siguiente Figura 32 se muestra la arquitectura del sistema de reconocimiento facial a modo de diagrama de bloques general; de esta manera se obtiene una mejor comprensión de las funciones llevadas a cabo en las 3 etapas mencionadas, las cuales se abordarán a detalle en las posteriores secciones. Todas las operaciones que son efectuadas en cada bloque de cada etapa serán implementadas sobre un servidor equipado con los componentes especificados en la sección previa de requerimientos de hardware y software.

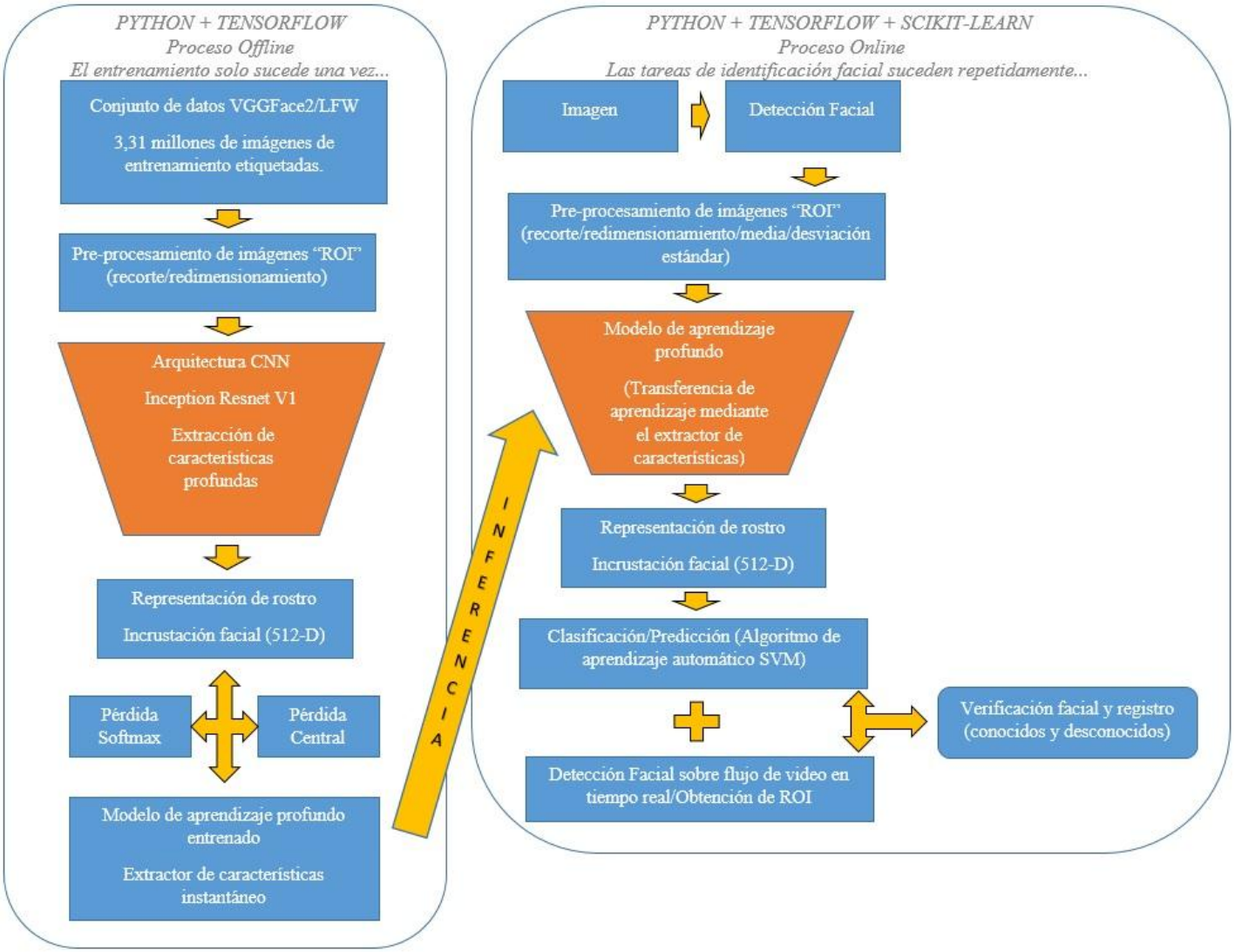


Figura 32. Arquitectura General del sistema.

Fuente: Autoría

3.4.1.2. Diagrama de bloques de la primera etapa

El diagrama de bloques que se presenta a continuación es el que comprende a la primera etapa, el cual se caracteriza por ser un proceso offline, enfocado a la construcción de un modelo de aprendizaje profundo para la extracción de incrustaciones faciales a través del entrenamiento intensivo de una arquitectura CNN (*Inception Resnet V1*) bajo GPU, empleando la tecnología de procesamiento paralelo CUDA y el conjunto de herramientas de la biblioteca cuDNN para el marco de aprendizaje automático de Tensorflow sobre un conjunto de datos de entrenamiento a gran escala, el cual se encuentra diseñado para encarar el problema del reconocimiento facial denominado VGGFace2. Es así que, la primera etapa está formada por 4 bloques los cuales a su vez contienen varios subprocesos; se han planteado los bloques dependiendo de las funciones que cada uno debe desarrollar. En la Figura 33 se puede observar el diagrama de bloques de la primera fase con cada uno de los subprocesos.

Figura 33. Diagrama de bloques de la primera etapa.

Fuente: Autoría

La primera fase del proyecto comienza con la adquisición de datos en el bloque 1, y este consiste en la obtención del conjunto de datos de rostros VGGFace2, el cual está disponible en la página oficial del proyecto (Visual Geometry Group, 2019). Este conjunto de datos está compuesto de una gran cantidad de imágenes de rostros de personajes influyentes de todo el mundo, que tienen grandes variaciones en postura, edad, iluminación, origen étnico y profesión.

En el bloque 2 se inicia el tratamiento a las imágenes de entrenamiento mediante la biblioteca OpenCV, el cual procede a realizar un pre-procesamiento de las imágenes del conjunto de datos convirtiéndolas de un formato BGR a RGB (convenciones para el orden de los diferentes canales de color) para la precisión del detector facial. Consecuentemente, la alineación del rostro se lleva a cabo por un proceso en el cual se busca recortar la zona de la detección del rostro (ROI) y redimensionar a una imagen con dimensiones de 182x182 pixeles, lo cual es un requerimiento necesario para la capa de entrada de la arquitectura CNN (Inception Resnet V1).

En el bloque 3 se realiza el entrenamiento intensivo de la red mediante el lenguaje de programación *Python* y la biblioteca de inteligencia artificial por excelencia *Tensorflow* empleando la distribución para GPU. El resultado de este entrenamiento es un modelo generalizado con la función de extraer un vector de incrustaciones faciales de 512 mediciones por cada rostro. Como la CNN se entrenó con una gran cantidad de imágenes de rostros, será capaz de generalizar de forma bastante precisa y única cualquier rostro.

Finalmente, el último bloque de esta primera etapa tiene el fin de comprobar o medir la precisión del modelo de aprendizaje profundo obtenido en los bloques previos mediante la evaluación cuantitativa en el conjunto de datos LFW (Labeled Faces in the Wild). La representación gráfica de la Figura 34 muestra de mejor manera los aspectos anteriormente mencionados:

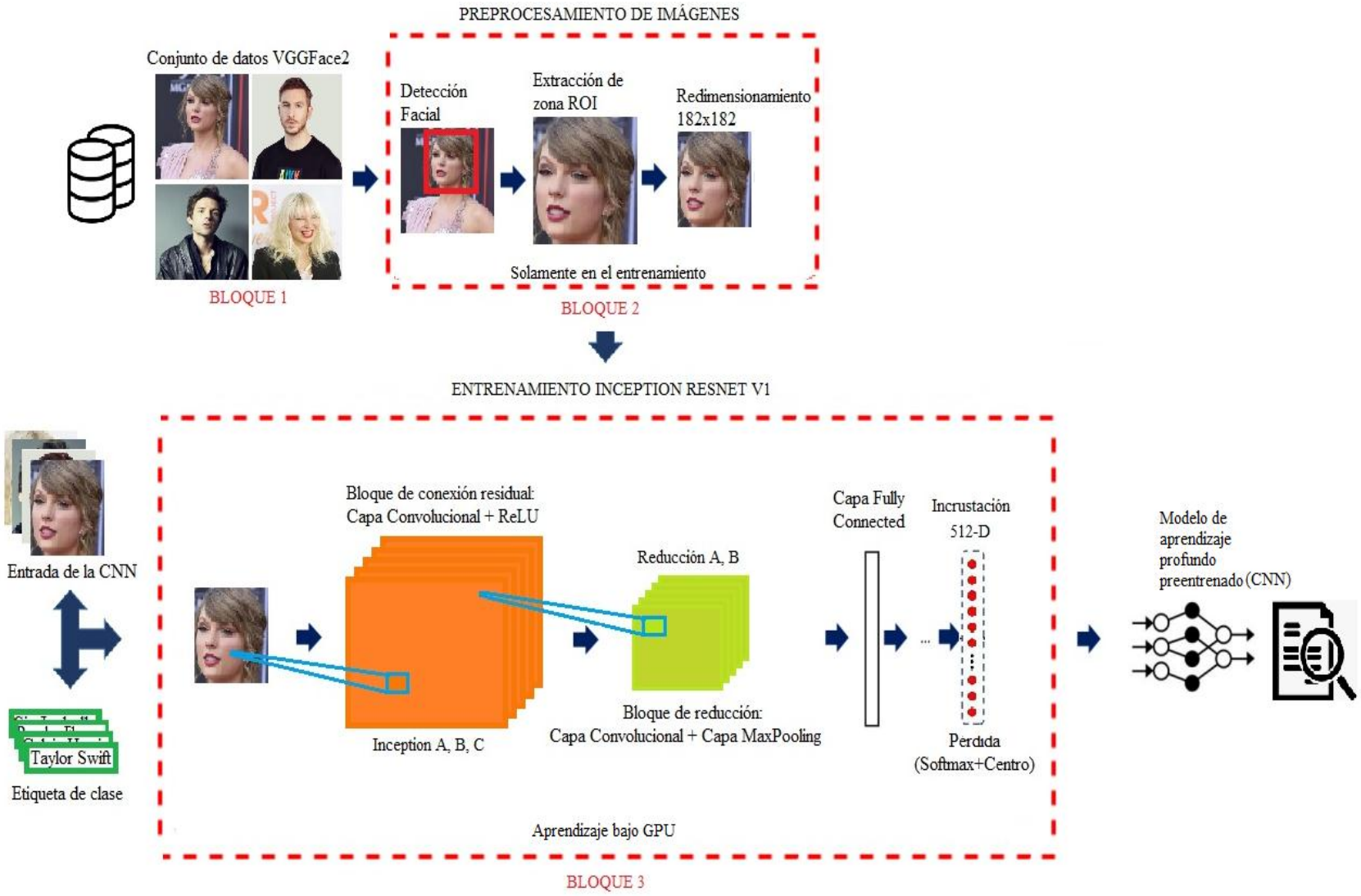


Figura 34. Representación gráfica del diagrama de bloques de la primera etapa.

Fuente: Autoría

3.4.1.3. Diagrama de bloques de la segunda etapa.

El diagrama de bloques de la segunda etapa del sistema se presenta con la finalidad de construir un modelo de aprendizaje automático capaz de clasificar las incrustaciones faciales generadas por el modelo de aprendizaje profundo en clases o identidades, a través del entrenamiento de un clasificador SVM[6] bajo GPU. Además, esta etapa también se caracteriza por ser un proceso offline. La Figura 34 muestra la disposición de bloques destinadas a esta segunda etapa.

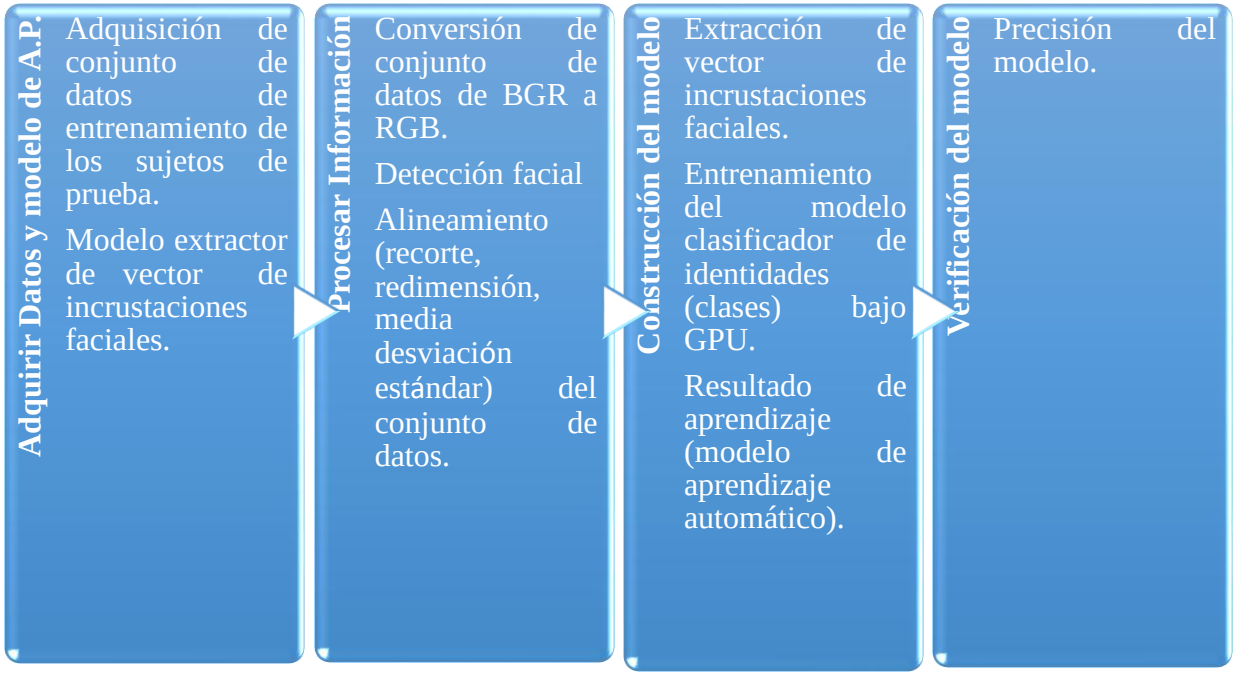


Figura 35. Diagrama de bloques de la segunda etapa.

Fuente: Autoría

[6] SVM: Support Vector Machines o Máquinas de Vector Soporte son un conjunto de algoritmos de aprendizaje supervisado propiamente relacionados con problemas de clasificación y regresión.

El bloque 1 de la segunda etapa consiste en la adquisición del conjunto de datos de entrenamiento personalizado de los sujetos de prueba del sistema en un proceso offline, además, se empleará el modelo extractor de vector de incrustaciones faciales obtenido en la primera etapa mediante la aplicación de la técnica de transferencia de aprendizaje.

En el bloque 2 se procede de la misma manera que en la capacitación del modelo de aprendizaje profundo, donde se hace un proceso de alineación y un pre-procesamiento (media y desviación estándar) de las imágenes de los rostros, en este caso se lo realiza sobre un conjunto de datos de entrenamiento personalizado.

En el bloque 3 se extraen los vectores de incrustaciones faciales correspondientes al conjunto de datos de entrenamiento del bloque 2 empleando el modelo de aprendizaje profundo, construido en la primera etapa del proyecto. Dichos vectores son clasificados en clases mediante un algoritmo de aprendizaje automático supervisado (SVM) tomando como datos de entrenamiento al conjunto de incrustaciones obtenidas de los rostros desde un directorio de imágenes y su etiqueta o identificación de clase correspondiente. Finalmente, en el último bloque se evalúa la precisión del modelo. La representación gráfica de la Figura 36 muestra de mejor manera los aspectos anteriormente mencionados:

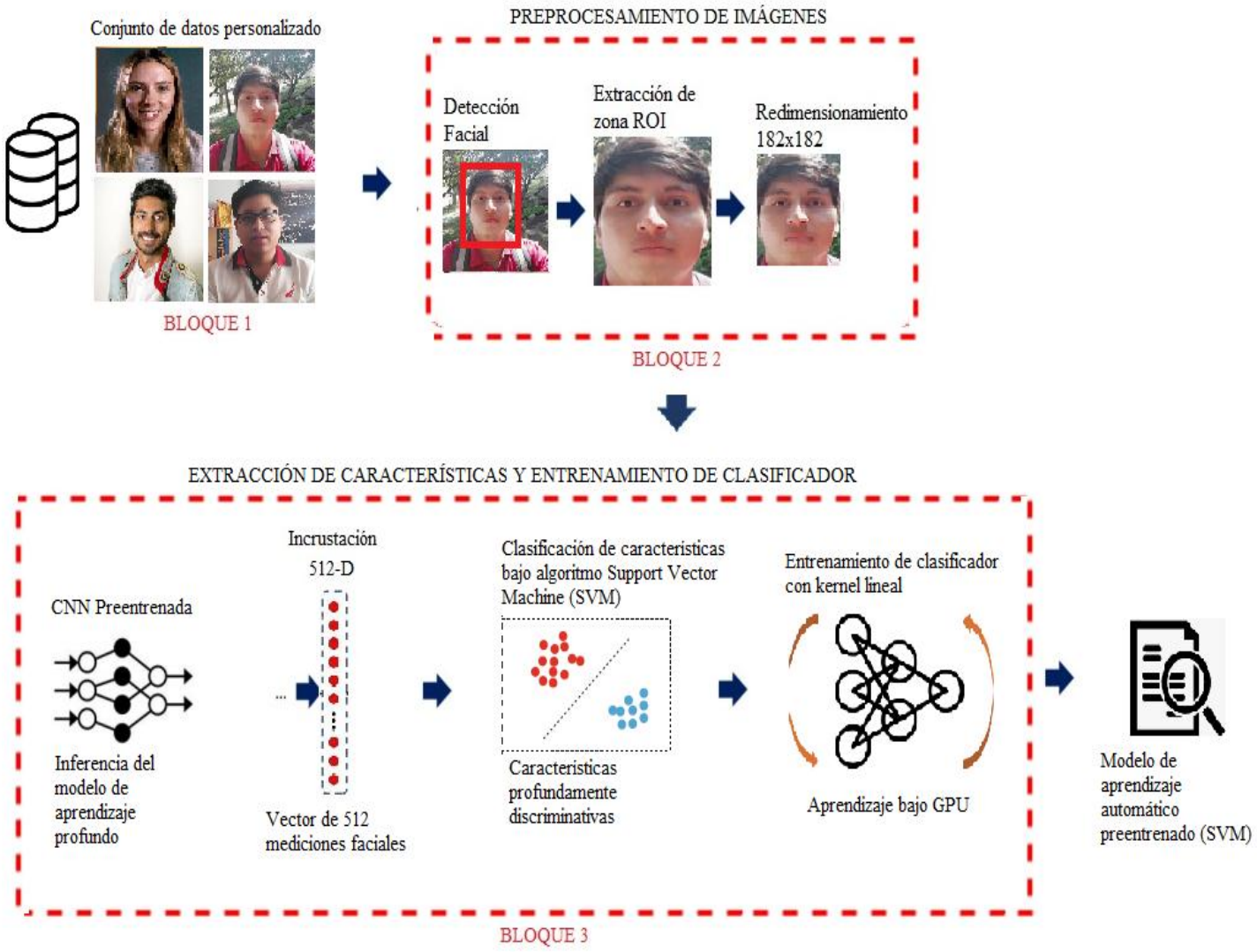


Figura 36. Representación gráfica del diagrama de bloques de la segunda etapa.

Fuente: Autoría

3.4.1.4. Diagrama de bloques de la tercera etapa.

En el diagrama de bloques que se presenta a continuación se establece el proceso general de funcionamiento del sistema en tiempo real bajo GPU, en el que se implementan los modelos entrenados previamente en la primera y segunda etapa para la tarea de identificación de personas. Además, esta etapa se caracteriza por ser un proceso online. La Figura 35 muestra la disposición de bloques destinadas a esta tercera etapa.

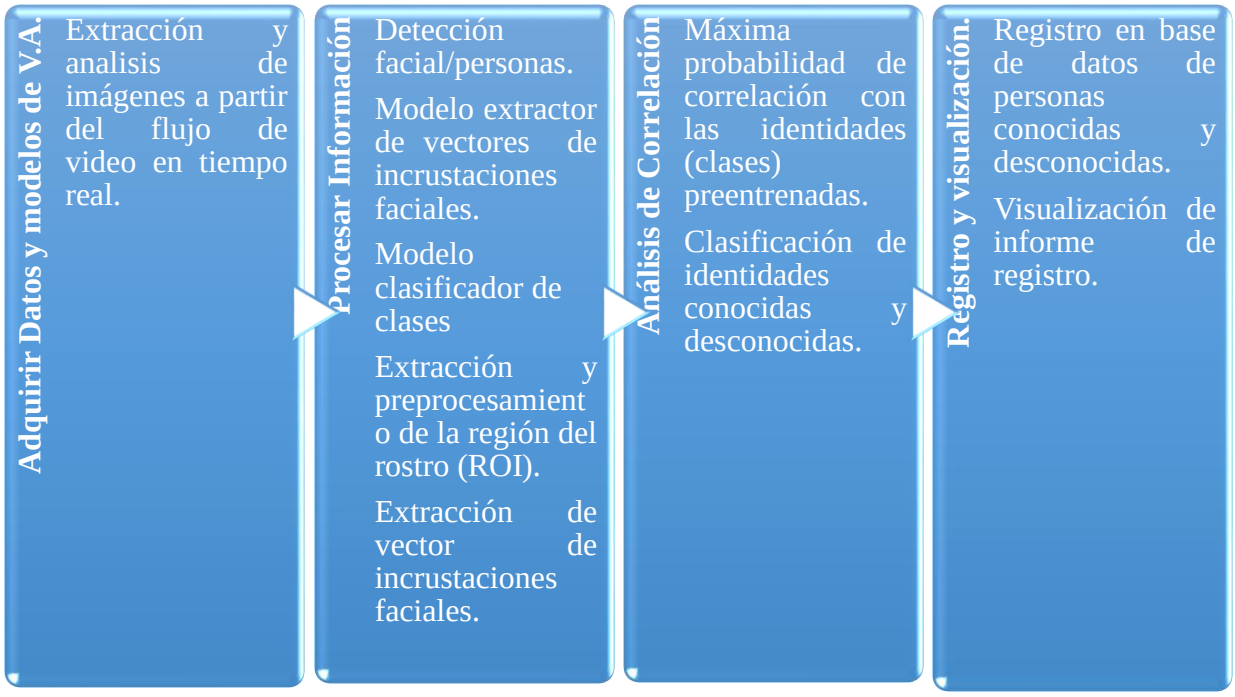


Figura 37. Diagrama de bloques de la tercera etapa.

Fuente: Autoría

En el bloque 1 se realiza el análisis de flujo de video en tiempo real, lo cual define el tratamiento de la secuencia de cuadros obtenidos de la cámara mediante algunas operaciones sobre la imagen con la librería OpenCV.

En el bloque 2 se define el proceso online de detección facial, la extracción de la región facial de interés ROI (“Region of Interest”) y el pre-procesamiento de la misma a través de un

proceso de conversión del espacio de colores RGB a YCbCr y una operación de ecualización del histograma de la ROI, recorte, redimensionamiento, y el cálculo de la media/desviación estándar (ingeniería de características para la normalización de características de ROI), para posteriormente extraer los vectores de incrustaciones faciales de los rostros obtenidos del flujo de video. En esta etapa se emplean los modelos de visión artificial generados en la primera y segunda etapa.

En el bloque 3 se correlaciona dicho conjunto de incrustaciones faciales obtenidas del bloque 2 con el modelo entrenado de aprendizaje automático supervisado de la segunda etapa. Específicamente, dicha correlación se basa en la selección de la máxima probabilidad de cercanía o similitud del conjunto de incrustaciones faciales obtenidas en tiempo real con el conjunto de incrustaciones previamente entrenadas. El resultado del proceso anterior muestra información de identidad en caso de existir una coincidencia, caso contrario se etiqueta como desconocida.

Finalmente, mediante una interfaz de usuario (GUI) se interpretan los resultados del bloque 3 y se registran en la base de datos de personas conocidas o desconocidas para la generación de un reporte, según sea el caso. La representación gráfica de la Figura 38 muestra de mejor manera los aspectos anteriormente mencionados:

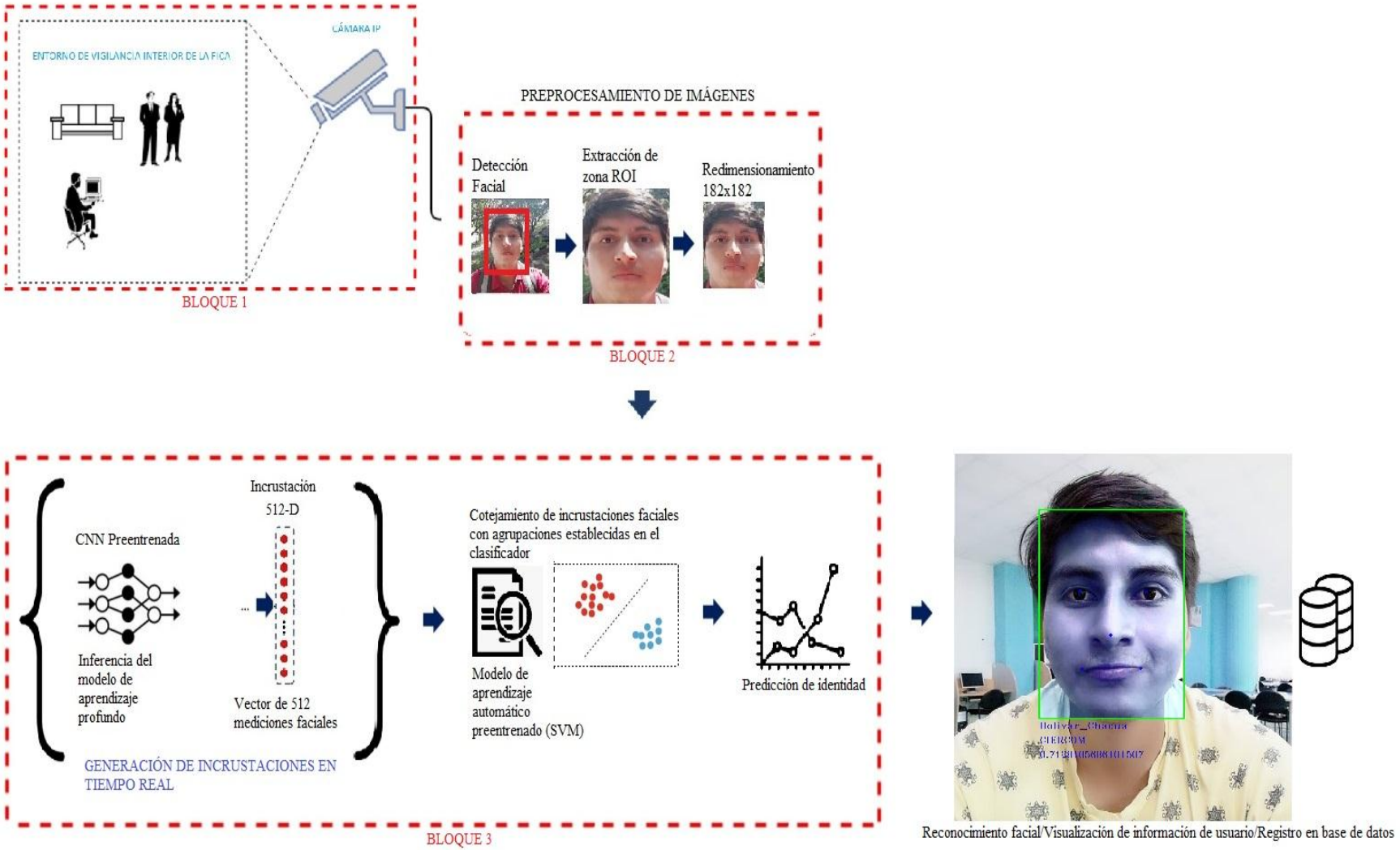


Figura 38. Representación gráfica del diagrama de bloques de la tercera etapa.

Fuente: Autoría

3.4.2. Desarrollo del software (Codificación).

En esta sección se procede a desarrollar los módulos que componen el software de programación en las etapas de capacitación de los modelos de visión artificial y la etapa de ejecución en tiempo real. Durante el desarrollo de software se realiza la codificación en una estación de trabajo con una arquitectura de sistema operativo Ubuntu 16.04, el cual se encargará de compilar toda la codificación realizada a través del lenguaje de programación interprete de Python, donde cada función del sistema poseerá un script o archivo ejecutable. Finalmente, para unificar dichas funciones mediante una interfaz amigable se hace uso de una plataforma de diseño de GUI´s. Se debe considerar que el desarrollo del software del sistema se realiza según los requerimientos especificados anteriormente.

3.4.2.1. Primera etapa.

Esta etapa se destina a la descripción del método de aprendizaje profundo empleado para la posterior extracción de características representativas de cualquier conjunto de datos provisto.

3.4.2.1.1. *Adquisición del conjunto de datos de entrenamiento.*

Para el entrenamiento del modelo de aprendizaje profundo a través de la arquitectura CNN (Inception Resnet V1) se ha seleccionado el conjunto de datos de rostros VGGFace2 (Visual Geometry Group, 2019), que se caracteriza por ser el más extenso que existe en la comunidad científica actualmente, al contener 3,31 millones de imágenes de 9131 sujetos. El conjunto de datos tiene un equilibrio de género aproximado, con un 59.3% de fotografías de hombres y 40.7% de fotografías de mujeres, variando entre 80 y 843 imágenes para cada identidad. La razón principal de esta elección es que las CNN´s requieren de una masiva cantidad de imágenes para entrenar

modelos bastante precisos en tareas de visión artificial, tales como la detección y reconocimiento facial, por lo que este conjunto de datos resulta ser muy adecuado.

En el siguiente enlace se puede acceder a la descarga del conjunto de datos de imágenes: https://bit.ly/2Oan955

3.4.2.1.2. *Preprocesamiento de imágenes.*

El primer paso a realizarse luego de obtener el conjunto de datos de imágenes es realizar una transformación y/o reorganización de los canales de color de las mismas, partiendo del espacio de color BGR a un espacio RGB, adecuado para la entrada de la arquitectura Inception Resnet V1, mediante la conversión a través del método CV_BGR2RGB del módulo OpenCV. El script "align_data_GUI.py" (Anexo 1) realiza esta tarea sobre todos los archivos antes de la fase de detección facial. Aunque la conversión que se realiza en este apartado no es el único tratamiento de imagen disponible dentro de las funciones de OpenCV, existen otras conversiones que merecen ser mencionadas tales como: CV_BGR2GRAY, CV_BGR2HSV, CV_RGB2YCrCb, CV_HSV2GRAY, entre otras. Cada una de estas conversiones tienen sus particularidades, ya que acompañadas con un adecuado proceso de realzado de características de la imagen pueden eliminar el ruido de la imagen original generado por cambios de iluminación o degradación afectando a la calidad de la misma, también pueden mejorar de manera significativa el contraste de los canales de color, entre otras ventajas. A continuación, se detallan las conversiones que se emplean en el proyecto:

CV_BGR2RGB: La conversión a RGB es una reorganización de los canales del color que no agrega una importancia sustancial a las características de la imagen, sin embargo, es necesaria para la entrada de la red. Para realizar este proceso se emplea el método cvtColor incorporada en

el módulo OpenCV, donde se establece la conversión CV_BGR2RGB como parámetro de conversión de imagen de salida.

CV_RGB2YCrCb: La conversión al espacio YCrCb codifica de forma no lineal el espacio RGB separando los valores de intensidad de los componentes de color de los canales RGB, esto es ideal para el proceso de ecualización de histograma de manera que los valores de intensidad no alteran el balance de color de la imagen. Para realizar este proceso se emplea el método cvtColor del módulo OpenCV donde se establece la conversión CV_RGB2YCrCb como parámetro de conversión de imagen de salida.

Dependiendo de los canales requeridos. OpenCV emplea el siguiente principio para la conversión al formato YCrCb, el color es representado por la luminancia (Y) y por dos componentes de color (Cb y Cr) (OpenCV Org, 2019); las formulaciones matemáticas de la Ecuación 7 muestran de manera detallada las operaciones realizadas sobre una imagen por dicha conversión:

$$Y \leftarrow 0.299 * R + 0.587 * G + 0.114 * B$$

$$Cr \leftarrow (R - Y) * 0.713 + delta$$

$$Cb \leftarrow (B - Y) * 0.564 + delta$$

$$R \leftarrow Y + 1.403 * (Cr - delta)$$

$$G \leftarrow Y - 0.714 * (Cr - delta) - 0.344 * (Cb - delta)$$

$$B \leftarrow Y + 1.773 * (Cb - delta)$$

Ecuación 7. Formulación matemática de la conversión de una imagen RGB a YCrCb.

Fuente: Adaptado de (OpenCV Org, 2019)

3.4.2.1.3. *Ecualización de histograma.*

La ecualización es un método que mejora el contraste en una imagen, con el fin de estirar el rango de intensidad. Es así que dada una imagen se busca agrupar los píxeles de manera que se normalizan los niveles de grises, aumentando el contraste entre las zonas oscuras y las zonas claras. En la Figura 39a se muestra una imagen RGB, a la que se le aplica la conversión a YCrCb como se indica en la Figura 39b, para posteriormente aplicar la ecualización de histograma y así mejorar su contraste como se indica en la Figura 39c, siendo esta operación óptima para el reconocimiento facial en entornos abiertos donde existe variación de iluminación, por lo que es utilizada en el proceso online del sistema. La instrucción que se emplea para la ecualización en OpenCV es: ***cv2.equalizeHist(src, dst).***

a) b) c)

Figura 39. a) Imagen RGB, b) conversión de imagen RGB a YCrCb y c) aplicación de ecualización de histograma.

Fuente: Autoría

3.4.2.1.4. *Detección facial.*

Una vez realizada la conversión de las imágenes a canales RGB, es momento de localizar la zona ROI con el modelo de detección facial propuesto por Kaipeng Zhang & Zhanpeng Zhang (2015) que establece una arquitectura CNN en cascada profunda definida en tres etapas

denominadas P-Net, R-Net y O-Net. El resultado del entrenamiento de esta red son tres archivos con extensión "*.npy*", que contienen los pesos de la red debidamente calibrados y ordenados en matrices tipo *numpy* (biblioteca enfocada al tratamiento de matrices) para cada una de las 3 etapas de la CNN. Dichos pesos pueden reconstruirse y accederse en cualquier computadora que disponga de un entorno de diseño de aplicaciones de inteligencia artificial.

El acceso a los pesos de la CNN, se lo realiza a través de Python y la biblioteca Tensorflow, con lo cual se crea un script denominado "detect_face.py" el cual provee los métodos para acceder a los pesos de la red y conseguir definir el cuadro delimitador (*bounding box*) de la zona ROI. En el Anexo 2 se muestra la estructura completa del script y en la Figura 40 se muestra una porción de código que define la forma en que se acceden a los pesos de la red pre-entrenada.

```
with tf.variable_scope('pnet'):#Crea una variable pnet
    data = tf.placeholder(tf.float32, (None,None,None,3), 'input')#Crea un tensor o grafo con una entrada
                                                                  #de dimensiones aleatorias para los 3 canales de color
    pnet = PNet({'data':data})
    pnet.load(os.path.join(model_path, 'det1.npy'), sess)#Carga los pesos de la red P-Net en el tensor
with tf.variable_scope('rnet'):#Crea una variable rnet
    data = tf.placeholder(tf.float32, (None,24,24,3), 'input')#Crea un tensor o grafo con una entrada
                                                              #de dimensiones 24x24 para los 3 canales de color
    rnet = RNet({'data':data})
    rnet.load(os.path.join(model_path, 'det2.npy'), sess)#Carga los pesos de la red R-Net en el tensor
with tf.variable_scope('onet'):#Crea una variable onet
    data = tf.placeholder(tf.float32, (None,48,48,3), 'input')#Crea un tensor o grafo con una entrada
                                                              #de dimensiones 48x48 para los 3 canales de color
    onet = ONet({'data':data})
    onet.load(os.path.join(model_path, 'det3.npy'), sess)#Carga los pesos de la red O-Net en el tensor
#
pnet_cara = lambda img : sess.run(('pnet/conv4-2/BiasAdd:0', 'pnet/prob1:0'),
                                  feed_dict={'pnet/input:0':img})
rnet_cara = lambda img : sess.run(('rnet/conv5-2/conv5-2:0', 'rnet/prob1:0'),
                                  feed_dict={'rnet/input:0':img})
onet_cara = lambda img : sess.run(('onet/conv6-2/conv6-2:0', 'onet/conv6-3/conv6-3:0', 'onet/prob1:0'),
                                  feed_dict={'onet/input:0':img})
return pnet_cara, rnet_cara, onet_cara#Retorna los 3 parametros de la red listos para usar
```

Figura 40. Acceso al módulo de detección facial.

Fuente: Autoría

A continuación, en la Figura 41 se muestra el resultado de emplear dichos métodos sobre el script "detect_face_landmarks_GUI.py" (Anexo 3) elaborado en Python, que muestra la zona del rostro detectada y debidamente enmarcada a través de un cuadro delimitador de color azul, además se denotan 5 puntos de referencia faciales (*landmarks*) de diferente color.

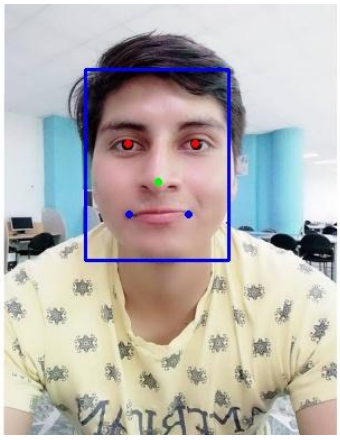


Figura 41. Prueba de detección de rostro con 5 puntos de referencia.

Fuente: Autoría

Para comprobar la eficacia de este detector facial se han hecho pruebas preliminares de detección a una distancia mayor a 3 metros con un grupo de 16 personas, los resultados obtenidos se muestran en la Figura 42:

Figura 42. Prueba preliminar de detección de 16 rostros.

Fuente: Autoría

3.4.2.1.5. *Alineamiento de rostro.*

Una vez realizada la detección facial, es preciso extraer la zona ROI y redimensionar a un tamaño especifico. Este proceso genera imágenes RGB en formato "*.png*" de dimensiones de 182x182 pixeles. El script "align_data_GUI.py" (Anexo 1) realiza este proceso, además en la Figura 43 se muestra una porción del código empleado:

```
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)#Conversion de la imagen a un espacio de color RGB
print('Dimensión de datos a rgb: ', img.ndim)
img = img[:, :, 0:3]
print('Después de la dimensión de datos: ', img.ndim)
#Retorna las 4 coordenadas del cuadro delimitador del rostro y los 5 puntos de referencia facial
bounding_boxes, _ = detect_face.detect_face(img, minsize, pnet, rnet, onet, threshold, factor)
nrof_faces = bounding_boxes.shape[0]#Retorna el numero de rostros detectados
print('Rostro_detectado: %d' % nrof_faces)
if nrof_faces > 0:#Condición de existencia de 1 rostro en la imagen
    det = bounding_boxes[:, 0:4]#Coloca las 4 coordenadas del rostro en la primera fila de una tupla
    img_size = np.asarray(img.shape)[0:2]#Coloca en un array las dimensiones de la imagen
    if nrof_faces > 1:#Condición de existencia de mas de 1 rostro en la imagen
        bounding_box_size = (det[:, 2] - det[:, 0]) * (det[:, 3] - det[:, 1])#Reescalado de la imagen
        img_center = img_size / 2#Dividiendo o reduciendo a la mitad las dimensiones de la imagen
        offsets = np.vstack([(det[:, 0] + det[:, 2]) / 2 - img_center[1],
                             (det[:, 1] + det[:, 3]) / 2 - img_center[0]])#Centralizando la imagen
        offset_dist_squared = np.sum(np.power(offsets, 2.0), 0)
        index = np.argmax(bounding_box_size - offset_dist_squared * 2.0)#Agregando pesos extras en el centro
        det = det[index, :]#Coloca las 4 coordenadas del rostro seleccionado
    det = np.squeeze(det)#Suprime entradas bidimensionales(suprimiendo un eje de la matriz a una sola)
    bb_temp = np.zeros(4, dtype=np.int32)#Variable para almacenar las 4 coordenadas de rostro

    bb_temp[0] = det[0]#Primera coordenada
    bb_temp[1] = det[1]#Segunda coordenada
    bb_temp[2] = det[2]#Tercera coordenada
    bb_temp[3] = det[3]#Cuarta coordenada
    try:
        cropped_temp = img[bb_temp[1]:bb_temp[3], bb_temp[0]:bb_temp[2], :]#Recorte de la zona del rostro de la imagen
        scaled_temp = misc.imresize(cropped_temp, (image_size, image_size), interp='bilinear')#Redimensionado de zona
                                                                                              #ROI a 182x182 pixeles
        nrof_successfully_aligned += 1#Contador de rostros debidamente alineados
        misc.imsave(output_filename, scaled_temp)#Guardado de la imagen con extension ".png"
    except Exception as e:
        os.remove(image_path)
else:
    print('No se puede alinear "%s"' % image_path)
    text_file.write('%s\n' % (output_filename))
```

Figura 43. Secuencia de instrucciones de código para el alineamiento de rostros.

Fuente: Autoría

Como resultado del script elaborado en Python, se muestra en la Figura 44 la zona ROI pre-procesada:

Figura 44. Prueba de alineamiento de rostro.

Fuente: Autoría

Es preciso mencionar que para el entrenamiento de la CNN, todas las imágenes del conjunto de datos de entrenamiento han sido transformadas al formato PNG; la razón principal de esto se debe a que: "*PNG utiliza un algoritmo de compresión sin pérdidas que tiene la capacidad de almacenar hasta 8 bits de información adicionales en cada pixel (transparencia) a diferencia del formato JPG"* (Idento, 2015); lo que podría determinar mediciones faciales ligeramente más precisas y diferenciables que con otros formatos que usan algoritmos de compresión con pérdidas tales como: JPG,GIF, entre otros.

Finalmente, cabe resaltar que el proceso de alineación de rostros conlleva mucho tiempo y una carga computacional considerable cuando la galería de fotos a pre procesar contiene una cantidad de 3,31 millones de imágenes de rostros (VGGFace2); es por esto que en esta etapa se toma en cuenta la rapidez y potencia de cómputo del módulo GPU (NVIDIA GTX 1080) especificado en la sección de requerimientos de hardware. Dicha tarea fue llevada a cabo en un lapso de 70 a 72 horas.

3.4.2.1.6. *Arquitectura CNN (Inception Resnet v1).*

El impresionante trabajo que ha hecho el aprendizaje profundo en el área de reconocimiento de objetos es vital en las tareas centrales de visión artificial (VA) e inteligencia artificial (IA) en general. Es así que, los modelos de VA son componentes clave para habilitar sistemas de IA que pueden procesar entradas visuales. En la Tabla 3 del Cap. 2 se puede constatar que la VA tiene muchos campos de aplicación, uno de ellos es el ámbito de la seguridad, el cual está muy arraigado a este proyecto cuyo objetivo es crear un sistema de reconocimiento facial que emplee como núcleo principal un modelo capaz de realizar la extracción de características profundas de un rostro a través de un vector de 512 mediciones únicas descriptibles. Esto es posible con una CNN cuidadosamente elaborada, como es el caso de la red Inception Resnet V1.

Dada la alta precisión que alcanzó la arquitectura Inception en la clasificación y/o reconocimiento de objetos sobre el conjunto de datos de referencia ImageNet en el desafío anual ILSVR (ImageNet Large Scale Visual Recognition) en el año 2015 (Khan, Rahmani, Shah, & Bennamoun, 2018), este proyecto seleccionó la arquitectura CNN denominada Inception-Resnet-V1 (Figura 45), la cual introduce el concepto de *conexiones residuales* sobre cada módulo de la red acelerando significativamente el proceso de entrenamiento. La red consta de 3 módulos principales: Inception-Resnet-A, Inception-Resnet-B, e Inception-Resnet-C con un tamaño de bloque: 35x35, 17x17, y 8x8 respectivamente.

La estructura de dicha CNN se puede visualizar en la Figura 45, la cual será explicada a detalle en los siguientes apartados.

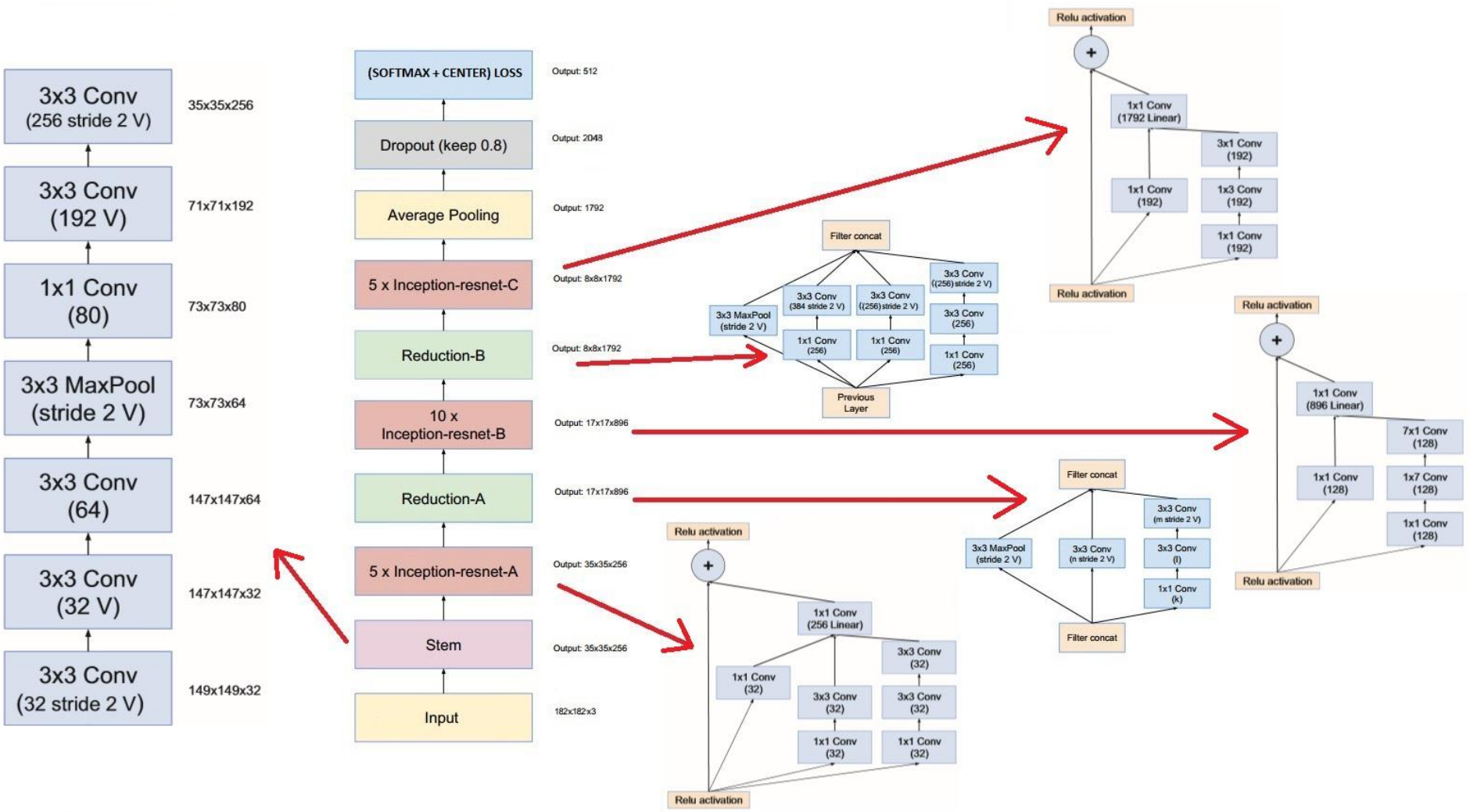


Figura 45. Arquitectura CNN Inception Resnet V1 general + Pérdida Softmax + Pérdida Central.

Fuente: Adaptado de (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017)

3.4.2.1.7. *Conexiones residuales.*

Antes de entrar a detalle en los aspectos de la arquitectura es importante definir la lógica que comprende el uso de *conexiones residuales*. La característica destacada de la arquitectura de red residual (*resnet*) es la identidad que omite las conexiones en los bloques residuales, lo que permite capacitar fácilmente arquitecturas CNN muy profundas.

En general esta red se diseñó teniendo en cuenta la eficiencia computacional, especialmente buscando minimizar la cantidad de RAM (CPU) y VRAM (GPU) para su entrenamiento y uso posterior a través de la inferencia (transferencia de aprendizaje). Su estructura consta de bloques o módulos *Inception*, los cuales sustituyen a una capa de convolución por varias capas de convoluciones más pequeñas y una última capa de filtro que se encarga de agrupar los resultados (*conexiones residuales*) (Figura 46). Estos módulos producen mejores resultados que las convoluciones más grandes, adicionalmente mejoran la velocidad de cálculo.

Para comprender estas *conexiones residuales* a profundidad, se considera el bloque residual de la Figura 46. Dada una entrada x, los pesos de las capas de la CNN implementan una función de transformación en esta entrada, representada por $F(x)$, que conlleva operaciones de capas de *convolución, max pooling y funciones de activación* (*ReLU, PreLU, Softmax, etc*) en la entrada y la salida del bloque. De esta manera estos módulos residuales se apilan uno sobre el otro para formar una red completa de extremo a extremo altamente robusta y efectiva.

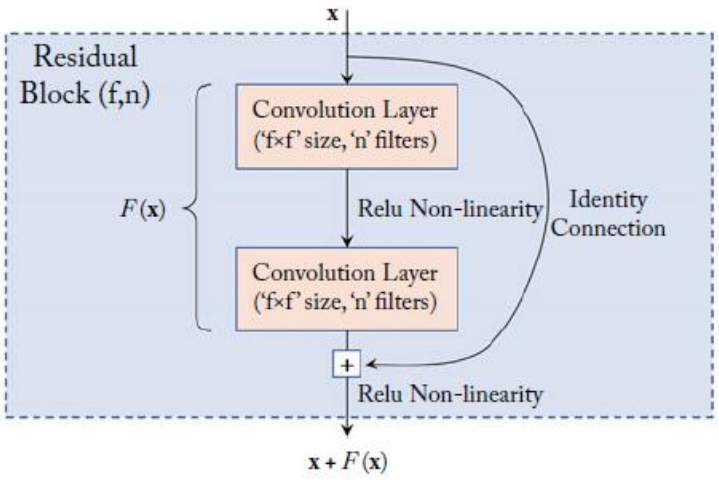


Figura 46. Estructura general de un bloque residual.

Fuente: Adaptado de (Khan, Rahmani, Shah, & Bennamoun, 2018)

3.4.2.1.8. *Función de activación.*

Como es común en la estructura de una CNN, las capas de *convolución, maxpooling y fully connected* a menudo son seguidas por una función de activación no lineal. Las funciones de activación toman una entrada de valores reales de la red y las comprime dentro de un rango pequeño como $[0,1]$ y $[-1,1]$. La aplicación de una función no lineal después de los pesos de dichas capas es muy importante ya que operan como un mecanismo de conmutación o selección, que decide si una neurona habilitará o no a todas sus entradas. Como se podrá apreciar en las siguientes secciones, todos los bloques residuales a lo largo de la CNN usan la función de activación *ReLU*, por lo que se ofrece una breve explicación a continuación:

Rectifier Linear Unit: ReLU es una función de activación simple que tiene una importancia práctica especial debido a su rápido cálculo a diferencia de técnicas comunes como: *sigmoide, tanh, LReLU, PReLU, entre otras*. Una función ReLU asigna un 0 si la entrada es negativa y mantiene su valor sin cambios si es positiva. Esto se puede representar de la siguiente manera en la Figura 47:

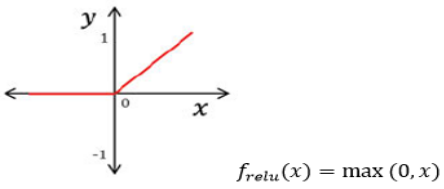


Figura 47. Función de activación ReLU.

Fuente: Adaptado de (Goodfellow, Bengio, & Courville, 2016)

3.4.2.1.9. *Bloques Inception.*

Una vez definido los componentes y funciones de un bloque residual se proceden a especificar los 3 principales bloques de la red denominados Inception. La Figura 48 muestra la distribución de cada una de ellas:

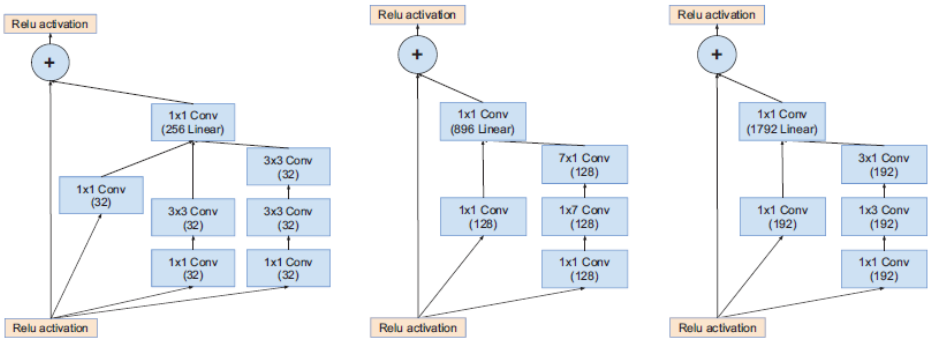


Figura 48. Bloques Inception A, B y C de conexiones residuales.

Fuente: Adaptado de (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017)

Antes cabe destacar que a partir de este apartado se emplea la biblioteca Tensorflow en su totalidad para el proceso de diseño y entrenamiento de la CNN. Además, como el conjunto de datos en muy extenso es imposible procesarlo en una sola pasada sobre la CNN, es por eso que se lo divide en mini-lotes para consecuentemente establecer tuberías (*pipelines*) adecuadas para la entrada de la CNN, donde se ha realizado un proceso de normalización o aumento de datos (*data augmentation: rotación, recorte, volteo, estandarización*) sobre las imágenes de los mini-lotes de entrenamiento, antes de realizar cualquier operación en la CNN. A continuación, se apreciarán algunas de las instrucciones de código de programación empleadas a lo largo de esta etapa.

3.4.2.1.10. *Inception A (Bloque 35x35).*

En esta primera sección se aborda el bloque Inception A, el cual tiene un diseño que consta de 3 ramificaciones y cada una de ellas contiene operaciones propias de una CNN sobre los tensores de entrada de los bloques residuales. Dichos tensores representan a un conjunto de datos multidimensionales que en este caso son imágenes RGB de 182x182 pixeles. A continuación, en la Figura 49 se puede apreciar una porción de código empleada en su implementación.

```
def block35(net, scale=1.0, activation_fn=tf.nn.relu, scope=None, reuse=None):
    """Construye el bloque resnet 35x35"""
    with tf.variable_scope(scope, 'Block35', [net], reuse=reuse):#Crea una variable "Block35"
        with tf.variable_scope('Branch_0'):#Primera variable de ramificación del bloque residual
            tower_conv = slim.conv2d(net, 32, 1, scope='Conv2d_1x1')#Convolución 1x1
        with tf.variable_scope('Branch_1'):#Segunda variable de ramificación del bloque residual
            tower_conv1_0 = slim.conv2d(net, 32, 1, scope='Conv2d_0a_1x1')#Convolución 1x1
            tower_conv1_1 = slim.conv2d(tower_conv1_0, 32, 3, scope='Conv2d_0b_3x3')#Convolución 3x3
        with tf.variable_scope('Branch_2'):#Tercera variable de ramificación del bloque residual
            tower_conv2_0 = slim.conv2d(net, 32, 1, scope='Conv2d_0a_1x1')#Convolución 1x1
            tower_conv2_1 = slim.conv2d(tower_conv2_0, 32, 3, scope='Conv2d_0b_3x3')#Convolución 3x3
            tower_conv2_2 = slim.conv2d(tower_conv2_1, 32, 3, scope='Conv2d_0c_3x3')#Convolución 3x3
        mixed = tf.concat([tower_conv, tower_conv1_1, tower_conv2_2], 3)#Concatena los tensores de las
                                                                        #3 ramificaciones en uno solo
        up = slim.conv2d(mixed, net.get_shape()[3], 1, normalizer_fn=None,#Convolución 1x1 final
                         activation_fn=None, scope='Conv2d_1x1')
        net += scale * up
        if activation_fn:
            net = activation_fn(net)#Agregando función de activación ReLU
    return net#Retorno de tensor del bloque residual
```

Figura 49. Construcción del bloque Inception A.

Fuente: Autoría

Además, en la Tabla 19 se muestran las dimensiones de cada bloque Inception de la red.

Tabla 19. Especificaciones del tamaño de cada bloque residual Inception.

Bloque residual Inception	**Tamaño de Filtro (filas*columnas*profundidad)**
Bloque Inception A	35x35x256
Bloque Inception B	17x17x896
Bloque Inception C	8x8x1792

Fuente: Autoría

Como se puede visualizar en el código de la Figura 49, a través de cada nodo de grafo[7] del bloque residual, se realizan operaciones de convolución y de activación (*ReLU*) de capa a través de cada ramificación de la CNN. Al final se concatenan dichos nodos y devuelven un nuevo tensor con características muy discriminantes de las imágenes del conjunto de datos.

A modo de ejemplo se muestra el código del bloque Inception A; las demás secciones se consiguen de manera similar, añadiendo o cambiando algunos nodos de grafo. El Anexo 4 muestra a mayor detalle las demás secciones.

3.4.2.1.11. *Reducción A (17x17).*

En esta sección se presenta el bloque de Reducción residual A, que consta de 2 ramificaciones con nodos de grafo muy similares a los bloques residuales anteriores (Figura 50). La única diferencia notable es el reemplazo de la operación de activación (*ReLU*) por una operación de capa *maxpooling* para cada sección de reducción. En la Tabla 20 se puede visualizar cada una de las dimensiones de cada bloque de reducción de la red.

Tabla 20. Especificaciones del tamaño de cada bloque de Reducción.

Reducción residual	**Tamaño de Filtro (filas*columnas*profundidad)**
Bloque de reducción A	17x17x806
Bloque de reducción B	8x8x1792

Fuente: Autoría

[7] Grafo: Representación de operaciones matemáticas codificadas (convolución, maxpooling, ReLU, PReLU, Softmax, entre otras).

Como se puede visualizar en el código de la Figura 50, a través de cada nodo de grafo del bloque de reducción, se realizan operaciones de *convolución* y de *maxpooling* a través de cada ramificación de la CNN. Al final se concatenan dichos nodos y devuelven un nuevo tensor con características mucho más discriminantes que los obtenidos en los bloques residuales *Inception* en cada una de sus variaciones. Las demás secciones se consiguen de manera similar, añadiendo o cambiando algunos nodos de grafo. El Anexo 4 muestra a mayor detalle las demás secciones.

```
def reduction_a(net, k, l, m, n):
    with tf.variable_scope('Branch_0'):#Primera variable de ramificación del bloque residual
        tower_conv = slim.conv2d(net, n, 3, stride=2, padding='VALID',
                                 scope='Conv2d_1a_3x3')#Convolución 3x3
    with tf.variable_scope('Branch_1'):#Segunda variable de ramificación del bloque residual
        tower_conv1_0 = slim.conv2d(net, k, 1, scope='Conv2d_0a_1x1')#Convolución 1x1
        tower_conv1_1 = slim.conv2d(tower_conv1_0, l, 3,
                                    scope='Conv2d_0b_3x3')#Convolución 3x3
        tower_conv1_2 = slim.conv2d(tower_conv1_1, m, 3,
                                    stride=2, padding='VALID',
                                    scope='Conv2d_1a_3x3')#Convolución 3x3
    with tf.variable_scope('Branch_2'):#Tercera variable de ramificación del bloque residual
        tower_pool = slim.max_pool2d(net, 3, stride=2, padding='VALID',
                                     scope='MaxPool_1a_3x3')#Agregando capa MaxPooling con filtro
                                                            #3x3
    net = tf.concat([tower_conv, tower_conv1_2, tower_pool], 3)#Concatena los tensores de las
                                                               #3 ramificaciones en uno solo
    return net#Retorno de tensor del bloque residual
```

Figura 50. Construcción del bloque de reducción A.

Fuente: Autoría

Una vez especificados los principales bloques de la red es momento de unificar todos y añadir una capa final *maxpooling, dropout y fully connected* de forma secuencial. El siguiente fragmento de código de la Figura 51 muestra dichas operaciones y a continuación se ofrece una breve explicación acerca de su funcionalidad dentro de la CNN.

```
    net = slim.avg_pool2d(net, net.get_shape()[1:3], padding='VALID',
                          scope='AvgPool_1a_8x8')#Capa MaxPooling promedio de
                                                 #todos los bloques residuales
                                                 #de la CNN
    net = slim.flatten(net)#Alineamiento de los parametros en un tensor unidimensional

    net = slim.dropout(net, dropout_keep_prob, is_training=is_training,
                       scope='Dropout')#Regularización de la CNN

    end_points['PreLogitsFlatten'] = net

net = slim.fully_connected(net, bottleneck_layer_size, activation_fn=None,
        scope='Bottleneck', reuse=False)#Capa de convolución 1x1 con una
                                        #profundidad de 512-D
```

Figura 51. Unificación de las salidas de los bloques de la red a través de las capas dropout y fully connected.

Fuente: Autoría

3.4.2.1.12. *Capa dropout.*

La capa *dropout* se basa en el principio de regularización de la red neuronal. Esto es necesario ya que durante el entrenamiento de la red usando un tamaño previamente establecido de mini-lote dentro del conjunto de entrenamiento, cada neurona activa una probabilidad fija (generalmente de 0.5, para este estudio se ha establecido en 0.4) que habitualmente genera un efecto de sobreajuste (overfitting[8]) de la red, con lo que la CNN no podrá generalizarse en nuevos ejemplos de imágenes de rostros y establecer una correcta predicción. La solución consiste en desconectar un porcentaje de las neuronas aleatoriamente en cada iteración del proceso de entrenamiento intencionalmente (Figura 52). Esta capa está presente en todos los modelos de redes neuronales por los buenos resultados que se obtiene en la mejoría de la generalización la red. Consecuentemente se obtendrán características descriptivas bastante generalizables de los rostros con los que ha sido entrenada la red y con rostros que no lo han sido.

[8] Overfitting: Efecto de sobre entrenamiento de un modelo, produciendo así predicciones erróneas (high bias).

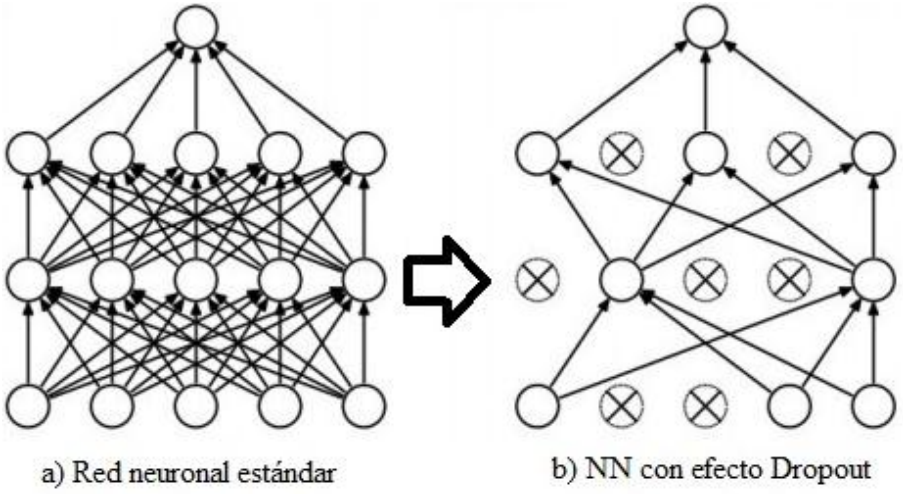


Figura 52. Efecto de la capa dropout sobre las conexiones de una red estándar.

Fuente: Autoría

3.4.2.1.13. *Capa fully connected.*

Las capas *fully connected* son esencialmente capas de convolución con tamaño de filtro 1x1. En una CNN típica, estas capas se colocan hacia el final de la arquitectura conectando densamente a todas las unidades de las capas anteriores. Esto es importante para la capacitación usando métodos o funciones de pérdida tales como Softmax y pérdida de centro, ya que se espera una única tubería (salida) que contenga las características extraídas hasta el momento de los rostros a través de toda la serie de capas interconectadas en la CNN. Para este estudio se ha establecido un tensor de salida de 512-D (mediciones/bytes) como capa fully connected, el cual es el tamaño total de la incrustación para cualquier rostro.

Finalmente, una vez que ha sido implementada la CNN en el intérprete de Python se procede a definir el entrenamiento del modelo bajo la supervisión conjunta de la pérdida de Softmax y la pérdida de centro especificadas en el trabajo de Yandong Wen & Kaipeng Zhang (2016).

3.4.2.1.14. *Funciones de pérdida.*

Las funciones de pérdida que se analizan a continuación se han empleado con el fin de crear un modelo de aprendizaje profundo bastante discriminatorio entre clases en la tarea de reconocimiento de rostros. En la capa final de la CNN propuesta se emplean un conjunto de funciones de pérdida (*costo o error*) para estimar la calidad de las predicciones realizadas por la red en los datos de entrenamiento, para los cuales se conocen las etiquetas de clase. Cabe recalcar que una función de pérdida cuantifica la diferencia entre la salida estimada del modelo (la predicción) y la salida correcta (la verdad). Además, las funciones de pérdida solamente se optimizan durante el proceso de entrenamiento de la red. Como se mencionará en el siguiente apartado existen dos funciones de pérdida que usadas en conjunto logran una alta eficiencia del sistema de identificación facial.

- ***Pérdida de Softmax.***

La función de *pérdida Softmax* en la variante de entropía cruzada, es un tipo de función de activación que permite interpretar la cadena de salidas de los bloques residuales como un vector discreto de distribución de probabilidad categórica multiclase muy útil en las CNN´s. Dicho enfoque ayuda a evaluar y trabajar valores bastante pequeños de la función de pérdida que puedan dificultar el ajuste de los pesos y bias de la red. Esto es esencial para este proyecto, ya que se busca separar de manera bastante precisa las características únicas que identifican a los individuos de un gran conjunto de clases (VGGFace2). La importancia del uso de esta función radica en que el entrenamiento de la CNN se la hace como un clasificador de clases con lo que se consigue obtener profundidad de características, aunque no lo suficientemente discriminatorias para la tarea de reconocimiento facial. La Ecuación 8 muestra el modelado matemático de esta función de pérdida.

$$L_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

Ecuación 8. Definición matemática de la pérdida Softmax.

Fuente: Adaptado de (Wen, Zhang, Li, & Qiao, 2016)

De la presente ecuación hay que mencionar que la expresión $W_{y_i}^T x_i + b_{y_i}$ generalmente representa la hipótesis del modelo de forma $Ax + B$, la cual es utilizada como base en casi todos los modelos de algoritmos de aprendizaje automático y profundo. En este caso en cualquier tamaño de mini-lote, x_i denota la i-ésima característica profunda que pertenece a la clase y_i, W_j denota la j-ésima columna de los pesos de la red y b es la unidad de bias. Además, el logaritmo que se emplea alrededor de la función de Softmax se denomina entropía cruzada. La implementación en Python de esta función de pérdida se puede apreciar en la Figura 53.

```
cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(
    labels=label_batch, logits=logits, name='cross_entropy_per_example')#Cálculo de la función softmax(cross-entropy)
cross_entropy_mean = tf.reduce_mean(cross_entropy, name='cross_entropy')#Calcula el promedio del tensor de salida softmax
```

Figura 53. Implementación de la pérdida Softmax en Tensorflow.

Fuente: Autoría

Tal y como se menciona en el trabajo de Yandong Wen & Kaipeng Zhang (2016) por sí sola la función de activación Softmax únicamente fomenta la separabilidad de las características y entrega un desempeño promedio en la tarea de reconocimiento facial, ya que las características resultantes no son lo suficientemente efectivas para la tarea de clasificación e identificación de rostros.

- ***Pérdida de centro.***

En esta sección se presenta una función de pérdida denominada *pérdida de centro*, la cual mejora de manera eficiente el poder discriminativo de las características profundamente aprendidas en la CNN en la etapa de la capa Softmax. Específicamente se aprende un centro, el cual es representado por un vector de dimensiones similares a las características profundas de cada clase. Durante el entrenamiento, se actualiza al mismo tiempo el centro y se minimizan las distancias entre las características profundas y sus centros de clase correspondientes. La Ecuación 9 muestra el modelado matemático de esta función de pérdida.

$$L_C = \frac{\lambda}{2}\sum_{i=1}^{m}\left\|x_i - c_{y_i}\right\|_2^2$$

Ecuación 9. Definición matemática de la pérdida Central.

Fuente: Adaptado de (Wen, Zhang, Li, & Qiao, 2016)

De la presente ecuación se concluye que c_{y_i} denota el y_i-ésimo centro de clase de características profundas promediadas en cada iteración basada en mini-lote y x_i es la i-ésima característica profunda tomada de la clase y_i, esto sirve para minimizar las variaciones dentro de la clase mientras se mantienen las características de las diferentes clases separables. Un detalle importante a destacar es que el hiperparámetro λ permite la distribución de características profundas aprendidas de mejor manera y equilibra las dos funciones de pérdida, dominando las variaciones intraclase según se ajuste su valor; el valor establecido para propósitos de este estudio oscila entre 0.95 a 1 (específicamente 0.95) debido a que los mejores resultados en términos de reconocimiento facial se obtienen entre ese rango de valores según el trabajo de Yandong Wen & Kaipeng Zhang (2016). La implementación en Python de esta función de pérdida se puede apreciar en la Figura 54.

```
nrof_features = features.get_shape()[1]#Tensor con las dimensiones del numero de caracteristicas
                                       #profundas
centers = tf.get_variable('centers', [nrof_classes, nrof_features], dtype=tf.float32,
    initializer=tf.constant_initializer(0), trainable=False)#Obteniendo datos de numero de clases
                                                            #y numero de caracteristicas profundas
                                                            #extraidas
label = tf.reshape(label, [-1])#Redimensionamiento de un tensor para las etiquetas de clase
centers_batch = tf.gather(centers, label)#Produce un tensor con las dimensiones de los centros
                                         #y etiquetas obtenidas de la actualizacion de los mini-lotes
diff = (1 - alfa) * (centers_batch - features)#Diferencia centro y caracteristicas profundas
centers = tf.scatter_sub(centers, label, diff)
with tf.control_dependencies([centers]):
    loss = tf.reduce_mean(tf.square(features - centers_batch))#Perdida de centro de clase
return loss, centers
```

Figura 54. Implementación de la pérdida Central en Tensorflow.

Fuente: Autoría

Al juntar las dos funciones de pérdida en la fase final de la CNN, el modelado matemático resultante es representado con la Ecuación 10:

$$L = L_S + \lambda L_C = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \left\| x_i - c_{y_i} \right\|_2^2$$

Ecuación 10. Definición matemática unificada de la pérdida Softmax y Central.

Fuente: Adaptado de (Wen, Zhang, Li, & Qiao, 2016)

Donde L_S indica la pérdida de Softmax y L_C indica la pérdida de centro de manera simplificada. Una vez implementada la supervisión conjunta de estas señales de pérdida, se desarrolla la etapa 1 de entrenamiento empleando una variación de algoritmo de descenso de gradiente estocástico (*SGD*), con el que se obtiene un modelo de aprendizaje profundo encargado de establecer 512 mediciones únicas representativas de un rostro lo suficientemente definidas que solo se pueden conseguir combinando dichas metodologías de vanguardia. En la etapa 2 se muestra sobre un espacio bidimensional (*2D*) la distribución de las características de cada rostro, en el cual se evidenciará su separabilidad entre clases, para su posterior clasificación mediante un algoritmo de aprendizaje automático supervisado denominado Máquinas de Vector Soporte (SVM), en este caso. Para obtener más detalles de la implementación de estas funciones de pérdida, el Anexo 5 muestra de manera detallada dichas configuraciones.

3.4.2.1.15. *Metodología de entrenamiento "Optimizador ADAM".*

Finalmente, una vez definidas todas las secciones que componen la arquitectura Inception-Resnet-V1 es preciso entrenar la CNN con un método muy común en la comunidad de aprendizaje automático y que también es aplicable para el aprendizaje profundo, este se denomina descenso de gradiente estocástico (*SGD*), el cual acelera el proceso de aprendizaje en búsqueda de la convergencia del modelo en la dirección del mínimo global a través de constantes derivaciones de la función de costo o de pérdida representativa de la red. En este proyecto se ha decidido emplear la variación de *SGD* denominado "*Adam*", el cual trae importantes mejoras con respecto a la rapidez en la convergencia de las CNN con grandes conjuntos de datos y/o espacios de alta dimensión; los principios matemáticos del algoritmo "*Adam*" se muestran a continuación en las ecuaciones (*a*,*b*,*c*,*d*) de la Figura 55.

$$(a)\ \ g_t = \nabla_\theta f_t(\theta_t - 1)$$

$$(b)\ \ m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t$$

$$(c)\ \ v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2$$

$$(d)\ \ \theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}}$$

Figura 55. Definición matemática del algoritmo de optimización ADAM.

Fuente: Adaptado de (Kingma & Ba, 2014)

Esencialmente Adam es un algoritmo para la optimización basada en gradientes de primer orden de las funciones objetivas estocásticas (g_t), en el cual se emplean estimaciones adaptativas de momentos de orden inferior (m_t, v_t) y se calculan individualmente las tasas de aprendizaje adaptativo (α) para los primeros y segundos momentos de los gradientes. El método tiene la ventaja de tratar con gradientes dispersos y la capacidad de lidiar con objetivos no estacionarios. El método es sencillo, computacionalmente eficiente y tiene pequeños requerimientos de memoria

(Kingma & Ba, 2014). Actualmente este método se encuentra disponible en la biblioteca Tensorflow y su implementación se la puede encontrar en el script "faceUTN_FICA.py" en el Anexo 5. Además, hay que mencionar que la CNN ha sido entrenada bajo el algoritmo de propagación hacia atrás en la búsqueda de los pesos (W_j^T) y bias (b_j) adecuados para que el valor de la función de pérdida conjunta sea mínima; los parámetros empleados por el algoritmo Adam para la optimización de la función de pérdida y el entrenamiento del modelo se aplican sobre 275 épocas[9] (1000 pasos por época) de forma variable con un tamaño de mini-lote de 90 ejemplos de entrenamiento. Las tasas de aprendizaje variable a tomar en cuenta sobre cada intervalo de entrenamiento son:

- Intervalo de 0 a 99 épocas: $\alpha = 0.05$
- Intervalo de 100 a 199 épocas: $\alpha = 0.005$
- Intervalo de 200 a 275 épocas: $\alpha = 0.0005$

En cuanto a los momentos de los gradientes:

- Primer y segundo momento de orden inferior: $m_t = 0.9$, $v_t = 0.999$

Además, el modelo se evalúa cada 5 épocas en el proceso de entrenamiento con un conjunto de validación de 33100 imágenes, correspondientes al 1% del conjunto de entrenamiento. Dicho conjunto de validación brinda un pequeño impulso en la precisión de modelos con arquitecturas muy profundas, además de evitar el sobreajuste (ajuste excesivo de parámetros) de la CNN. Finalmente mencionar que, en este caso en particular el entrenamiento se ha completado en su totalidad bajo el uso intensivo de potencia de cómputo paralelo de los módulos CPU/GPU del

[9] Época: Cada iteración que actualiza los parámetros de SGD utilizando el conjunto de entrenamiento completo.

servidor especificado en la sección de requerimientos de hardware, alcanzando una precisión y pérdida de entrenamiento de 92.9% y 2.1 respectivamente, Como se apreciará en las secciones 3.4.2.1.17 y 4.4, el modelo presenta un muy buen desempeño en diferentes conjuntos de datos de prueba. Finalmente, las múltiples operaciones que se realizan en cada capa de la CNN de extremo a extremo, requieren aproximadamente de 168 a 175 horas de entrenamiento continuo para obtener un modelo extractor de características de 512-D (bytes) generalizable para cualquier rostro.

3.4.2.1.16. *Modelo de incrustaciones faciales de 512-D.*

Al final del entrenamiento exhaustivo de toda la arquitectura CNN se obtiene un archivo constructor de incrustaciones faciales genérico pre-entrenado. Como se puede visualizar en la Figura 56 determina 512 mediciones únicas para una nueva entrada.

Figura 56. Conjunto de 512 mediciones obtenidas de un rostro.

Fuente: Autoría

3.4.2.1.17. *Precisión.*

Para evaluar cuantitativamente la precisión del modelo se ha empleado el método *cross validation (10 fold),* que consiste en la partición o división de un conjunto de datos, en un conjunto

de entrenamiento y prueba para evaluar el rendimiento del modelo; en este caso el conjunto de datos VGGFace2 fue seleccionado para el entrenamiento previo por parte de la CNN propuesta y el conjunto de datos LFW (Labeled Faces in the Wild) para la realización de las pruebas respectivas. LFW es un conjunto de datos muy conocido, el cual contiene aproximadamente 13000 fotografías de 5000 individuos, lo que lo hace bastante adecuado para evaluar la precisión con la que las incrustaciones faciales son generadas para una gran cantidad de identidades (clases). Los resultados obtenidos del test de precisión del modelo se pueden visualizar en la Figura 57 y a continuación se presentan algunos detalles adicionales de la forma de evaluación del modelo.

Figura 57. Precisión de modelo extractor de características faciales.

Fuente: Autoría

Primeramente, es importante mencionar que se ha empleado un listado (*.txt*), donde se define el conjunto de pares de imágenes a seleccionar del conjunto de imágenes LFW (Figura 58), ya sea de una misma clase o de diferentes clases; dicho listado especifica los nombres de los directorios (etiquetas o nombres de clase) y los índices de ordenación de las imágenes en cada directorio para la posterior extracción de los pares de incrustaciones faciales (x, y).

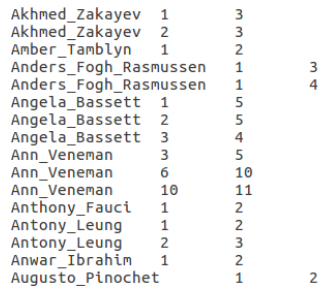


Figura 58. Ejemplo de listado de conjunto de pares.

Fuente: Autoría

En la evaluación del modelo se utilizó la métrica de "*similitud del coseno*" para medir la distancia de los pares de incrustaciones faciales extraídas y verificar la similitud o disparidad entre el par de imágenes dadas, esto quiere decir que si hay una distancia bastante cercana ($dcoseno \cong 0$) las imágenes corresponden a una misma persona, caso contrario son imágenes de diferentes identidades ($dcoseno \cong 1$); la Ecuación 11 muestra el modelo matemático de la distancia del coseno. Finalmente se calcula el promedio total de las distancias obtenidas y se presentan los resultados de la precisión del modelo como se puede apreciar en la Figura 57.

$$dcoseno(x, y) = \frac{\cos^{-1}\left(\frac{x^T y}{\|x\|\|y\|}\right)}{\pi}$$

Ecuación 11. Distancia del coseno.

Fuente: Adaptado de (Nguyen & Bai, 2010)

En conclusión, en esta etapa se han desarrollado e implementado los conceptos necesarios de aprendizaje profundo para el entrenamiento de un modelo de incrustaciones faciales con una tasa de precisión bastante aceptable de 0.99433% (99.433%) y con un error de mínimo de generalización de 0.00271%(0.271%), lo que resulta bastante conveniente para el objetivo de este proyecto. Las implementaciones en Python de los parámetros (Conjunto LFW, modelo de aprendizaje profundo, listado de pares de imágenes) usados para la evaluación del modelo se detallan en el Anexo 6.

3.4.2.2. Segunda etapa.

Esta etapa detalla el procedimiento para la construcción del conjunto de datos personalizado, y la descripción del modelo de aprendizaje automático empleado para la clasificación de las características que se obtienen de cada clase del conjunto de datos.

3.4.2.2.1. *Conjunto de datos de entrenamiento personalizado.*

Esta sección tiene el propósito de detallar el procedimiento general para la recolección del conjunto de datos personalizado, que básicamente conlleva la creación de una pequeña base de datos, la cual contiene fotografías de los rostros pertenecientes a los usuarios directos e indirectos del sistema en la FICA; para lograr esto se emplea el script "dataset_creator_GUI.py" (Anexo 7). Esta tarea conlleva gran importancia debido a que una de las funciones del sistema a través de este módulo implica la captura y recolección de alrededor de 100 a 200 imágenes de todos los ángulos faciales de cada individuo registrado en el sistema. Como es de conocimiento general en el aprendizaje profundo, cuando se emplean conjuntos de datos más extensos y diversos en el entrenamiento de una arquitectura CNN la precisión del modelo tiende a aumentar significativamente, es por eso que el modelo de aprendizaje automático también debe entrenarse con una cantidad promedio *(m>=1*00) de ejemplos de entrenamiento para lograr una precisión aceptable. Además, es preciso indicar que el registro de información personal tales como: nombres, apellidos, número de cédula, edad, carrera, y nivel es llevado a cabo sobre una tabla de la base de datos local empleada. Finalmente, una vez que el conjunto de datos ha sido elaborado, este se encuentra listo para la posterior extracción de incrustaciones faciales con la finalidad de construir un modelo de aprendizaje automático (clasificación) para la identificación facial, el cual se implementará en las siguientes secciones. La Figura 59 muestra las fotografías de un usuario almacenados en un subdirectorio del sistema en formato ".jpg"; siguiendo las pautas y/o características del conjunto de datos empleado en el entrenamiento de la arquitectura CNN, se realiza el proceso de alineamiento, convirtiéndolas al formato ".png".

El Anexo 7 muestra la implementación completa de las funciones previamente mencionadas.

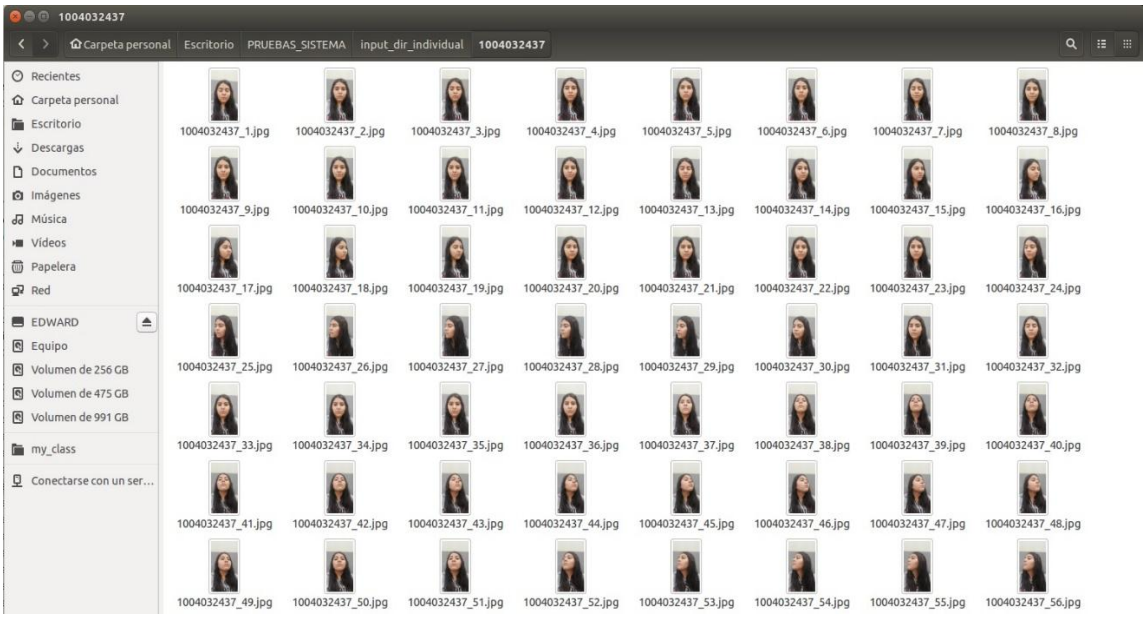


Figura 59. Construcción de conjunto de imágenes faciales.

Fuente: Autoría

3.4.2.2.2. *Máquinas de Vector Soporte (SVM).*

Dentro de esta sección se analiza el algoritmo de aprendizaje automático a emplear para la clasificación de cada conjunto de incrustaciones faciales correspondientes a cada individuo registrado en el sistema. A partir de este apartado se hará mención de los conceptos que involucran la implementación del algoritmo de clasificación de datos a usar en este proyecto y que además resulta ser el método más efectivo y utilizado dentro de muchas aplicaciones de aprendizaje automático, el cual se denomina: Máquinas de Vector Soporte ("Support Vector Machine").

Es importante destacar que el algoritmo de SVM fue seleccionado debido al respaldo que tiene en varios trabajos de impacto sobre reconocimiento facial profundo, tales como: el trabajo de Zhu (2014) & Schroff (2015), donde se mencionan que "*es bastante conveniente usar SVM en última etapa de la CNN, ya que logra mayor precisión de clasificación de características profundamente aprendidas en la tarea de verificación facial*".

Básicamente SVM es un algoritmo de aprendizaje automático supervisado muy potente y versátil que se usa para problemas de clasificación o regresión, lo cual es perfecto para este proyecto, ya que SVM funciona al encontrar un hiperplano[10] lineal ideal que separa el conjunto de datos de entrenamiento en dos o más clases, dependiendo de la distribución general de las características de los sujetos de entrenamiento y de sus etiquetas. Como existen muchos de estos hiperplanos lineales debido a la cantidad de sujetos que se registran en el sistema, el algoritmo SVM intenta encontrar el hiperplano de separación óptimo, que se logra de manera intuitiva cuando la distancia (también conocida como margen) a las muestras de datos de entrenamiento más cercanas es la más grande posible. Esto se debe a que, en general, cuanto mayor sea el margen, menor será el error de generalización del modelo, por lo que consecuentemente se obtiene mayor precisión en la tarea de clasificación e identificación facial.

Matemáticamente SVM se define como un modelo lineal de margen máximo que se denota a través de un hiperplano expresado mediante la ecuación de la forma $w^T x_i + b = 0$, donde w es el vector normal al hiperplano, x_i representa a los datos de entrenamiento, y b se denomina el término de bias. Adaptando este modelo a nuestro problema de clasificación: a continuación se le proporciona el conjunto de datos de incrustaciones faciales de n ejemplos de entrenamiento de la siguiente forma: $\{(x_i, y_i), \dots, (x_n, y_n)\}$, donde x_i es un vector de 512 mediciones, y y_i es la clase o identidad a la que pertenece x_i. A modo de ejemplo en la Figura 60 se puede apreciar claramente la clasificación de dos clases de datos característicos a través de un hiperplano lineal (*decision boundary*) trazado por el algoritmo SVM conjuntamente con sus vectores de soporte.

[10] Hiperplano: Distribución lineal de una función óptima de clasificación.

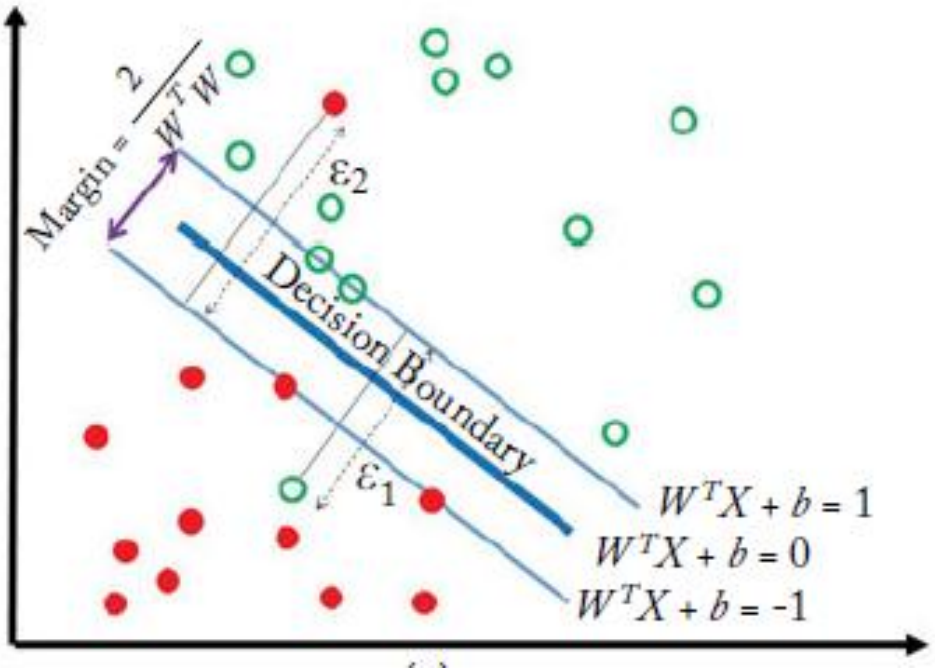


Figura 60. Clasificación de clases mediante el uso del algoritmo SVM.

Fuente: Adaptado de (Scikit-Learn Org, 2018)

Para fines de este proyecto, después de obtener el modelo extractor de incrustaciones faciales de la etapa 1 y posteriormente usar sobre un conjunto de datos que nunca ha visto el modelo dentro de su etapa de entrenamiento, es imprescindible obtener los 512 valores numéricos generados por el modelo a partir de los vectores de mediciones faciales de cada clase y emplear sobre ellos el algoritmo SVM (one vs. one) para clasificar agrupamientos característicos de identidad de cada individuo, tal y como se puede apreciar en la Figura 61, donde se distribuyen dentro de un espacio bidimensional los vectores de incrustación facial, lo cual denota secciones susceptibles a la clasificación con bastante prominencia. Para la representación de este gráfico se ha realizado una reducción de las dimensionalidades del número de mediciones de cada vector, disminuyendo a solo una medición por cada muestra de clase (imagen), mediante una función de la biblioteca Scikit-Learn denominada Incrustación de Vecinos Estocásticos Distribuidos en t (t-SNE) (van der Maaten & Hinton, 2008).

La técnica t-SNE es usada a menudo para obtener una representación visual (gráfico 2D) de las características extraídas de las imágenes de entrenamiento de un conjunto de datos en la penúltima capa de una CNN, Esto es importante ya que proporciona una visión holística del conjunto de datos y muestra la calidad de las representaciones de características aprendidas para diferentes clases o agrupaciones (Khan, Rahmani, Shah, & Bennamoun, 2018).

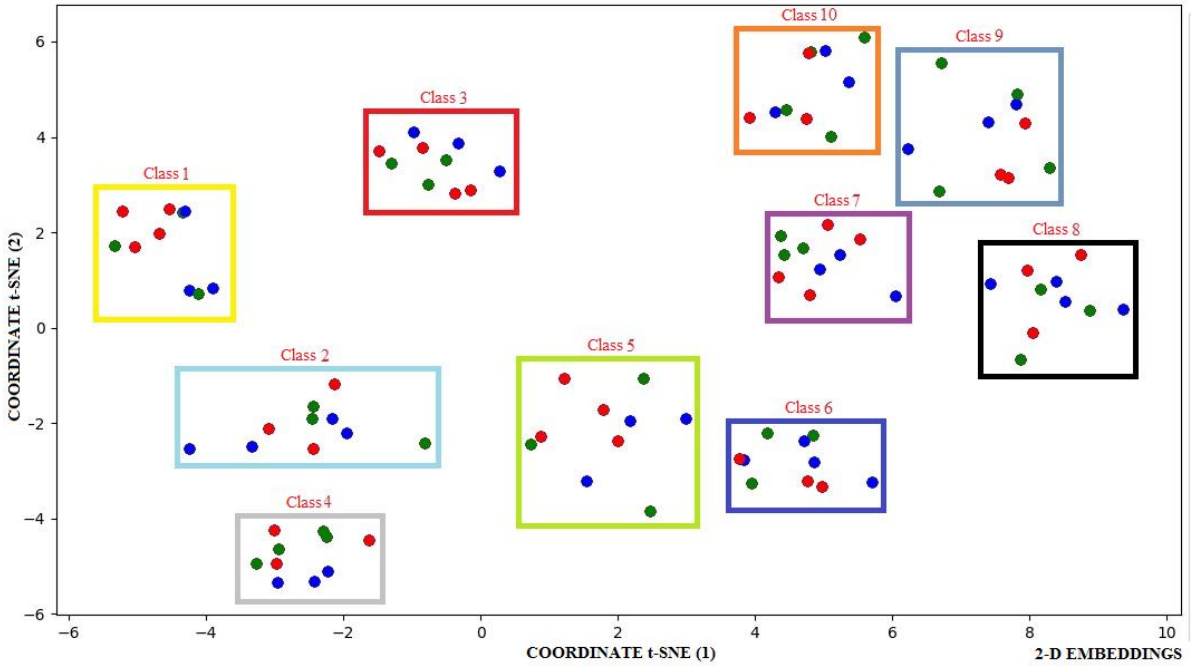


Figura 61. Distribución de incrustaciones faciales en un espacio bidimensional (2D).

Fuente: Autoría

Finalmente, es conveniente mencionar que: resulta bastante beneficioso el uso de esta técnica, puesto que, ayuda a determinar el tipo de método ideal a usar en un problema específico de clasificación, donde el conjunto de características (x) y etiquetas (y) de entrenamiento pueden variar indistintamente. En este caso en particular, según el gráfico de la Figura 61, se pudo corroborar visualmente que, las incrustaciones muestran zonas de clasificación que pueden ser abordadas exitosamente por el clasificador SVM.

3.4.2.2.3. *Entrenamiento de clasificador.*

El módulo Scikit-Learn para aprendizaje automático en Python implementa el clasificador SVM, por lo que en este punto se procede a cargar los datos de entrenamiento que en este caso son los vectores de incrustación facial con sus respectivas etiquetas de clase. Para entrenar el algoritmo se emplea la función "*.fit*" que recibe los datos de entrenamiento mencionados anteriormente (Figura 62) y crea un modelo a modo de fichero (.pkl) que contiene codificadas las incrustaciones y las etiquetas.

```
# Entrenando clasificador
print('Entrenando clasificador SVM Support Vector Machine')
model = SVC(C=1,kernel='linear', probability=True)#Definición del modelo SVM lineal
model.fit(emb_array, labels)#Entrenamiento del modelo
```

Figura 62. Entrenamiento del algoritmo SVM con Scikit-Learn.

Fuente: Autoría

Hay que mencionar que el clasificador SVM dispone de algunas variantes en el tipo de función de *kernel* que se puede usar, tales como: *lineal, sigmoidal, y polynomial*. Por lo que se ha realizado una serie de pruebas preliminares en las cuales se ha determinado el kernel óptimo con la mejor precisión de clasificación. La sección 3.4.2.2.4 muestra a detalle las pruebas realizadas. Para fines de este proyecto se escogió el kernel lineal.

Para obtener más detalles de la implementación del clasificador SVM, el Anexo 8 muestra de manera detallada las configuraciones mencionadas.

3.4.2.2.4. *Precisión.*

Para evaluar la precisión de este modelo se ha usado el método *cross validation (10 fold)* con el cual se ha empleado pequeñas porciones del conjunto de datos VGGFace2. Cada porción consta de una muestra de tamaño de 100 clases, la cual se incrementa hasta alcanzar una agrupación de 400 clases. Hay que mencionar que cada porción seleccionada (100,200,300 y 400

clases) ha sido dividida en un conjunto de entrenamiento y un conjunto de pruebas, donde el 80% de las imágenes de cada clase son usadas para el entrenamiento del clasificador y el 20% restante para comprobar la precisión del mismo. El conjunto de pruebas contiene 20 fotografías dentro de cada clase, donde se aplica la función de Scikit-Learn denominada "*.predict_proba*" con la finalidad de verificar la correlación del conjunto de pruebas y el conjunto de datos de entrenamiento provisto y establecido previamente sobre el modelo de aprendizaje automático.

SVM dispone de algunas funciones de kernel con diferentes formulaciones matemáticas, tal y como se puede apreciar en la Tabla 21. Las tasas de precisión obtenidas con conjuntos de datos de diferente tamaño difieren con cada kernel analizado, esto se debe a que al aumentar el número de clases también se incrementan el número de incrustaciones a clasificar lo que consecuentemente aumenta la complejidad del clasificador en espacios de alta dimensión con funciones de clasificación distintas. Además, el valor del hiperparámetro C se ha establecido en 1 para regularizar más las estimaciones del modelo, con propósitos de obtener clasificaciones satisfactorias en todos los experimentos.

Tabla 21. Precisión de clasificación de incrustaciones faciales con diferente kernel SVM.

Kernel	**Formulación matemática**	**Conjunto de datos (100 clases)**	**Conjunto de datos (200 clases)**	**Conjunto de datos (300 clases)**	**Conjunto de datos (400 clases)**
Lineal	$\langle x, x' \rangle$	0.953(%)	0.932(%)	0.924(%)	0.915(%)
Sigmoidal	$\tanh(y\langle x, x' \rangle + r)$	0.945(%)	0.933(%)	0.918(%)	0.907(%)
Polynomial	$(y\langle x, x' \rangle + r)^d$	0.010(%)	0.005(%)	0.003(%)	0.001(%)

Fuente: Autoría

Los resultados del test de precisión del modelo con la mejor precisión encontrada se pueden visualizar en la Figura 63 con la variación de kernel lineal.

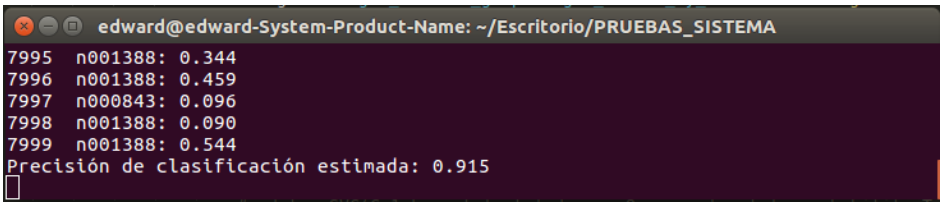


Figura 63. Prueba de precisión alcanzada por el modelo SVM lineal.

Fuente: Autoría

Como se puede apreciar en la Tabla 21 y la Figura 63, la precisión del clasificador SVM lineal alcanza una tasa del 0.915% (91.5%) de predicciones correctas en el conjunto de datos de 400 clases, por lo que se puede concluir con que el sistema puede entrenar y registrar a una cantidad considerable de usuarios.

Es preciso resaltar que la precisión del clasificador puede mejorarse con un conjunto de imágenes estándar de mejor calidad (720p, 1080p), puesto que, cada porción de clases contiene imágenes que han sido recolectados de internet a través del proyecto VGGFace2, las cuales poseen resoluciones bastante variables. No obstante, como un análisis preliminar se ha demostrado la efectividad de clasificación de las SVM´s sobre una gran cantidad de datos.

Hay que mencionar que en el Capítulo 4 se corroborará el correcto funcionamiento del clasificador, a través de un análisis cualitativo y cuantitativo, sobre rostros de individuos captados en secuencias de video en tiempo real recolectados desde la cámara IP especificada en la sección de requerimientos de hardware. La tarea mencionada es equivalente a realizar un test de prueba de desempeño del modelo, donde se obtendrán resultados de precisión, error, sensibilidad y especificidad de clasificación e identificación facial en ese instante, a medida que se analiza el flujo de video.

3.4.2.3. Tercera etapa.

En la última etapa se detallan las configuraciones realizadas para el acceso al flujo de video procedente de la cámara IP, y el conjunto de operaciones llevadas a cabo en el pre-procesamiento de la imagen. Además, se muestran las distintas funciones y/o módulos implementados bajo una interfaz de usuario (GUI).

3.4.2.3.1. *Configuración de cámara IP*.

El módulo recolector de fotogramas (cámara) especificado en la sección de requerimientos de hardware consta de una cámara tipo IP de la marca "HIKVISION" (HIKVISION Digital Technology Co., 2019), la cual se caracteriza por poseer una conexión cableada directa mediante puerto ethernet y facilidad de administración a través del protocolo IP. La Figura 64 ilustra la conexión física de los principales componentes del sistema, donde se puede visualizar la interconexión del servidor y la cámara IP.

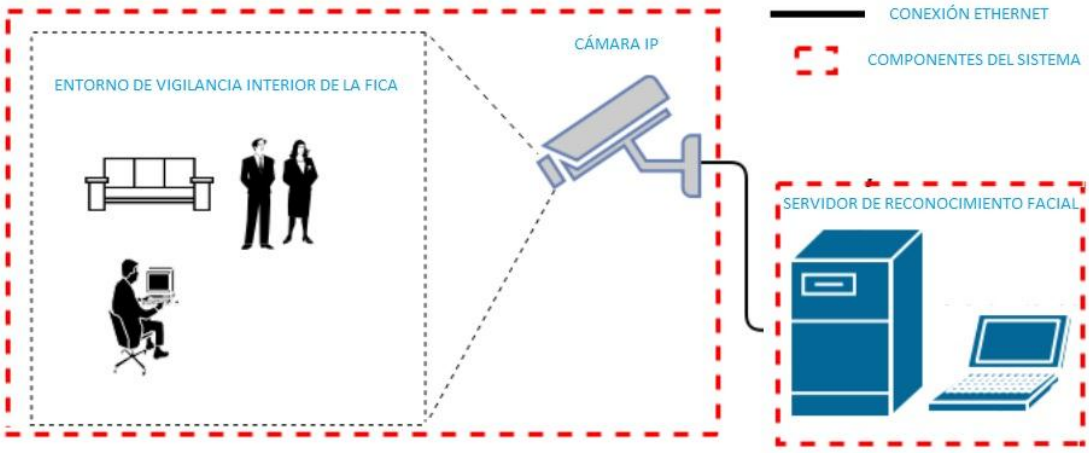


Figura 64. Esquema preliminar de conexión física del sistema.

Fuente: Autoría

Una vez establecida dicha conexión es necesario configurar el acceso mediante un número de puerto y una dirección IP dentro de la red local; estos parámetros son importantes ya que deben ser especificados en el método *VideoCapture(IP)* de la biblioteca OpenCV, el cual permite el acceso al flujo de video de la cámara en el intérprete de Python. La siguiente instrucción lleva a cabo lo anteriormente mencionado: **cv2.VideoCapture('protocol://IP:port/1').**

De existir autentificación, se debe especificar adicion-almente el username y password configurados en la cámara, tal y como muestra la siguiente instrucción: **cv2.VideoCapture('protocol://username:password@IP:port/1').**

3.4.2.3.2. *Preprocesamiento de la imagen.*

Como se mencionó en la sección de "Preprocesamiento de imágenes" de la etapa 1, la conversión de la zona ROI dado en el formato RGB al formato YCrCb permite el uso de la técnica de ecualización de histograma. Esta operación no altera el balance de color de la imagen, sino más bien mejora las condiciones de la misma en entornos cambiantes con respecto a la iluminación del lugar; esto es crucial para contrarrestar los efectos negativos de la luminosidad, ya que afectan la calidad de la imagen debido a los cambios bruscos de iluminación que se dan en el transcurso del día, así de esta manera el sistema se beneficia en gran manera.

3.4.2.3.3. *Desarrollo de la interfaz de usuario (GUI).*

Para el desarrollo de la interfaz de usuario del sistema se utilizó la biblioteca *Tkinter*, que contiene un kit completo de herramientas estándar para el desarrollo de GUI's en el intérprete de Python. A continuación, se mostrará de manera unificada a través de una interfaz de usuario, cada función implementada en el sistema de reconocimiento facial. El script "InterfaceSystemGUI.py" muestra la codificación empleada para el diseño de la ventana principal del sistema (Anexo 9).

En la Figura 65 se puede visualizar el conjunto de funciones disponibles en el sistema y son las siguientes: Habilitar cámara, Obtener IP, Detección facial, Crear conjunto de datos, Alineación del conjunto de datos, Entrenamiento de Clasificador, Reconocimiento Facial en Vivo, y Reconocimiento Facial en vivo + Detección de personas. En la siguiente sección se hará mención de las funciones que conllevan cada botón del sistema.

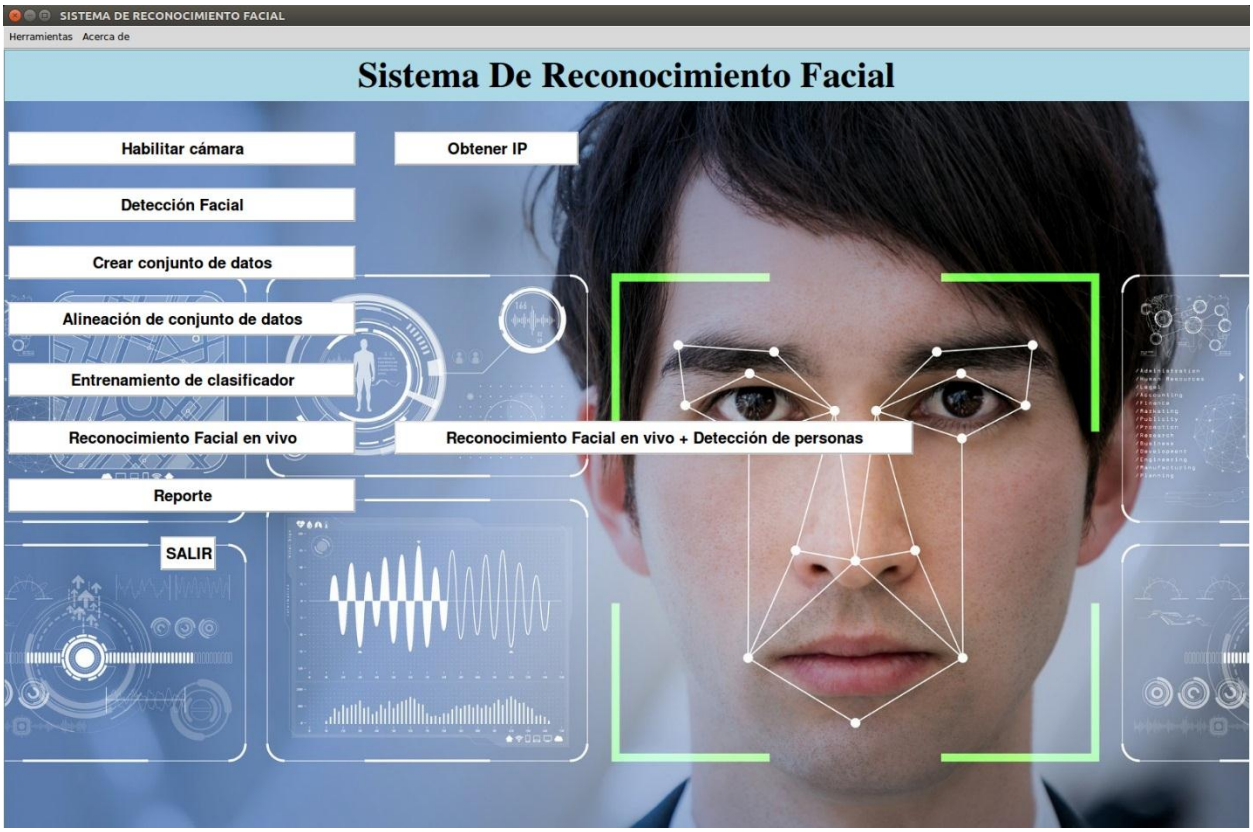


Figura 65. Interfaz de sistema de reconocimiento facial.

Fuente: Autoría

- ***Habilitar cámara.***

La opción de "Habilitar cámara" despliega una ventana que permite ingresar la dirección IP de la cámara y registrarla en el sistema (Figura 66). El botón "Probar" realiza una comprobación de la conexión de la cámara con el sistema para posteriormente registrarla con el botón "Registrar".

Figura 66. Ventana de ingreso de la dirección IP de la cámara.

Fuente: Autoría

- ***Detección facial.***

Esta opción solamente habilita el módulo de detección facial, el cual muestra un cuadro delimitador sobre la zona ROI y 5 puntos de referencia facial (Figura 67).

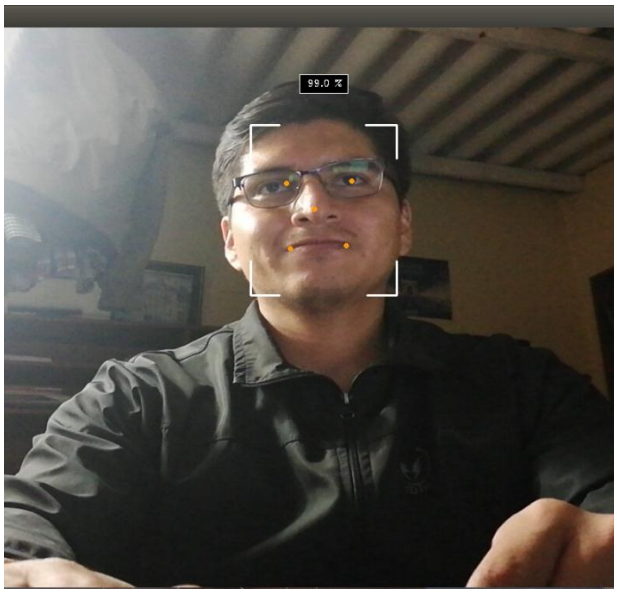


Figura 67. Módulo de Detección Facial.

Fuente: Autoría

- ***Crear conjunto de datos.***

Esta opción despliega un formulario de registro, donde se introducirán los datos de los usuarios del sistema (Figura 68). Los campos de información personal que se han determinado son: Cédula, Apellidos, Nombres, Carrera, Nivel, Email, Género, Edad, País y Ocupación. Una vez ingresada la información a través del botón "Registrar", el sistema procede a realizar la captura de fotografías del usuario para ser entrenado en el mismo.

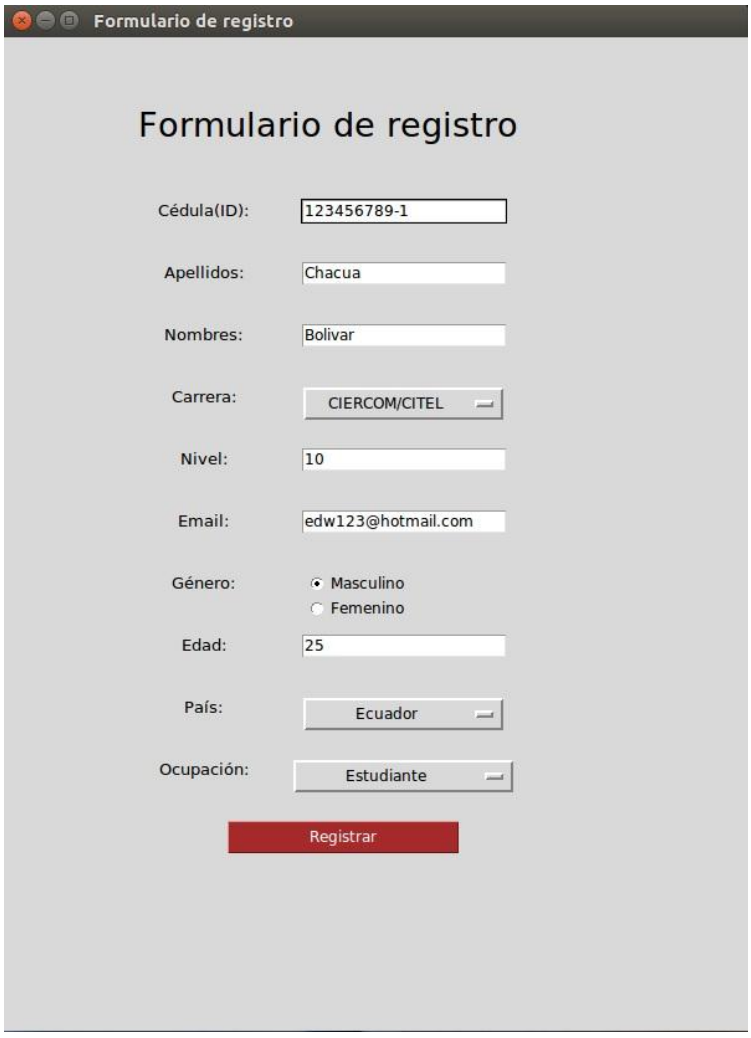


Figura 68. Ventana de formulario de registro.

Fuente: Autoría

- ***Alineación del conjunto de datos.***

En esta opción se realiza el procedimiento para la alineación del conjunto de imágenes recolectadas de los usuarios del sistema previamente. El botón “Consultar” muestra la cantidad de clases y el número total de imágenes encontradas, además, el botón “Iniciar” inicia el proceso de alineación de las imágenes y muestra el total de imágenes que han finalizado el preprocesamiento con éxito (Figura 69).

Figura 69. Ventana de preprocesamiento de datos.

Fuente: Autoría

- ***Entrenamiento de Clasificador.***

En esta opción se realiza el procedimiento para entrenar el conjunto de datos alineado previamente, mediante el algoritmo SVM. El botón “Consultar” muestra la cantidad de clases y el número total de imágenes encontradas. El botón “Iniciar” tiene dos variantes, que son: “ENTRENAMIENTO” y “EVALUACIÓN”; donde la primera opción se entrena el clasificador y la segunda opción presenta la precisión de clasificación estimada del mismo (Figura 70).

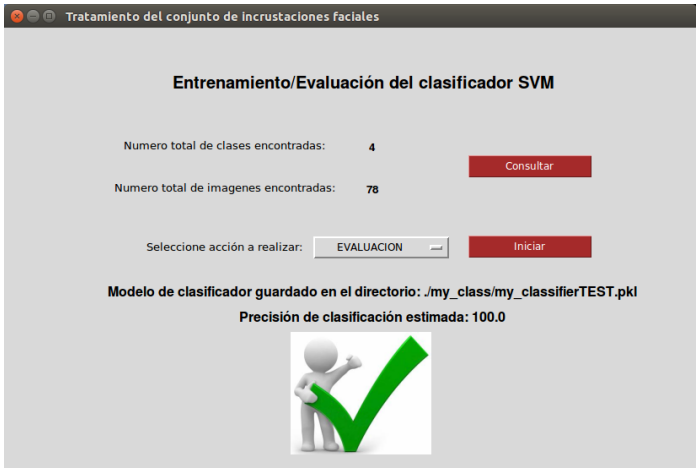


Figura 70. Ventana de entrenamiento de clasificador.

Fuente: Autoría

- ***Reconocimiento Facial en tiempo real.***

En esta opción se despliega una ventana donde se puede visualizar el funcionamiento del sistema sobre el flujo de video en tiempo real (Figura 71), donde se muestra la información de identidad (cédula, nombre) encontrada de un usuario.

Figura 71. Ventana de Reconocimiento Facial en tiempo real.

Fuente: Autoría

- ***Reconocimiento Facial en tiempo real + Detección de personas.***

En esta opción se muestra adicionalmente un módulo que ha sido implementado a modo de prueba, donde además de desempeñar la tarea de reconocimiento facial se determina la zona ROI de detección de una o varias personas (Figura 72).

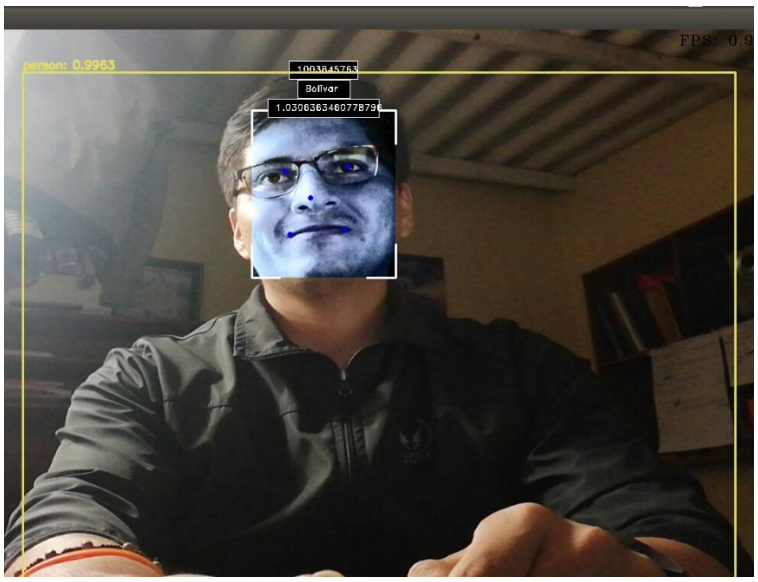


Figura 72. Ventana de Reconocimiento Facial + Detección de persona.

Fuente: Autoría

- ***Registro.***

Finalmente, la opción "Registro" guardará sobre unos archivos con extensión *".txt"* y *".xlsx"* la información de reconocimiento de los usuarios (cédula, nombres, apellidos, fecha y hora) que han sido registrados por parte del sistema. Además, se almacenará en un directorio del sistema las capturas de rostro de las personas que no han sido reconocidas.

4. CAPÍTULO IV. Implementación y Desarrollo de Pruebas.

Este capítulo describe el proceso de implementación del sistema de reconocimiento facial desarrollado en el Cap. 3 utilizando una de las cámaras del sistema CCTV disponible en la FICA-UTN. Además, se identifica la población objetivo del proyecto, con la cual se realiza las pruebas de desempeño del sistema y finalmente se presentan resultados y/o conclusiones del mismo.

4.1. Identificación de la población (muestra)

Para determinar una muestra representativa de la FICA, hay que extraer un número de elementos o sujetos del total de la población ($N \cong 2500$), para lo cual se emplea la Ecuación 12. Donde se considera un coeficiente de confianza adicional (P_α) al error de muestreo a cometer (tasa de error absoluto "e"), para obtener un tamaño de muestra proporcional (Pérez López, 2005).

$$n = \frac{\lambda_\alpha^2 * NPQ}{e^2(N-1) + \lambda_\alpha^2 PQ}$$

Ecuación 12. Cálculo del tamaño de muestra para una población finita.

Fuente: Adaptado de (Pérez López, 2005)

Donde:

n = Tamaño de la muestra.

λ_α= Nivel de confianza.

N = Tamaño de la población.

e = Error porcentual absoluto.

P = Probabilidad a favor.

Q = Probabilidad en contra.

Para este proyecto de estudio se ha considerado un tamaño de población de 2500 individuos, que se conforman en su gran mayoría de estudiantes matriculados en el periodo académico Abril-Agosto 2019 en todas las siete carreras de la FICA, además de docentes y personal administrativo de la misma facultad. Los parámetros considerados son los siguientes: el valor de N corresponde al tamaño aproximado de la población, que para este estudio en particular es de 2500 individuos, para el nivel de confianza λ_α se ha tomado un valor del 98% ($\lambda_\alpha = 2.33$), esperando que el 10% de los resultados obtenidos no correspondan a información verídica; tanto para el valor de la probabilidad a favor P como el valor de la probabilidad en contra Q se estableció el valor de 0.5 ya que aún no se conoce la efectividad y/o fracaso del sistema en diversas condiciones de iluminación y poses de los individuos. El error porcentual absoluto e con un valor de 0.10 teniendo un porcentaje de respuestas correctas del ±10% de los resultados totales obtenidos. El resultado del cálculo de la muestra aplicando la Ecuación 12 se presenta a continuación:

$$n = \frac{2.33^2 * 2500 * 0.5 * 0.5}{0.1^2 * (2500 - 1) + 2.33^2 * 0.5 * 0.5} \cong 128$$

Con el resultado de la muestra que se obtuvo se procede a aplicar las pruebas a un numero de 128 individuos para posteriormente analizar los resultados. La muestra se seleccionó utilizando el criterio de muestreo por conveniencia debido a que la población total de la FICA para este estudio es demasiado grande (N≅2500), por lo que resulta imposible incluir cada individuo en las fases de entrenamiento y pruebas del sistema para la fecha de culminación de este estudio. Los individuos de la muestra seleccionada constan de estudiantes de las carreras de ingeniería en: electrónica y redes de comunicaciones (CIERCOM), telecomunicaciones (CITEL), y sistemas computacionales (CISIC), los cuales se incluyen en la base de datos del sistema.

4.2. Implementación del sistema

Previo al desarrollo de las pruebas del sistema, es preciso establecer el lugar donde el servidor desarrollará las tareas de reconocimiento facial. La Figura 73 muestra la disposición de los elementos que conforman este proyecto en el primer piso de la FICA; para efectos de prueba el servidor ha sido alojado de manera temporal en la parte inferior del Laboratorio de Computo 1 y la cámara de vigilancia ha sido posicionada enfocando al pasillo principal y las escaleras del primer piso, donde existe mayor afluencia de personas.

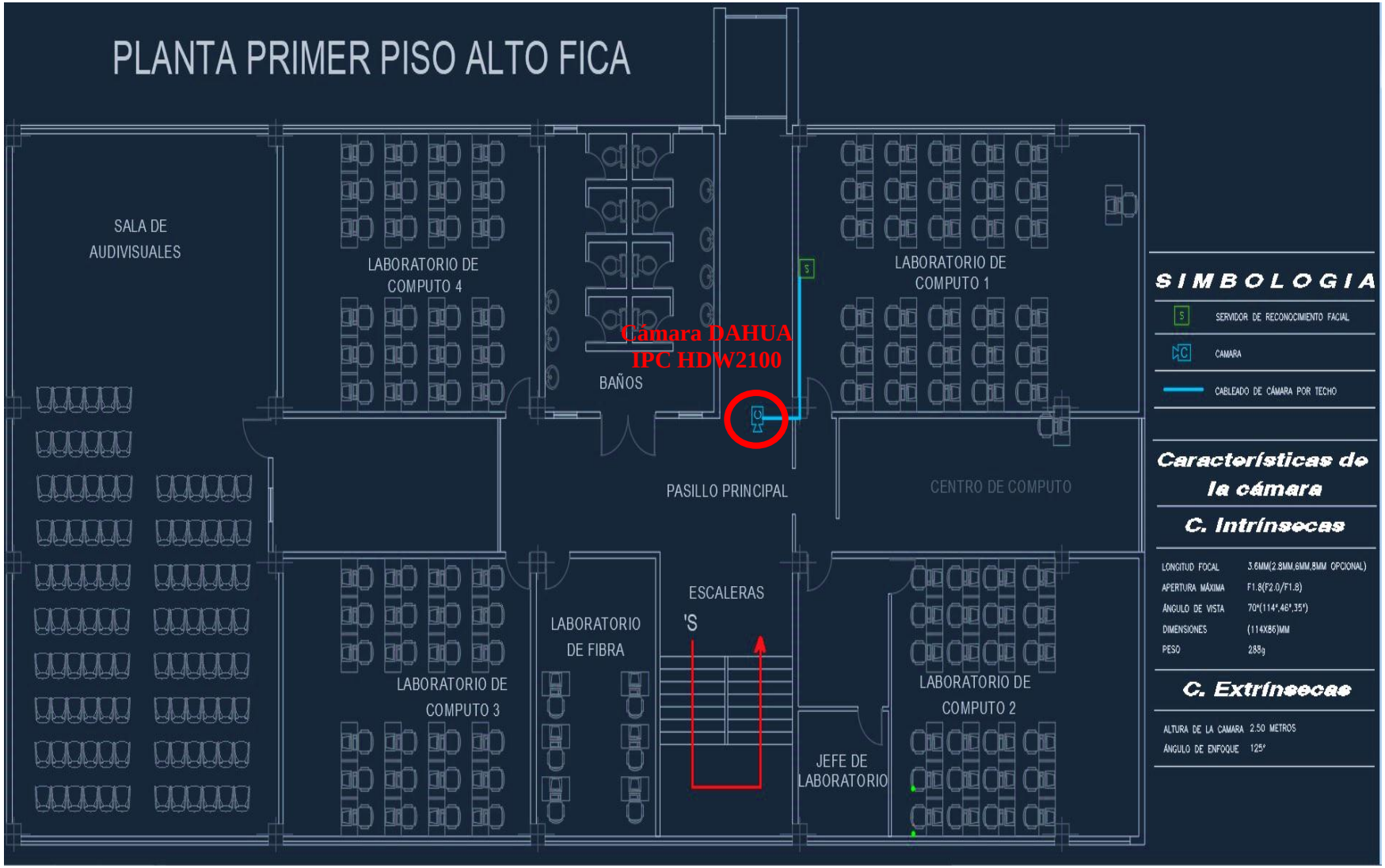


Figura 73. Plano del primer piso de la FICA.

Fuente: Autoría

4.2.1. Diagrama de conexión general del sistema

En esta sección se materializa el diseño del proyecto y la unificación de los diferentes elementos del sistema. Para ello se considera el diagrama de conexión de la Figura 74, donde se muestra que, por medio de un router (Tabla 22) el sistema establece conexión con 2 módulos, el primer módulo se encarga de la recolección de fotografías faciales para la construcción del conjunto de datos de entrenamiento mediante una conexión inalámbrica con un Smartphone; el segundo módulo se encarga de la adquisición del flujo de video de la cámara IP para realizar las tareas de detección y reconocimiento facial a través de una conexión cableada directa.

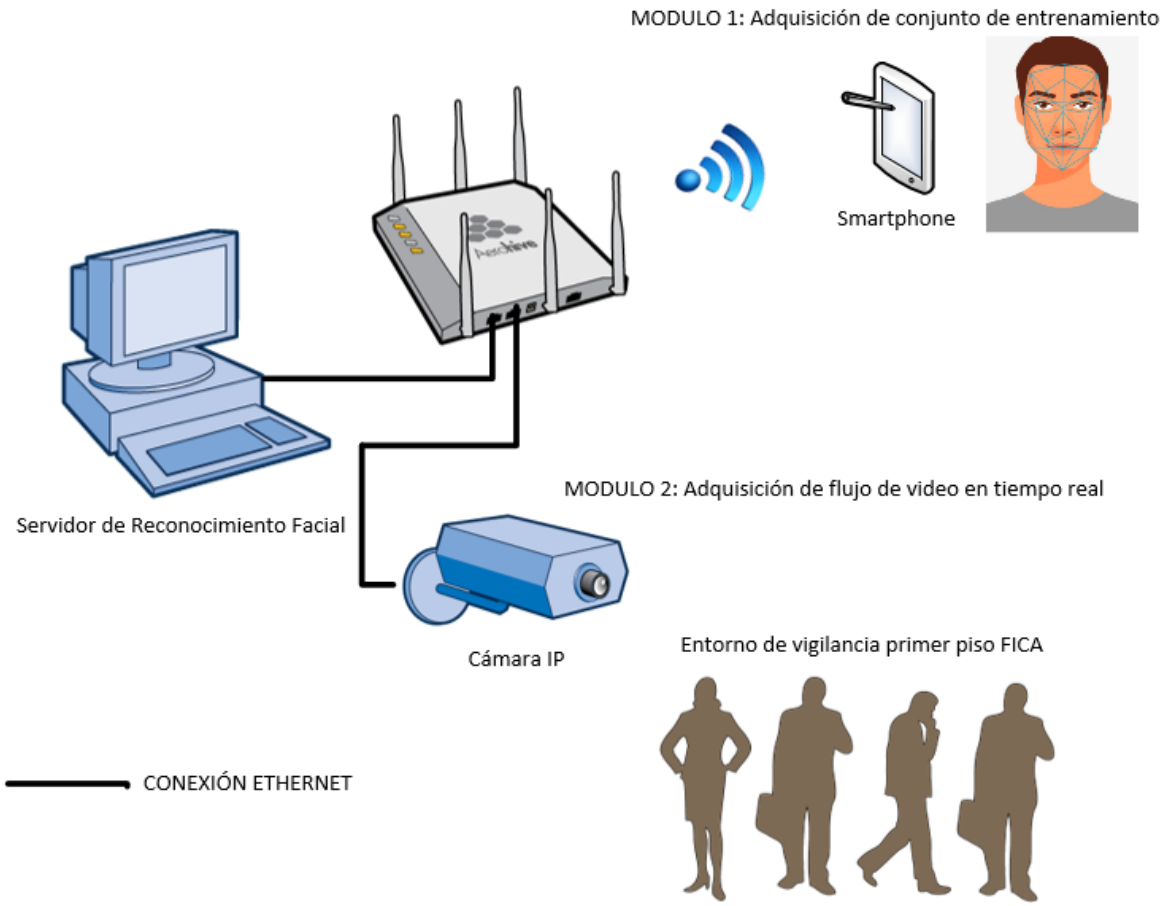


Figura 74. Esquema general de conexión del sistema.

Fuente: Autoría

El conjunto de características generales del router usado se especifican a continuación en la Tabla 22.

Tabla 22. Especificaciones técnicas del router.

Router TP-Link WR841N	
INTERFACE	4 PUERTOS LAN 10/100Mbps 1 PUERTO WAN10/100Mbps
PUERTOS	4 × 10/100Mbps LAN Ports, 1 × 10/100Mbps WAN Port
FUENTE DE ALIMENTACIÓN	9VDC / 0.6A
ESTÁNDARES INALÁMBRICOS	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
DIMENSIONES (W x D x H)	7.6 x 5.3 x 1.3 pulg.(192 x 134 x 33 mm)
ANTENA	2 antenas fijas omnidireccionales de 5dBi
SEGURIDAD INALÁMBRICA	64/128/152-bit WEP / WPA / WPA2,WPA-PSK / WPA2-PSK
PROTOCOLOS	Soporta IPv4 & IPv6
FRECUENCIA	2.4-2.4835 GHz

Fuente: Adaptado de (TP-Link, 2019)

4.2.1.1.Módulo de adquisición de conjunto de datos de entrenamiento

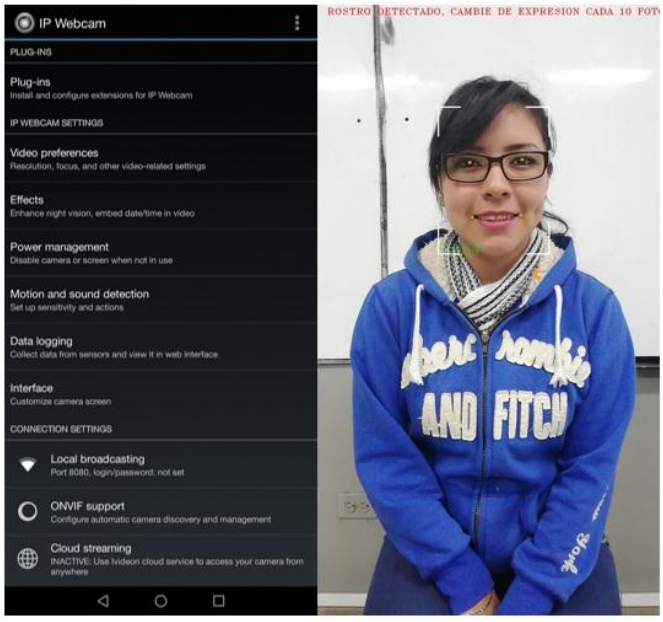
El objetivo de esta sección es especificar las herramientas usadas para la recolección del conjunto de datos para el entrenamiento del clasificador de máquinas de vector soporte (SVM). Anteriormente en la sección 3.4.2.2.1 se especificaron algunas funciones que conlleva esta tarea a través del script del Anexo 7, no obstante, también se ha hecho uso de la cámara de un Smartphone y una aplicación de Android. El conjunto de características generales del Smartphone usado para la recolección del conjunto de datos se especifican en la Tabla 23:

Tabla 23. Especificaciones técnicas del Smartphone.

HUAWEI MATE 10 LITE	
PANTALLA	5,9 pulgadas FullHD+, resolución de 2160x1080 píxeles
CPU/GPU	Kirin 659 de ocho núcleos a 2,36GHz/ GPU Mali T830 MP2
RAM/ALMACENAMIENTO	4 GB/64GB más microSD de hasta 256GB
CÁMARAS	Dual, 16MP+2MP, autofocus por detección de fase, flash LED, HDR, video 1080p@30fps, cámara frontal 13MP+2MP

Fuente: Adaptado de (HUAWEI, 2018)

En cuanto a la aplicación empleada, esta se denomina "IP Webcam" y puede obtenerse de manera gratuita en la PlayStore de Android. Esta aplicación simula una cámara de red (IP), por lo que permite obtener el flujo de video de la cámara de cualquier dispositivo Android en tiempo real (Figura 75) a través de una conexión inalámbrica. La Figura 75 muestra al lado izquierdo (a) el panel de configuración de la aplicación y al lado derecho (b) los resultados de aplicar el script del Anexo 7 al flujo de video obtenido desde el Smartphone por medio de la aplicación.



a) b)

Figura 75. a) Ventana de configuración IP Webcam y b) adquisición de conjunto de datos (fotografías).

Fuente: Autoría

4.2.1.2.Módulo de adquisición de flujo de video

En esta sección es preciso mencionar que, debido a algunas limitaciones y/o restricciones encontradas en la configuración del sistema CCTV de la FICA, no se pudo obtener el flujo de video principal de las cámaras de mayor interés para este estudio. Entre dichas limitaciones se encuentran:

- Denegación de acceso a flujo de video (mediante protocolo RTSP/HTTP).
- Denegación de acceso a las configuraciones individuales de cada cámara.
- Desconexión automática del equipo CCTV (NVR) de la red de datos cada cierto periodo de tiempo (4 horas).
- Restricciones de acceso de la red de datos (segmento VLan CCTV).

Por las razones expuestas, fue necesario el uso de una cámara marca DAHUA, especificada como segunda opción en la sesión de requerimientos de hardware, la cual formaba parte del sistema CCTV de la FICA antes de una reubicación de equipos realizada recientemente (año 2018). La Tabla 23 muestra de forma general las características principales de la cámara, las especificaciones completas se encuentran en el Anexo 15.

Tabla 24. Especificaciones técnicas de la cámara IP.

ESPECIFICACIONES	**Propiedades**
Sensor de imagen	1/3" 1.3Megapixels Aptina CMOS
Máxima resolución de imagen	1280 × 960
Resoluciones disponibles	1.3M(1280×960) / 720P(1280×720) / D1(704×576) / CIF(352×288)
Interfaz de comunicación	Interfaz Ethernet RJ45 (10/100Base-T)
Fuente de alimentación	12 VDC ± 25%, PoE (802.3af Class3)
Compresión de video	DC12V, PoE (802.3af)

Fuente: Adaptado de (Dahua, 2012)

Dicha cámara se encontraba almacenada en las oficinas del centro de cómputo, de modo que fue necesario instalarla nuevamente en el primer piso de la FICA. La Figura 76 muestra la disposición de la cámara.

Figura 76. Orientación de la cámara en el primer piso de la FICA apuntando simultáneamente al pasillo y a las escaleras de subida y bajada.

Fuente: Autoría

4.3. Validación y métricas de eficiencia del sistema

Una vez finalizada la etapa de diseño e implementación del sistema se procede a realizar la etapa de pruebas, donde se valida la efectividad del sistema. La validación de un sistema inteligente consiste en la comprobación de si el producto es el correcto de acuerdo con el rendimiento esperado; lo que requiere examinar la validez de si los resultados suministrados por el sistema son correctos, y la constatación del cumplimiento de las necesidades y requisitos del usuario (Palma Mendez & Marín Morales, 2004).

La validación del sistema se realiza en torno a diferentes consideraciones, tomadas en cuenta en la configuración de un ambiente controlado y no controlado; donde un ambiente controlado se caracteriza por poseer cambios de iluminación mínimos y con poca cantidad de personas; en su contraparte, un ambiente no controlado se diferencia por poseer mayores cambios de iluminación y distancia variable, además de contener una gran cantidad de personas en un lugar amplio. Los cambios de pose en referencia al rostro, también son evaluados, de forma somera.

La herramienta fundamental de evaluación del desempeño de clasificación del sistema son las **matrices de confusión**, que particularmente proporcionan una idea de cómo está clasificando el sistema, a partir de un conteo de aciertos y errores de cada una de las clases. La matriz de confusión (Figura 77) considera los siguientes índices: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).

		Clasificador	
		+	-
Valor real	+	TP	FN
	-	FP	TN

Figura 77. Matriz de confusión.

Fuente: Adaptado de (Koldo, 2018)

Donde:

- TP: Son el número de verdaderos positivos, es decir, de predicciones correctas para la clase positiva + (rostro).
- TN: Son el número de verdaderos negativos, es decir, de predicciones correctas para la clase negativa – (no rostro).
- FP: Son el número de falsos positivos, es decir, una predicción positiva cuando realmente debería ser negativo.
- FN: Son el número de falsos negativos, es decir, una predicción negativa cuando realmente debería ser positiva.

Métricas de eficiencia

A continuación, se presentan las distintas métricas utilizadas para medir la eficiencia del sistema a partir de la matriz de confusión:

- *Precisión*

Es la proporción del número total de predicciones que fueron correctamente clasificadas, basándose en los verdaderos positivos y negativos frente a todas las detecciones del sistema. La precisión se calcula mediante la Ecuación 13:

$$Pr = \frac{TP + TN}{Total}$$

Ecuación 13. Fórmula de la precisión.

Fuente: Adaptado de (Koldo, 2018)

- *Tasa de error*

Es la proporción del número total de predicciones que fueron incorrectamente clasificadas, basándose en los falsos positivos y negativos frente a todas las detecciones del sistema. La tasa de error se calcula mediante la Ecuación 14:

$$Er = \frac{FP + FN}{Total}$$

Ecuación 14. Fórmula de la tasa de error.

Fuente: Adaptado de (Koldo, 2018)

- *Sensibilidad*

También se la llama recall o tasa de verdaderos positivos obtenidos por el sistema. Proporciona la probabilidad de que, dada una observación realmente positiva, el modelo la clasifique así. La sensibilidad se calcula mediante la Ecuación 15:

$$Rec = \frac{TP}{TP + FN}$$

Ecuación 15. Fórmula de la sensibilidad.

Fuente: Adaptado de (Koldo, 2018)

- *Especificidad*

La especificidad es el número correcto de detecciones negativas obtenidas por el sistema. Proporciona la probabilidad de que, dada una observación realmente negativa, el modelo la clasifique así. La especificidad se calcula mediante la Ecuación 16:

$$Esp = \frac{TN}{TN + FP}$$

Ecuación 16. Fórmula de la especificidad.

Fuente: Adaptado de (Koldo, 2018)

4.4. Eficiencia del sistema

Antes de comprobar la eficiencia del sistema de reconocimiento facial, es preciso recalcar que, los modelos del aprendizaje profundo y automático fueron evaluados preliminarmente en las secciones 3.4.2.1.17 y 3.4.2.2.4, donde se lograron resultados de precisión alentadores sobre conjuntos de datos de entrenamiento de dominio público (VGGFace2, LFW), mediante el método *cross-validation*. Por lo que se pudo concluir que, en un ambiente de pruebas de laboratorio, los modelos se desempeñan bastante bien y alcanzan tasas de éxito aceptables.

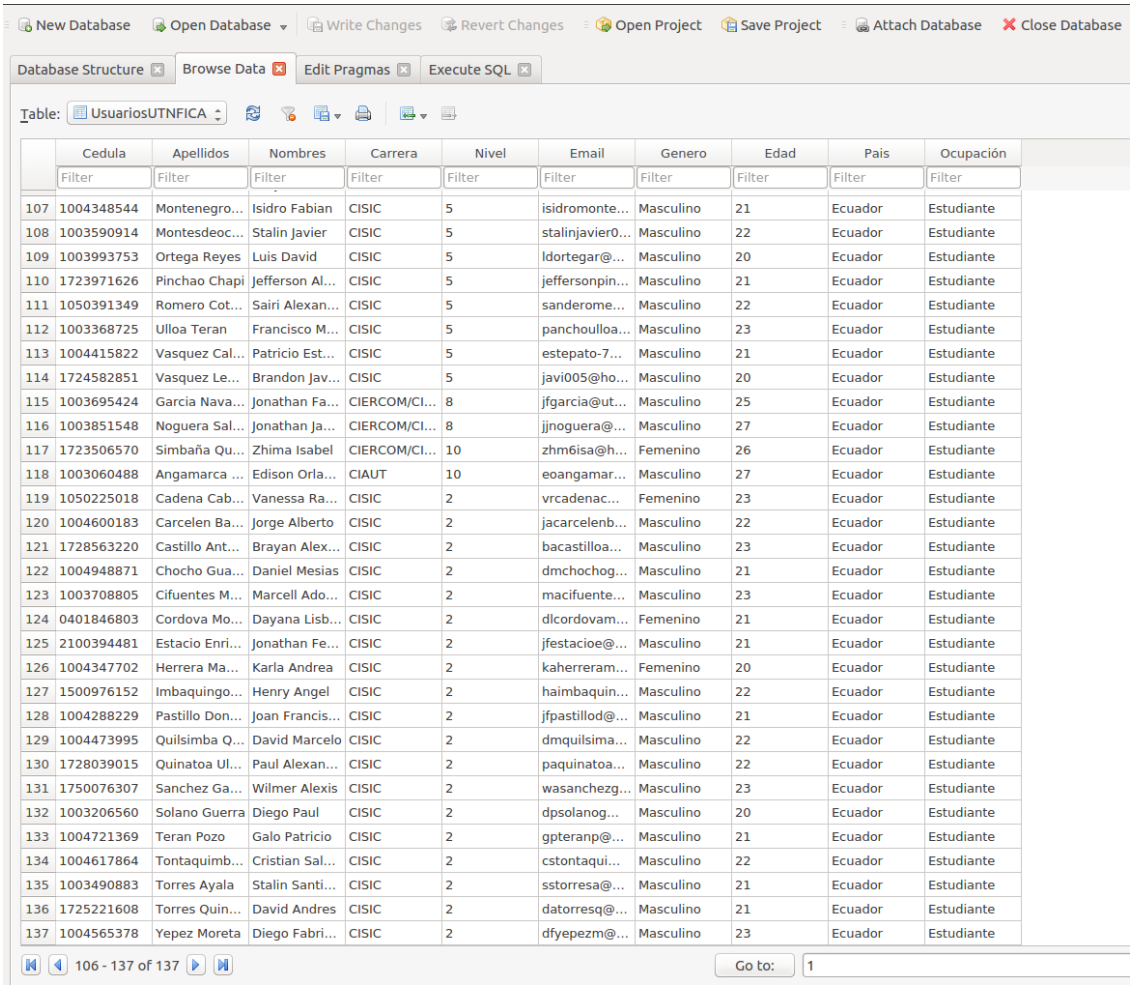
Sin embargo, es preciso mencionar que dichas pruebas de precisión solamente muestran un aproximado del desempeño del sistema en condiciones de prueba favorables (proceso offline en entorno controlado), difiriendo de una aplicación de la vida real (proceso online en entorno no controlado). Es por eso que en esta sección se analiza el sistema de manera cualitativa y cuantitativa empleando las métricas definidas en la sección 4.3. Finalmente, destacar que se ha establecido un umbral de predicción para limitar personas conocidas y desconocidas en el sistema. Esto se debe a que el clasificador del sistema ofrece un valor de predicción alto para individuos que se encuentran registrados (conocidos), caso contrario la predicción es muy baja para individuos no registrados (desconocidos). Además, los valores del umbral se han adecuado dependiendo de la calidad de las imágenes obtenidas en tiempo real en cada situación.

4.4.1. Análisis Cualitativo y Cuantitativo

Para este tipo de análisis se consideran 2 situaciones: ambiente controlado y ambiente no controlado con 24 y 128 sujetos de prueba respectivamente. Como punto de partida para la ejecución de las pruebas en cualquiera de los entornos mencionados, se realizó la recopilación de la base de datos de rostros y la información de cada individuo en la base de datos (sección 4.2.1.1).

Para efectuar este proceso se cumplió con la captura de fotografías a estudiantes y docentes de la FICA, llegando a construir una base de datos de 136 individuos con alrededor de 100 muestras, debidamente ordenadas en directorios para cada clase (sujeto). Como se mencionó en la sección 2.4.2.2.1, las fotografías se encuentran almacenadas en formato “.jpg” con una resolución 1080p (1920x1080) para el posterior proceso de alineamiento.

El registro de la información personal de cada individuo de la base de datos se muestra en la Figura 78.



	Cedula	Apellidos	Nombres	Carrera	Nivel	Email	Genero	Edad	Pais	Ocupación
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
107	1004348544	Montenegro...	Isidro Fabian	CISIC	5	isidromonte...	Masculino	21	Ecuador	Estudiante
108	1003590914	Montesdeoc...	Stalin Javier	CISIC	5	stalinjavier0...	Masculino	22	Ecuador	Estudiante
109	1003993753	Ortega Reyes	Luis David	CISIC	5	ldortegar@...	Masculino	20	Ecuador	Estudiante
110	1723971626	Pinchao Chapi	Jefferson Al...	CISIC	5	jeffersonpin...	Masculino	21	Ecuador	Estudiante
111	1050391349	Romero Cot...	Sairi Alexan...	CISIC	5	sanderome...	Masculino	22	Ecuador	Estudiante
112	1003368725	Ulloa Teran	Francisco M...	CISIC	5	panchoulloa...	Masculino	23	Ecuador	Estudiante
113	1004415822	Vasquez Cal...	Patricio Est...	CISIC	5	estepato-7...	Masculino	21	Ecuador	Estudiante
114	1724582851	Vasquez Le...	Brandon Jav...	CISIC	5	javi005@ho...	Masculino	20	Ecuador	Estudiante
115	1003695424	Garcia Nava...	Jonathan Fa...	CIERCOM/CI...	8	jfgarcia@ut...	Masculino	25	Ecuador	Estudiante
116	1003851548	Noguera Sal...	Jonathan Ja...	CIERCOM/CI...	8	jjnoguera@...	Masculino	27	Ecuador	Estudiante
117	1723506570	Simbaña Qu...	Zhima Isabel	CIERCOM/CI...	10	zhm6isa@h...	Femenino	26	Ecuador	Estudiante
118	1003060488	Angamarca ...	Edison Orla...	CIAUT	10	eoangamar...	Masculino	27	Ecuador	Estudiante
119	1050225018	Cadena Cab...	Vanessa Ra...	CISIC	2	vrcadenac...	Femenino	23	Ecuador	Estudiante
120	1004600183	Carcelen Ba...	Jorge Alberto	CISIC	2	jacarcelenb...	Masculino	22	Ecuador	Estudiante
121	1728563220	Castillo Ant...	Brayan Alex...	CISIC	2	bacastilloa...	Masculino	23	Ecuador	Estudiante
122	1004948871	Chocho Gua...	Daniel Mesias	CISIC	2	dmchochog...	Masculino	21	Ecuador	Estudiante
123	1003708805	Cifuentes M...	Marcell Ado...	CISIC	2	macifuente...	Masculino	23	Ecuador	Estudiante
124	0401846803	Cordova Mo...	Dayana Lisb...	CISIC	2	dlcordovam...	Femenino	21	Ecuador	Estudiante
125	2100394481	Estacio Enri...	Jonathan Fe...	CISIC	2	jfestacioe@...	Masculino	21	Ecuador	Estudiante
126	1004347702	Herrera Ma...	Karla Andrea	CISIC	2	kaherreram...	Femenino	20	Ecuador	Estudiante
127	1500976152	Imbaquingo...	Henry Angel	CISIC	2	haimbaquin...	Masculino	22	Ecuador	Estudiante
128	1004288229	Pastillo Don...	Joan Francis...	CISIC	2	jfpastillod@...	Masculino	21	Ecuador	Estudiante
129	1004473995	Quilsimba Q...	David Marcelo	CISIC	2	dmquilsima...	Masculino	22	Ecuador	Estudiante
130	1728039015	Quinatoa Ul...	Paul Alexan...	CISIC	2	paquinatoa...	Masculino	22	Ecuador	Estudiante
131	1750076307	Sanchez Ga...	Wilmer Alexis	CISIC	2	wasanchezg...	Masculino	23	Ecuador	Estudiante
132	1003206560	Solano Guerra	Diego Paul	CISIC	2	dpsolanog...	Masculino	20	Ecuador	Estudiante
133	1004721369	Teran Pozo	Galo Patricio	CISIC	2	gpteranp@...	Masculino	21	Ecuador	Estudiante
134	1004617864	Tontaquimb...	Cristian Sal...	CISIC	2	cstontaqui...	Masculino	22	Ecuador	Estudiante
135	1003490883	Torres Ayala	Stalin Santi...	CISIC	2	sstorresa@...	Masculino	21	Ecuador	Estudiante
136	1725221608	Torres Quin...	David Andres	CISIC	2	datorresq@...	Masculino	21	Ecuador	Estudiante
137	1004565378	Yepez Moreta	Diego Fabri...	CISIC	2	dfyepezm@...	Masculino	23	Ecuador	Estudiante

Figura 78. Registro de información en la base de datos.

Fuente: Autoría

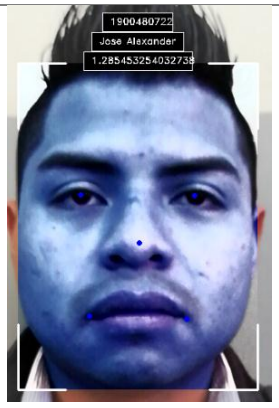
4.4.1.1.Pruebas en ambiente controlado (Laboratorio).

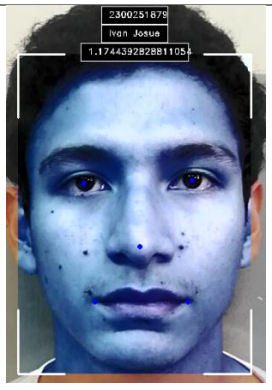
Para el desarrollo de este tipo de pruebas se ha reunido a 24 individuos en un lugar con óptimas condiciones de iluminación y con leves cambios de postura facial. El umbral de predicción del clasificador se ha establecido en un valor alto de 0.75 para este tipo de prueba. Los resultados obtenidos con el primer grupo de individuos se presentan en la Tabla 24:

Tabla 25. Verificación facial individual.

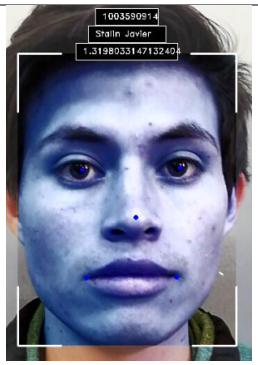
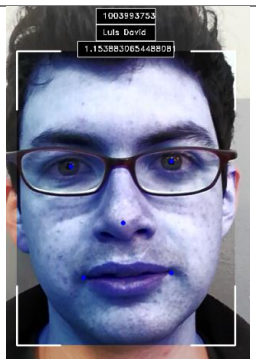
Num. Sujeto	**Imagen procesada (SALIDA)**	**Num. de rostros detectados**	**Etiqueta correcta (SI/NO)**	**Verificación de perfil izquierdo, derecho y frontal. Cantidad de aciertos (3/3)**	**Porcentaje de verificación alcanzado**
1		1	SI	3/3	100%
2		1	SI	3/3	100%

3		1	SI	3/3	100%
4		1	SI	3/3	100%
5		1	SI	3/3	100%
6		1	SI	3/3	100%

7		1	SI	3/3	100%
8		1	SI	3/3	100%
9		1	SI	3/3	100%
10		1	SI	3/3	100%

11		1	SI	3/3	100%
12		1	SI	3/3	100%
13		1	SI	3/3	100%
14		1	SI	3/3	100%

15		1	SI	3/3	100%
16		1	SI	3/3	100%
17		1	SI	3/3	100%
18		1	SI	3/3	100%

19		1	SI	3/3	100%
20		1	SI	3/3	100%
21		1	SI	3/3	100%
22		1	SI	3/3	100%

23		1	SI	3/3	100%
24		1	SI	3/3	100%

Fuente: Autoría

En la Tabla 24 se muestran los resultados finales alcanzados por el sistema para cada individuo registrado a través de una imagen procesada en tiempo real (online), donde se puede apreciar el cuadro delimitador de la región de interés (ROI) del rostro, el nombre del individuo en la parte superior; la cantidad de rostros detectados, los aciertos en la verificación de etiqueta de clase de perfil izquierdo, derecho y frontal, además del porcentaje alcanzado en las tres pruebas realizadas. Las pruebas también consideraron el uso de lentes.

Continuando con la evaluación del sistema, se establecieron 5 grupos de individuos, donde se puede corroborar el desempeño del sistema con más de un individuo. La variación de la cantidad de individuos en cada grupo ha sido determinada de forma experimental para conocer la capacidad de identificación facial simultanea del sistema en un ambiente grupal.

- ***Grupo Nº 1***

El primer grupo de personas está compuesto por 5 individuos (clases) (Figura 79).

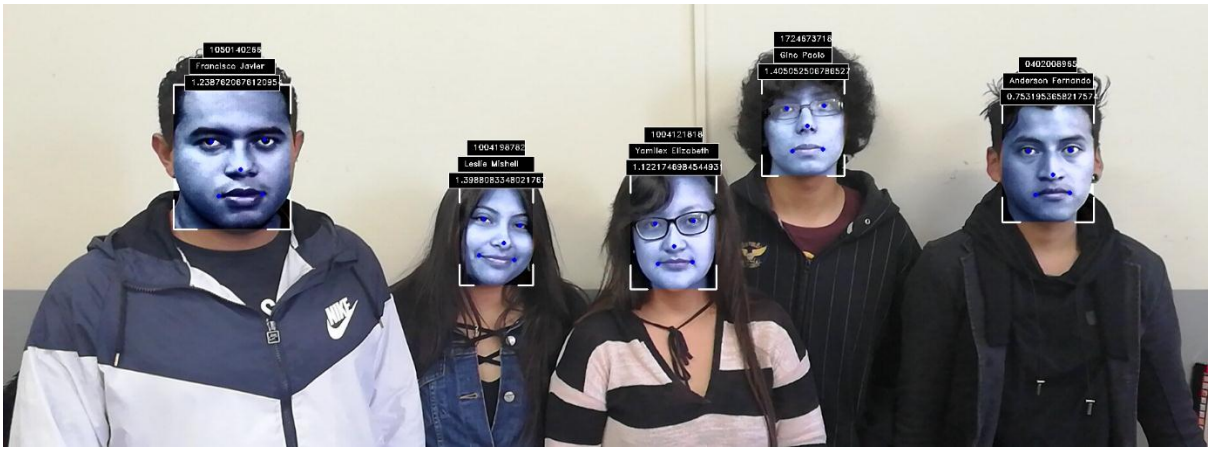


Figura 79. Grupo Nº1 de pruebas.

Fuente: Autoría

- ***Grupo Nº 2***

El segundo grupo de personas está compuesto por 10 individuos (clases), donde se encuentran incluidos los sujetos del primer grupo (Figura 80).

Figura 80. Grupo Nº2 de pruebas.

Fuente: Autoría

- ***Grupo Nº 3***

El tercer grupo de personas está compuesto por 5 individuos (clases) diferentes a los del grupo 1 y 2 (Figura 81).

Figura 81. Grupo Nº3 de pruebas.

Fuente: Autoría

- ***Grupo Nº 4***

El cuarto grupo de personas está compuesto por 10 individuos (clases), donde se encuentran incluidos los sujetos del tercer grupo (Figura 82).

Figura 82. Grupo Nº4 de pruebas.

Fuente: Autoría

- ***Grupo Nº 5***

El quinto grupo de personas está compuesto por 5 individuos (clases), diferentes a los del grupo 1,2,3, y 4 (Figura 83).

Figura 83. Grupo Nº5 de pruebas.

Fuente: Autoría

En cada situación el sistema se desempeña de manera correcta en tiempo real, considerando cambios de expresión, pose, y un cambio de iluminación mínimo. Los resultados alcanzados se pueden apreciar en la Tabla 26. Específicamente, la Tabla 26 muestra la eficiencia del clasificador SVM, cuando se ha entrenado sobre una base de datos de 136 individuos. Las métricas calculadas ante cada grupo de individuos, indican un valor alto de verdaderos positivos, a excepción del grupo 5, donde se originó un falso negativo obtenido de manera casual en el momento de esta captura en particular, que puede deberse a algunas de las siguientes variantes, tales como: calidad de la óptica, calidad de imagen, aberraciones cromáticas o imperfecciones encontradas en la imagen, ángulo de captura del rostro o un cambio drástico de apariencia facial, ocasionando que el clasificador interprete de forma errónea las facciones de un individuo. Cabe recalcar que en la mayoría del flujo de video este individuo fue reconocido. No obstante, estos resultados aún demuestran una alta correlación de los datos obtenidos frente a cada usuario (etiquetas correctas), además se demuestra que el detector de rostros identifica la zona ROI de forma fiable en todos los grupos.

Tabla 26. Evaluación de resultados en entorno controlado con 24 personas.

CLASIFICADOR(SVM)		**GRUPO 1 (cara:etiqueta/ cara:no etiqueta)**		**GRUPO 2 (cara:etiqueta/ cara:no etiqueta)**		**GRUPO 3 (cara:etiqueta/ cara:no etiqueta)**		**GRUPO 4 (cara:etiqueta/ cara:no etiqueta)**		**GRUPO 5 (cara:etiqueta/ cara:no etiqueta)**		
		+	-	+	-	+	-	+	-	+	-	
VALOR REAL(cara:etiqueta/ no cara:no etiqueta)	+	5	0	10	0	5	0	10	0	4	1	
	-	0	0	0	0	0	0	0	0	0	0	
MÉTRICAS		**Valores obtenidos GRUPO 1**		**Valores obtenidos GRUPO 2**		**Valores obtenidos GRUPO 3**		**Valores obtenidos GRUPO 4**		**Valores obtenidos GRUPO 5**		**MÉTRICAS GLOBALES**
Precisión (Pr)		100%		100%		100%		100%		80%		96%
Tasa de error (Er)		0%		0%		0%		0%		20%		4%
Sensibilidad o Recall (Rec)		100%		100%		100%		100%		80%		96%
Especificidad (Esp)		0%		0%		0%		0%		0%		0%

Fuente: Autoría

4.4.1.2.Pruebas en ambiente no controlado (FICA).

Para el desarrollo de este tipo de pruebas se han organizado algunos grupos de individuos, que posteriormente fueron situados en el pasillo del primer piso de la FICA para la ejecución de una serie de pruebas a diferentes distancias frontales, donde las condiciones de iluminación y los cambios de postura facial son reales. El umbral de predicción del clasificador se ha establecido en un valor bajo de 0.25 para este tipo de prueba, debido a que existe una degradación de la calidad de la imagen. En la Figura 84 se puede apreciar que la cámara CCTV se situó apuntando a una zona donde la afluencia de personas es constante.

Figura 84. Pasillo primer piso FICA con la cámara apuntando a las escaleras de subida y bajada.

Fuente: Autoría

Es preciso mencionar que, en esta sección se considera un factor muy importante a tomar en cuenta, el cual es la precisión de reconocimiento a diferentes distancias. Si bien, la detección facial puede ser desarrollada casi sin ningún problema por el detector planteado en la sección

3.4.2.1.4 (Figura 85) en entornos desafiantes, a distancias cercanas (1 metro) o lejanas (6 metros o más), y con imágenes de baja y alta resolución; no sucede lo mismo para la extracción de características faciales a larga distancia y a bajas resoluciones, ya que conlleva algunas restricciones que dependen completamente del tipo de óptica usada, afectando el desempeño del sistema y por lo tanto la identificación de las personas. Dichos aspectos y/o limitaciones se explicarán a detalle más adelante en la sección 4.5, donde se evaluarán los resultados obtenidos por el sistema frente a las condiciones anteriormente mencionadas, y se brindarán soluciones y/o recomendaciones como trabajo futuro.

Figura 85. Detección facial pasillo primer piso FICA.

Fuente: Autoría

Como punto de partida, es preciso indicar los aspectos que serán tomados en cuenta para el desarrollo de pruebas con algunos grupos de individuos, donde de forma general se evalúa la cantidad de aciertos alcanzados en la tarea de detección y reconocimiento facial de forma grupal en un rango de 5 segundos a distancias variables de 4.5, 3.5, 2.5 y 1.5 metros. Además, para efectos de este estudio, se ha establecido un tamaño (ancho y alto) o umbral de detección facial de ($w = 25, h = 25$) pixeles, donde solo se detectarán rostros con iguales o mayores dimensiones dentro de la imagen principal. Esto debido a que se han seguido algunas pautas y/o recomendaciones presentadas en los estudios realizados por Marciniak (2013) y Li (2019). El estudio de Marciniak (2013) concluye que: "*el tamaño mínimo para conseguir un correcto desempeño en las tareas de detección y reconocimiento facial es de 21x21 pixeles*". Desde luego, en el estudio se menciona que, de cierta manera dicha recomendación puede encontrar un uso generalizado en el análisis de imágenes CCTV, ya que las pruebas realizadas en el estudio se limitan a evaluar conjuntos de datos faciales con ciertas restricciones (cambios de iluminación, pose, y expresión controlados; redimensionamiento manual de imagen facial), por lo que, quedan dudas de la validez de los resultados ante un entorno sin restricciones en tiempo real, que de todas maneras serán corroborados con este estudio. Por el contrario, en el estudio realizado por Li (2019) se menciona que: "*Los rostros obtenidos a distancias lejanas brindan muy baja calidad (menores a 20x20 pixeles), especialmente en conjuntos de datos proporcionados por sistemas de vigilancia basados en video (CCTV), por lo que tienden a tener un tamaño bastante limitado. A pesar de que algunos enfoques de aprendizaje profundo son muy sólidos frente a algunos desafíos del reconocimiento facial (iluminación, leve desenfoque, leves cambios de expresión y pose, entre otros), son muy pobres en otros casos (imágenes de baja resolución y/o calidad, alto desenfoque, compresión, entre otros); es por eso que, en escenarios desafiantes (CCTV) es una tarea difícil y propensa a*

errores, por lo que se deben explorar enfoques adicionales que solventen los problemas de imágenes de baja calidad". Las recomendaciones del estudio de Li, indican que para mejorar la eficiencia de los sistemas de reconocimiento facial en imágenes de baja resolución y/o calidad se deben implementar técnicas SR(*Super-Resolution*), que involucran el uso de algunas arquitecturas de redes neuronales profundas, con el fin de generar imágenes de mejor calidad, tales como las GAN´s (*Generative Adversarial Networks*), WGAN, entre otras (Li, Flynn, Prieto, & Mery, 2019). La Figura 86 muestra algunos ejemplos de imágenes faciales de alta y baja calidad.

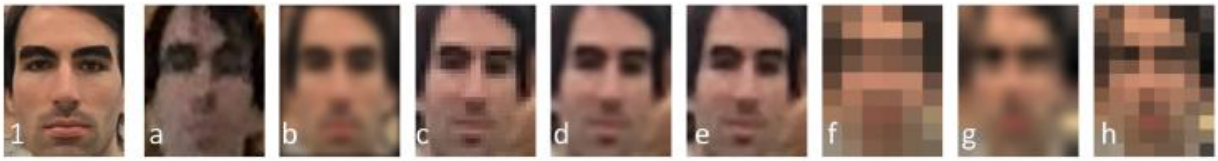


Figura 86. Comparación de imágenes faciales de alta (1) y baja (a-h) calidad.

Fuente: Adaptado de (Li, Flynn, Prieto, & Mery, 2019)

Los comentarios proporcionados en dichos estudios, permiten obtener una idea de los resultados negativos que el sistema obtendrá frente a imágenes faciales con dimensiones iguales o menores a 25x25 pixeles, por lo que se limitará a analizar imágenes faciales obtenidas de hasta alrededor de 4.5 metros de distancia para algunos casos de prueba (10 y 16 individuos aglomerados). Dichas elecciones son resultado de realizar un análisis de las dimensiones faciales obtenidas en el entorno de la FICA (Tabla 27), donde se manifiesta que la zona ROI del rostro medida a una distancia máxima de 4.5 metros se encuentra entre el rango de $(\mathrm{w} = 20 - 30)$ pixeles de ancho y de $(\mathrm{h} = 30 - 40)$ pixeles de alto, mientras que a una distancia mínima de 1.5 metros es de $(\mathrm{w} = 60 - 70)$ pixeles de ancho y de $(\mathrm{h} = 70 - 80)$ pixeles de alto aproximadamente. Las mediciones indican que las dimensiones de la zona ROI a máxima distancia se encuentran entre el rango de tamaño mínimo considerable para la verificación de un individuo. Adicionalmente se consideran algunos aspectos de la cámara antes de iniciar las respectivas pruebas del sistema, tales

como: el tamaño máximo configurable de la imagen proporcionada por la cámara es de 1280x960 (1.3 M) y el flujo de cuadros a analizar cada segundo se limita a 2 cuadros por segundo (fps).

Tabla 27. Dimensiones ROI medidas a diferentes distancias.

	Dimensiones ROI (pixeles)	
Distancia	**Ancho(w)**	**Alto(h)**
4.5 y 4 metros	20-30	30-40
3.5 y 3 metros	30-40	40-50
2.5 y 2 metros	40-50	50-60
1.5 metros	60-70	70-80

Fuente: Autoría

Dada ciertas limitaciones en cuanto a la disponibilidad de los 128 individuos de prueba, se ha seleccionado una submuestra aleatoria de 20 individuos, los resultados pretenden reflejar lo capacidad de clasificación del sistema ante un pequeño (5) o gran número (16) de individuos de prueba aglomerados en el pasillo de la FICA, por lo que se intuirán los resultados como un todo global.

La forma de evaluación para cada grupo considera diferentes distancias desde la posición de los individuos hacia la cámara, para lo cual se ha definido el primer grupo que consta de 4 subgrupos de 5 individuos cada uno, ordenados en una fila a 3 diferentes distancias. De forma similar se ha definido un segundo y tercer grupo con 10 y 16 individuos respectivamente aglomerados, donde la distancia de cada fila de 5 individuos consecuentemente varía, por lo que para cada situación se toman como referencia las distancias de cada fila con un espaciamiento entre ellos de 1 metro, abarcando así diferentes distancias de forma simultánea. Los grupos evaluados se presentan a continuación:

- ***Grupo 1 (4 Subgrupos de 5 individuos)***

Primer subgrupo

Para el primer subgrupo se ha considerado las distancias de 3.5, 3 y 2 metros desde la cámara a la posición de los individuos. La Figura 87 muestra la disposición de los individuos en el pasillo del primer piso de la FICA y en la Tabla 28 se muestran los resultados obtenidos.

Tabla 28. Evaluación de métricas del Subgrupo 1.

			Subgrupo 1			
			Métricas Promedio del Subgrupo 1			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	3.5 mts.	1-10	18%	82%	18%	0%
2	3 mts.	1-10	66%	34%	66%	0%
3	2 mts.	1-10	70%	30%	70%	0%

Fuente: Autoría

Figura 87. Verificación del Subgrupo 1.

Fuente: Autoría

Segundo subgrupo

Para el segundo subgrupo se ha considerado las distancias de 3.5, 3 y 2 metros desde la cámara a la posición de los individuos. La Figura 88 muestra la disposición de los individuos en el pasillo del primer piso de la FICA y en la Tabla 29 se muestran los resultados obtenidos.

Tabla 29. Evaluación de métricas del Subgrupo 2.

			Subgrupo 2			
			Métricas Promedio del Subgrupo 2			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	3.5 mts.	1-10	20%	80%	20%	0%
2	3 mts.	1-10	56%	44%	56%	0%
3	2 mts.	1-10	80%	20%	80%	0%

Fuente: Autoría

Figura 88. Verificación del Subgrupo 2.

Fuente: Autoría

Tercer subgrupo

Para el tercer subgrupo se ha considerado las distancias de 3.5, 3 y 2 metros desde la cámara a la posición de los individuos. La Figura 89 muestra la disposición de los individuos en el pasillo del primer piso de la FICA y en la Tabla 30 se muestran los resultados obtenidos.

Tabla 30. Evaluación de métricas del Subgrupo 3.

			Subgrupo 3			
			Métricas Promedio del Subgrupo 3			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	3.5 mts.	1-10	32%	68%	32%	0%
2	3 mts.	1-10	52%	48%	52%	0%
3	2 mts.	1-10	70%	30%	70%	0%

Fuente: Autoría

Figura 89. Verificación del Subgrupo 3.

Fuente: Autoría

Cuarto subgrupo

Para el tercer subgrupo se ha considerado las distancias de 3.5, 3 y 2 metros desde la cámara a la posición de los individuos. La Figura 90 muestra la disposición de los individuos en el pasillo del primer piso de la FICA y en la Tabla 31 se muestran los resultados obtenidos.

Tabla 31. Evaluación de métricas del Subgrupo 4.

			Subgrupo 3			
			Métricas Promedio del Subgrupo 3			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	3.5 mts.	1-10	12%	88%	12%	0%
2	3 mts.	1-10	50%	50%	50%	0%
3	2 mts.	1-10	68%	32%	68%	0%

Fuente: Autoría

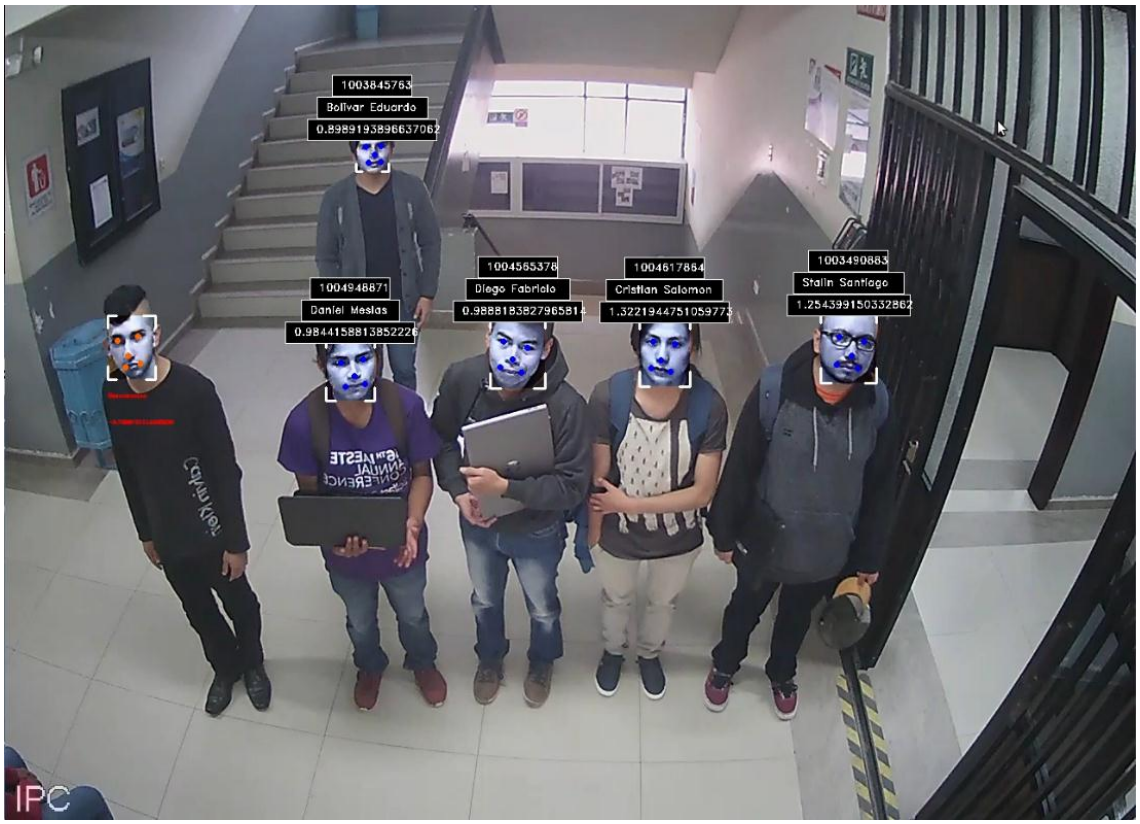


Figura 90. Verificación del Subgrupo 4.

Fuente: Autoría

- ***Grupo 2 (10 individuos)***

En esta sección se evalúa a un grupo de 10 individuos, donde se ha considerado las distancias de 4.5, 3.5, 2.5 y 1.5 metros desde la cámara a la posición de los individuos distribuidos en 2 filas de 5 (Figura 91). En la Tabla 32 se muestran los resultados obtenidos. Cabe recalcar que en la situación 3 se realizaron pruebas con 2 sujetos que no forman parte de la base de datos del sistema, esto debido a que se encontraban en la toma realizada en ese momento en particular.

Tabla 32. Evaluación de métricas del Grupo 2.

			Subgrupo 10 individuos			
			Métricas Promedio			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	4.5 y 3.5 mts.	1-10	34%	66%	34%	0%
2	3.5 y 2.5 mts.	1-10	58%	42%	58%	0%
3	2.5 y 1.5 mts.	1-10	74.8%	25.2%	75.79%	73.35%

Fuente: Autoría

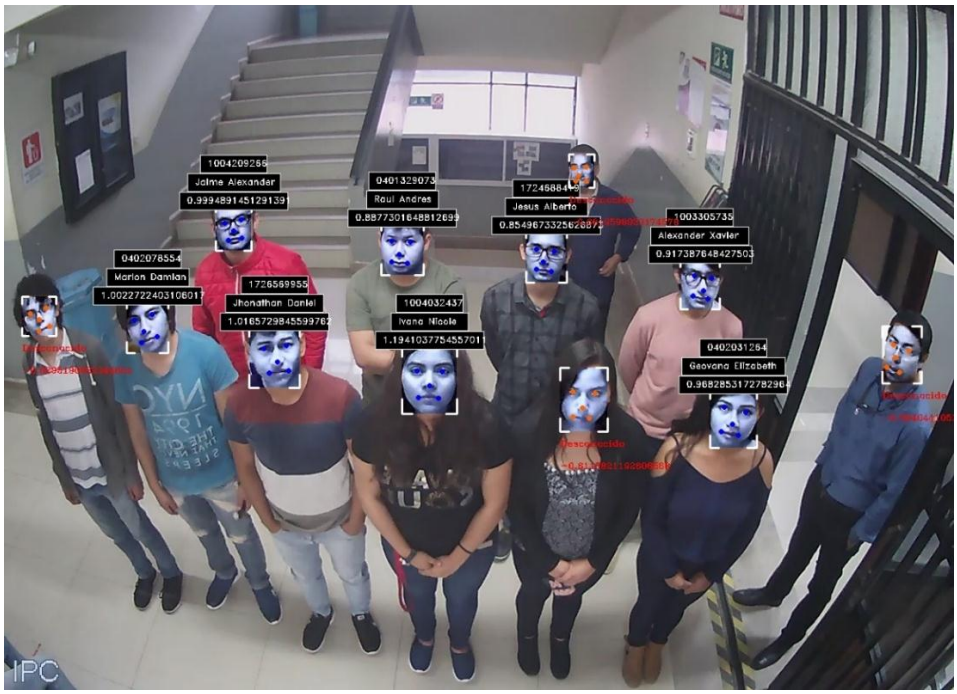


Figura 91. Verificación del Grupo 2.

Fuente: Autoría

- ***Grupo 3 (16 individuos considerando un sujeto desconocido)***

En esta sección se evalúa a un grupo de 16 individuos, donde se ha considerado las distancias de 4.5, 3.5, 2.5 y 1.5 metros desde la cámara a la posición de los individuos distribuidos en 3 filas de 5 (Figura 92), y considerando la presencia de 1 sujeto que no forma parte de la base de datos del sistema. En la Tabla 33 se muestran los resultados obtenidos. En este caso como el grupo es numeroso, solo existen 2 situaciones, donde se han abarcado todas las posibles distancias.

Tabla 33. Evaluación de métricas del Grupo 3.

			Subgrupo 16 individuos			
			Métricas Promedio			
Situación #	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
1	4.5, 3.5 y 2.5 mts.	1-10	45.6%	54.4%	43.54%	75%
2	3.5, 2.5 y 1.5 mts.	1-10	67.5%	32.5%	65.3%	100%

Fuente: Autoría

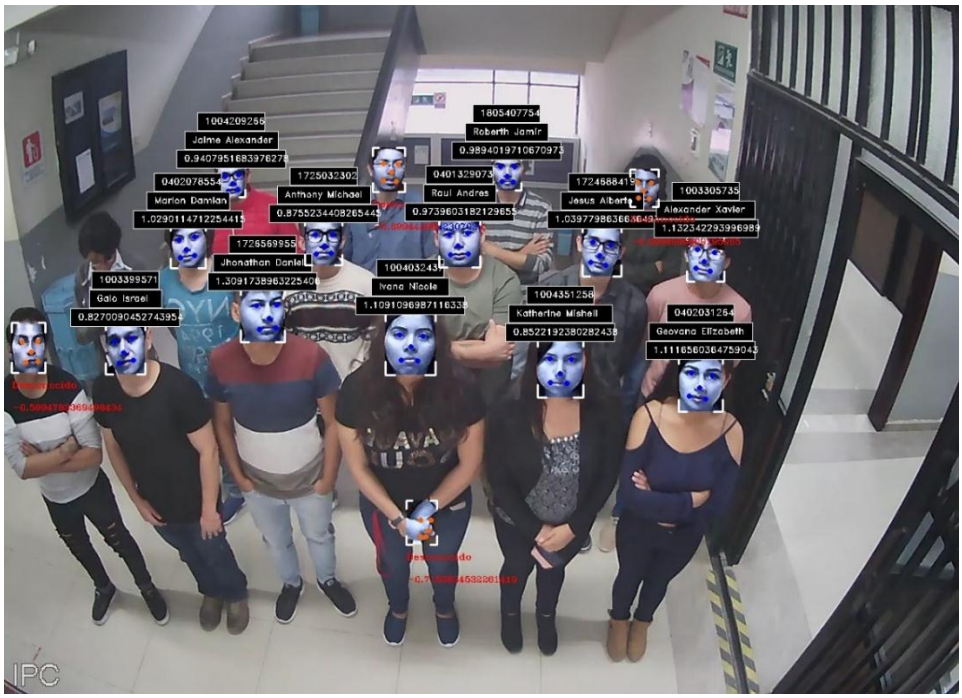
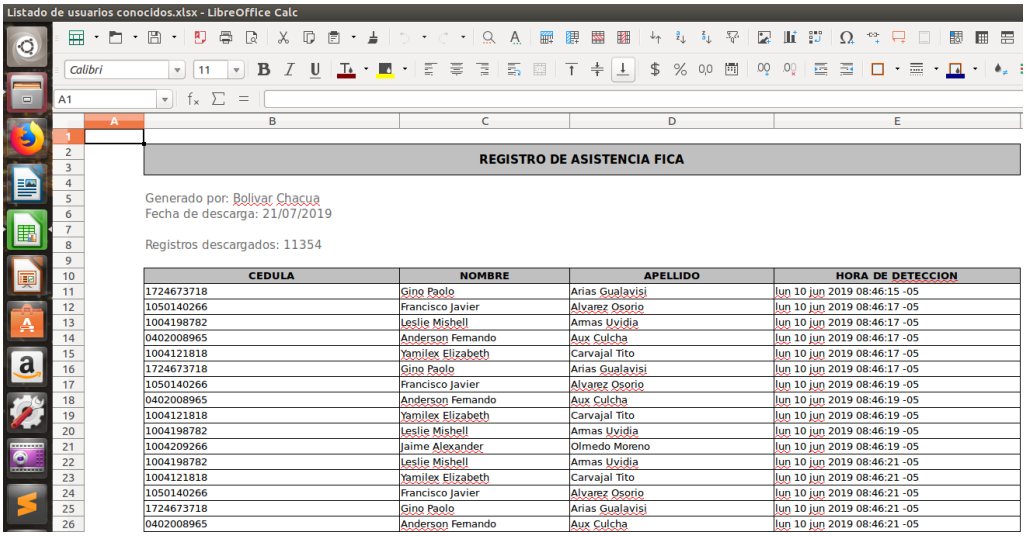


Figura 92. Verificación del Grupo 3.

Fuente: Autoría

4.5. Reporte

En cuanto al reporte generado se ha establecido el registro de los siguientes campos para personas conocidas, tales como: cedula, nombres, apellidos, fecha y hora de detección. La Figura 93 muestra el reporte generado por el sistema en formato ".xlsx".



Listado de usuarios conocidos.xlsx - LibreOffice Calc

REGISTRO DE ASISTENCIA FICA

Generado por: Bolivar Chacua
Fecha de descarga: 21/07/2019

Registros descargados: 11354

CEDULA	NOMBRE	APELLIDO	HORA DE DETECCION
1724673718	Gino Paolo	Arias Gualavisi	lun 10 jun 2019 08:46:15 -05
1050140266	Francisco Javier	Alvarez Osorio	lun 10 jun 2019 08:46:17 -05
1004198782	Leslie Mishell	Armas Uvidia	lun 10 jun 2019 08:46:17 -05
0402008965	Anderson Fernando	Aux Culcha	lun 10 jun 2019 08:46:17 -05
1004121818	Yamilex Elizabeth	Carvajal Tito	lun 10 jun 2019 08:46:17 -05
1724673718	Gino Paolo	Arias Gualavisi	lun 10 jun 2019 08:46:17 -05
1050140266	Francisco Javier	Alvarez Osorio	lun 10 jun 2019 08:46:19 -05
0402008965	Anderson Fernando	Aux Culcha	lun 10 jun 2019 08:46:19 -05
1004121818	Yamilex Elizabeth	Carvajal Tito	lun 10 jun 2019 08:46:19 -05
1004198782	Leslie Mishell	Armas Uvidia	lun 10 jun 2019 08:46:19 -05
1004209266	Jaime Alexander	Olmedo Moreno	lun 10 jun 2019 08:46:19 -05
1004198782	Leslie Mishell	Armas Uvidia	lun 10 jun 2019 08:46:21 -05
1004121818	Yamilex Elizabeth	Carvajal Tito	lun 10 jun 2019 08:46:21 -05
1050140266	Francisco Javier	Alvarez Osorio	lun 10 jun 2019 08:46:21 -05
1724673718	Gino Paolo	Arias Gualavisi	lun 10 jun 2019 08:46:21 -05
0402008965	Anderson Fernando	Aux Culcha	lun 10 jun 2019 08:46:21 -05

Figura 93. Reporte de personas reconocidas por el sistema.

Fuente: Autoría

A continuación, en la Figura 94 se presenta una galería de fotografías con capturas del rostro de los individuos reconocidos por el sistema (conocidos).

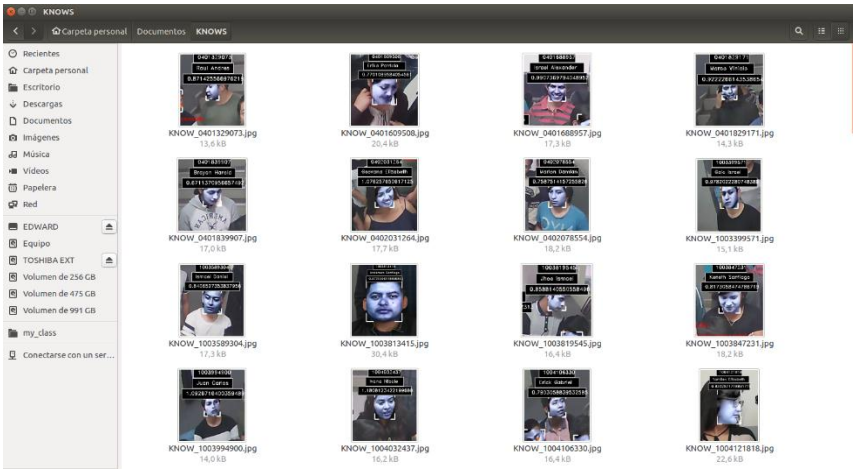


Figura 94. Capturas de fotografías de personas reconocidas por el sistema.

Fuente: Autoría

Para personas desconocidas se genera un reporte de la misma manera, con los siguientes campos, tales como: número de captura, fecha y hora de detección, además de una captura de rostro del individuo. La Figura 95 muestra el reporte generado por el sistema en formato ".xlsx".

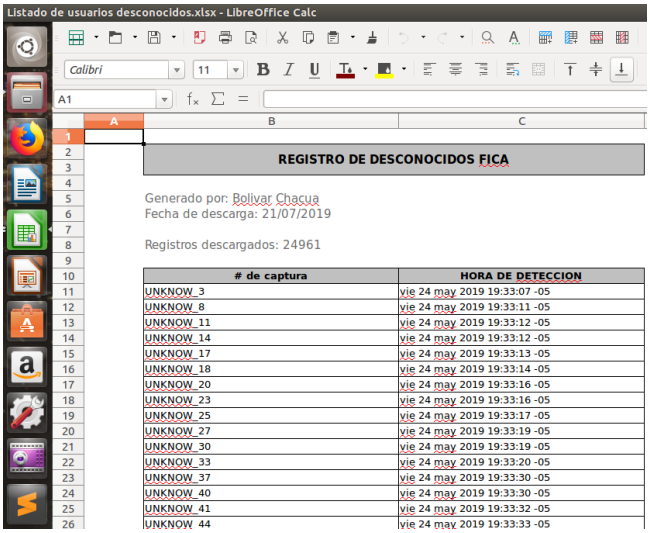


Figura 95. Reporte de personas no reconocidas por el sistema.

Fuente: Autoría

A continuación, en la Figura 96 se presenta una galería de fotografías con capturas del rostro de los individuos no reconocidos por el sistema (desconocidos).

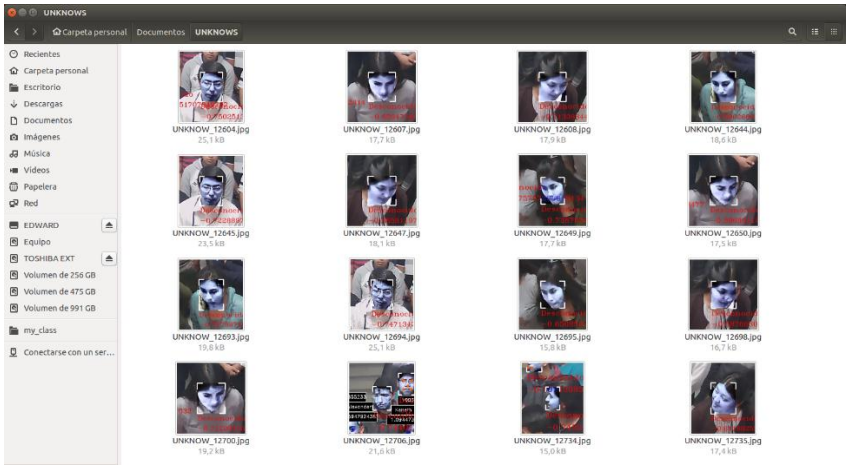


Figura 96. Capturas de fotografías de personas no reconocidas por el sistema.

Fuente: Autoría

4.6. Discusión de resultados

El reconocimiento facial puede ser una tarea fácil para los humanos y las máquinas en casos simples como el abordado en la sección 4.4.1.1, donde se ha obtenido un desempeño de un 96%; sin embargo, en escenarios desafiantes sin restricciones es una tarea difícil y propensa a errores. A pesar de que los modelos de aprendizaje profundo de última generación como las CNN´s, han logrado un rendimiento de clasificación a nivel humano en algunas tareas, actualmente no parecen ser capaces de tener un excelente desempeño sobre imágenes con bajas resoluciones; tal y como lo demuestra este estudio, claramente se ha obtenido un desempeño promedio de entre 65% y 75% en los escenarios expuestos en la sección 4.4.1.2 para diferentes grupos de 5 o más individuos aglomerados a cortas distancias, la Tabla 34 muestra de manera resumida las métricas que se obtuvieron:

Tabla 34. Métricas globales.

Métricas Globales						
			Promedio de mejores resultados			
Situación	**Distancia**	**# Frames**	**(Pr)**	**(Er)**	**(Rec)**	**(Esp)**
5 individuos	2 mts	1-10	72%	28%	72%	0%
12 individuos	2.5 y 1.5 mts.	1-10	74.8%	25.2%	75.79%	73.35%
16 individuos	3.5, 2.5 y 1.5 mts.	1-10	67.5%	32.5%	65.3%	100%
Global			71.43%	28.57%	71.03%	86.67%

Fuente: Autoría

Como era de esperarse los mejores resultados en cuanto a la precisión se encuentran entre un rango de distancias cortas de 1.5 a 2.5 mts, mientras que, en cualquier situación con distancias lejanas de 3.5 o más mts, resulta evidente el bajo desempeño del sistema, obteniendo así los peores resultados. Esto se debe a que, en varias ocasiones no se logra identificar al individuo o se genera

una identificación errónea por la resolución promedio de la zona ROI. Otro factor adicional es que la cámara ofrece imágenes con ciertas deformaciones en los bordes, que consecuentemente deforma los rostros que se obtienen en esas zonas (Figura 97).

Figura 97. Zona de la imagen con deformaciones.

Fuente: Autoría

Debido a los escasos detalles que existen de la forma en que funcionan internamente los sistemas de reconocimiento facial comerciales actuales, este estudio ha considerado la cantidad de identificaciones correctas con individuos aglomerados de forma simultánea en una colección de varios cuadros de video, lo cual ha arrojado porcentajes de precisión alentadores.

Es importante destacar que, las métricas obtenidas representan una aproximación de la efectividad del sistema en un ambiente sin restricciones, en el cual no se han considerado otras secuencias de video, donde la identificación facial sucede en algún/os frame/s de video de forma casual o persistente en algún/os individuo/s; esto sucede en gran proporción a cambios abruptos de pose y bajas resoluciones de la zona ROI. Debido a la labor que conlleva realizar un análisis

sobre múltiples cuadros, no se los ha realizado de forma exhaustiva, por lo que, si se incluyera dicho análisis, el porcentaje de precisión podría mejorar sutilmente.

Los resultados obtenidos son comparables al estudio realizado por Schroff (2015), donde alcanzaron tasas de precisión de un 76.3%, entrenando un modelo de aprendizaje profundo bajo una señal de supervisión denominada pérdida de triplete con una arquitectura CNN levemente diferente a la expuesta en este estudio y utilizando un conjunto de datos ligeramente más pequeño de 2.6 millones de imágenes faciales. Dicho estudio también afirma científicamente que, para obtener un considerable impulso de precisión en conjuntos de prueba, se debe emplear conjuntos de datos con más ejemplos de entrenamiento (variedad de clases) de hasta decenas de millones de imágenes faciales (Tabla 35).

Tabla 35. Precisión de un modelo de aprendizaje profundo tras aumentar la cantidad de imágenes faciales en un conjunto de datos de entrenamiento.

# Imágenes de entrenamiento	**Precisión**
2.600.000	76.3%
26.000.000	81.5%
52.000.000	85.1%
260.000.000	86.2%

Fuente: Adaptado de (Schroff, Kalenichenko, & Philbin, 2015)

En el mismo estudio se atribuye la importancia de emplear imágenes de alta calidad, la Tabla 36 muestra el tamaño de imágenes en pixeles y la precisión alcanzada en cada caso. Como se puede evidenciar, la tasa de precisión aumenta positivamente al elevar el tamaño de la imagen (resoluciones mayores a 80x80 pixeles); además, es preciso mencionar que el entrenamiento con imágenes faciales de menor resolución y el uso de técnicas SR (Super-Resolution) también podría mejorar los resultados obtenidos.

Tabla 36. Precisión obtenida a diferentes cantidades de pixeles.

#Pixeles	Precisión
1600	37.8%
6400	79.5%
14400	84.5%
25600	85.7%
65536	86.4%

Fuente: Adaptado de (Schroff, Kalenichenko, & Philbin, 2015)

Finalmente, la conclusión a la que se puede llegar con el análisis realizado, es que los mayores desafíos encontrados en el sistema se relacionan al tipo de óptica empleada, ya que se ha limitado el análisis de imágenes a una resolución promedio de 1280x960 pixeles y con ciertas deformaciones en los bordes, atribuidas a la ingeniería de la cámara. Todo lo mencionado anteriormente en esta sección refleja de cierta manera los resultados obtenidos en este estudio en particular, por lo que el uso de una cámara con mayor resolución de imagen (iguales o mayores a 1080p), prescindiendo de ciertas imperfecciones propias de la misma, podría mejorar los resultados obtenidos en este estudio.

4.7. Limitaciones del sistema

De acuerdo con el análisis cualitativo y cuantitativo realizado previamente, se presentan algunas de las conclusiones en base a las limitaciones encontradas en el sistema.

- ***Resolución de la cámara***

Como en cualquier proyecto de visión artificial, la óptica juega el rol más importante. Esto se debe a que la imagen que se obtiene a través de los dispositivos ópticos debe poseer una alta calidad para obtener buenos resultados mediante el tratamiento con algoritmos de extracción de características profundas, como el desarrollado en este estudio. Tal y como se mencionó en la sección 4.4.1.2, la resolución de los rostros capturados a distancias lejanas es demasiada baja, lo que conlleva a que el sistema no logre reconocer al/los individuos en la escena. La Figura 98 muestra las diferencias de calidad entre dos secciones de una imagen obtenida a diferentes distancias.

a) b)

Figura 98. Capturas de rostro realizadas a una distancia a) lejana y b) corta.

Fuente: Autoría

Por lo que se apreciar, la Figura (98a) contiene a 6 individuos, de los cuales 5 han sido detectados correctamente a una distancia mayor a 4 mts, mientras que la Figura (98b) contiene a 4

individuos, de los cuales 4 han sido detectados correctamente a una distancia de 2 mts; en la situación (98a) se denota una evidente pérdida de calidad de la imagen a largas distancias, y aunque el detector facial aún puede lidiar con la baja calidad/resolución de la imagen, la identificación facial resulta ser efectiva solamente a cortas distancias, tal y como lo corrobora este estudio. Los resultados también pueden atribuirse a: aberraciones cromáticas o imperfecciones encontradas en la imagen.

- ***Cambio de pose***

Según los análisis realizados, el cambio de pose no resulta ser un problema para la identificación facial en ambientes controlados, ya que todas las pruebas realizadas en la sección 4.4.1.1 resultaron exitosas. La situación cambia en un entorno no controlado, debido a que existen muchas variaciones de postura de los individuos. Las pruebas realizadas demostraron efectividad cuando los rostros de los individuos estaban parcialmente de frente, mas no cuando se encontraban completamente de lado a cualquier distancia; esto es evidente, debido a que, la baja resolución de la imagen nuevamente presenta inconvenientes en poses laterales. Las variaciones laterales del rostro afectan el rendimiento de un sistema de reconocimiento facial, puesto que el entrenamiento del modelo de aprendizaje profundo fue alimentado por un conjunto de datos de entrenamiento que presenta imágenes faciales de forma frontal en su gran mayoría. Para lidiar con el cambio de pose se podría aplicar un proceso de alineación facial completo, similar al propuesto en el estudio de Arcoverde (2014), el cual propone el uso de puntos de referencia de los ojos (*landmarks*) para lograr una alineación canónica de la zona ROI del rostro basada en la traslación, rotación y escala, de manera que los ojos encuentren una línea horizontal con coordenadas similares en el eje *y*. El proceso de alineación facial expuesto queda a consideración en un trabajo futuro.

La Figura 99 muestra el bajo rendimiento del sistema a largas distancias y con variaciones de pose lateral.

Figura 99. Identificación facial con bajo desempeño.

Fuente: Autoría

Es conveniente mencionar que, los sistemas de reconocimiento facial que operan con CCTV ofrecen buen rendimiento con rostros obtenidos de frente, mientras que los rostros de lado no presentan buenos resultados y son almacenados para fines de análisis forense. Un ejemplo claro de lo mencionado es el software de monitoreo facial "SenseFace Surveillance System" desarrollado por el gigante chino en desarrollo de aplicaciones de inteligencia artificial *Sensetime*. El siguiente enlace muestra una demo de la robustez del sistema desarrollado por dicha empresa:

https://bit.ly/2LAlcNp

Finalmente, una vez presentado el análisis cualitativo y cuantitativo, el diseño del sistema de reconocimiento facial empleando técnicas de inteligencia artificial de vanguardia ofrece una precisión (promedio) máxima de identificación de 71.43% de 1.5 a 3.5 metros de distancia.

El siguiente enlace muestra una pequeña demostración de los resultados obtenidos por el sistema desarrollado en este estudio:

https://bit.ly/2lUFDJd

5. CAPÍTULO V. Conclusiones y Recomendaciones.

5.1. Conclusiones

En la actualidad el aprendizaje profundo es un campo de la inteligencia artificial bastante prominente en la rama de la visión por computador debido a que emplean técnicas/enfoques bastante robustos para la extracción de características profundas y reconocimiento de patrones en imágenes, videos, voz, datos, entre otros. Las redes neuronales convolucionales (CNN´s) son una herramienta poderosa para la extracción de características profundas sobre imágenes y videos, ya que este tipo de enfoque consta de decenas de capas ocultas, con el fin de descubrir y reconocer patrones en los objetos del mundo que nos rodea.

Las arquitecturas CNN Inception Resnet logran altos porcentajes de precisión en tareas de visión por computador, debido a que agrupan varios filtros de convolución/max-pooling, y funciones de activación en un solo bloque o modulo, por lo que la CNN puede apilar muchos de estos bloques permitiéndole obtener más profundidad y por lo tanto alta precisión frente a otras arquitecturas.

Un factor determinante para contribuir a la precisión de un modelo de aprendizaje profundo es la selección de un conjunto de datos de entrenamiento con una gran cantidad de ejemplos, puesto que las arquitecturas CNN alcanzan mucho mayor éxito en grandes volúmenes de datos.

El conjunto de datos VGGFace contiene una gran cantidad de ejemplos de entrenamiento con el objetivo de hacer frente a los retos que impone el reconocimiento facial en entornos desafiantes, permitiendo realizar investigaciones de alto nivel sin la necesidad de elaborar un conjunto de datos propio.

Es importante el uso de un conjunto de datos de entrenamiento (training set) y de prueba (test set) en cualquier enfoque de aprendizaje automático y profundo, particularmente en este estudio el primer conjunto sirve de base para generalizar un modelo mediante el ajuste de pesos y bias (sesgo) adecuados para múltiples clases o identidades a través de incrustaciones de 512 bytes únicas, y el segundo conjunto determina cual es la probabilidad de que dadas un par de incrustaciones generadas a partir de rostros nunca vistos por la red sean o no la misma persona.

Antes de emplear cualquier conjunto de datos para el entrenamiento de un modelo de aprendizaje profundo, este debe ser preparado mediante algunas técnicas de aprendizaje automático aplicables a imágenes, con la finalidad de obtener un conjunto de datos adecuado para la entrada de la red y con mejores características que puedan generalizar el modelo y evitar el sobreajuste de la CNN, tales como: cálculo de la media, desviación estándar; recorte, redimensionamiento y rotación aleatoria de la zona ROI, entre otras.

La conversión del espacio de colores RGB a YCbCr y la adición de la ecualización de histograma en las imágenes de prueba en tiempo real puede brindar resultados positivos en ambientes controlados y no controlados; tal y como lo evidencia este estudio, un sistema de reconocimiento facial puede presentar una mejor robustez frente a condiciones variantes de iluminación.

CUDA (Arquitectura Unificada de Dispositivos de Cómputo) es una tecnología de procesamiento paralelo tremendamente importante en el área de la computación de altas prestaciones (HPC), ya que aprovecha la potencia de cálculo de los núcleos físicos de una GPU Nvidia para el tratamiento de cantidades masivas de datos enfocados al desarrollo de investigaciones y/o aplicaciones en diferentes campos científicos, incluyendo la visión por computador como se evidencia en este estudio.

El uso de la biblioteca cuDNN acelera la capacitación de redes neuronales profundas en el marco de aprendizaje profundo de Tensorflow, puesto que proporciona implementaciones altamente optimizadas para rutinas estándar, tales como: capas de convolución, pooling, normalización, y activación en diferentes arquitecturas CNN, especialmente las Resnet.

El uso de las señales de pérdida de Softmax y Central en la etapa de entrenamiento del modelo de aprendizaje profundo son de vital importancia, ya que dichas señales permiten obtener características altamente discriminativas para cada clase del conjunto de datos de entrenamiento y por consiguiente ofrecer un buen rendimiento en conjuntos de datos de prueba.

El entrenamiento de la CNN mediante el cálculo de tasas de aprendizaje adaptativo y el tratamiento de gradientes dispersos que propone el optimizador ADAM, permite obtener los mínimos locales de cada mini-lote de entrenamiento de manera más eficiente y parcialmente más rápida que con otros métodos tradicionales de SGD.

Luego de entrenar una red demasiado grande como lo es una CNN esta puede ser usada para ejecutar un proceso de inferencia o transferencia de aprendizaje, donde se encarga de clasificar, reconocer y procesar nuevas entradas con una latencia baja, por lo que para este proceso los pesos y bias del modelo previamente entrenado se encuentran congelados y se inicializan, evitando realizar muchas de las operaciones de la CNN nuevamente.

Las incrustaciones faciales que genera el modelo de aprendizaje profundo son fácilmente abordables por cualquier clasificador como KNN(K-Vecinos-Cercanos) y SVM (Máquinas de Vector Soporte), ya que las características discriminativas de cada clase son ventajosamente agrupables y ocupan un espacio único diferenciable sobre un plano bidimensional.

De acuerdo al análisis del rendimiento del clasificador realizado en este estudio, se ha logrado comprobar que el enfoque SVM con kernel lineal consigue entrenar múltiples clasificadores de hasta más de 400 clases con una variación de 80 ejemplos de entrenamiento y 20 ejemplos de prueba, ofreciendo una alta eficiencia en el conjunto de pruebas de 91.5% (Tabla 21).

Las incrustaciones faciales obtenidas a distancias lejanas ofrecen bajo desempeño en la tarea de reconocimiento facial en sistemas de video vigilancia con imágenes de baja resolución, ya que suprimen una gran cantidad de características, las cuales no pueden ser correctamente correlacionadas en el clasificador, así se tratasen de una clase entrenada previamente en el mismo.

El uso de un umbral de reconocimiento facial limita las identidades conocidas y desconocidas, esto es así dado que, el clasificador ofrece una alta probabilidad cuando existe correlación a una clase entrenada en el mismo, caso contrario establece una baja probabilidad; dichas probabilidades dependen en gran manera de la calidad de las imágenes que se ponen a prueba.

Existen muchos procesos bajo el funcionamiento de un sistema de reconocimiento facial comercial, ya que están destinados a procesar enormes cantidades de imágenes y datos, por lo que es común que operen con servidores locales altamente robustos o mediante el arrendamiento de servidores en la nube con baja latencia.

La forma de evaluar los modelos de aprendizaje profundo y automático fueron llevados a cabo mediante el método cross validation y el análisis de medidas cuantitativas de calidad como la matriz de confusión, a través de métricas estadísticas como la exactitud o precisión, tasa de error, sensibilidad, y especificidad.

El reconocimiento facial es un área de estudio de gran interés en el campo de investigación de la inteligencia artificial, ya que representa una de las tecnologías más disruptivas y con un gran avance económico, sin embargo, aún es propensa a errores y una línea abierta de investigación.

Los experimentos realizados en este estudio arrojaron mejores resultados en cuanto a la precisión cuando los individuos se encontraban entre un rango de distancias cortas de 1.5 a 2.5 mts (Figura 92), mientras que, en cualquier otra situación con distancias lejanas de 3.5 mts o más, resulto evidente el impacto negativo que sufre el sistema, tal y como lo demuestra este estudio, bajo condiciones controlables el rendimiento del sistema alcanza un 96% de precisión (Tabla 26), no sucede lo mismo bajo condiciones no controlables donde alcanza un 71.43% de precisión a distancias cortas (Tabla 34).

5.2. Recomendaciones

Antes de comenzar cualquier investigación en el marco de la inteligencia artificial es importante poseer todo el conjunto de herramientas de software y hardware, así como también de un conocimiento general de los principales algoritmos de aprendizaje supervisado y no supervisado.

Es muy recomendable analizar a detalle la arquitectura y funcionamiento de una red neuronal con la finalidad de comprender el funcionamiento general de cualquier variación de las mismas, tales como lo son las redes neuronales perceptron multicapa (MLP), redes neuronales convolucionales (CNN´s), redes neuronales recurrentes (RNN), redes adversarias generativas (GAN), entre otras.

Al existir muchas arquitecturas de redes neuronales convolucionales es recomendable emplear una arquitectura que no demande de muchos recursos computacionales y posea muy buenas referencias de éxito en diferentes estudios de impacto relacionados al tema a abordar, ya que el simple uso de una CPU convencional podría llevar semanas o meses de entrenamiento, lo que no sucede con un equipo que posea altos recursos de cómputo como por ejemplo el uso de una CPU de última generación o una GPU dedicada con tecnología de procesamiento paralelo.

En lo relacionado a la precisión del modelo de aprendizaje profundo, sería muy conveniente extender el conjunto de datos al doble o triple del usado en este estudio, con el fin de generar incrustaciones faciales mucho más discriminativas ante una enorme diversidad de clases, algunos de los conjuntos de datos que pueden emplearse son los siguientes: CASIA-WebFace (10.575 clases), MegaFace Dataset (690.572 clases), MS-Celeb-1M (100.000 clases), entre otros.

Es muy recomendable usar el método de ajuste fino (fine-tunning) en redes neuronales convolucionales, puesto que permite utilizar modelos de aprendizaje profundo pre-entrenados para reconocer clases en las que no fueron entrenados originalmente, por lo que de esta manera es posible ampliar el aprendizaje previo de un modelo pre-entrenado mediante el reemplazo de las capas completamente conectadas (fully connected) del mismo, por un nuevo conjunto de capas que se encargan de aprender los patrones de las capas convolucionales aprendidas anteriormente en la red y la actualización de pesos y bias del nuevo conjunto de datos a través del algoritmo de propagación hacia atrás (backpropagation), sin penalizar su rendimiento y minimizando el tiempo de entrenamiento que supondría emplear varios conjuntos de datos nuevamente, además, este método puede llevar a una mayor precisión.

El impacto de usar diferentes formatos de imágenes con pérdida (JPG) y sin pérdida (GIFF) de información en redes neuronales convolucionales es un campo poco analizado, ya que la mayoría de investigaciones usan formatos de imagen con pérdidas como lo son JPG, JPG2000, entre otros, por lo que se recomienda usar de forma experimental diferentes formatos de imágenes y comprobar los niveles de precisión que conlleva usar cada uno.

Se recomienda el uso de la plataforma TensorRT para el proceso de inferencia o transferencia de aprendizaje en aplicaciones de aprendizaje profundo. Esta herramienta puede minimizar la latencia de cualquier aplicación, aumentando la rapidez de inferencia en cualquier GPU Nvidia, lo cual es requisito fundamental para muchos servicios en tiempo real, o aplicaciones automáticas e integradas en dispositivos con limitada capacidad de cómputo.

Se recomienda variar el hiperparámetro de ajuste de la función de centro λ con diferentes valores (0.90 a 1) y emplear diferentes optimizadores de funciones de pérdida (SGD Estandar,

RMSprop, Adadelta, Adagrad) de forma experimental, ya que la precisión de verificación de las características profundamente aprendidas puede verse beneficiada en algún caso.

Variar el tamaño de la incrustación puede tener un impacto positivo o negativo en la tarea de reconocimiento facial y por consiguiente en el consumo de recursos de hardware, por lo que se recomienda probar con algunas variaciones de tamaño del tensor de salida de la capa fully connected, con la finalidad de obtener incrustaciones de un tamaño de 128, 256, 1024, o 2048 bytes y verificar la precisión de cada una de ellas.

Construir e implementar con éxito modelos de alto nivel de abstracción en el campo del aprendizaje profundo resulta ser un desafío e inclusive un arte. Es por eso que existe una gama de marcos o frameworks que permiten simplificar la excesiva programación y la complejidad de este campo de la inteligencia artificial, por lo que se recomienda probar con diferentes marcos tales como: Caffe2, Tensorflow 2.0, Microsoft Cognitive Toolkit, Pytorch, Mxnet, Keras, entre otros.

La gran mayoría de las técnicas empleadas en este estudio son usadas en la tecnología de reconocimiento facial, es por eso que si se requiere realizar una implementación en una situación de la vida real se recomienda usar diferentes tipos de cámaras para diferentes entornos y un servidor medianamente potente; por ejemplo, una cámara de resolución promedio de 360p, 480p y 720p puede obtener buenos resultados en entornos controlados, por el contrario, una cámara de resolución alta de 1080p, 2k o inclusive 4k puede resultar mucho mejor para entornos no controlados.

Teniendo en cuenta las limitaciones del sistema, una buena manera de superarlas es recolectando una base de datos de rostros de los individuos en el entorno en el que se movilizan de forma masiva, estoy sería muy beneficioso ya que es posible aplicar técnicas de clustering de

forma no supervisada, con el fin de agrupar conjuntos de datos faciales no etiquetados y generar nuevos conjuntos de datos por individuo para posteriormente alimentar un nuevo modelo de aprendizaje profundo con características de individuos mucho más reales al entorno de despliegue del sistema.

Se recomienda ampliamente el desarrollo en producción de esta herramienta para fines de control de seguridad, ya que el enfoque es discriminante ante individuos conocidos y desconocidos, además de ser una aplicación funcional adecuada bajo una óptica óptima.

Por último y no menos importante, si se requiere más velocidad en el tiempo de ejecución de código del sistema, valdría la pena la migración del lenguaje interpretado de Python a un lenguaje compilado C++. Actualmente Tensorflow dispone de un envoltorio que contiene todos los mecanismos para la construcción y ejecución de cualquier flujo de datos en código C++ mediante TensorFlow's C++ API.

Referencias

Alahi, A., Vandergheynst, P., Bierlaire, M., & Kunt, M. (2010). Cascade of Descriptors to Detect and Track Objects Across Any Network Cameras. *Computer Vision and Image Understanding*, 624-640.

Álvarez López, G. (2016). *Desarrollo de una aplicación para ordenadores para la detección y reconocimiento facial de alumnos para el posterior control de asistencia.* Gandia.

Alvear Puertas, V. E. (2016). *SISTEMA ELECTRÓNICO CON APLICACIÓN IoT PARA MONITOREO FACIAL QUE BRINDE ESTIMADORES DE DESCONCENTRACIÓN DEL ESTUDIANTE UNIVERSITARIO EN EL AULA A ESCALA DE LABORATORIO.* Ibarra.

Arcoverde Neto, E. N., Duarte, R. M., Barreto, R. M., Magalhães, J. P., M. Bastos, C. C., Ing Ren, T., & C. Cavalcanti, G. D. (2014). Real-Time Head Pose Estimation for Mobile devices. *ResearchGate*.

Atiqur, R. (2015, Marzo 27). *Mathworks*. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/50077-face-detection-using-viola-jones-algorithm

Aurélien , G. (2017). *Hands-On Machine Learning with Scikit-Learn & Tensorflow.* O'Reilly Media.

Behzad, H., & Mohammad H., M. (2017). Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 30-40.

Buitink, L., Louppe, G., Blondel, M., Pedregosa, F., Muller, A. C., Grisel, O., . . . Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv* .

Caballero Barriga, E. R. (2017). *APLICACIÓN PRÁCTICA DE LA VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE ROSTROS EN UNA IMAGEN, UTILIZANDO REDES NEURONALES Y ALGORITMOS DE RECONOCIMIENTO DE OBJETOS DE LA BIBLIOTECA OPENCV.* Bogota.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). VGGFace2: A dataset for recognising faces across pose and age. *IEEE*, 67-74.

Casallas, R., & Yie, A. (2016). *Universidad de los Andes.* Retrieved from https://profesores.virtual.uniandes.edu.co/~isis2603/dokuwiki/lib/exe/fetch.php?media=principal:isis2603-modelosciclosdevida.pdf

Chazallet, S. (2016). *Python 3 Los fundamentos del lenguaje 2ª Edición.* Barcelona: Ediciones ENI.

Corsair Components. (2016). *CORSAIR*. Retrieved from https://www.corsair.com/es/es/Power/Plug-Type/cxm-series-2015-config/p/CP-9020061-NA

Dahua. (2012). DH-IPC-HDW2100.

Deng, J., Zhou, Y., & Zafeiriou, S. (2017). Marginal Loss for Deep Face Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 60-68.

Domínguez Pavón , S. (2017). *Reconocimiento facial mediante el Anáisis de Componentes Principales (PCA)* . Sevilla.

Espinoza Olguín, D. E., & Jorquera Guillen, P. I. (2015). *Reconocimiento Facial.* Valparaíso.

Fernández García, N. L. (2016). *Introducción a la Visión Artificial.* Córdoba.

Flores, E. (2016). *Ingenieria de Software*. Retrieved from http://ingenieriadesoftware.mex.tl/61885_Modelo-V.html

García Mateos, G. (2007). *Procesamiento de caras humanas mediante integrales proyectivas.* Murcia .

García Santillán, I. D. (2008). *Visión Artificial y Procesamiento Digital de Imágenes usando Matlab.* Ibarra.

García, M. A., & Martínez, J. M. (2013). Enhanced people detection combining appearance and motion information. *Electronics Letters*, 256-258.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* Cambridge: MIT press.

Guo, S., Chen, S., & Li, Y. (2016). Face recognition based on convolutional neural network and support vector machine. *IEEE International Conference on Information and Automation (ICIA)*, 1787-1792.

HIKVISION Digital Technology Co. (2019). *HIKVISION*. Retrieved from https://www.hikvision.com/es-la/Products/Network-Camera/EasyIP-2.0/4MP/DS-2CD2142FWD-I(W)(S)

Huang, D., Shan, C., Ardebilian, M., Wang, Y., & Chen, L. (2011). Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 765-781.

Huang, J., Rathod, V., Sun, C., Zhu, M., & Korattikara, A. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7310-7311.

HUAWEI. (2018). *HUAWEI Latin*. Retrieved from https://consumer.huawei.com/latin/phones/mate10-lite/

Idento. (2015). *Idento Es*. Retrieved from https://www.idento.es/blog/desarrollo-web/png-vs-jpg-que-formato-de-imagen-es-mejor-para-la-web/

Intel Corporation. (2017). *INTEL*. Retrieved from https://ark.intel.com/es/products/126686/Intel-Core-i7-8700-Processor-12M-Cache-up-to-4-60-GHz-

ISO/IEC/IEEE. (2011). *IEEE. 29148: 2011-Systems and software engineering-Requirements engineering.* IEEE.

Jain, A. K., Dass, S. C., & Nandakumar, K. (2004). Can soft biometric traits assist user recognition? *International Society for Optics and Photonics.*, 561-573.

Jimenez Encalada, J. C. (2015). *Implementación de técnicas de identificación de objetos aplicados al reconocimiento facial en videovigilancia del SIS-ECU-911*. Cuenca.

Jung, M. Y. (2006). *Biometric Market and Industry Overview.* Bruselas : International Biometric Group.

Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., & Brossard, E. (2016). The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. *Conferencia IEEE sobre Visión por Computador y Reconocimiento de Patrones*, 4873-4882.

Khan, S., Rahmani, H., Shah, S., & Bennamoun, M. (2018). *A guide to convolutional neural networks for computer vision.* Crawley: Morgan & Claypool Publishers.

Kingma, D., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *arXiv*.

Koldo. (2018, Abril 22). *koldo Pina*. Retrieved from https://koldopina.com/matriz-de-confusion/

Learned Miller, E., Huang, G., RoyChowdhury, A., Li, H., & Hua, G. (2016). Labeled Faces in the Wild: A Survey. *Springer*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning.* New York: Nature.

Li, D., & Dong, Y. (2014). *Deep Learning: Methods and Applications.* Now (the essence of Knowledge).

Li, P., Flynn, P. J., Prieto, L., & Mery, D. (2019). Face Recognition in Low Quality Images: A Survey. *ResearchGate*.

Lior, W., Tal, H., & Itay, M. (2011). Face Recognition in Unconstrained Videos with Matched Background Similarity. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). SphereFace: Deep Hypersphere Embedding for Face Recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212-220.

Lutz, M. (2013). *Learning Python.* Sepastopol: O´REILLY.

Marciniak, T., Chmielewska, A., Weychan, R., Parzych, M., & Dabrowski, A. (2013). Influence of low resolution of images on reliability. *Springer*.

Martínez Guerrero, M. (2018). *Reconocimiento facial para la identificación de usuarios.* Valencia.

Melekhov, I., Juho, K., & Esa, R. (2016). Image patch matching using convolutional descriptors with euclidean distance. *Asian Conference on Computer Vision. Springer*, 638-653.

Meng, R., Shengbing, Z., Yi, L., & Meng, Z. (2014). CUDA-based Real-time Face Recognition System. *IEEE*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antanoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv.org*.

Morales, E., & Escalante, H. J. (2012). *Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).* Retrieved from https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/refuerzo.pdf

Nefian, A. V., & Hayes, M. H. (1998). Hidden Markov Models for Face Recognition. *Proceedings of the 1998 IEEE International Conference*, 2721-2724.

Nguyen, H. V., & Bai, L. (2010). Cosine Similarity Metric Learning for Face Verification. *Springer*.

Nilsson, N. (2001). *Inteligencia Artificial.* España: Mc Graw HILL.

NVIDIA Corporation. (2016). *NVIDIA*. Retrieved from https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080/

Nvidia Developer. (2019). *Nvidia Developer*. Retrieved from https://developer.nvidia.com/cudnn

Nvidia High Performance Computing. (2019). *Nvidia High Performance Computing*. Retrieved from https://developer.nvidia.com/cuda-zone

OpenCV Org. (2019). *OpenCV*. Retrieved from https://docs.opencv.org/2.4.13.7/index.html

Pajares Martinsanz, G., & Santos Peñas, M. (2006). *Inteligencia Artificial e Ingenieria del Conocimiento.* Mexico: Alfaomega.

Palma Mendez, J. T., & Marín Morales, R. (2004). *Inteligencia Artificial.* Madrid: Mc Graw Hill.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. *BMVC*.

Pérez García, A. (2013). Software Requirements Specification.

Pérez López, C. (2005). *MUESTREO ESTADÍSTICO.* Madrid: PEARSON EDUCACIÓN.

Pérez Montero, M. A. (2014). *SlideShare.* Retrieved from https://es.slideshare.net/Marco1984/diseo-de-software-modelo-lineal-presentacion

Pérez, C., & Santín, D. (2007). *Minería de Datos. Técnicas y Herramientas.* Madrid: Thomson .

Platero Dueñas, C. (2009). Apuntes de visión artificial. *Departamento de Electrónica, Automática e Informática Industrial*.

Sadhya, D., Gautam, A., & Singh, S. K. (2017). Performance Comparison of Some Face Recognition Algorithms on Multi-covariate Facial Databases. *IEEE,* 1-5.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. *IEEE*.

Scikit-Learn Org. (2018). *Scikit Learn Org*. Retrieved from https://scikit-learn.org/stable/modules/svm.html

Shai, S.-S., & Shai, B.-D. (2014). *Understanding Machine Learning: From Theory to Algorithms.* New York: Cambridge University Press.

Sobrado Malpartida, E. (2003). *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot.* Lima.

Stan Z, L., & Anil K, J. (2011). *Handbook of Face Recognition.* Londres: Springer.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. *AAAI*.

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701-1708.

TP-Link. (2019). *TP-Link Argentina*. Retrieved from https://www.tp-link.com/ar/home-networking/wifi-router/tl-wr841n/#overview

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of machine learning research*, 2579-2605.

Villegas Quezada, C. (2005). *Reconocimiento de rostros utilizando análisis de componentes principales: limitaciones del algoritmo.* Mexico, D.F.

Viola, P. A., & Jones, M. J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE CVPR* .

Viola, P., & Jones, M. (2004). Robust Real-time Object Detection. *SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF*. Vancouver.

Visual Geometry Group. (2019). *Robots @Oxford - University of Oxford*. Retrieved from http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A Discriminative Feature Learning Approach for Deep Face Recognition. *Springer*.

Wu, Z., Shen, C., & van den Hengel, A. (2019). Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Revisiting the resnet model for visual recognition. Pattern Recognition*, 119-133.

Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning Face Representation from Scratch. *arXiv*.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2015). Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE*.

Zhang, K., Zhang, Z., Wang, H., Li, Z., Qiao, Y., & Liu, W. (2017). Detecting Faces Using Inside Cascaded Contextual CNN. *IEEE*.

Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover Canonical-View Faces in the Wild with Deep Neural Networks. *arXiv preprint arXiv*.

Anexos

- Anexo 1: https://bit.ly/2HByBkS
- Anexo 2: https://bit.ly/2Qhzzpl
- Anexo 3: https://bit.ly/2JW15HI
- Anexo 4: https://bit.ly/2YzP0w2
- Anexo 5: https://bit.ly/2HCUY9R
- Anexo 6: https://bit.ly/2YLILFt
- Anexo 7: https://bit.ly/30zBYjL
- Anexo 8: https://bit.ly/2M9SSCG
- Anexo 9: https://bit.ly/2HNklVn
- Anexo 10: https://bit.ly/2QnmRW8
- Anexo 11: https://bit.ly/30JhyFg
- Anexo 12: https://bit.ly/2HwRNAy
- Anexo 13: https://bit.ly/2JTSnK5
- Anexo 14: https://bit.ly/2VI240q
- Anexo 15: https://bit.ly/2xvJD5z