

**UNIVERSIDAD NACIONAL DANIEL ALCIDES CARRION**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE FORMACION PROFESIONAL DE INGENIERIA DE**

**SISTEMAS Y COMPUTACIÓN**



**TESIS**

**Algoritmos de reconocimiento facial mediante aprendizaje automático**

**para la identificación de personas en una institución educativa de**

**Pasco - 2021**

**Para optar el título profesional de:**

**Ingeniero de Sistemas y Computación**

**Autor: Bach. Cristian Brayan MUÑOZ VILLANUEVA**

**Asesor: Mg. Marco Antonio DE LA CRUZ ROCCA**

**Cerro De Pasco – Perú - 2022**

**UNIVERSIDAD NACIONAL DANIEL ALCIDES CARRION**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE FORMACION PROFESIONAL DE INGENIERIA DE**

**SISTEMAS Y COMPUTACIÓN**



**TESIS**

**Algoritmos de reconocimiento facial mediante aprendizaje automático**

**para la identificación de personas en una institución educativa de**

**Pasco - 2021**

**Sustentada y aprobada ante los miembros del jurado:**

---

Mg. Raúl Delfín CONDOR BEDOYA  
PRESIDENTE

---

Mg. Teodoro ALVARADO RIVERA  
MIEMBRO

---

Mg. Pit Frank ALANIA RICARDI  
MIEMBRO

## **DEDICATORIA**

A mi Mamá por todo su amor e impulso  
durante la realización de la tesis

## **AGRADECIMIENTO**

Agradecido eternamente con **DIOS** por haberme otorgado una familia maravillosa, quienes siempre han creído en mí, enseñándome el valor del sacrificio, a enfocarme en mis metas. A todos ellos dedico este trabajo porque han fomentado en mí el deseo de superación. Lo que me ayudo a realizar esta investigación. Espero contar con su valioso e incondicional apoyo.

## RESUMEN

Pongo a vuestra consideración la Tesis “Algoritmos de reconocimiento facial mediante aprendizaje automático para la identificación de personas en una institución educativa de Pasco - 2021”. El objetivo que persigue esta investigación es determinar el algoritmo de reconocimiento facial que mejor resulte en la identificación de personas de una institución educativa de Pasco – 2021, para ello se realizó una investigación de tipo experimental, con una muestra de tipo intencionada no probabilística conformada 310 imágenes diversas de 10 estudiantes de la Facultad de Ingeniería UNDAC.

Se evaluaron 3 algoritmos de reconocimiento facial: Eigenface, Fisherface y Local Binary Pattern (LBP) mediante las métricas de exactitud y precisión de aprendizaje automático, el experimento se realizó en tres escenarios de entrenamiento de los modelos algorítmicos con diferente número de imágenes, para luego testear cada modelo con imágenes de los 10 estudiantes, los resultados indican que el algoritmo de reconocimiento facial LBP, es el que mejor porcentaje de aciertos ofrece en cuanto a la métrica de exactitud con un 94.37%, del mismo modo en las pruebas para evaluar el porcentaje de precisión el algoritmo LPB presenta el valor más alto con 97.83%, ocupando el segundo lugar el algoritmo Fisherface.

De lo que se desprende que el algoritmo de reconocimiento facial LBP resulta siendo el que mejores resultados ofrece en porcentajes de exactitud y precisión en la identificación de personas.

**Palabras clave:** Algoritmos de reconocimiento facial, aprendizaje automático, identificación de personas.

El Autor.

## ABSTRACT

I present for your consideration the Thesis "Facial recognition algorithms through automatic learning for the identification of people in an educational institution of Pasco - 2021". The objective of this research is to determine the facial recognition algorithm that best results in the identification of people in an educational institution of Pasco - 2021, for this an experimental type of research was conducted, with a non-probabilistic purposive sample of 310 different images of 10 students of the Faculty of Engineering UNDAC.

Three face recognition algorithms were evaluated: Eigenface, Fisherface and Local Binary Pattern (LBP) through the metrics of accuracy and precision of machine learning, the experiment was conducted in three training scenarios of algorithmic models with different number of images, and then test each model with images of the 10 students, the results indicate that the LBP face recognition algorithm, is the one that offers the best percentage of hits in terms of accuracy metrics with 94.37%, likewise in the tests to evaluate the percentage of accuracy the LPB algorithm presents the highest value with 97.83%, occupying the second place the Fisherface algorithm. It follows that the LBP face recognition algorithm is the one that offers the best results in percentages of accuracy and precision in the identification of persons.

**Keywords:** Face recognition algorithms, machine learning, persons identification.

## INTRODUCCIÓN

Con el avance tecnológico acelerado que se ha dado en este nuevo milenio, muchas herramientas y métodos computacionales se han desarrollado enormemente, una de las áreas más beneficiadas con esto ha sido la inteligencia artificial, donde campos como el aprendizaje automático han visto un sin fin de posibilidades que faciliten la vida y el trabajo diario.

En esta variedad de posibilidades se encuentra la identificación biométrica y dentro de ella el reconocimiento facial. Los rostros humanos son únicos y no pueden ser replicados, teniendo esto presente, es que mediante la programación por computadora se pueden ejecutar algoritmos orientados a este tema y, generar modelos que aprendan a identificar personas por sí mismo, sin intervención humana.

En ese contexto se ha desarrollado esta investigación, analizando algoritmos de reconocimiento facial mediante aprendizaje automático con el fin de determinar aquel que mejor logre identificar a una persona,

La investigación consta de 4 capítulos, a groso modo se describe cada uno de ellos:

En el Capítulo I, se presenta la definición y formulación del problema, los objetivos de la investigación, la justificación del estudio entre otros aspectos necesarios apegados a la estructura según el reglamento de grados y títulos.

En el Capítulo II, se muestra todo lo relacionado al marco teórico necesario para comprender el tema de investigación, implica describir antecedentes relacionados con el tema. las bases teóricas para el desarrollo del estudio, así como la definición de términos, la formulación de hipótesis investigativas, las variables de investigación y la descripción de indicadores.

En el Capítulo III, refiere toda la metodología propuesta y desarrollada, comprende el tipo de investigación, el diseño, las técnicas e instrumentos de recolección de datos, entre otros necesarios.

En el Capítulo IV, en este apartado se describe todo el proceso de la investigación realizada, se describe y evalúa los resultados obtenidos, se valida las hipótesis y se discute los resultados.

Finalmente se muestra conclusiones, recomendaciones, la bibliografía empleada y los anexos necesarios.

No dudo que la presente tesis será un aporte significativo a la academia de investigadores y a la comunidad de la Universidad Nacional Daniel Alcides Carrión.

EL AUTOR.

## ÍNDICE

DEDICATORIA

AGRADECIMIENTO

RESUMEN

ABSTRACT

INTRODUCCIÓN

ÍNDICE

### CAPITULO I

#### PROBLEMA DE INVESTIGACIÓN

1.1. Identificación y determinación del problema.....	1
1.2. Delimitación de la Investigación.....	3
1.3. Formulación del problema.....	4
1.3.1. Problema general.....	4
1.3.2. Problemas específicos.....	4
1.4. Formulación de objetivos.....	4
1.4.1. Objetivo general.....	4
1.4.2. Objetivos específicos.....	5
1.5. Justificación de la investigación.....	5
1.6. Limitaciones de la Investigación.....	6

## **CAPITULO II**

### **MARCO TEÓRICO**

2.1. Antecedentes de estudio .....	7
2.2. Bases teóricas - científicas .....	11
2.3. Definición de términos básicos .....	49
2.4. Formulación de hipótesis .....	52
2.4.1. Hipótesis General.....	52
2.4.2. Hipótesis Específicas.....	52
2.5. Identificación de Variables.....	52
2.6. Definición Operacional de las variables e indicadores .....	52

## **CAPITULO III**

### **METODÓLOGA Y TÉCNICAS DE INVESTIGACIÓN**

3.1. Tipo de investigación .....	54
3.2. Nivel de investigación .....	54
3.3. Métodos de la Investigación.....	54
3.4. Diseño de investigación.....	55
3.5. Población y Muestra.....	55
3.6. Técnicas e instrumentos de recolección de datos. ....	55
3.7. Selección, validación y confiabilidad de los instrumentos de investigación .....	56

3.8. Técnicas de procesamiento y análisis de datos.....	59
3.9. Tratamiento Estadístico .....	60
3.10. Orientación ética filosófica y epistémica .....	60

## **CAPITULO IV**

### **RESULTADOS Y DISCUSIÓN**

4.1. Descripción del trabajo de campo .....	61
4.2. Presentación, análisis e interpretación de resultados .....	71
4.3. Prueba de hipótesis.....	90
4.4. Discusión de resultados .....	95

CONCLUSIONES

RECOMENDACIONES

REFERENCIAS BIBLIOGRAFICAS

ANEXOS

## INDICE DE FIGURAS

Figura 1 .....	2
Nivel de confianza en seguridad biométrica.....	2
Figura 2 .....	8
Diseño e implementación del sistema. ....	8
Figura 3 .....	13
El desarrollo del reconocimiento facial, tecnologías relacionadas y características de las diferentes etapas en la evolución del reconocimiento facial.....	13
Figura 4 .....	14
Etapas del reconocimiento facial.....	14
Figura 5 .....	15
Modelo del proceso de reconocimiento facial desarrollado. ....	15
Figura 6 .....	17
Matrices de imágenes en escala de grises y en RGB.....	17
Figura 7 .....	19
Características de Haar. ....	19
Figura 8 .....	21
Ejemplo de procedimiento de AdaBoost.....	21
Figura 9 .....	22
Esquema de Cascada de decisión. ....	22
Figura 10 .....	24
Matriz de una imagen. ....	24
Figura 11 .....	25
Reducción PCA de los datos multidimensionales. ....	25
Figura 12 .....	26
Comparación entre PCA y LDA. ....	26
Figura 13 .....	27
Geometría de un rostro, mapa de profundidad y modelo 3D de técnicas basadas en Modelos.....	27
Figura 14 .....	29
Dataset de imágenes. ....	29
Figura 15 .....	30
Formateo de la imagen en píxeles y escala de colores. ....	30
Figura 16 .....	30

Transformación de matriz a vector para aplicar Eigenface. ....	30
Figura 17 .....	31
Matriz y vectores de imágenes de tamaño $Z \times K^2$ . ....	31
Figura 18 .....	33
Imágenes con distinto nivel de iluminación. ....	33
Figura 19 .....	35
Proyección y segmentación de cada punto del espacio bidimensional.....	35
Figura 20 .....	36
Histograma con diferencia lumínica en LBP.....	36
Figura 21 .....	37
Imagen en escala de grises con su código de color para LBP. ....	37
Figura 22 .....	38
Proceso de codificación para imágenes en LBP. ....	38
Figura 23 .....	39
Valores de los pixeles resultante del proceso de codificación para LPB. ....	39
Figura 24 .....	40
Proceso de Mapeo en LBP. ....	40
Figura 25 .....	40
Etiquetas numéricas de Mapeo en LBP.....	40
Figura 26 .....	41
Matrices de etiquetas de imagen, codificación y mapeado en LBP para elejemplo dado. ....	41
Figura 27 .....	41
Obtención del histograma en LBP.....	42
Figura 28 .....	47
Página oficial de Python.....	47
Figura 29 .....	48
Página oficial de Anaconda.....	48
Figura 30 .....	49
Página oficial Visual Studio Code.....	49
Figura 31 .....	57
Cámara fotográfica y de video digital Nikon. ....	57
Figura 32 .....	58
Estructura de carpetas de la base de datos de imágenes. ....	58

Figura 33 .....	65
Página web The Extended yale fase database B.....	65
Figura 34 .....	66
Imágenes de rostro de la persona yaleB17 .....	66
Figura 35 .....	68
Imágenes de rostro de la base de datos “Master” de un alumno de la Facultad de Ingeniería de la UNDAC.....	68
Figura 36 .....	73
Exactitud y precisión del modelo Eigenface .....	73
Figura 37 .....	74
Tiempo de entrenamiento y testeo del modelo Eigenface en función del número de imágenes de entrenamiento .....	74
Figura 38 .....	76
Exactitud y precisión del modelo Eigenface. ....	76
Figura 39 .....	78
Exactitud y precisión del modelo Eigenface. ....	78
Figura 40. ....	80
Indicadores de exactitud y precisión del modelo Fisherface.....	80
Figura 41 .....	82
Indicadores de exactitud y precisión del modelo Fisherface.....	82
Figura 42 .....	84
Indicadores de exactitud y precisión del modelo Fisherface.....	84
Figura 43 .....	86
Indicadores de exactitud y precisión del modelo LPB en función del número de imagen de entrenamiento para 10 personas.....	86
Figura 44 .....	88
Indicadores de exactitud y precisión del modelo LBP. ....	88
Figura 45 .....	90
Indicadores de exactitud y precisión del modelo LBP. ....	90
Figura 46 .....	92
Comparativa de las métricas de exactitud de los algoritmos Eigenface,Fisherface y LBP.....	92
Figura 47 .....	94
Comparativa de las métricas de precisión de los algoritmos Eigenface,Fisherface y LBP.....	94

## INDICE DE TABLA

Tabla 1 .....	53
Definición operacional de las variables e indicadores. ....	53
Tabla 2.....	62
Software empleado en el trabajo de investigación .....	62
Tabla 3.....	63
Librerías adicionales empleado en la investigación. ....	63
Tabla 4.....	64
Hardware empleado en el trabajo de investigación. ....	64
Tabla 5.....	72
Resultados algoritmo Eigenface. ....	72
Tabla 6.....	75
Resultados algoritmo Eigenface. ....	75
Tabla 7.....	77
Resultados algoritmo Eigenface. ....	77
Tabla 8.....	79
Resultados algoritmo Fisherface. ....	79
Tabla 9.....	81
Resultados algoritmo Fisherface. ....	81
Tabla 10.....	83
Resultados algoritmo Fisherface para test de estudiantes. ....	83
Tabla 11 .....	85
Resultados algoritmo LBP para test de estudiantes.....	85
Tabla 12.....	87
Resultados algoritmo LBP para test de estudiantes.....	87
Tabla 13.....	89
Resultados algoritmo LBP para test de estudiantes.....	89

## **CAPITULO I**

### **PROBLEMA DE INVESTIGACIÓN**

#### **1.1. Identificación y determinación del problema**

La suplantación de identidad y los ciberataques en los diferentes sectores productivos, ha ocasionado pérdidas cuantiosas en el mundo, un estudio realizado por Javelin (2021, citado en Mitek Systems, 2021) en los Estados Unidos, indica que "el 38% de las víctimas de suplantación de identidad terminaron cerrando las cuentas bancarias en donde tuvo lugar el fraude".

El mismo estudio indica que la confianza en la seguridad mediante contraseñas ha disminuido, notándose un mayor interés por la autenticación biométrica (huella dactilar, reconocimiento facial y escaneo de iris) como alternativa, véase figura 1.

## Figura 1

### Nivel de confianza en seguridad biométrica.

Nota. En esta se aprecia que un 63% de la población ve de forma positiva



al reconocimiento facial como medida de seguridad alterna o complementaria al uso de contraseñas citado en Mitek Systems, 2021.

Fuente: Javelin (2021).

La pandemia de COVID-19 sentó las bases para que los delincuentes se aprovecharan y explotaran las vulnerabilidades de las organizaciones, obligadas a un trabajo remoto y virtual, en nuestro país, hasta abril del 2021, la División de Investigación de Delitos de Alta Tecnología (Divindat) de la Policía realizó más de 1118 investigaciones, estando entre los casos más representativos la suplantación de identidad (El peruano, 2021).

Siendo el sector educación, en esta virtualidad, quien en forma íntegra se ha volcado a una enseñanza y gestión educativa virtual, llega a ser un sector muy vulnerable, encontrándose limitados o sin los medios tecnológicos, sin el suficiente conocimiento y capacitación de las tecnologías para la educación virtual y con escasos recursos económicos, sobre todo en las instituciones del estado, la educación estatal, que permitan hacer frente al tema del robo o suplantación de

identidad, la escasa o nula relación entre profesor y alumno, propia de este escenario, obliga a confirmar la identidad del alumno, el método más común es el de usuario y contraseña, practicado en casi todas las instituciones educativas del país, sin embargo muchos alumnos comparten sus credenciales con otros, para que puedan rendir sus pruebas, acceder a lugares web privados como bibliotecas, simuladores, software especializado, produciéndose vulnerabilidades y suplantación de identidad, en otros casos los estudiantes han sufrido el robo de sus credenciales siendo víctimas de alteración de su estatus académico, entre otros muchos malos manejos.

En la actualidad existen tecnologías que pueden complementar en seguridad al uso de usuario y contraseña, mediante la identificación biométrica como la huella dactilar o la identificación facial, mecanismos que son empleados en: Amazon ML, Azure ML Studio, Google Prediction API, Google ML Engine, entre otros, pero que son de difícil implementación y de elevados costos (La Madrid, D y Barriga, M., 2019).

Ante esta situación, se requiere un medio alternativo o complementario que permita evitar la suplantación de una persona, identificándolo con seguridad, por eso el presente proyecto evalúa algoritmos de reconocimiento facial en aprendizaje automático para determinar el que mejor precisión y/o exactitud muestre en la identificación de personas que están permitidos o que pueden acceder a los ambientes de la institución como laboratorios, aulas de clase, ambientes restringidos y otros similares.

## **1.2. Delimitación de la Investigación**

### **1.2.1. Delimitación Espacial**

El presente trabajo de investigación, se realizó en el distrito de Yanacancha,

provincia y departamento de Pasco, en una institución educativa de Pasco, específicamente la Universidad Nacional Daniel Alcides Carrión.

### **1.2.2. Delimitación Temporal**

La investigación fue planificada para ser ejecutada en un periodo de 04 meses, comprendido entre noviembre 2021 a febrero 2022, sin embargo, se tuvo que ampliar un mes por la complejidad para el procesamiento de datos.

### **1.2.3. Delimitación Conceptual**

La investigación se circunscribe dentro de la evaluación de algoritmos de reconocimiento facial en aprendizaje automático que permita determinar aquel que mejor resultado brinde al momento de identificar a personas en una institución pública.

## **1.3. Formulación del problema**

### **1.3.1. Problema general**

¿Qué se consigue evaluando algoritmos de reconocimiento facial mediante aprendizaje automático en la identificación de personas de una institución educativa de Pasco- 2021?

### **1.3.2. Problemas específicos**

¿Qué se consigue evaluando algoritmos de reconocimiento facial en aprendizaje automático en relación a la exactitud en la identificación de personas?

¿Qué se consigue evaluando algoritmos de reconocimiento facial en aprendizaje automático en relación a la precisión en la identificación de personas?

## **1.4. Formulación de objetivos**

### **1.4.1. Objetivo general**

Determinar el algoritmo de reconocimiento facial que mejor resulte en la identificación de personas de una institución educativa de Pasco - 2021.

#### **1.4.2. Objetivos específicos**

Determinar el algoritmo de reconocimiento facial en aprendizaje automático con mayor exactitud en la identificación de personas

Determinar el algoritmo de reconocimiento facial en aprendizaje automático con mayor precisión en la identificación de personas.

#### **1.5. Justificación de la investigación**

En estos últimos años la tecnología de procesamiento de información ha evolucionado notablemente, han surgido procesadores de cuatro, ocho, dieciséis núcleos, que realizan procesamientos en paralelo y, se encuentran al alcance del usuario común, por sus costos reducidos, esto ha permitido a disciplinas como el aprendizaje automático (Machine Learning), probar algoritmos que requieren muchos recursos de procesamiento y tiempo de espera en cálculos, pasando por ejemplo de días a sólo horas, inmerso en esto se encuentra el reconocimiento facial, técnica que ha sido y es un tema de sumo interés para los investigadores.

A la par el Estado peruano viene invirtiendo en las instituciones educativas públicas, especialmente en la superior universitaria, ha transfiriendo cuantiosas sumas de dinero para infraestructura y equipamiento en el marco del Licenciamiento de las universidades. La Universidad Nacional Daniel Alcides Carrión ha sido una de las beneficiadas en este aspecto, habiendo implementado sus laboratorios, aulas de clase, ambientes administrativos y otros de uso pedagógico con equipos, muchos de ellos costosos, siendo responsabilidad de la institución velar por el cuidado, conservación y mantenimiento, sin embargo la posibilidad de daños, sustracción, pérdida o robo de estos, es una constante, ante ello se propone evaluar algoritmos de reconocimiento facial de aprendizaje automático y determinar el que mejor resultado arroje en la identificación de

personas para que puedan tener o ser autorizadas de ingresar a los ambientes y discriminar de aquellos no autorizados, algoritmos que pueden ser implementados en un aplicativo móvil o software para computadora. Esto contribuiría a fortalecer la seguridad en la universidad y disminuir el gasto que pudiese ocasionar como consecuencia de acceso de personas no autorizadas a los ambientes de la institución.

#### **1.6. Limitaciones de la Investigación**

Dadas la situación que se viven por las características actuales debido al confinamiento por el Covid – 19 se establecen las siguientes limitaciones:

- **Limitación temporal:** los datos analizados para el estudio corresponden al año 2021, periodo de noviembre a diciembre.
- **Limitación geográfica:** se circunscribió en la Región Pasco.
- **Limitación económica:** El trabajo es autofinanciado en su totalidad.

## **CAPITULO II**

### **MARCO TEÓRICO**

#### **2.1. Antecedentes de estudio**

Dentro del ámbito del reconocimiento de imágenes y la identificación de objetos diversos se han propuesto estudios que abarcan desde algorítmicos en minería de datos, pasando por la inteligencia artificial y hoy en día el aprendizaje automático y profundo, donde se vienen dando aportes importantes en lo que a estudios predictivos se refiere. A continuación, se describe algunos, que sirven como antecedentes para esta investigación, realizados a nivel internacional, nacional y regional.

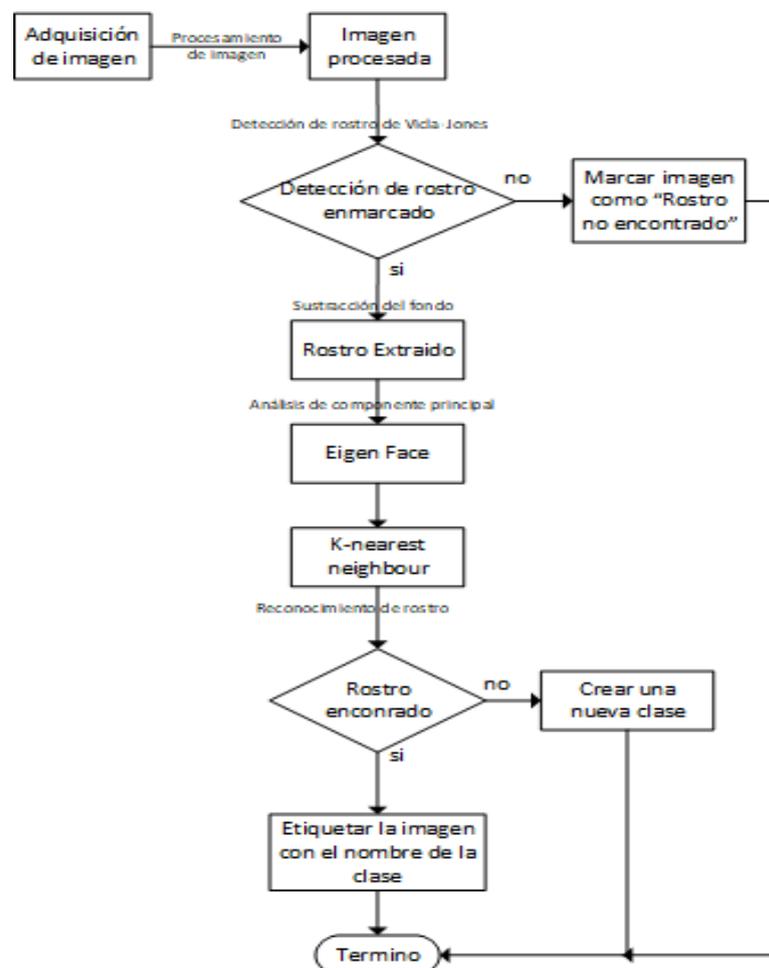
##### **2.1.1. Internacional**

En el ámbito internacional existen investigaciones como la de Myna A.N., Deepthi, K. y Prathibba, G. (2021) denominada “Classification of photographs based on face recognition”, esta investigación prueba algoritmos para el reconocimiento de patrones de rostro, su objetivo se centra en ayudar al problema de clasificación de fotografías según el rostro presente en la imagen, pudiendo el usuario encontrar todas las imágenes de cualquier persona en particular. Concluye

que, el reconocimiento facial mediante las técnicas existentes debe aún mejorarse para que sean lo suficientemente robustas en el manejo de características propias de estas técnicas como es el manejo de errores de escalado, rotación y traslación. Siendo de interés para nuestra investigación el modelo que emplea para el diseño y la implementación del reconocedor facial desarrollado (ver figura 2), donde la imagen tomada a la persona se envía como entrada al módulo de detección de rostro siendo procesada mediante el algoritmo de Viola - Jones, procesada y si se detecta un rostro es recién transferida para el reconocimiento facial mediante los algoritmos Eigenface y K-nearest (K-vecinos).

**Figura 2**

**Diseño e implementación del sistema.**



Nota. La figura muestra el diseño y la implementación del reconocedor facial

mediante el algoritmo de Viola – Jones, Eigenface y K-nearest, Fuente: MynaA., Deepthi, K. y Prathibba, G. (2021).

Dentro del proceso de reconocimiento facial, una parte fundamental es la detección de rostros, presente también en el diseño de la figura 2, en este caso emplea el algoritmo basado en la técnica Viola-Jones, sin embargo, existe además otras técnicas muy empleadas actualmente, Rahmad et al. (2020) en su artículo “Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection” realiza comparación entre el algoritmo Viola-Jones Haar Cascade Classifier (V-J) y el Histogram of Oriented Gradients (HOG) para detectar rostros humanos en las imágenes que se evalúan. El resultado obtenido indica que ambos sistemas detectaron con éxito el rostro bajo 6 diferentes condiciones permitiendo comparar los resultados. La investigación concluye que el algoritmo V-J alcanza una precisión del 75,33% frente a un 80,22%, sin embargo, menciona que el algoritmo V-J presenta una menor complejidad en su codificación, aspecto a tener en cuenta para el desarrollo de esta investigación.

Así mismo tenemos a la investigación desarrollada por Espinoza, D. y Jorquera, P. (2015) titulada “Reconocimiento facial”. Tesis de pregrado donde se enfoca en analizar e implementar un sistema de reconocimiento facial mediante herramientas informáticas como el OpenCV/EmguCV, librerías del entorno .Net desarrollada por Intel, para ello desarrollaron una base de datos local, que permite almacenar las imágenes fotográficas de rostros, ya sacadas al usuario, para que en el paso posterior puedan reconocer este rostro cuando la misma persona sea analizada de nuevo, poniendo a prueba 3 algoritmos: Eigenface, Fisherface y LBP (Local Binary Pattern), todos estos basados en la técnica de apariencia de carácter lineal para determinar cuál de ellos resulta más eficiente al momento de identificar al

usuario, el trabajo concluye que los algoritmos LBP y Fisherface son los más eficientes, con grados de acierto superior al 80% , por lo que es pertinente considerarlos en la investigación.

De otro lado, se tiene el trabajo realizado por Castaño, D. y Alonso, J. (2019), titulado “Sistema de reconocimiento facial para controlar el acceso a viviendas” tesis de pregrado de la universidad católica de Colombia. El fin de esta investigación es desarrollar un sistema de reconocimiento facial el cual permitirá el control de acceso a una casa, para ello crea un prototipo de un dispositivo controlador electrónico apoyado en la programación en Python de un reconocedor facial con el algoritmo Eigenface PCA, debido a su baja complejidad computacional y su poca utilización de recursos informáticos, obteniendo una tasa de acierto del 97%. Entre sus principales conclusiones indica que el porcentaje de reconocimiento vistos en las pruebas de eficacia dependen mucho de la cantidad de fotos de entrenamiento que se realicen, entre más se hagan, será mejor el reconocimiento facial. Así mismo, las fotos que se evalúen no deben poseer grados de inclinación porque eso reduce los niveles de eficiencia del reconocimiento facial, estos puntos deben ser considerados al momento de la recopilación de los datos de la muestra.

### **2.1.2. Nacional**

A nivel nacional llama la atención la tesis desarrollada por La Madrid, D. (2019) titulada “Modelo Tecnológico de Reconocimiento Facial para la Identificación de Pacientes en el Sector Salud” de la Universidad Peruana de Ciencias Aplicadas. La investigación habla sobre el desarrollo de un modelo tecnológico cuyo objetivo es la identificación de pacientes mediante un servicio cognitivo de reconocimiento facial en la nube (cloud) para satisfacer la

necesidad que tienen el sector salud en el tema de suplantación de la identidad, proponiendo adicionalmente una alerta de emergencia mediante mensaje de texto a los parientes del paciente. Atendiendo a sus conclusiones, el autor indica que gracias al análisis desarrollado sobre las herramientas tecnológicas de reconocimiento facial en el mercado, identificó a Azure de Microsoft como el servicio de reconocimiento facial más adecuado por factores de eficiencia, portabilidad, usabilidad entre otros, a pesar de los costos elevados de acceso a esta plataforma en la nube, siendo este aspecto de utilidad para implementar el análisis comparativo de los algoritmos para el reconocimiento de imágenes en nuestra investigación.

### **2.1.3. Local**

Revisado el repositorio de la Universidad Nacional Daniel Alcides Carrión no se encuentra investigación alguna realizada sobre este tema o relacionados directos como visión por computadora, visión artificial, detección y reconocimiento facial o detección de personas empleando técnicas de aprendizaje automático.

## **2.2. Bases teóricas - científicas**

### **2.2.1. Reconocimiento facial**

El ser humano en su camino evolutivo lleva consigo características que definen su identidad, como su nombre, apodo, etc, pero es su rostro una de las características que lo hace único, Espinoza, D. y Jorquera, P. (2015) definen reconocimiento facial como “una herramienta que nos permite identificar a una persona automáticamente por medio de una imagen digital. Es una forma de seguridad biométrica “.

Para Li, Lixian et al. (2020). El reconocimiento facial es “una tecnología biométrica, que se basa en la identificación de características faciales de una

persona. Las personas recogen las imágenes faciales, y el equipo de reconocimiento procesa automáticamente las imágenes”.

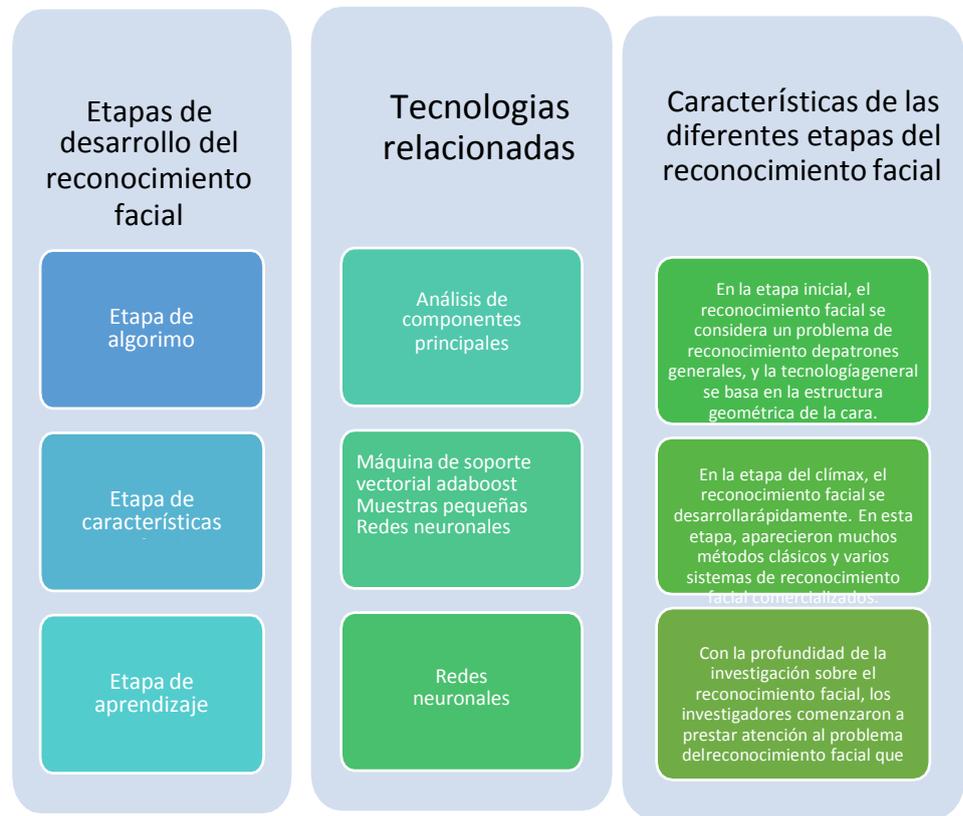
En base a estos conceptos se puede decir que el reconocimiento facial es una técnica para verificar la identidad de un individuo mediante su rostro, ya sea por fotos, videos o en tiempo real.

Conforme ha pasado el tiempo el campo de acción en este aspecto ha crecido enormemente, encontrando espacio principalmente en seguridad, tanto en la alerta temprana de situaciones sospechosas como en el rastreo de personas sospechosas.

A continuación, se muestra (figura 3) el desarrollo del reconocimiento facial, tecnologías relacionadas y características de las diferentes etapas en la evolución del reconocimiento facial (Li, Lixian et al.,2020).

**Figura 3**

**El desarrollo del reconocimiento facial, tecnologías relacionadas y características de las diferentes etapas en la evolución del reconocimiento facial**



Nota. La figura muestra las tecnologías relacionadas, las características y las diferentes etapas del desarrollo del reconocimiento facial, Fuente: Li, Lixian et al. (2020).

Obsérvese que, la evolución en las etapas del desarrollo del reconocimiento facial, pasa del análisis de la geometría básica de un rostro, limitado por la insipiente tecnología de la época, hasta nuestros tiempos, donde gracias al gran salto tecnológico en el hardware de procesamiento, se puede analizar los rostros en tiempo real. Sin embargo, en ese trance las etapas que conforman un reconocimiento facial, siguen siendo esencialmente los mismos (García, M.

,2009). Vea figura 4.

### Figura 4

#### Etapas del reconocimiento facial.



Nota. La figura muestra las etapas del reconocimiento facial, Fuente:

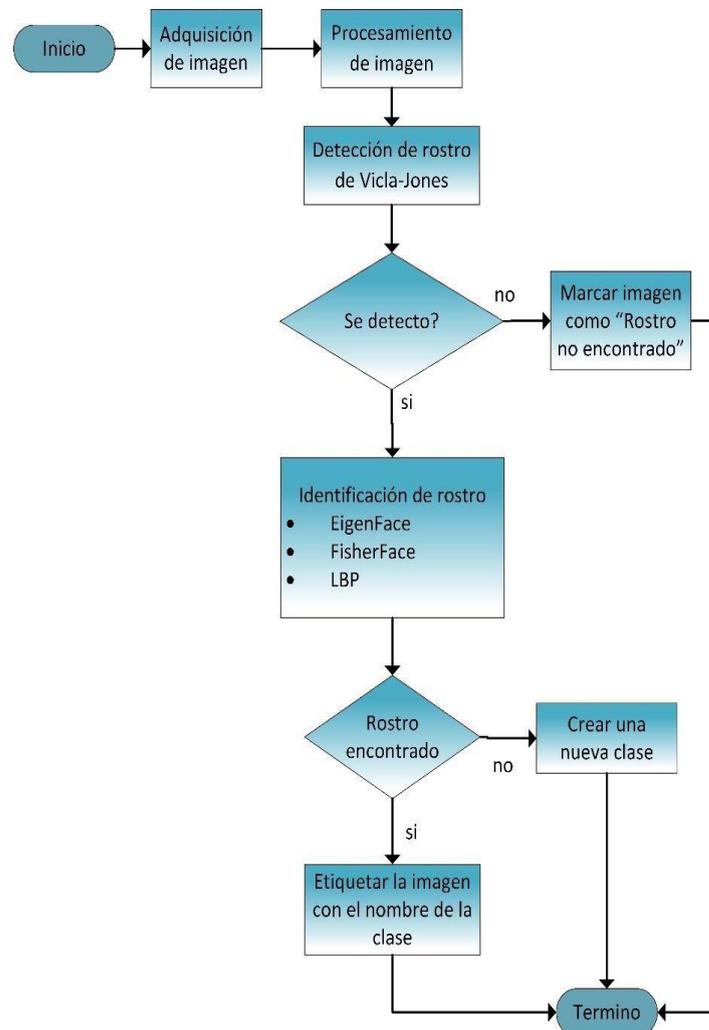
García,M. (2009).

#### 2.2.1.1. Modelo de reconocimiento facial.

Teniendo en cuenta las etapas del reconocimiento facial: detección de rostros y verificación de rostros de la figura 4, así como el modelo de diseño e implementación de un sistema de reconocimiento facial propuesto (Myna et al., 2021), figura 3 del apartado de antecedentes, se elaboró el modelo del proceso de reconocimiento facial para esta investigación, obsérvese la figura 5.

**Figura 5**

**Modelo del proceso de reconocimiento facial desarrollado.**



Nota. En la figura se puede ver todo el proceso de reconocimiento facial desde la adquisición de la imagen hasta su respectiva identificación aplicando cada algoritmo, Fuente: Elaboración propia.

A continuación, se describe los componentes principales del modelo.

**a. Adquisición de imágenes.**

La adquisición es el proceso a través del cual se obtiene una imagen. Este proceso implica seleccionar el medio con el cual se realizar las características a considerar para la captura de las imágenes y el acto

mismo de la captura de la imagen. Es decir, se establece el instrumento que permite capturar la imagen: cámaras fotográficas, filmadoras, radiografías, etc.

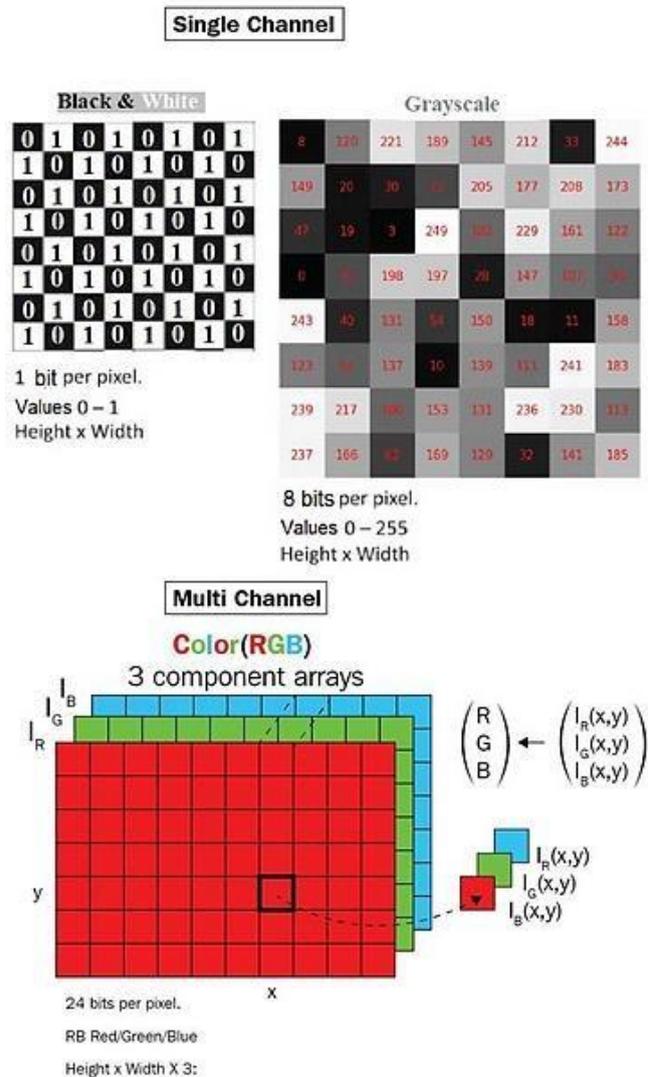
**b. Procesamiento de imágenes**

El procesamiento de imágenes consiste en el conjunto de técnicas para mejorar la apariencia visual de una imagen a una forma más apropiada para el observador o computador, operando sobre los píxeles que lo conforman (Szeliski, 2022)

En este apartado se ingresan todas las imágenes a las que se requiere aplicar el proceso de detección de rostro, cada imagen es tratada como una matriz de dos dimensiones  $(x,y)$  para cada una de las componentes RGB si la imagen es a color, o una sola matriz  $(x,y)$  para imágenes en escala de grises, de aquí se desprende que las imágenes a escala de grises ocupan menos espacio que las imágenes a color, siendo más simple de analizar ya que consumen menos tiempo y emplean menos recursos de memoria y almacenamiento del computador (Juárez, C. y Serrano, J. , s.f.). Véase la figura siguiente:

**Figura 6**

**Matrices de imágenes en escala de grises y en RGB.**



Nota. La imagen muestra la matriz de las imágenes a escala y cuando es a color se tiene una matriz tridimensional haciendo referencia a cada canal de color, Fuente: Sandipan (2018).

Generalmente para la extracción de algunas de las características de la imagen capturada se debe convertir la imagen de color a escala de grises, como por ejemplo para la extracción de características que requieren el uso del histograma. Cada uno de los componentes de la matriz es el valor de intensidad de un punto de la imagen o pixel

(Juárez y Serrano, s.f.).

Esta conversión es idónea para un trabajo menos complejo al momento de detectar o identificar un rostro.

### **c. Detección de rostro**

La detección de rostro o detección facial “es una tecnología de visión computarizada que determina el lugar y el tamaño del rostro humano en imágenes o videos. Es uno de los subtipos de detección de objeto” (Viola & Jones, 2001, citado en Espinoza y Jorquera, 2015).

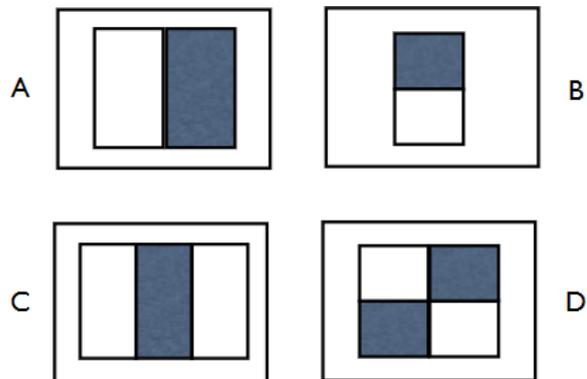
**El Detector de rostros de Viola y Jones.** Uno de los métodos fundamentales en la detección de rostros que ha tenido un gran impacto es el de Viola y Jones (2004) quienes presentaron un marco de detección de rostros que puede detectarlas a 15 fotogramas por segundo. Los autores distinguieron su trabajo por tres contribuciones clave en el desarrollo de este detector de rostros: 1) Una representación de la imagen basada en la imagen integral 2) un algoritmo de aprendizaje, basado en AdaBoost y la combinación de 3) Clasificadores Complejos en Cascada (Espinoza y Jorquera, 2015).

Seguidamente se describen cada uno:

- **Imagen integral**, “es una representación de la imagen original, permitiendo extraer las características de la imagen mediante filtros de Haar. Esta imagen es obtenida a partir de un algoritmo que genera la suma de los valores de un rectángulo dentro de una imagen” (Espinoza y Jorquera, 2015). Las características de Haar se observan en la figura siguiente.

## Figura 7

### Características de Haar.



Nota. Esta imagen es obtenida a partir de un algoritmo que genera la suma de los valores de un rectángulo dentro de una imagen”,

Fuente: Espinoza y Jorquera (2015).

Estos son los cuatro tipos de características de Haar que Viola y Jones utilizaron en su investigación. El valor de una característica de dos rectángulos es la diferencia entre la suma de los píxeles dentro de las dos regiones rectangulares. Estas regiones son horizontal o verticalmente adyacentes y del mismo tamaño. La característica de tres rectángulos calcula la diferencia entre la suma de las dos regiones exteriores y el rectángulo central. La característica de cuatro rectángulos halla la diferencia entre los pares de rectángulos diagonales.

Las características rectangulares son algo primitivas y sensibles a la presencia de bordes, barras (ejes) y otra estructura de imagen simple. Viola y Jones generaron un conjunto muy grande y variado de características rectangulares (Viola & Jones, 2004).

- **Ada Boost**, “es un algoritmo de aprendizaje automático que consiste en la extracción de características por medio de

clasificadores. Basado en la mejor forma de ejecutar los clasificadores para detectar de forma favorable el rostro humano” (Espinoza y Jorquera, 2015).

AdaBoost es un proceso de selección codicioso. Se utiliza un sistema de ponderación para asociar pesos a los clasificadores con un peso grande asociado a cada función de clasificación buena y un peso pequeño con las funciones pobres.

AdaBoost es una técnica eficaz para seleccionar un pequeño conjunto de funciones buenas. El objetivo es separar los ejemplos positivos y negativos de forma que un número mínimo de ejemplos se clasifiquen erróneamente (Espinoza y Jorquera, 2015).

Para conseguirlo, se diseña un algoritmo de aprendizaje débil para seleccionar una única función rectangular que consiga mejorar esta separación entre los ejemplos positivos y los negativos. Este clasificador débil determina la función de umbral óptima para minimizar los errores de clasificación. Un clasificador débil puede presentarse como:

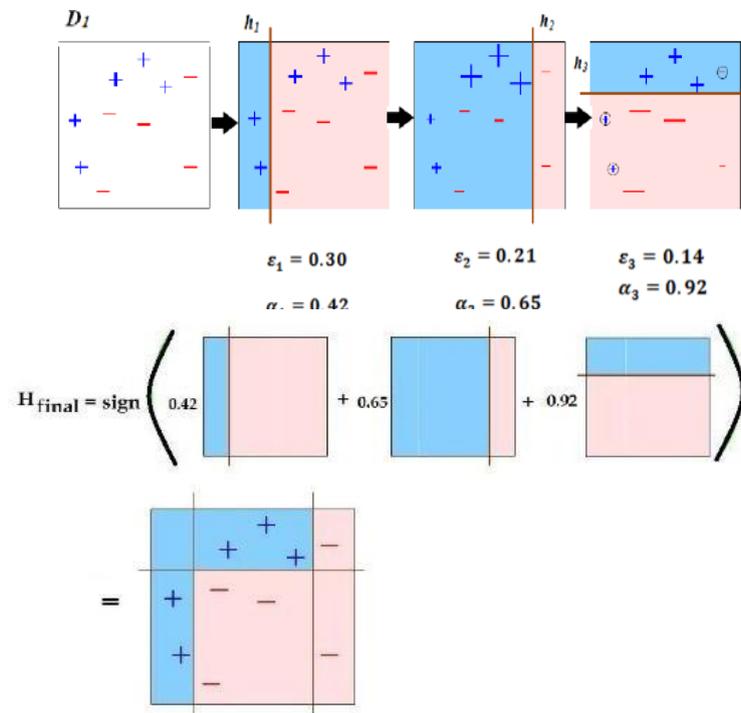
$$h(x, f, p, \theta) = \begin{cases} 1 & \text{si } pf(x) < p\theta \\ 0 & \text{en otro caso} \end{cases}$$

donde  $h(x, f, p, \theta)$  es un clasificador débil compuesto por la característica  $f$ , un umbral  $\theta$  y una polaridad  $p$  y una sub ventana de  $24 \times 24$  píxeles,  $x$ , de una imagen (Viola & Jones, 2004).

La siguiente figura esquematiza de forma práctica como opera AdaBoost:

**Figura 8**

**Ejemplo de procedimiento de AdaBoost.**



Nota. La figura muestra la clasificación en dos grupos que hace el algoritmo AdaBoost de una imagen, Fuente: Morales (2015).

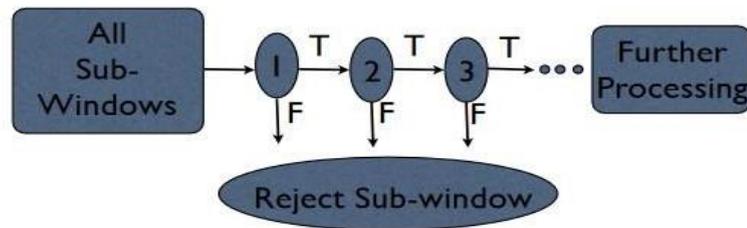
Se tiene el conjunto de datos en donde un tipo de datos se representa por símbolos “+” azules y el otro por símbolos “-” en rojo. El algoritmo AdaBoost debe clasificarlos por tipo. En su primer intento, el clasificador divide en 2 grupos los datos, pero realiza tres errores: deja dos datos negativos dentro del conjunto positivo y un dato positivo no lo toma en cuenta. En un segundo intento asigna mayores pesos a los valores que no fueron considerados en el grupo azul, pero se sigue dando errores, por lo que sigue haciendo más intentos hasta conseguir el resultado final que se observa.

- **Clasificadores Complejos en Cascada**, un clasificador no es

suficiente para detectar rostros de manera eficiente, por eso se implementa una cascada de clasificadores entrenados y ajustados, véase la figura 9.

**Figura 9**

**Esquema de Cascada de decisión.**



Nota. En la figura se muestra el esquema de cascada de decisión, donde T=true y F = false,

Fuente: Espinoza y Jorquera (2015).

El objetivo de construir una cascada de filtros es una mayor eficiencia y, por tanto, una reducción del tiempo de cálculo. Viola y Jones sugirieron que se puede construir un número menor de clasificadores eficientes y potenciados de clasificadores eficientes y potenciados. Estos clasificadores potenciados serán más eficientes a la hora de rechazar muchas de las subventanas negativas y detectar casi todas las instancias positivas. La mayoría de las subventanas son rechazadas por los clasificadores más sencillos, y la reducción de la tasa de falsos positivos se consigue utilizando clasificadores más complejos (Espinoza y Jorquera, 2015).

Se utiliza AdaBoost para crear etapas de entrenamiento de los clasificadores. La primera etapa se compone de un clasificador

fuerte de dos características. Para construir un filtro facial eficaz, el umbral del clasificador fuerte se ajusta para minimizar los falsos negativos (Espinoza y Jorquera,2015).

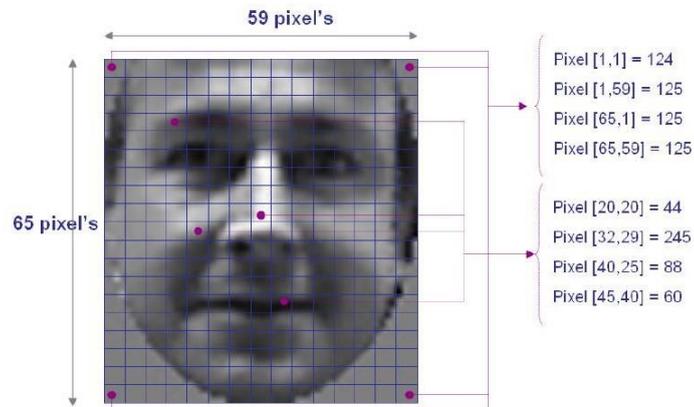
**d. Identificación de rostro**

Consiste en procesar la imagen capturada y compararla con las imágenes almacenadas en una base de datos. Si se encuentra una coincidencia, se identifica al individuo. Si no se encuentra ninguna coincidencia, el individuo es reportado como no identificado (Çarıkçı y Özen, 2012).

Según Stan y Jain (citado en Cadena et al., 2018) el reconocimiento facial, al que ellos describen como análisis computarizado de imágenes de rostros, comprende una señal visual que es registrada digitalmente como una matriz de valores de píxeles. Estos valores en los píxeles representan color o solo intensidad (Cadena et al., 2018), en esta investigación se asumen imágenes en escala de grises, ya que solo toma como base a una sola matriz adiferencia de las de color que requieren 3 matrices (una para el Rojo, otra para el Verde y otra para Azul, el famoso **RGB**). El valor de cada píxel está en el rango de 0 a 255 (tonos de gris) y la dimensión del espacio de la imagen estaba dada por la matriz de  $m \times n$  píxeles, con  $m=59$  y  $n=65$  (Ramon, 2011). Vea figura 10.

## Figura 10

### Matriz de una imagen.



Nota. La imagen muestra el ejemplo de descriptor basado en píxeles,  
Fuente:Ramon (2011).

A su vez, para cada imagen hemos de generar un vector con dimensión igual al producto de las dimensiones de la matriz de la imagen, para el ejemplo de la figura sería un vector de 3835 posiciones (Ramon, 2011), este vector, así como la matriz correspondiente, servirán para trabajar los diferentes algoritmos de reconocimiento facial.

Estos algoritmos emplean técnicas: 1) basado en apariencias y 2) técnicas basadas en modelos (Espinoza y Jorquera, 2015).

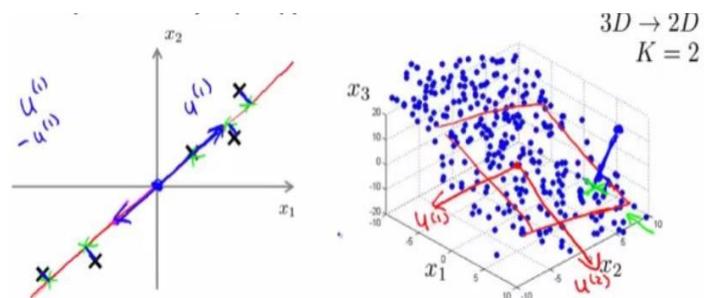
Según Gimeno (2010) “los sistemas basados en la apariencia se utilizan directamente sobre las imágenes sin hacer uso de modelos 3D”, para poder representar un objeto con este tipo de sistemas se requiere de una gran cantidad de vistas del rostro de la imagen (diferentes poses, ángulos, iluminación, etc). El método basado en la

apariencia puede clasificarse como lineal o no lineal. Ex-PCA, LDA, DCT se usa en el enfoque lineal, mientras que Kernel PCA se usa en el enfoque no lineal (Espinoza y Jorquera, 2015). A continuación, describiremos brevemente cada una de ellas:

- **Análisis de componentes principales – PCA**, es uno de los métodos estadísticos más utilizados y citados. Un procedimiento matemático realiza una reducción de dimensionalidad al extraer el componente principal de los datos multidimensionales (Dwidevi, 2018). En la figura siguiente se muestra una reducción de tres dimensiones a dos aplicando PCA.

**Figura 11**

**Reducción PCA de los datos multidimensionales.**

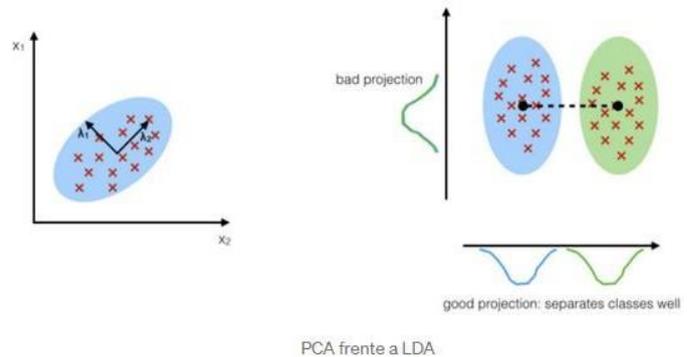


Nota. La figura muestra una reducción de tres dimensiones a dos aplicando PCA, Fuente: Dwidevi (2018).

- **Análisis Discriminante Lineal – LDA**, se usa ampliamente para encontrar la combinación lineal de características mientras se preserva la separabilidad de clases. A diferencia de PCA, la LDA trata de modelar la diferencia entre niveles. Para cada nivel se obtiene la LDA diferenciada en múltiples vectores de proyección (Dwidevi, 2018).

**Figura 12**

**Comparación entre PCA y LDA.**



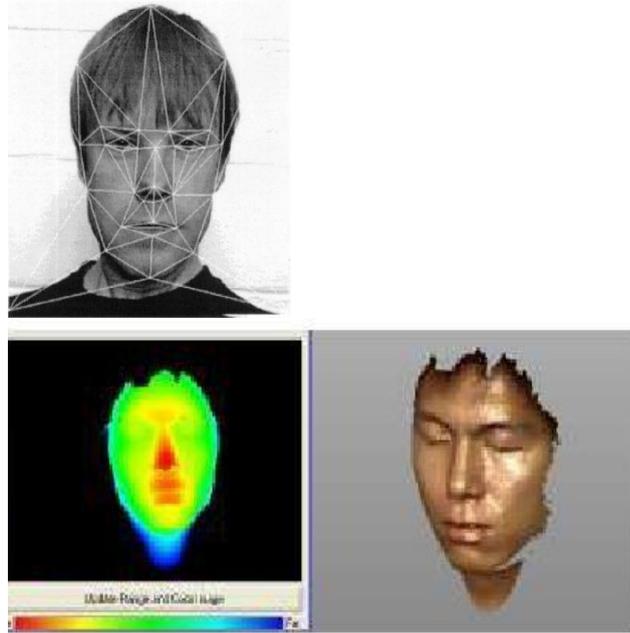
Nota. La figura muestra que la diferencia entre PCA Y LDA,  
Fuente: Dwidevi(2018).

- **Transformada de coseno discreta – DCT**, la transformada discreta del coseno es clave en la compresión JPEG. Es una variación de la transformada discreta de Fourier donde la imagen se descompone en sumas de cosenos y no de senos y cosenos (Dwidevi, 2018).
- **Kernel PCA**, Scholkopf introdujo el uso de funciones Kernel para realizar PCA no lineal. Su metodología básica es aplicar un mapeo no lineal a la entrada y luego resolver un PCA lineal en el subespacio de características resultante.

Por el otro lado, los sistemas basados en Modelos “intentan construir un modelo lo más descriptivo posible de la cara humana capaz de detectar con precisión las variaciones faciales” (Gimeno, 2010). Este tipo de sistemas buscan obtener características tridimensionales de las imágenes más complejas como el grosor de la nariz, las escalas térmicas para profundidad, etc, para ello requiere de imágenes de alta resolución, véase la siguiente figura:

**Figura 13**

**Geometría de un rostro, mapa de profundidad y modelo 3D de técnicas basadas en Modelos.**



Nota. La imagen muestra las características tridimensionales de las imágenes, grosor, escalas térmicas, etc, Fuente: Villegas (2005) y Gimeno (2011).

**e. Creación y/o etiquetado de imágenes**

Una vez identificado el rostro se procede a etiquetarlo en base a la información que se tiene en la base de datos de imágenes que se maneja, en el caso que no exista la imagen o no haya sido identificada la imagen se procede a generar una nueva carpeta y se etiqueta la imagen nueva en función del formato establecido para este fin.

**2.2.1.2. Algoritmos de reconocimiento facial.**

Cuando se tiene un escenario en el que la identidad de una persona es desconocida, siendo necesaria la identificación para que ingrese o transite, se requiere realizar una búsqueda en una base de datos de imágenes

para comparar, y establecer si esa persona puede acceder o estar en las instalaciones o zonas críticas, para esa identificación se emplean algoritmos de reconocimiento facial, entonces se puede indicar que un algoritmo de reconocimiento facial: consiste en una serie de pasos a seguir para identificar el rostro de una persona. Según la literatura revisada los algoritmos más utilizados y que mejores resultados arrojaron en el reconocimiento facial son Fisherface, Eigenface y LBP (Espinoza y Jorquera, 2015; Rahmad et al., 2020; Castaño y Alonso, 2019), según los autores, son los que consumen menos recursos de cómputo y requieren menos cantidad de cálculos en el proceso, estando inmerso todos estos algoritmos en la técnica basada en la apariencia lineal.

**a. Algoritmo Eigenface**

Este algoritmo fue presentado por vez primera por Sirovich y Kirby en su artículo de 1987 para que luego en 1991 sea formalizado por Turk y Pentland en su artículo Face recognition using Eigenface (Rosebrock, 2021). Se basa en la técnica PCA- Análisis de Componentes Principales (Stan & Jain, 2011, citado en Cadena et al., 2018).

El PCA es un método de proyección a un subespacio y es ampliamente utilizado en el reconocimiento de patrones. Este método busca dos objetivos principales 1) la sustitución de vectores correlacionados de grandes dimensiones por vectores no correlacionados de dimensiones más pequeñas y 2) calcular una base para el conjunto de datos. La estrategia del método Eigenface consiste en extraer los rasgos característicos de la cara y representar la cara

en cuestión como una combinación lineal de las llamadas "Eigenface" (rostros fantasmas), obtenidas del proceso de los rasgos faciales. (Çarıkçı y Özen, 2012).

El procedimiento general como opera Eigenface es el siguiente (Rosebrock, 2021):

1. Se ingresa un conjunto de datos de N imágenes de rostros, donde varias imágenes pueden pertenecer a una misma persona, se recomienda estandarizar a un número de imágenes por igual de cada persona que se incluya en el conjunto de datos. Vea la siguiente figura de muestra.

#### **Figura 14**

##### **Dataset de imágenes.**

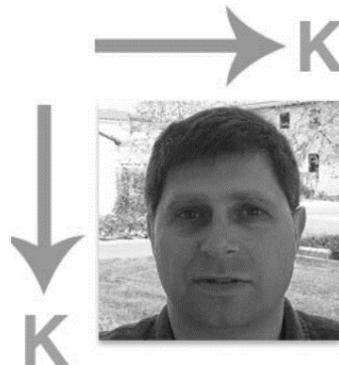


Nota. Se muestran diferentes imágenes de personas a diferentes escalas, Fuente: Rosebrock (2021).

2. Todas las imágenes se redimensionan a un formato de trabajo apropiado de K x K píxeles, pero los valores pueden ser diferentes si es conveniente para el caso y, se cambia el color de la imagen a un tono de grises.

**Figura 15**

**Formateo de la imagen en píxeles y escala de colores.**

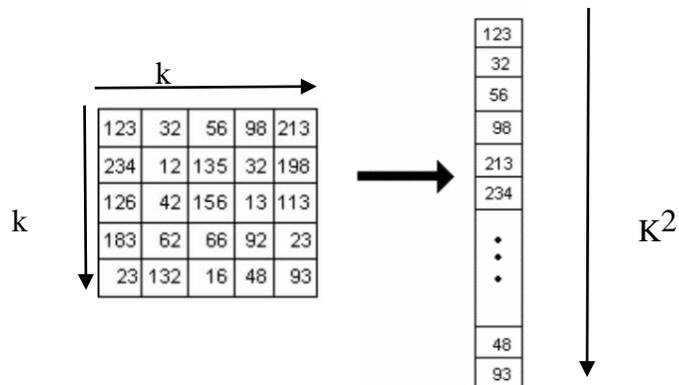


Nota. La figura muestra una imagen a escala de grises de igual altura y ancho, Fuente: Rosebrock (2021).

3. Se construye un equivalente matricial con los valores numéricos de cada píxel, que oscilan entre 0 y 255, que equivale a los tonos de gris. Para aplicar el algoritmo Eigenface, necesitamos formar un solo vector a partir de la imagen. Esto se logra "aplanando" cada imagen en un vector  $K^2$ , vea la imagen. Este paso se hace para cada imagen que se tiene.

**Figura 16**

**Transformación de matriz a vector para aplicar Eigenface.**



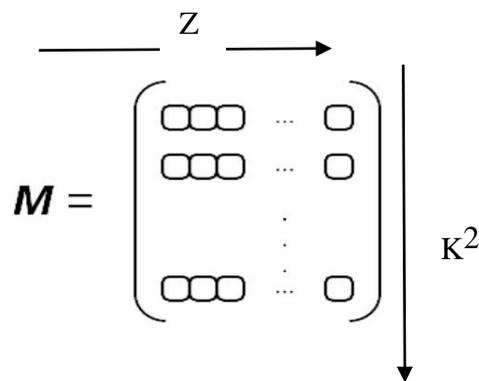
Nota. En la imagen se puede ver la transformación de una

matriz de  $k \times k$  a unvector  $K^2$  para la aplicación del algoritmo de Eigenface, Fuente: Elaboración propia.

- Después de aplanar cada imagen, se forma una matriz de vectores, cada vector de tamaño  $K^2$  es una imagen, como se ve en la figura 17, donde  $Z$  es el número total de imágenes en nuestro conjunto de datos.

**Figura 17**

**Matriz y vectores de imágenes de tamaño  $Z \times K^2$ .**



Nota: En la figura se puede apreciar una matriz de  $Z \times K^2$ , dentro de la matriz se encuentran  $K^2$  vectores, Fuente: Elaboración propia.

- Calcular la media  $u_i$  de cada fila en la matriz, dándonos el valor de intensidad de píxel promedio del conjunto de datos de la imagen.
- Restar el  $u_i$  de cada columna  $Z_i$ , esto significa centrar los datos en la media y es un paso obligatorio cuando se realiza PCA en Eigenface. Se obtiene una matriz  $M$  centrada.
- Se calcula la matriz de covarianza.

$$\text{Matriz de covarianza: } \mathbf{M} \cdot \mathbf{M}^T$$

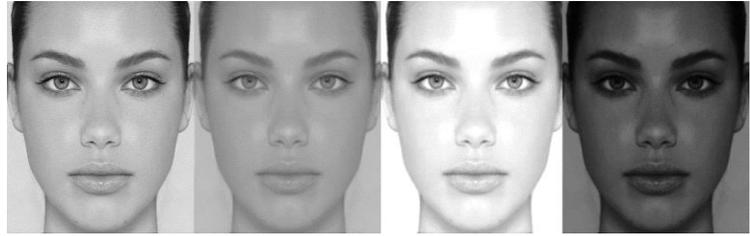
8. Se realiza una descomposición de valores propios en la matriz de covarianza para obtener los valores propios  $\lambda_i$  (magnitud) y vectores propios  $X_i$  (sentido).
9. Clasificar  $X_i$  por  $|\lambda_i|$ , de mayor a menor.
10. Tome los N vectores propios superiores con la magnitud de valor propio correspondiente más grande.
11. Transforme los datos de entrada proyectándolos (es decir, tomando el producto escalar) en el espacio creado por los N vectores propios superiores; estos eigenvectores se llaman nuestras Eigenface.
12. Se calcula el eigenface de la imagen a evaluar.
13. Se calcula la distancia Euclidiana entre la imagen evaluada y los Eigenface, para determinar a qué imagen de la base de datos correspondecaso contrario se indicará que no se encuentra una coincidencia.

**b. Algoritmo Fisherface**

Eigenface considera que la iluminación es una característica importante de una imagen y con mayor razón en un rostro, como se aprecia en la figura siguiente, pero existe algoritmos que no consideran este aspecto como fundamental. (Yashquero, s.f.)

## Figura 18

### Imágenes con distinto nivel de iluminación.



Nota: En la figura se puede ver una imagen con diferentes iluminaciones, Fuente: Elaboración propia.

Al considerar estas iluminaciones como una característica importante, puede descartar las características de otras personas considerándolas menos útiles, esto tiende a generar situaciones desfavorables en el aprovechamiento de la data para la identificación de imágenes de rostros.

Podemos solucionar esto ajustando las caras propias de modo que se extraiga las características de todos los individuos por separado en lugar de mirarlos como un todo, como es el caso de los Eigenface. Entonces, ahora, incluso si los datos de la cara de una persona tienen cambios de iluminación altos, no afectará las características de otras personas. (Yashquiero, s.f.).

El algoritmo de Fisherface extrae los componentes principales que separan a un individuo de otro. Entonces, ahora las características de un individuo no pueden dominar las características de otra persona. Este algoritmo fue inventado por el gran estadístico Sir RA Fisher, quien lo usó con éxito para clasificar flores, basado en la técnica de Análisis Discriminante Lineal o LDA, por sus siglas en inglés.

El reconocimiento de imágenes que usa este algoritmo se sustenta en la reducción de las dominaciones del espacio facial usando PCA y luego aplicando LDA (Yashquero, s.f.)

El algoritmo Fisherface se puede describir en los siguientes pasos (Bhattacharyya y Rahul, 2013):

### **Paso 1**

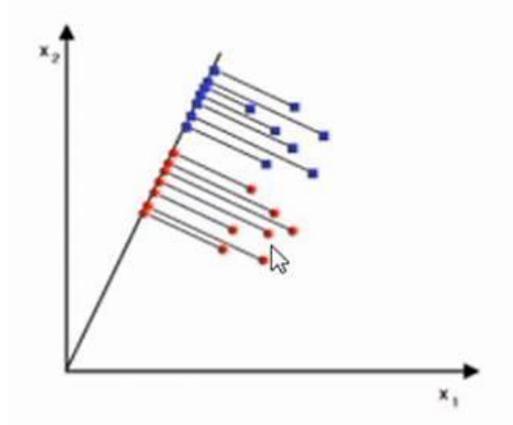
Necesitamos un conjunto de entrenamiento compuesto por un grupo amplio de imágenes de sujetos con diversas características faciales (con pequeñas variaciones en el ángulo de visión y expresiones). La base de datos debe contener varios ejemplos de imágenes faciales para cada sujeto en el conjunto de entrenamiento y al menos un ejemplo en el conjunto de prueba. Todas las imágenes ya están normalizadas en matrices  $m \times n$  y que sólo contienen las regiones de la cara y no gran parte del cuerpo de los sujetos.

### **Paso 2**

Para cada imagen, partiendo de la matriz bidimensional  $M \times N$  de valores de intensidad  $I(x,y)$ , construimos la expansión vectorial  $\Phi \in \mathbb{R}^{m \times n}$ . Este vector corresponde a la representación inicial del rostro. Así, el conjunto de todos los rostros en el espacio de características se trata como un espacio vectorial de alta dimensión, en la figura siguiente se muestra cómo se pueden separar en dos grupos o clases, los valores matriciales dentro del espacio vectorial (rojos vs azules), para esto es necesario encontrar una dirección, que pueda proyectar adecuadamente cada punto del espacio bidimensional sobre el vector (recta).

**Figura 19**

**Proyección y segmentación de cada punto del espacio bidimensional**



Fuente: Yashquero, s.f.

### **Paso 3**

Al definir todas las instancias de la cara de la misma persona como pertenecientes a una clase y las caras de diferentes sujetos como pertenecientes a diferentes clases para todos los sujetos del conjunto de entrenamiento, establecemos un marco para realizar un análisis de separación de clústeres (clases o grupos) en el espacio de características. Además, una vez etiquetadas todas las instancias del conjunto de entrenamiento y definidas todas las clases, calculamos las matrices de dispersión dentro de la clase y entre las clases.

Es decir, tomando como ejemplo la figura 19, se busca que la media de valores del grupo rojo se encuentre lo más separado posible de la media del grupo azul, así mismo se quiere que la dispersión dentro de cada clase sea lo menor posible.

Luego una nueva imagen que se desea identificar con este algoritmo, lo que hará es buscar a que subconjunto o clase dentro de las ya

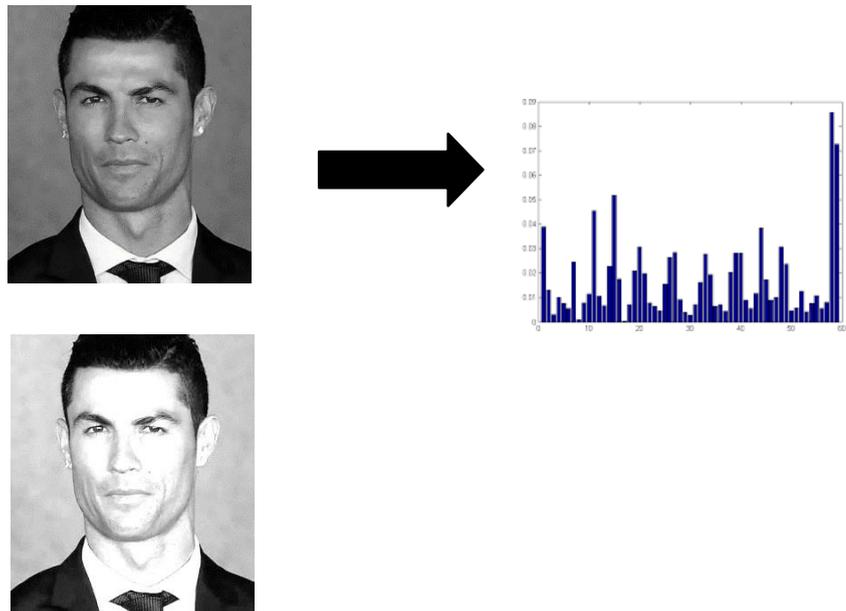
definidas sea más probable que pertenece.

**c. Algoritmo LBP (Local Binary Pattern)**

El algoritmo Local Binary Pattern o Patrón Binario Local es un operador de textura básicamente que etiqueta los píxeles de una imagen por umbral, partiendo de una escala de grises se crea una imagen binaria. Una de las propiedades más de este algoritmo es su indiscriminación frente a los cambios en la escala de grises como por ejemplo la luminosidad, vea figura 20, ya que emite el mismo patrón gráfico (histograma) para 2 o más imágenes idénticas con diferencia lumínica (Espinoza y Jorquera, 2015).

**Figura 20**

**Histograma con diferencia lumínica en LBP.**

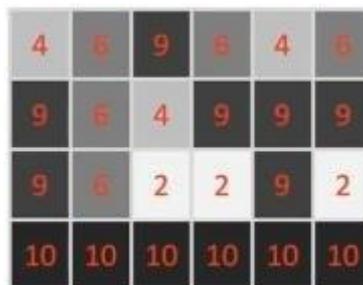


Nota. En la figura se muestra la aplicación del algoritmo LBP a una imagen y el mismo patrón gráfico para definir la similitud en las imágenes a pesar de su luminosidad, Fuente: Elaboración propia.

LBP etiqueta los píxeles de una imagen con números decimales a los que se le denomina Patrones Binarios Locales o códigos LBP de sus siglas en inglés. Para proceder con el algoritmo, previamente la imagen ha pasado por un preprocesamiento, donde se ha estandarizado a una dimensión de  $M \times N$  píxeles, aquí  $M$  puede ser igual a  $N$ , adicionalmente la imagen se ha convertido a escala de grises, donde cada pixel toma un valor según el tono de color gris que representa, véase la figura siguiente, que servirá como modelo para la descripción del algoritmo LBP.

**Figura 21**

**Imagen en escala de grises con su código de color para LBP.**



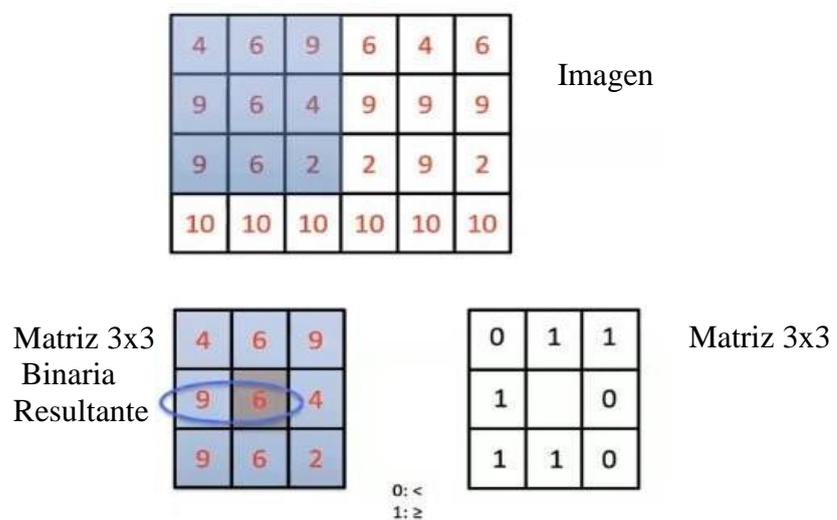
Nota. La figura muestra una imagen con un código de color según la intensidad del color, Fuente: Mery (2020).

El investigador Mery (2020) de la Universidad Católica de Chile indica que el algoritmo LBP posee tres etapas definidas 1) Codificación, 2) Mapeado y 3) Histograma. Esta corresponde a una forma más didáctica de explicar los pasos de desarrollo de LBP, sin dejar ninguna parte del proceso del algoritmo.

En la primera etapa de **codificación** se codifica la estructura local alrededor de cada píxel que se evalúa, para el ejemplo el píxel de valor

6, vea figura 22, cada píxel se compara con sus ocho vecinos en una vecindad de 3x3 valores, variantes del algoritmo toman otros valores de NxN, si el vecino resultamayor o igual al pixel central esa posición toma el valor 1, caso contrario se le asigna 0. Luego los valores de la matriz binaria, partiendo del valor que se encuentra en la esquina superior izquierda y rotando en sentido horario, se multiplican por las potencias de 2, comenzando por la potencia cero y se suman todos los valores obtenidos, la suma resultante es la etiqueta de la matriz equivalente para la celda evaluada (inicialmente con valor 6), se procede de la misma forma para los demás pixeles. Al final se obtiene los resultados de la figura 23, obsérvese que no se ha trabajado los pixeles del borde de la imagen, esto no genera ningún problema ya que la perdida de datos es insignificante, si nos ponemos a pensar que en las imágenes reales corresponde al ancho de un pixel en el borde.

**Figura 22**  
**Proceso de codificación para imágenes en LBP.**



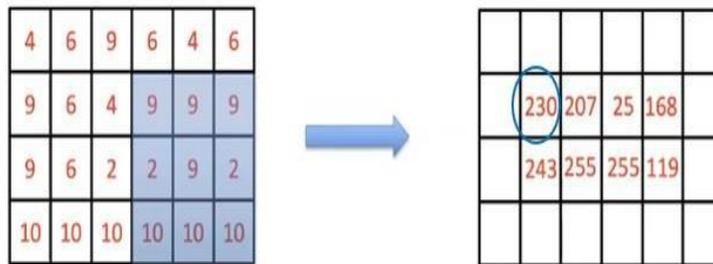
1	2	4
128	+	8
64	32	16

$$= 2+4+32+64+128 = 230$$

Nota. Proceso de codificación para imágenes en Patrones Binarios Locales, Fuente: Mery (2020).

**Figura 23**

**Valores de los pixeles resultante del proceso de codificación para LPB.**

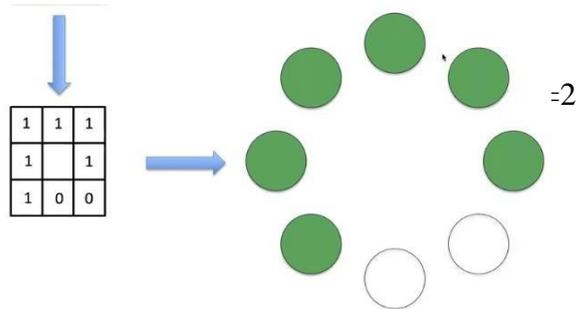


Nota. En la imagen se muestra el resultado del proceso de codificación para imágenes, Fuente: Mery (2020).

La siguiente etapa del LPB es el **mapeado**, para este paso se parte de la matriz binaria obtenida al momento de evaluar un pixel, suponiendo que se tiene la matriz binaria, vea la figura 24, donde se Gráfica 8 círculos concéntricos que corresponde a cada celda, el color verde representa el uno y los blancos el cero, lo que se trata de determinar es cuantas veces se producen transiciones de color, partiendo de cualquiera de los puntos, avanzando en sentido horario, en la imagen se observa que existen 2 transiciones (U), pasa del verde al blanco y luego del blanco al verde.

**Figura 24**

**Proceso de Mapeo en LBP.**

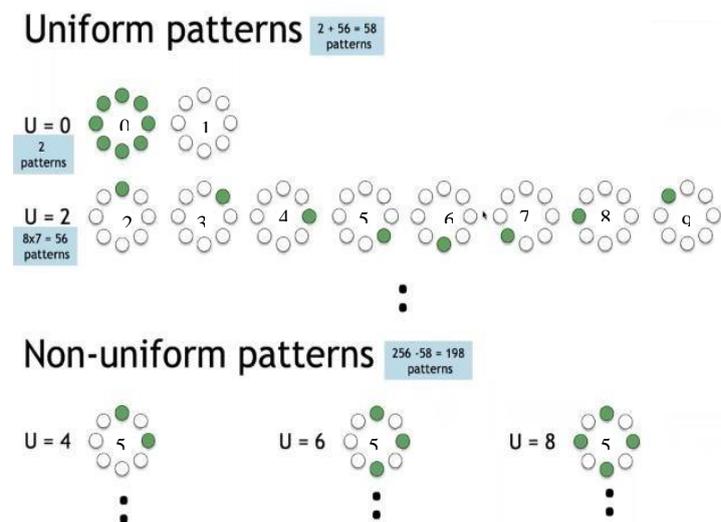


Nota. En la figura se muestra 8 círculos concéntricos que corresponde a cada celda suponiendo que se tiene una matriz binaria, el color verde representa los 1 y el color blanco representa lo ceros, lo que se trata de determinar es cuantas veces se producen transiciones de color, Fuente: Mery (2020).

Mery en su exposición académica, muestra una estandarización de etiquetado numérico, figura 25, para cada tipo de transición de colores, que es usado para generar el histograma de la última etapa del algoritmo LBP.

**Figura 25**

**Etiquetas numéricas de Mapeo en LBP.**

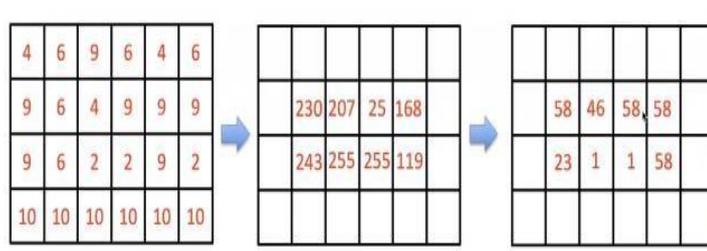


Nota. En la figura se muestra una estandarización de etiquetado numérico paracada tipo de transición de colores, que es usado para generar el histograma dela última etapa del algoritmo LBP, Fuente: Mery (2020).

Entonces si consideramos nuestro ejemplo donde se evalúa el pixel que tiene el valor 6, tiene un código transformado correspondiente al valor 230 y un patrón de transición igual a 58 ya que tiene 4 transiciones de color y esconsiderado no uniforme.

**Figura 26**

**Matrices de etiquetas de imagen, codificación y mapeado en LBP para elejemplo dado.**

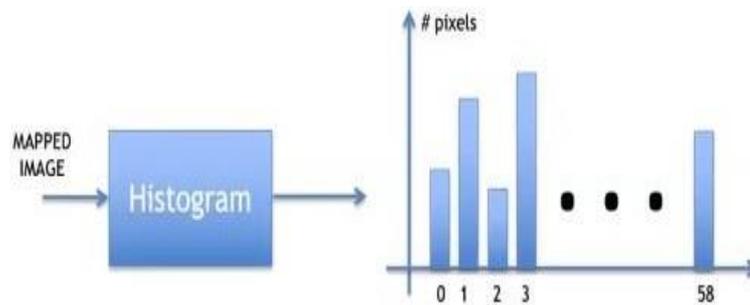


Fuente: Mery (2020).

La última etapa **histograma**, consiste en contar cuantos pixeles de la matriz de mapeado, tienen el valor 0, luego cuantas el valor 1, cuantas el valor 2 y así sucesivamente hasta llegar al valor 58, finalmente estas frecuencias halladas se Grafican en un histograma al que se le denomina “feature”. Vea figura 27.

**Figura 27**

**Obtención del histograma en LBP.**



Nota. En la figura se observa un histograma que consiste en contar pixeles dela matriz de mapeado, con esto se grafica el histograma que se denomina “Feature”, Fuente: Mer (2020).

Una vez obtenido el histograma de cada imagen este se emplea para ser comparada con el histograma de la imagen a reconocer y determinar si se encuentra en la base de datos.

**2.2.1.3. Aprendizaje automático**

En última década, el crecimiento de tecnologías emergentes ha sido muy amplia. El número de dispositivos habilitados para la red en los dominios de la internet de las cosas, y especialmente en lo que hoy se denomina la computación en el borde (edge computing) está aumentando de manera espectacular, lo que genera una amplia producción de datos electrónicos. Estos datos contienen información valiosa que puede utilizarse en diversos sectores, como la industria, educación, los negocios entre otros. (IBM, 2020). Para analizar esta información y hacer que los dispositivos Smart (dispositivos inteligentes) sean inteligentes, se requiere entrar en el mundo de aprendizaje automático (machine learning) y los algoritmos que emplean.

La definición de aprendizaje automático que IBM realiza indica que “es una

rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para imitar la forma en que los humanos aprenden, mejorando gradualmente su precisión” (IBM, 2020).

Fue en los finales de los años cincuenta que Arthur Samuel, científico de IBM, desarrollo un programa que hiciera que la computadora juegue consigo mismo a las damas, siendo el programa capaz de mejorar su rendimiento mediante la experiencia, lo que indica una forma de aprendizaje, naciendo entonces el Aprendizaje Automático o Machine Learning (IBM, 2020).

**a. Tipos de Aprendizaje Automático (IBM, 2020).**

**Aprendizaje supervisado**, trabaja sobre modelos predictivos, es decir modelos que predicen el futuro a partir de datos históricos, datos que se encuentran etiquetados para entrenar los algoritmos. A medida que los datos son introducidos en el modelo, este realiza ajustes en los pesos hasta que se obtenga un ajuste correcto en el modelo, siempre tratando de evitar los famosos sobreajustes o los ajustes insuficientes. Un ejemplo de uso en la vida real son los clasificadores de spam en el correo electrónico.

**Aprendizaje no supervisado**, este tipo de aprendizaje emplea algoritmos para analizar y agrupar conjuntos de datos que no se encuentran etiquetados. Se emplean para descubrir patrones ocultos o agrupaciones sin la necesidad de que intervenga un ser humano, muy útil en el análisis exploratorio de datos, como la segmentación de clientes por ejemplo el **reconocimiento de imágenes**. En otros casos específicos se emplea para reducir la dimensionalidad simplificando

la cantidad de funciones que se debe emplear para las predicciones.

**b. Exactitud y precisión de un algoritmo en aprendizaje automático para reconocimiento facial.**

Después de aplicar los algoritmos de aprendizaje automático, necesitamos algunas herramientas para saber si han hecho bien su trabajo. A estas herramientas se les conoce como métricas de evaluación del rendimiento. En los diferentes estudios se ha introducido un número importante de métricas, cada una de las cuales considera determinados aspectos del rendimiento de un algoritmo.

En esta investigación, utilizamos dos métricas comunes a los problemas de clasificación para obtener información valiosa sobre el rendimiento de los algoritmos y realizar un análisis comparativo. Estas métricas son la precisión y la exactitud (IArtificial.net, 2020).

**Precisión.**

La precisión o precisión en inglés, refiere en qué medida acierta el modelo cuando pronostica un buen resultado, Si fuera un estudio de detección de cáncer, se relacionaría los pacientes en los que se acertó al indicar que tenían cáncer (TP) sobre la suma de pacientes en los que se acertó más los pacientes a los que se indicó que tenían cáncer cuando no era verdad (FP). Para calcular el valor se aplica la siguiente formula:

Donde:

$$precision = \frac{TP}{TP + FP}$$

TP: Verdaderos positivos

FP: Falsos positivos

Por ejemplo, si usamos un modelo con un bajo nivel de precisión, a muchos pacientes se les dirá que tienen una enfermedad, lo que podría dar lugar a algunos diagnósticos erróneos.

### **Exactitud.**

La exactitud o accuracy en inglés, mide el porcentaje de casos que el modelo ha logrado acertar. Es decir, cuántos aciertos positivos y aciertos negativos tubo en relación al total de casos analizados. Si fuera un estudio de detección de cáncer, se relacionaría la suma de los pacientes en los que se acertó al indicar que tenían cáncer (TP) más los pacientes en los que se acertó al indicar que no tenían cáncer (TN) todo esto sobre el número total de pacientes que pasaron la prueba.

Para calcular la exactitud empleamos la siguiente formula:

Donde:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP: Verdaderos positivosFP: Falsos positivos

TN: Verdaderos negativosFN: Falsos negativos

El problema con la exactitud es que nos puede llevar al engaño, es decir,puede hacer que un modelo malo parezca que es mucho mejor de lo que en realidad es, esto sucede cuando las clases están desbalanceadas, es decir cuando existen pocos registros para entrenamiento en una de las categorías, siendo insuficientes para generar un modelo confiable, Para problemas con clases desbalanceadas es mucho mejor usar precisión. (IArtificial.net, 2020).

Por lo tanto, se podría decir que la precisión resulta ser un mejor indicador al momento de evaluar un modelo.

### **2.2.2. Identificación de personas**

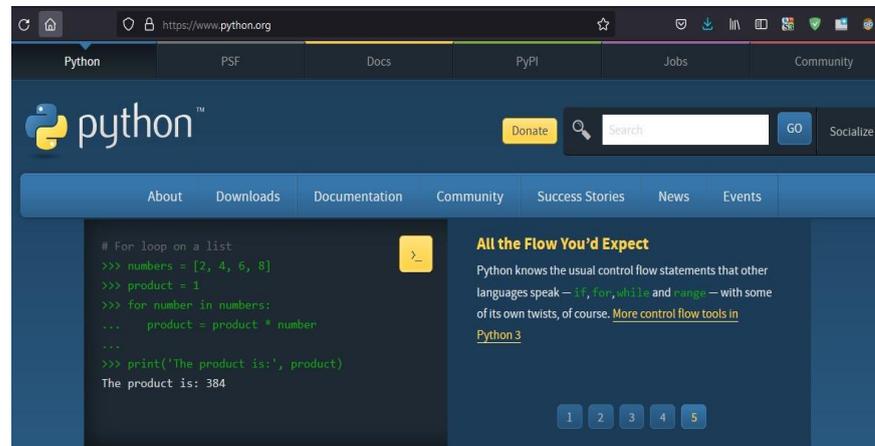
El concepto de identificación es confundido en muchas ocasiones con el de reconocimiento e incluso con el de verificación, sin embargo, existen diferencias entre uno y otro, particularmente el tema que nos llama en este apartado es la definición del término identificación y por consiguiente el de identificación de personas. Según el blog en Tecnología biométrica de Sady (2016) “Identificación es la tarea mediante la cual el sistema biométrico intenta determinar la identidad de un individuo. Los datos biométricos son comparados contra todos los datos que se albergan en la base de datos”. Entendiéndose por identidad al “conjunto de rasgos propios de un individuo o de una colectividad que los caracterizan frente a los demás.” (Real Academia Española, s.f., definición 2.)

### **2.2.3. Herramientas aplicadas en el aprendizaje automático.**

**a. Python:** Es un lenguaje de programación orientada a objetos no propietario. Cuenta con un intérprete de comandos muy intuitivo y versátil, posee una extensa biblioteca, está disponible libremente para todas las plataformas desde su sitio web <https://www.python.org/>. (ver figura 28). El mismo sitio también contiene distribuciones y punteros a muchos módulos, programas y herramientas de terceros gratis, y documentación adicional.

**Figura 28**

**Página oficial de Python**



Nota. En la figura se puede apreciar la página del lenguaje de programación Python, Fuente: [www.python.org](http://www.python.org)

- b. Anaconda:** Es una tecnología creada para trabajar en la ciencia de los datos, tanto para estudiantes como empresas que requieren integración con trabajos de Aprendizaje automático o Machine Learning y también emplea requerimientos en Inteligencia Artificial. Se integra cómodamente con el lenguaje de programación Python y el sistema operativo Windows, puede ser usada y descargada en su versión personal desde su dirección de internet [www.anaconda.com](http://www.anaconda.com) (ver figura 29). El mundo actual de la ciencia de datos increíblemente activo, todos los días, surgen desafíos, así como innovaciones increíbles. En ese sentido las organizaciones deben brindar a sus colaboradores la tecnología de aprendizaje automático y ciencia de datos de vanguardia para que puedan innovar de forma rápida.

**Figura 29**

**Página oficial de Anaconda**



Nota. En la figura se muestra la página web que nos ayuda con la obtención de paquetes para ciencia de datos, Fuente:

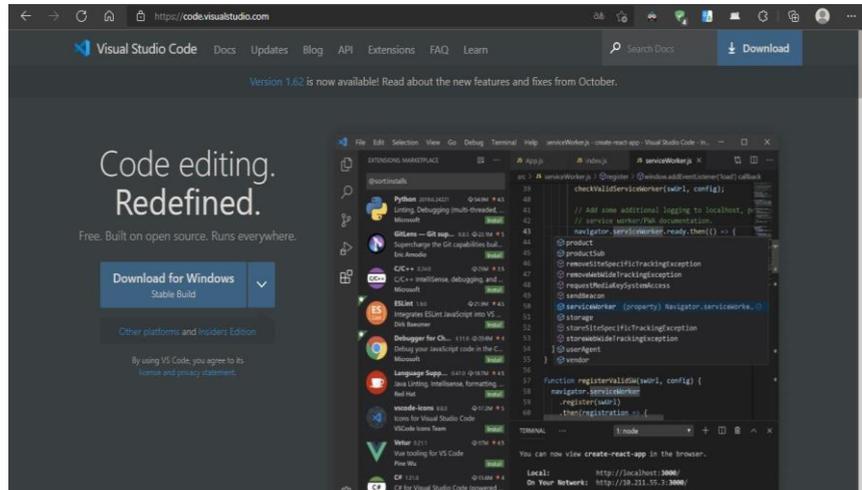
[www.anaconda.com](http://www.anaconda.com)

- c. Visual Studio Code:** Es un editor de código fuente ágil y potente, se ejecuta en el escritorio principal del sistema operativo y está disponible para Windows, Linux y macOS. Soporta múltiples lenguajes de programación como Python, JavaScript, C ++, PHP, entre otros. Esta libre para ser descargado desde la internet y puede ser accedida desde su página web <https://code.visualstudio.com> (ver figura 30),

Visual Studio Code es hoy por hoy uno de los editores de código más extendido en el mercado tecnológico, personalizable, amigable, pero si requiere de pericia básica para su manejo.

**Figura 30**

**Página oficial Visual Studio Code**



Nota. La figura muestra al editor de código Visual Studio Code,

Fuente:code.visualstudio.com

- d. **OpenCV:** Es una biblioteca de uso libre para temas de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). Apareció en enero de 1999, ha sido empleada en un sin número de aplicaciones y es una de las bibliotecas más importantes empleadas en codificación para el reconocimiento de objetos, detección de movimiento, reconstrucción 3D a partir de imágenes que son sólo algunos ejemplos de aplicaciones de OpenCV. Esta biblioteca fue desarrollada totalmente en C++ sin embargo existen conectores mediante su API para Python, java, Matlab y otros lenguajes.

**2.3. Definición de términos básicos**

- A. **Algoritmo.** “Es un conjunto de pasos o instrucciones a seguir para alcanzar solucionar un problema” (Cernánides, H. & Zapata, E., 2006)
- B. **Dato:** Información dispuesta de manera adecuada para su tratamiento por una computadora (Laudon, 2016).

- C. Datos de entrenamiento.** - Los datos de entrenamiento son aquellos que contienen la respuesta correcta, que se conoce como destino o atributo de destino. El algoritmo encuentra patrones en los datos de entrenamiento (imágenes) y genera un modelo de aprendizaje automático que captura dichos patrones para reconocer nuevas imágenes (Aws, s.f.).
- D. Datos de testeo.** – Los datos de testeo o prueba son los datos que se reservan para comprobar si el modelo generado por los algoritmos de reconocimiento facial, a partir de los datos de entrenamiento funcionan (Aws, s.f.).
- E. Eficiencia.** - Según el Diccionario de la Real Academia Española, “eficiencia es la ‘capacidad de disponer de alguien o algo para conseguir el cumplimiento adecuado de una función. No debe confundirse con la eficacia, que se define como ‘la capacidad de lograr el efecto que se desea o se espera’” (“Eficiencia”, 2021).
- F. Identificación Biométrica.** - “Es la utilización de los métodos de la biometría para verificar la identidad de un individuo” (Cernánides, H. & Zapata, E., 2006). Entre las técnicas de autenticación biométrica tenemos: Reconocimiento de la firma, Reconocimiento de la forma de la mano, Reconocimiento Facial, Reconocimiento de Voz, reconocimiento de la Huella Dactilar, Reconocimiento del iris.
- G. Falsa Aceptación.** - “Cuando se acepta a alguien que NO es” (Cernánides, H. & Zapata, E., 2006); por ejemplo, cuando se acepta el ingreso de una persona no autorizada por una inadecuada identificación, se le conoce también como falso positivo.

- H. Falso Rechazo.** - “Consiste en no aceptar a alguien que SI es” (Cernández, H. & Zapata, E., 2006); termino conocido también como falso negativo, se incurre en este tipo de situación cuando se rechaza a una persona en el sistema cuando debió ser admitido. Por ejemplo, cuando en una dependencia bancaria una persona quiere retirar una suma considerable de dinero, pero el identificador de huella dactilar no lo reconoce a pesar de ser el titular de la cuenta bancaria.
- I. Fotograma.** – “Un fotograma es cada una de las imágenes impresas químicamente en la tira de celuloide del cinematógrafo o bien en la película fotográfica”, (Fotograma, 2021). Cuando una secuencia de fotogramas es visualizada de acuerdo a una determinada frecuencia de imágenes por segundo se genera una sensación de movimiento dando lugar a los videos y películas cinematográficas.
- J. FPS.** - Fotogramas por segundo o cuadros por segundo (CPS), es la velocidad por la cual se muestra una secuencia de fotogramas (García, 2009).
- K. Overfitting.** – Traducido del inglés como “sobreajuste”, hace referencia al fallo de nuestro modelo al generalizar -hacer encajar- el conocimiento que pretendemos que se aprenda. Cargar con información adicional al sistema hace que en vez de mejorar sus cálculos empeore o disminuya su precisión.
- L. Ruido.** - El ruido dentro del contexto de un conjunto de datos o bases de datos puede definirse como “la alteración parcial o total de la información recogida en una instancia causada por un factor externo. El ruido conduce a modelos excesivamente complejos con un rendimiento

degradado”, (Garcial-Gil, Leungo, Garcia, y Herrera, s.f.). Esta alteración puede ser producto de datos no completados, borrados y modificados.

## **2.4. Formulación de hipótesis**

### **2.4.1. Hipótesis General.**

Evaluando algoritmos en reconocimiento facial de aprendizaje automático se determinará aquella que mejor resulte en la identificación de personas de una institución educativa de Pasco - 2021.

### **2.4.2. Hipótesis Específicas.**

Evaluando algoritmos de reconocimiento facial en aprendizaje automático se determinará aquella con mejor resultado de exactitud en la identificación de personas.

Evaluando algoritmos de reconocimiento facial en aprendizaje automático se determinará aquella con mejor resultado de precisión en la identificación de personas.

## **2.5. Identificación de Variables.**

**Variable independiente:** Algoritmos en Reconocimiento facial

**Variable dependiente:** Identificación de personas.

**Variable interviniente:** Aprendizaje automático

## **2.6. Definición Operacional de las variables e indicadores**

**Tabla 1**

**Definición operacional de las variables e indicadores.**

<b>VARIABLE INDEPENDIENTE</b>	<b>INDICADOR</b>	<b>VALOR FINAL</b>	<b>TIPO DE VARIABLE</b>
-------------------------------	------------------	--------------------	-------------------------

ALGORITMOS DE RECONOCIMIENTO FACIAL	<ul style="list-style-type: none"> <li>Algoritmos de reconocimiento facial implementado</li> </ul>	1. Si 2. No	Dicotómica
-------------------------------------	--	----------------	------------

<b>VARIABLE DEPENDIENTE</b>	<b>INDICADOR</b>	<b>VALOR FINAL</b>	<b>TIPO DE VARIABLE</b>
IDENTIFICACIÓN DE PERSONAS	<ul style="list-style-type: none"> <li>Exactitud en la identificación de personas</li> <li>Precisión en la identificación de personas</li> </ul>	0 a 100%	Cuantitativa

Nota. Definición de variables e indicadores de la presente investigación, Fuente:

Elaboración propia.

## **CAPITULO III**

### **METODÓLOGA Y TÉCNICAS DE INVESTIGACIÓN**

#### **3.1. Tipo de investigación**

La presente investigación es de tipo aplicada ya que permitirá solucionar problemas prácticos en el corto plazo aplicando para ello los conocimientos adquiridos (Hernández et al., 2010).

#### **3.2. Nivel de investigación**

La investigación se encuentra en el nivel descriptivo y explicativo, es descriptiva porque en la investigación se caracterizó cada una de las variables en estudio, es explicativa porque responde a la cuestión de ocurrencia del fenómeno y en qué condiciones se presenta éste (Hernández et al., 2010).

#### **3.3. Métodos de la Investigación**

Para el presente tema de investigación se aplicó el método hipotético deductivo, este método parte de una hipótesis o premisa explicativa, para más adelante conseguir conclusiones particulares de ella, y que posteriormente serán comprobadas.

### **3.4. Diseño de investigación.**

Para realizar esta investigación se empleó el diseño de tipo experimental, existiendo manipulación de la variable independiente para analizar sus efectos en la variable dependiente. Teniéndose para efectos del experimento un solo grupo de análisis por lo que se encuentra dentro de la subcategoría cuasiexperimental.

### **3.5. Población y Muestra**

#### **3.5.1. Población**

El universo poblacional está conformado por las imágenes de estudiantes de una institución educativa de Pasco, específicamente estudiantes de la Universidad Nacional Daniel Alcides Carrión (UNDAC) sede Pasco

#### **3.5.2. Muestra**

La muestra de estudio es intencionada no probabilística, se tomó 10 estudiantes de la Facultad de Ingeniería UNDAC, al azar de quienes se obtuvo 31 imágenes faciales para la investigación. Siendo en total 310 imágenes como muestra para el trabajo.

### **3.6. Técnicas e instrumentos de recolección de datos.**

La obtención de los datos para esta investigación, así como la información correspondiente, se ha realizado empleando ciertas técnicas e instrumentos que a continuación se enumeran.

#### **3.6.1. Técnicas**

- La observación experimental para elaborar datos de las variables en estudio
- El Análisis Documental de las fuentes bibliográficas correspondientes.
- Captura de imágenes
- Métricas de reconocimiento facial

#### **3.6.2. Instrumentos**

- En base a la lectura
- Textos
- Cámara fotográfica y de video digital
- Base de datos de imágenes
- Registro de datos para métricas de exactitud y precisión
- Software de programación

### **3.7. Selección, validación y confiabilidad de los instrumentos de investigación**

#### **3.7.1. Selección de instrumentos**

Esta investigación se basa en recolectar imágenes del rostro de personas, para luego extraer y analizar características mediante algoritmos programados en computadora, que aplican aprendizaje automático y elaborados en el lenguaje de programación (Python), y determinar el grado de exactitud y precisión con el que estos algoritmos identifican a las personas que quieren acceder o que se encuentran en algún ambiente determinado.

Por ello fue necesario que la recolección de datos, en este caso las imágenes de rostros de las personas, se realizara mediante una cámara fotográfica y de video digital (figura 31) certificación ISO, las características de este instrumento son:

#### **CAMARA DIGITAL**

Marca: NIKON Modelo: D3200

Píxeles efectivos: 24.2 millones (megapíxeles) Tamaño de sensor: 23.2 mm x 15.4 mm Sensibilidad: Hi-1 (ISO 12800)

Video: Full HD (30 cps), HD (60 cps), VGA (30 cps)

### **Figura 31**

#### **Cámara fotográfica y de video digital Nikon.**



Nota. En la figura se muestra la cámara con la que se trabajó en la investigación, Fuente: Elaboración propia.

Así mismo se empleó una base de datos de carpetas que contienen archivos de las imágenes fotográficas, para la investigación se guardó 31 imágenes de cada persona, el formato de imagen es JPG, formato reconocido en cualquier sistema operativo, la codificación del etiquetado de cada imagen es la siguiente:

#### **Apellidos+Nombres+ YYYY.JPG**

Donde:

YYYY            código de orden de la imagen de la persona va de  
01a 999

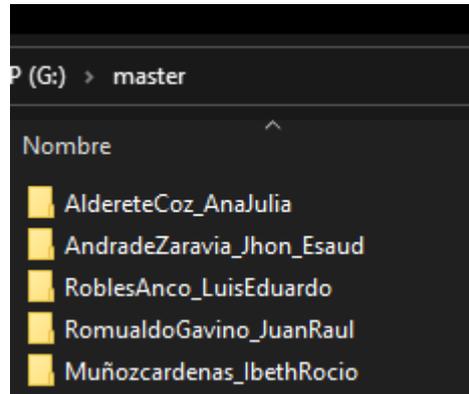
Apellidos        Apellido paterno y materno de la persona

Nombres        Nombres de la persona

Obsérvese la figura 32, donde se muestra la estructura parcial de la base de datos de imágenes, contenidas en una carpeta principal denominada “master”, cada carpeta corresponde a una persona, el nombre de la carpeta se etiqueta con los apellidos y nombres de la misma.

**Figura 32**

**Estructura de carpetas de la base de datos de imágenes.**



Nota. En la figura se muestra la estructura que se usó para almacenar las imágenes obtenidas, donde cada carpeta corresponde a una persona, Fuente:Elaboración propia.

El almacenamiento de imágenes en la base de datos se realizó mediante la aplicación de un programa desarrollado en lenguaje de programación Python, teniendo como característica fundamental el dimensionamiento homogéneo de todas las imágenes almacenadas a 150 x 150 píxeles de forma tal que no genere inconsistencia o errores al momento de ejecutar los algoritmos de reconocimiento facial.

Finalmente se empleó un formato para el Registro de datos para ser usado en el cálculo de las métricas de exactitud y precisión. El formato del documento se indica en el anexo 2, consta de 6 campos (columnas) donde se registra.

- Nombre de estudiante evaluado
- Código de la imagen evaluada
- TP: El algoritmo identifica positivamente la imagen de la persona en la base de datos.
- FP El algoritmo identifica positivamente la imagen de la persona que no le

corresponde en la base de datos.

- TN: El algoritmo identifica negativamente la imagen de la persona porque no existe en la base de datos.
- FN El algoritmo identifica negativamente la imagen de la persona apesar que existe en la base de datos.

### **3.7.2. Validación**

Para Hernández, Fernández y Baptista (2006) “la validez en términos generales, se refiere al grado en que un instrumento realmente mide la variable que quiere medir”.

Para el caso de la cámara fotográfica digital Nikon D3200, posee certificaciones ISO que validan los aspectos a ser considerados para una fotografía como resolución, brillo, formato de guardado de imagen conforme se indica en la descripción que se hace en el apartado 3.6.1. y que garantizan la calidad de la imagen.

En cuanto a la estructura de la base de datos de imágenes empleadas, esta se ha adecuado de otras investigaciones en reconocimiento facial (La Madrid, D. et. al (2019), García, M. (2009) y Espinoza, D. et. al (2015)), donde no se indica dificultad en el uso, por lo que se considera apropiada.

Finalmente, en cuanto al Registro de datos para ser usado en el cálculo de las métricas de exactitud y precisión, este es un simple formato Excel que contiene 6 campos con la información requerida para calcular las métricas y que no requiere mayor validación.

### **3.8. Técnicas de procesamiento y análisis de datos.**

Para el procesamiento de datos se empleó la técnica de preprocesamiento de imágenes, detección, reconocimiento y clasificación de rostros. Esta técnica

implica el empleo de diversos algoritmos de aprendizaje automático.

### **3.9. Tratamiento Estadístico**

El tratamiento estadístico de datos se realizó mediante el empleo de métricas de evaluación de aprendizaje automático como son: exactitud, precisión y tiempo de procesamiento.

Así mismo nos valimos de gráficos ilustrativos, tablas y figuras.

### **3.10. Orientación ética filosófica y epistémica**

Para la realización de la investigación se pidió el consentimiento informado a cada uno de las personas que conforman la muestra de estudio, indicándoles que la información que proporcionen será de uso exclusivo para la investigación, reservándose cada una de ellas el derecho de retirarse de la misma, si así lo creía conveniente.

## **CAPITULO IV**

### **RESULTADOS Y DISCUSIÓN**

#### **4.1. Descripción del trabajo de campo**

El desarrollo de la investigación implicó primeramente abordar dos puntos necesario, por un lado, el software y por el otro el hardware empleado en el procesamiento de la información. Posteriormente se describió la base de datos de imágenes empleada y finalmente se detalló el proceso de trabajo de campo realizado, basado en el flujograma base planteado en el capítulo 2 (subtitulo 2.2.1.1) del modelo de reconocimiento facial.

##### **4.1.1. Software**

El software empleado en el trabajo investigativo se muestra en la tabla siguiente (tabla 2), detallando las características de cada una.

**Tabla 2**

**Software empleado en el trabajo de investigación**

<b>Software</b>	<b>Tipo de licencia</b>	<b>Características</b>
<b>VsCode</b>	Microsoft corporación. (versión de libre acceso)	Entorno de desarrollo integrado (IDE), se emplea con diversos lenguajes de programación, en este caso Python
<b>Python 3.9.0</b>	Software libre	Lenguaje de programación
<b>Excel 2019</b>	Microsoft Corporation.	Hoja de cálculo para trabajo estadístico y gráficos.

Nota: Datos sobre los software y lenguaje de programación usados, Fuente:  
Elaboración propia.

El empleo del VSCode de Microsoft resulto de mucha ayuda, al ser una herramienta muy versátil, fácilmente puede implementarse con otros lenguajes de programación, así como con librerías sin conflictos entre versiones, para la implementación del código de programación en Python 3.9.0. se generó el entorno de desarrollo apropiado en esta IDE para todo el proyecto.

Se vio por conveniente organizar toda la información de trabajo en un solo directorio (carpeta) subdividido en directorios, uno dedicado a la programación, otro para la base de datos de imágenes y uno tercero para objetos extras.

Se instalaron apropiadamente las librerías necesarias, las más importantes: OpenCV y TensorFlow. OpenCV que como ya se describió en el capítulo tres,

presta las funcionalidades necesarias para la ejecución de los algoritmos de reconocimiento facial, por el otro lado TensorFlow permite emplear funcionalidades complementarias en cuanto al preprocesamiento de imágenes, esto es dimensionamiento estándar de los archivos de imagen y tratamiento de aspectos como luminosidad y conversión a escala de gris.

En cuanto al empleo de otras librerías adicionales, estas se muestran y describen en la tabla la tabla 3.

**Tabla 3**

**Librerías adicionales empleado en la investigación.**

<b>Librería</b>	<b>Descripción</b>
<b>numpy</b>	Posee funcionalidades para el procesamiento numérico matemático.
<b>matplotlib</b>	Da la funcionalidad para la creación de gráficos personalizados de ser necesario
<b>time</b>	Proporciona funcionalidades relacionadas con el manejo del tiempo

Nota: Datos sobre las librerías usadas en el lenguaje Python, Fuente:

Elaboración propia.

#### **4.1.2. Hardware**

En cuanto al hardware empleado se trabajó con una computadora portátil marca HP Pavilion, de procesador AMD, con memoria suficiente para realizar los cálculos matemáticos asociados a los algoritmos dados en el lenguaje de programación Python, estas características se mencionan en la tabla 4.

**Tabla 4**

**Hardware empleado en el trabajo de investigación.**

<b>Característica</b>	<b>Descripción</b>
<b>Modelo</b>	HP Pavilion 15''
<b>Procesador</b>	AMD Ryzen 5
<b>RAM</b>	8 Gb de 2400 MHz
<b>Sistema operativo</b>	Windows 10 6 bits Home Edition

Nota: Datos sobre el equipo usado en la presente investigación,

Fuente:Elaboración propia.

#### **4.1.3. Base de datos**

Para esta investigación se emplearon dos bases de datos, la primera de ellas es la base de datos Extended Yale Face Database B que contiene 16,128 imágenes de 28 personas, quienes fueron fotografiadas bajo 9 poses diferentes y 64 condiciones de iluminación, todas las imágenes están en escala de grises y en un tamaño de 480 x 640 pixeles, en formato de imagen es **pgm<sup>1</sup>**.

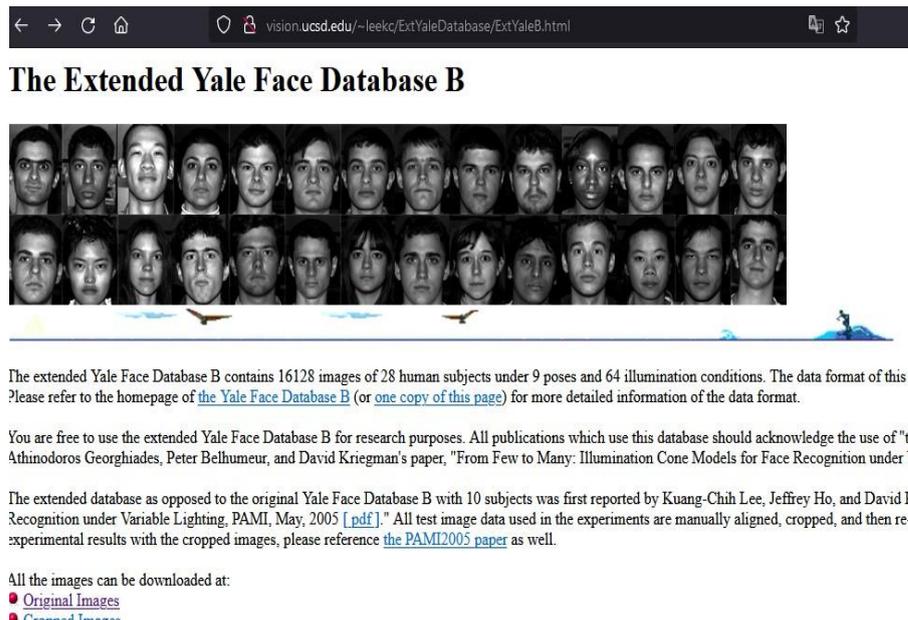
Es decir, cada una de las imágenes se encuentran ya adecuadas para ser empleadas en una matriz de números naturales con valores comprendidos entre 0 y 255, donde cada valor representa un pixel con su grado de intensidad lumínica, siendo material ya dispuesto para ser empleado de forma directa para el entrenamiento de los

algoritmos que se estudian en esta investigación (Fisherface, Eigenface y LBP-Local Binary Pattern).

A continuación, se muestra una vista de la página oficial de donde se descargó la base de datos de imágenes (véase figura 33).

**Figura 33**

### **Página web The Extended yale fase database B**



Fuente: <http://vision.ucsd.edu>

<sup>1</sup> Un archivo PGM o Portable Gray Map (mapa gris portátil) almacena imágenes en 2D en escala de grises. Cada pixel de las imágenes contiene solo uno o dos bytes de información.

Fuente: [www.Adobe.com](http://www.Adobe.com)

Así mismo, se muestra 12 imágenes de una de las personas que se encuentran en esta base de datos y que corresponde a la etiqueta YaleB17.zip (archivo comprimido) con 594 imágenes diferentes de la misma persona.

**Figura 34**

**Imágenes de rostro de la persona yaleB17**



Nota. En la figura se puede ver una lista de imágenes todas correspondientes a la misma persona en diferentes ángulos y enfoques,

Fuente: <http://vision.ucsd.edu>

Es necesario mencionar que ha sido observado (Georghiades et al., 2001) que existen mayores variaciones en las imágenes de una misma persona por variaciones en la iluminación que entre las imágenes que corresponde a diferentes personas al momento de realizar pruebas, lo que indica un buen grado de heterogeneidad en la variedad de imágenes de la base de datos.

Para el trabajo de entrenamiento de los algoritmos en esta investigación se consideraron a las 28 personas que contiene la base de datos, tomando 30 imágenes de cada una, haciendo un total de 840 imágenes analizadas en la fase de

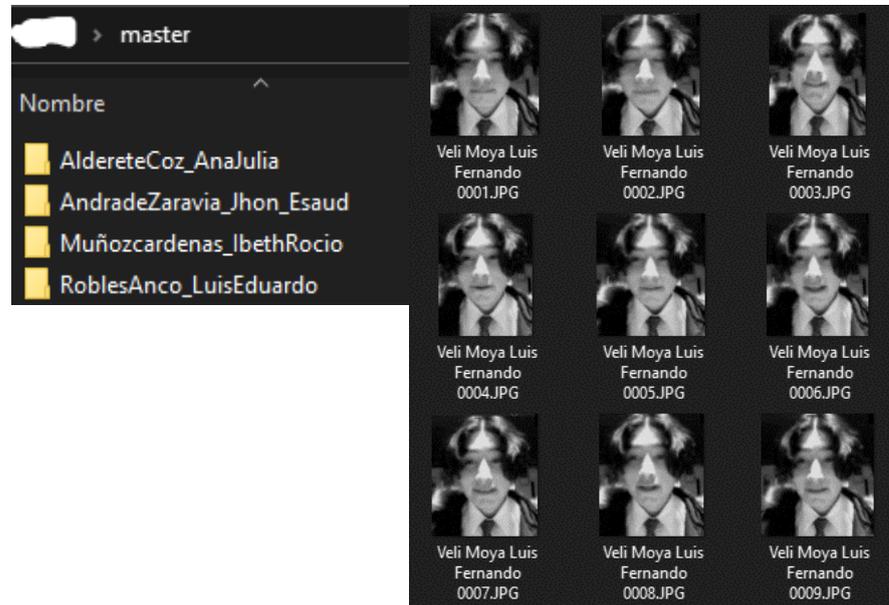
entrenamiento algorítmico. Las 30 imágenes se seleccionaron de forma manual y aleatoria para cada uno de las personas que integra la base de datos, y se redimensiono cada imagen a 150x150 pixeles, cabe mencionar que se toma 30 imágenes para no saturar la capacidad de cálculo del equipo informático y evitar disminución en la performance del procesador por calentamiento excesivo.

La segunda base de datos, corresponde a imágenes de alumnos de la UNDAC de la carrera de Ingeniería de Sistemas y Computación, a quienes se les explico y solicito su colaboración para desarrollar el experimento, la base de datos de imágenes denominada “master” (ver figura 35) y que conforma la data de imágenes que sirvió para la fase de testeo (prueba) de los algoritmos de esta investigación, contiene a 10 personas con 31 imágenes de cada una, también estandarizada a un formato de pixeles de 150 x 150, todas a escala de grises, haciendo un total de 310 imágenes que fueron empleadas para la validación de las hipótesis planteadas.

Cada una de las imágenes paso por un preprocesamiento que permitió transformarlas al formato descrito, ya que inicialmente tenían diferentes tamaños en cuanto a número de pixeles y la resolución de colores que no era escala de grises.

**Figura 35**

**Imágenes de rostro de la base de datos “Master” de un alumno de la Facultad de Ingeniería de la UNDAC.**



Nota: En la figura se muestra la base de datos de las imágenes de los alumnos de ingeniería, Fuente: Elaboración propia

Como se mencionó en el capítulo 3 (subtítulo 3.6.1) el etiquetado de cada imagen tiene la forma **Apellidos+nombres+código.jpg**, contenidas en carpetas individuales, la carpeta se identifica por los Apellidos y Nombres de los estudiantes.

#### **4.1.4. Procedimiento del trabajo de campo**

Habiendo culminado con la descripción de las herramientas de software, hardware y base de datos empleadas para la investigación se procede a describir el procedimiento seguido para el experimento:

1. Instalación del software VSCode v.1.6 sobre la plataforma del sistema operativo Windows 10 de 64 bits, tras el cual se procede a instalar en el IDE del VSCode el lenguaje de programación Python v3.9.0 y las

librerías que se indicaron en el apartado anterior. En esta última parte se debe tener cuidado de instalar la versión adecuada de librerías ya que pueden ocasionar conflictos entre versiones, para ello es recomendable visitar la página oficial de Python y de las librerías que dan soporte.

2. Instalado las aplicaciones necesarias se crea la estructura de carpetas de trabajo en la unidad C: donde se ubica la base de datos de imágenes, los programas de cada uno de los algoritmos de reconocimiento facial y demás archivos necesarios.
3. Creada la estructura de carpetas, se copian las dos bases de datos de imágenes: “Extended Yale Face Database B” con imágenes de 28 personas (16,128 imágenes para entrenamiento de los modelos algoritmos), y la base de datos “master” con imágenes de 10 estudiantes de la Facultad de Ingeniería de la UNDAC (310 imágenes para el testeo de los modelos algoritmos).
4. Posteriormente se procedió a la programación de los algoritmos para el reconocimiento facial, primero con la codificación para detección de rostro mediante Viola-Jones, esto determina, como se vio en el capítulo dos, si la imagen que se analiza contiene un rostro humano. Luego se continuó con la programación de los algoritmos Fisherface, Eigenface y LBP, para la identificación de rostros, los códigos se pueden apreciar en el anexo 3.
5. A continuación, se realizó el procesamiento de imágenes de las bases de datos, de forma manual, de tal manera que todos tengan el mismo tamaño (150x150 píxeles) y con la misma escala de colores a tonos de gris. Se tubo finalmente 840 imágenes filtradas, para la parte de entrenamiento del modelo, en la base de datos “Extended Yale Face Database B” y, 300

imágenes para el testeo o prueba en la base de datos “master”. De acuerdo a diferentes investigaciones para poder saber si está funcionando bien el modelo, debemos separar el conjunto de datos en 2: conjunto de entrenamiento (train) y conjunto de pruebas (test). Por lo general se divide haciendo “70-30”, es decir 70% del conjunto de datos para entrenamiento y el 30% para las pruebas.

6. Con los programas para reconocimiento facial terminado se procedió a realizar el entrenamiento de los modelos Fisherface, Eigenface y LBP, mediante las 840 imágenes de la base de datos Extended Yale. Subdividiendo este entrenamiento en tres categorías, una primera con 10 personas (30 imágenes de cada una) de la base de datos filtrada, una segunda con 20 personas y una tercera con la todas las personas de la base de datos (28 en total), el porqué de esta forma de trabajo radica primero, en poder observar si el modelo de reconocimiento facial mejora o no con la cantidad de imágenes de entrenamiento que se aplican y, segundo, obtener el tiempo de procesamiento del modelo en el entrenamiento de los datos para ser referenciado comparativamente con el tiempo empleado al momento de realizarel test de las imágenes de alumnos de la Facultad de Ingeniería de la UNDAC.
7. Para cada uno de estas tres categorías se probó los modelos algorítmicos Fisherface, Eigenface y LBP, con las imágenes de la base de datos “master” (300 imágenes, 250 en entrenamiento y 60 para test) para así establecer los niveles de exactitud y precisión de cada algoritmo y validar las hipótesis de investigación. Se obtuvo un número amplio de resultados que se describen en el siguiente apartado.

#### 4.2. Presentación, análisis e interpretación de resultados

A continuación, se muestra los resultados del experimento, registrados en tablas y gráficas, para cada uno de los 3 algoritmos: Eigenface, Fisherface y LBP, como se dijo en el apartado anterior (paso 6 y 7), cada algoritmo se evaluó en tres segmentos, primero con imágenes procesadas de 10 personas para entrenamiento, luego 20 y finalmente 28 personas de la base de datos Extended Yale Face Database B, en cada uno de esos segmentos se testeó a continuación las imágenes de la base de datos “master” de alumnos de la Facultad de Ingeniería - UNDAC.

El formato de las tablas tiene la siguiente estructura:

<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento o(s)</b>	<b>tiempo del test (s)</b>	<b>% de exactitud test</b>	<b>% de precisión test</b>
...				

- La columna de **imágenes en entrenamiento**, indica el número de imágenes que se empleó para el entrenamiento del modelo algorítmico con la base de datos “Extended Yale...”.
- La columna de **tiempo de entrenamiento(s)**, indica el tiempo en segundos que demora el entrenamiento del modelo algorítmico con la base de datos “Extended Yale...”.
- La columna de **tiempo del test(s)**, indica el tiempo en segundos que demora el testeó de las imágenes de la base de datos “Master”.
- La columna **% de exactitud test**, indica la ratio de exactitud que se obtiene de las imágenes testeadas, este indicador es necesario para la validación de las hipótesis.

- La columna **% de precisión test**, indica la ratio de precisión que se obtiene de las imágenes testeadas, este indicador es necesario para la validación de las hipótesis.

#### 4.2.1. Resultados del algoritmo Eigenface

##### Modelo con 10 personas entrenadas

A continuación, se describen los resultados para cada una de las 3 pruebas realizadas para el modelo de reconocimiento facial con el algoritmo Eigenface, esta primera tabla (tabla 5) presenta resultados para imágenes de 10 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 5**

##### **Resultados algoritmo Eigenface.**

<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento (s)</b>	<b>tiempo del test (s)</b>	<b>% de exactitud test</b>	<b>% de precisión test</b>
4	28,4	30,4	65,30%	66,28%
8	56,8	60,8	66,14%	67,46%
12	85,2	95,4	67,35%	68,63%
16	113,6	123,8	68,14%	69,64%
20	142,0	156,2	69,35%	71,50%
24	170,4	187,4	69,62%	71,77%
27	191,7	214,7	69,62%	72,54%
30	213,0	232,2	68,46%	69,69%

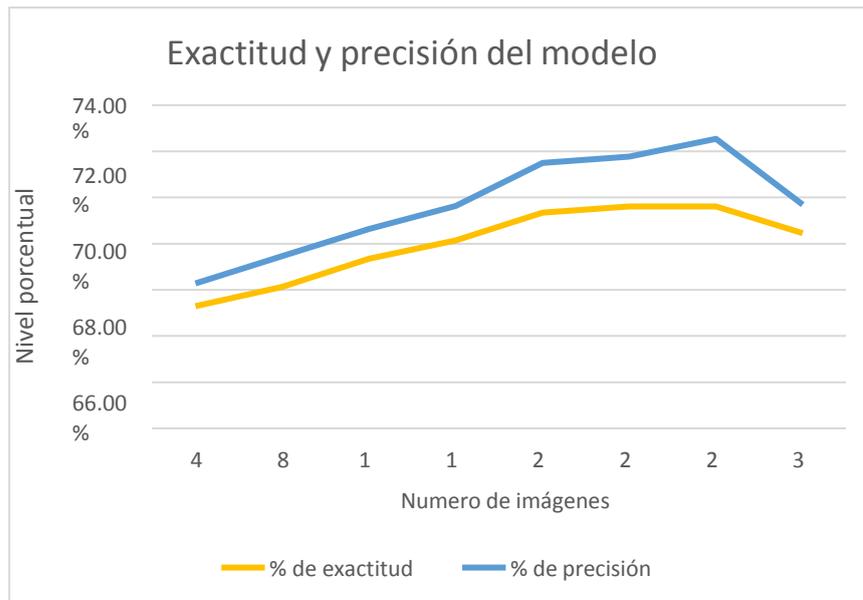
Nota. Test de estudiantes en función del número de imágenes de

entrenamiento de 10 personas, Fuente: Elaboración propia

Con esta información se presenta gráficas, que permiten entender mejor cada uno de los indicadores, mostrados en las columnas.

**Figura 36**

**Exactitud y precisión del modelo Eigenface**



Nota. Indicadores de exactitud y precisión del modelo Eigenface en función del número de imágenes de entrenamiento para 10 personas,

Fuente: Elaboración propia

Lo primero que se analiza son los niveles porcentuales de exactitud y precisión del modelo algorítmico Eigenface en la detección de rostros de los estudiantes, cuando es testeado después de que el modelo fuera entrenado con una cantidad incremental de imágenes de 10 personas.

Se puede observar que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de **exactitud** del modelo hasta llegar a un valor tope, establecido en 69,62%, después del cual este porcentaje disminuye a pesar que sigue incrementándose el número de

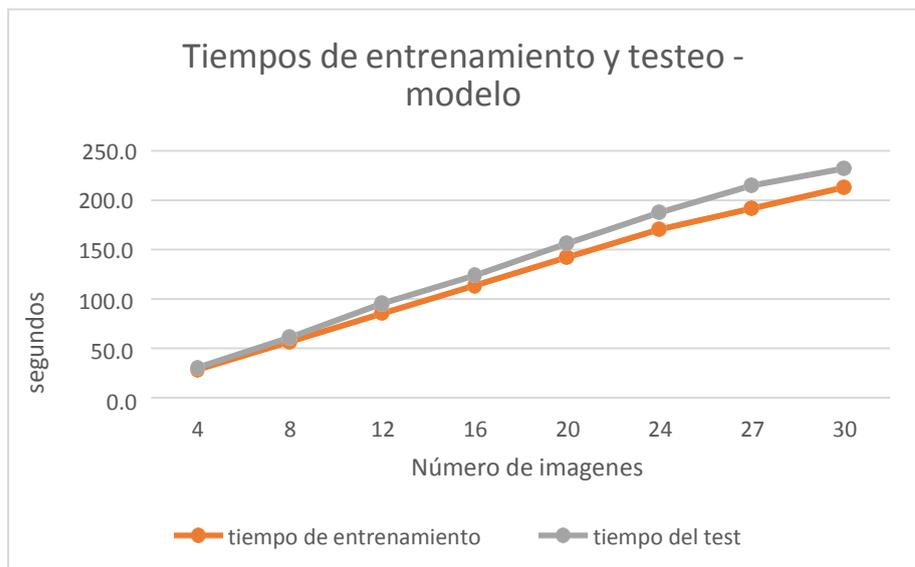
imágenes empleadas en el entrenamiento, lo que indica que se está produciendo un caso de sobreajuste (overfitting), significa que a pesar de tener más información, no se incrementa el porcentaje de exactitud como se podría esperar.

En el caso del nivel de **precisión** también se logra observar un comportamiento similar al del indicador “exactitud”, alcanzando un valor límite de 72,54%, después del cual experimenta una tendencia a disminuir en su valor produciéndose también un sobreajuste en el modelo.

En cuanto a análisis de los tiempos consumidos por el algoritmo Eigenface para 10 personas entrenadas y 10 estudiantes testados, se muestra la gráfica siguiente:

**Figura 37**

**Tiempo de entrenamiento y testeo del modelo Eigenface en función del número de imágenes de entrenamiento**



Nota. La imagen muestra el tiempo de entrenamiento y testeo del modelo Eigenface en función del número de imágenes de entrenamiento, Fuente:Elaboración propia

Se puede observar que la relación entre el tiempo consumido en el entrenamiento y en el testeo de imágenes es directamente proporcional al número de imágenes entrenadas en el modelo, es decir a mayor número de imágenes mayor tiempo empleado. Este comportamiento resulta siendo similar en las demás pruebas que se realizan para este algoritmo y los otros analizados, por lo que solo se presentara este tipo de gráfico en esta parte a manera de evidenciar como se vinculan el número de imágenes entrenadas y los tiempos consumidos por el modelo, en los demás resultados se presentaran solo los datos de la tabla.

**Modelo con 20 personas entrenadas**

Esta segunda tabla (tabla 6) presenta resultados para imágenes de 20 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 6**

**Resultados algoritmo Eigenface.**

<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento (s)</b>	<b>tiempo del test (s)</b>	<b>% de exactitud test</b>	<b>% de precisión test</b>
4	56,0	33,0	68,90%	71,38%
8	112,0	61,8	69,48%	70,38%
12	168,0	98,2	70,79%	72,00%
16	224,0	126,1	72,06%	72,99%
20	280,0	157,8	74,11%	74,93%
24	336,0	188,7	75,32%	77,35%

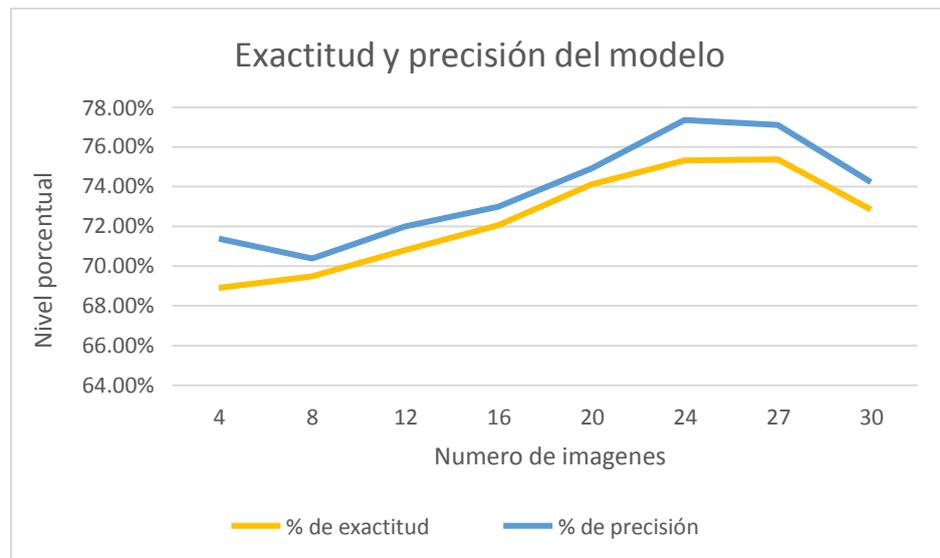
27	378,0	215,9	75,37%	77,11%
30	420,0	233,2	72,85%	74,23%

Nota. Test de estudiantes en función del número de imágenes de entrenamiento de 20 personas, Fuente: Elaboración propia

A continuación, se muestra la figura 38 donde se visualiza mejor la relación entre porcentajes de exactitud y precisión del modelo cuando este es entrenado con 20 personas.

**Figura 38**

**Exactitud y precisión del modelo Eigenface.**



Nota. La figura muestra, Indicadores de exactitud y precisión del modelo Eigenface en función del número de imágenes de entrenamiento para 20 personas, Fuente: Elaboración propia

En este caso se ha incrementado el número de personas para el entrenamiento del modelo a 20. Se puede observar que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de **exactitud** del modelo hasta llegar a un valor tope, establecido

en 75,37%, después del cual este porcentaje disminuye a pesar que sigue incrementándose el número de imágenes empleadas en el entrenamiento, lo que indica que se está produciendo a partir de aquí un caso de sobreajuste (overfitting).

En el caso del nivel de **precisión**, sucede algo curioso, inicialmente parece que el modelo no mejora en sus resultados, pero luego se logra observar un comportamiento incremental en sus aciertos, alcanzando un valor límite de 77,35%, después del cual experimenta una tendencia a disminuir en su valor produciéndose también un sobreajuste en el modelo.

**Modelo con 28 personas entrenadas**

Esta segunda tabla (tabla 7) presenta resultados para imágenes de 28 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 7**

**Resultados algoritmo Eigenface.**

imágenes en entrenamiento	tiempo de entrenamiento (s)	tiempo del test (s)	% de exactitud test	% de precisión test
4	116,4	34,3	67,70%	68,58%
8	232,8	64,1	69,59%	71,27%
12	349,2	99,5	70,44%	72,97%
16	465,6	127,8	71,44%	73,44%
20	582,0	160,8	72,02%	73,46%

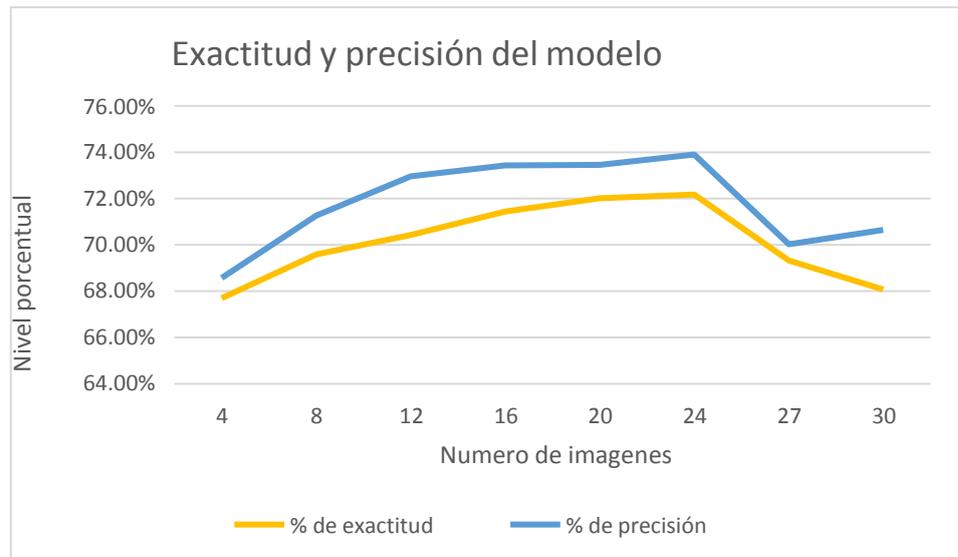
24	698,4	190,8	72,17%	73,91%
27	785,7	217,9	69,33%	70,02%
30	873,0	234,7	68,07%	70,66%

Nota. Test de estudiantes en función del número de imágenes de entrenamiento de 28 personas, Fuente: Elaboración propia

A continuación, se muestra la gráfica donde se visualiza mejor la relación entre porcentajes de exactitud y precisión del modelo cuando este es entrenado con 28 personas.

**Figura 39**

**Exactitud y precisión del modelo Eigenface.**



Nota. Indicadores de exactitud y precisión del modelo Eigenface en función del número de imágenes de entrenamiento para 28 personas, Fuente: Elaboración propia

En este caso se ha incrementado el número de personas para el entrenamiento del modelo a 28. Se aprecia que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de

**exactitud** del modelo hasta llegar a un valor tope, establecido en 72.17%, después del cual este porcentaje disminuye a pesar que sigue incrementándose el número de imágenes empleadas en el entrenamiento, lo que indica que se está produciendo a partir de aquí un caso de sobreajuste (overfitting).

En el caso del nivel de **precisión**, se logra observar un comportamiento incremental en sus aciertos, alcanzando un valor máximo de 73,91%, después del cual experimenta una tendencia a disminuir en su valor causado seguramente por el sobreajuste en el modelo.

#### 4.2.2. Resultados del algoritmo Fisherface

##### Modelo con 10 personas entrenadas

Esta primera tabla (tabla 8) presenta resultados para imágenes de 10 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 8**

##### **Resultados algoritmo Fisherface.**

imágenes en entrenamiento	tiempo de entrenamiento	tiempo de testing	% de exactitud	% de precisión
4	26,4	28,2	57,23%	62,23%
8	52,8	56,0	79,15%	87,15%
12	79,2	84,7	83,32%	88,32%
16	105,6	114,0	84,45%	91,45%
20	132,0	142,6	85,55%	90,55%
24	158,4	172,7	86,02%	95,02%

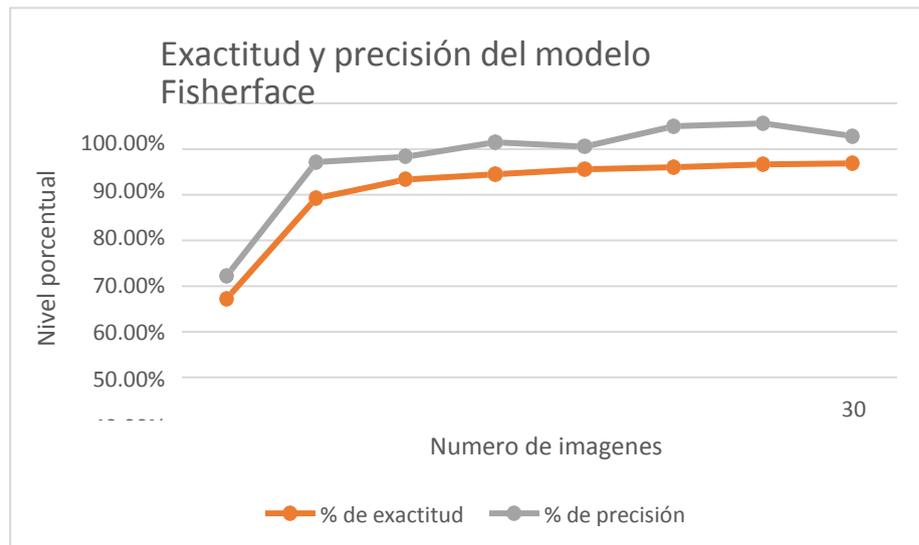
27	178,2	188,9	86,62%	95,62%
30	198,0	211,9	86,85%	92,85%

Nota. Test de estudiantes en función del número de imágenes de entrenamiento de 10 personas, Fuente: Elaboración propia

En base a esta información se construyen el grafico correspondiente donde se analizan los indicadores porcentuales de exactitud y precisión para el algoritmo Fisherface.

**Figura 40.**

**Indicadores de exactitud y precisión del modelo Fisherface.**



Nota. La figura muestra los Indicadores de exactitud y precisión del modelo Fisherface en función del número de imágenes de entrenamiento para 10 personas, Fuente: Elaboración propia

Se visualiza que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de **exactitud** del modelo, esto de forma exponencial, hasta llegar a un valor tope, establecido en 86.85%, con una tendencia a mantenerse en ese nivel a pesar del número incremental de imágenes empleadas en el entrenamiento.

En el caso del nivel de **precisión** también se logra observar un comportamiento similar al del indicador “exactitud”, con una tendencia en el crecimiento de forma exponencial alcanzando un valor límite de 95.62%, después del cual experimenta una tendencia a disminuir en su valor produciéndose un sobreajuste en el modelo.

### **Modelo con 20 personas entrenadas**

La tabla 9 presenta resultados para imágenes de 20 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 9**

#### **Resultados algoritmo Fisherface.**

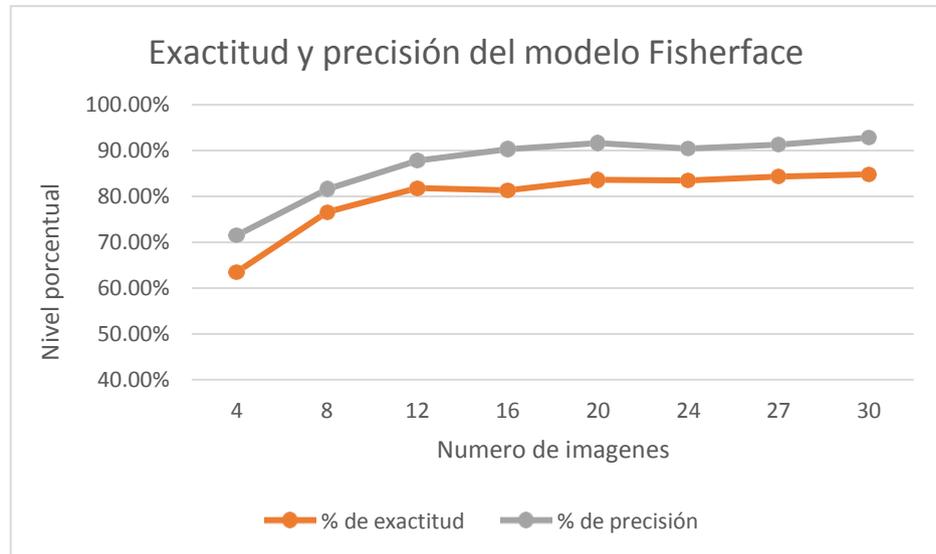
<b>imágenes en entrenamien to</b>	<b>tiempo de entrenamien to</b>	<b>tiempo de testing</b>	<b>% de exactitud</b>	<b>% de precisión</b>
4	53,6	27,9	63,53%	71,53%
8	107,2	55,3	76,59%	81,59%
12	160,8	84,0	81,83%	87,83%
16	214,4	112,6	81,29%	90,29%
20	268,0	141,2	83,64%	91,64%
24	321,6	171,0	83,46%	90,46%
27	361,8	186,9	84,30%	91,30%
30	402,0	210,4	84,82%	92,83%

Nota. Test de estudiantes en función del número de imágenes de

entrenamiento de 20 personas, Fuente: Elaboración propia

**Figura 41**

**Indicadores de exactitud y precisión del modelo Fisherface.**



Nota. La imagen muestra los Indicadores de exactitud y precisión del modelo Fisherface en función del número de imágenes de entrenamiento para 20 personas Fuente: Elaboración propia

Para este escenario se ha incrementado el número de personas para el entrenamiento a 20. Se observa que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de **exactitud** del modelo en una proporción exponencial, similar al escenario anterior, hasta un máximo de 84.82%, con una tendencia a estabilizarse alrededor de este valor porcentual.

En el caso del nivel de **precisión**, sucede del mismo modo, de manera exponencial crece su nivel hasta llegar a un valor de 92.83%, con una tendencia a mantenerse alrededor de este valor.

**Modelo con 28 personas entrenadas**

Esta tabla (tabla 10) presenta resultados para imágenes de 28 personas en

entrenamiento y 10 estudiantes testeados con el algoritmo Fisherface.

**Tabla 10**

**Resultados algoritmo Fisherface para test de estudiantes.**

imágenes en entrenamiento	tiempo de entrenamien to	tiempo de testing	% de exactitud	% de precisión
4	78,4	28,9	56,81%	65,81%
8	156,8	54,3	70,34%	79,34%
12	235,2	83,2	74,75%	83,75%
16	313,6	111,9	75,04%	84,04%
20	392,0	140,2	76,14%	82,14%
24	470,4	172,0	78,88%	84,88%
27	529,2	187,1	76,38%	81,38%
30	588,0	211,6	77,74%	82,74%

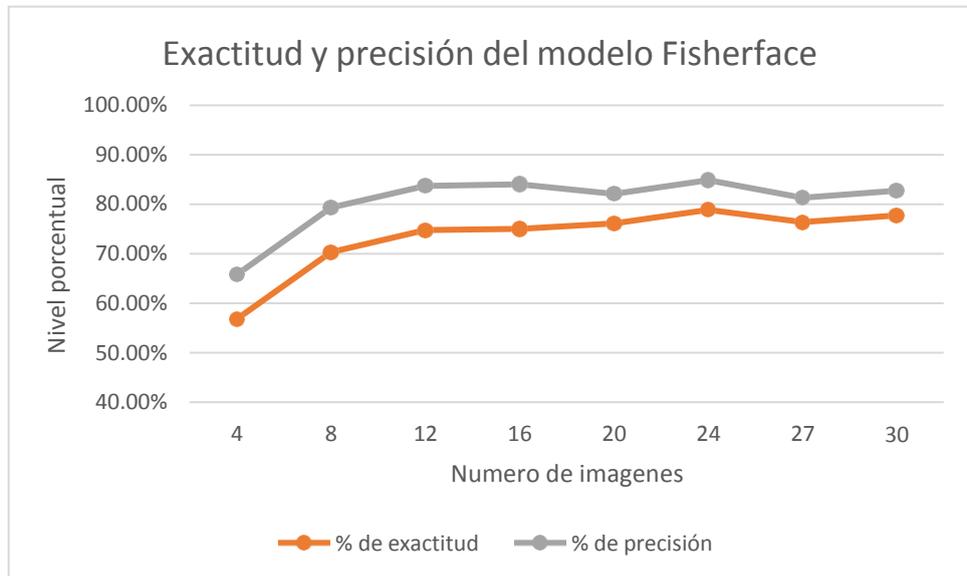
Nota. Resultados algoritmo Fisherface para test de estudiantes en función del número de imágenes de entrenamiento de 28 personas,

Fuente: Elaboración propia

A continuación, se muestra la gráfica con la relación entre porcentajes de exactitud y precisión del modelo cuando este es entrenado con 28 personas.

**Figura 42**

**Indicadores de exactitud y precisión del modelo Fisherface.**



Nota. Indicadores de exactitud y precisión del modelo Fisherface en función del número de imágenes de entrenamiento para 28 personas,

Fuente: Elaboración propia

Se observa que a medida que aumenta el número de imágenes en el entrenamiento, incrementa exponencialmente, de forma no muy pronunciada, marcando el porcentaje de **exactitud** del modelo un valor tope de 77.74%, establecido en 72.17%, mostrando una tendencia a estabilizarse en este nivel.

En el caso del nivel de **precisión**, se observa un comportamiento incremental exponencial similar al indicador anterior, llega a tomar un valor de 82.74, con tendencia estable. Cabe mencionar que estos valores obtenidos para este escenario resultan siendo menores a los dos escenarios anteriores para este modelo.

**4.2.3. Resultados del algoritmo LBP (Local Binary Pattern)**

**Modelo con 10 personas entrenadas**

La tabla 11 presenta resultados al aplicar el algoritmo LBP (Patrón Binario Local en español) para imágenes incrementales de 10 personas en entrenamiento y 10 estudiantes testeados.

**Tabla 11**

**Resultados algoritmo LBP para test de estudiantes**

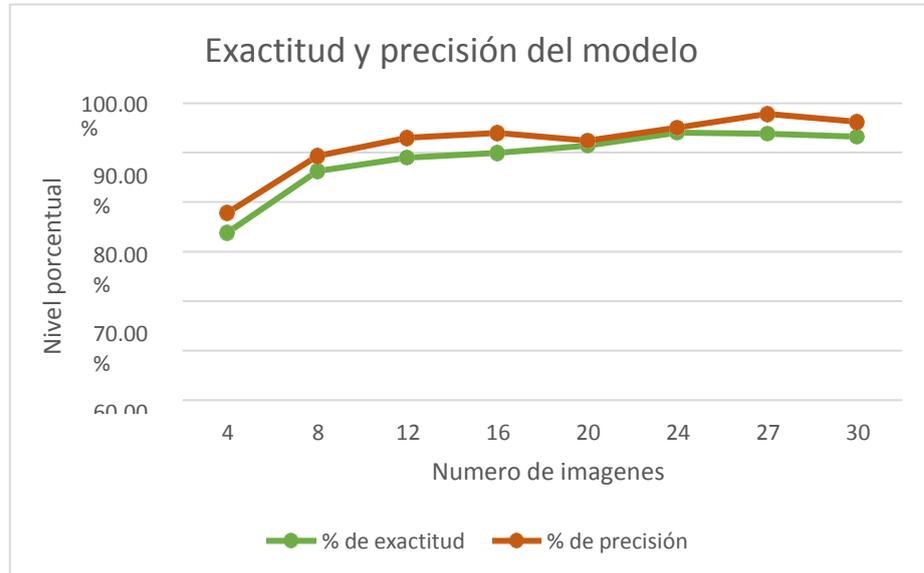
<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento</b>	<b>tiempo de testing</b>	<b>% de exactitud</b>	<b>% de precisión</b>
4	27,6	29,8	73,86%	77,86%
8	55,2	60,2	86,29%	89,29%
12	82,8	87,8	88,99%	92,99%
16	110,4	115,9	89,98%	93,98%
20	138,0	150,4	91,42%	92,42%
24	165,6	177,2	94,05%	95,05%
27	186,3	203,1	93,83%	97,83%
30	207,0	221,5	93,24%	96,24%

Nota. Resultados algoritmo LBP para test de estudiantes en función del número de imágenes de entrenamiento de 10 personas, Fuente: Elaboración propia

Partiendo de esta tabla con información se construyen el gráfico correspondiente donde se analizan los indicadores porcentuales de exactitud y precisión para el algoritmo LBP.

**Figura 43**

**Indicadores de exactitud y precisión del modelo LPB en función del número de imagen de entrenamiento para 10 personas**



Nota: En la figura se muestra la una grafica de la exactitud y precisión de LBP en función de un numero de imágenes de entramiento, Fuente: Elaboración propia

Se visualiza que a medida que aumenta el número de imágenes en el entrenamiento, aumenta el porcentaje de **exactitud** del modelo, de forma constante, hasta llegar a un valor máximo de 93.83%, con una tendencia a mantenerse cerca a ese valor, aunque se incremente el número de imágenes empleadas en el entrenamiento.

En el caso del nivel de **precisión** también se logra observar un comportamiento parecido al del indicador “exactitud”, con una tendencia en el crecimiento de forma sostenida alcanzando un valor límite de 97.83%, después del cual experimenta una ligera disminución

### **Modelo con 20 personas entrenadas**

La tabla 12 presenta resultados para imágenes en entrenamiento de 20 personas e imágenes de 10 estudiantes testeados.

**Tabla 12.**

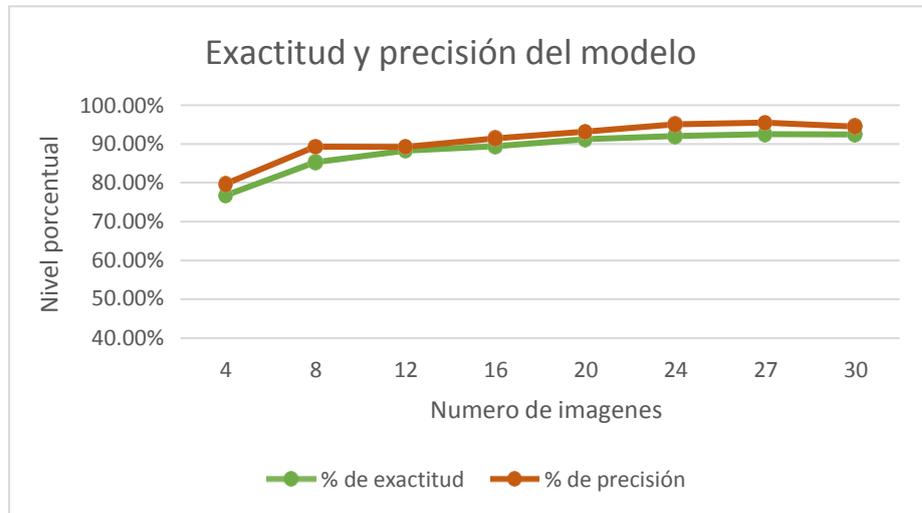
**Resultados algoritmo LBP para test de estudiantes.**

<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento</b>	<b>tiempo de testing</b>	<b>% de exactitud</b>	<b>% de precisión</b>
4	51,2	29,0	76,72%	79,72%
8	102,4	59,8	85,29%	89,29%
12	153,6	86,2	88,22%	89,22%
16	204,8	115,9	89,43%	91,43%
20	256,0	148,3	91,20%	93,20%
24	307,2	176,0	92,05%	95,05%
27	345,6	202,8	92,49%	95,49%
30	384,0	220,0	92,46%	94,46%

Nota. Resultados algoritmo LBP para test de estudiantes en función del número de imágenes de entrenamiento de 20 personas, Fuente: Elaboración propia

**Figura 44**

**Indicadores de exactitud y precisión del modelo LBP.**



Nota: En la figura se muestra la una gráfica de la exactitud y precisión de LBP en función del número de imágenes de entrenamiento para 20 personas, Fuente: Elaboración propia

Para este escenario se ha incrementado el número de personas para el entrenamiento a 20. Se observa que a medida que aumenta el número de imágenes en el entrenamiento, incrementa el porcentaje de **exactitud** del modelo llegando a un valor de 92.49%, con una tendencia a estabilizarse alrededor de este valor porcentual.

En el caso del nivel de **precisión**, se da una situación similar, crece su nivel hasta llegar a valer de 95.49%, con una tendencia a mantenerse alrededor de este valor.

**Modelo con 28 personas entrenadas**

Esta tabla 13 presenta resultados para imágenes de 28 personas en entrenamiento y de 10 estudiantes testeados con el algoritmo LBP.

**Tabla 13**

**Resultados algoritmo LBP para test de estudiantes.**

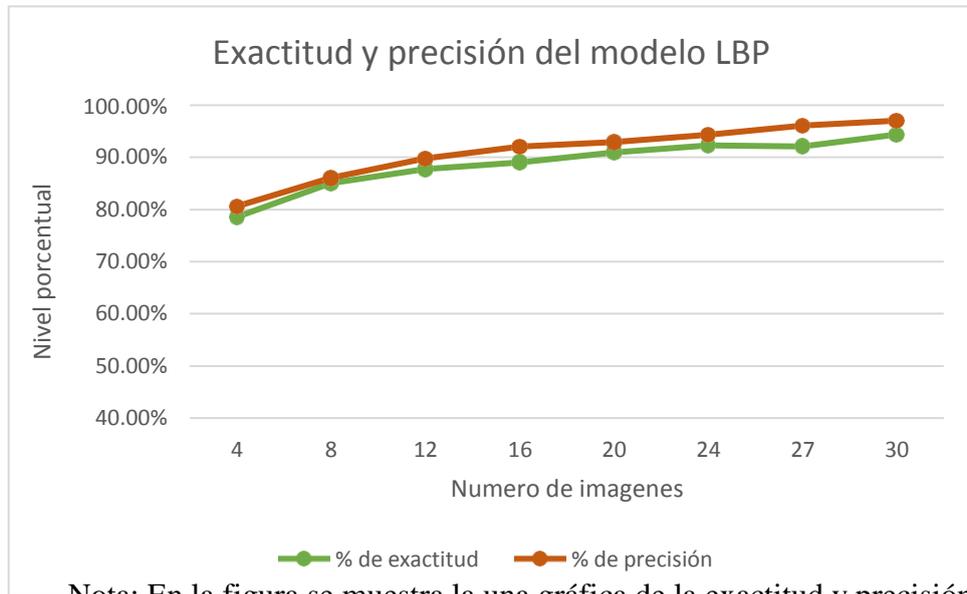
<b>imágenes en entrenamiento</b>	<b>tiempo de entrenamiento</b>	<b>tiempo de testing</b>	<b>% de exactitud</b>	<b>% de precisión</b>
4	77,7	30,1	78,57%	80,57%
8	155,4	62,4	85,05%	86,05%
12	233,2	88,0	87,75%	89,75%
16	310,9	116,3	89,03%	92,03%
20	388,6	151,0	90,92%	92,92%
24	466,3	179,0	92,30%	94,30%
27	524,6	202,6	92,07%	96,07%
30	582,9	219,4	94,37%	97,02%

Nota. Resultados algoritmo LBP para test de estudiantes en función del número de imágenes de entrenamiento de 28 personas, Fuente: Elaboración propia

A continuación, se muestra la gráfica con la relación entre porcentajes de exactitud y precisión del modelo cuando este es entrenado con 28 personas para el algoritmo LBP.

**Figura 45**

**Indicadores de exactitud y precisión del modelo LBP.**



Nota: En la figura se muestra la una gráfica de la exactitud y precisión de LBP en función del número de imágenes de entrenamiento para 28 personas, Fuente: Elaboración propia

Se observa que a medida que aumenta el número de imágenes en el entrenamiento, incrementa de manera constante, sobre todo al inicio de las pruebas el porcentaje de **exactitud** del modelo, llegando a un valor máximo de 94.37%, , mostrando una tendencia a posiblemente incrementar.

En el caso del nivel de **precisión**, se observa un comportamiento similar al indicador anterior, llega a tener un valor final de 97.02%, con tendencia incremental.

**4.3. Prueba de hipótesis.**

A continuación, se pasa a contrastar las hipótesis de la investigación, para ello se procede a mencionar una a una y establecer su validez.

**4.3.1. Hipótesis específica 1.**

La primera hipótesis de investigación:

“Evaluando algoritmos de reconocimiento facial en aprendizaje automático se determinará aquella con mejor resultado de exactitud en la identificación de personas”

Para determinar aquel algoritmo que brinda mejor resultado de exactitud se evaluó el rendimiento en la métrica de exactitud para cada algoritmo, los resultados se muestran en las tablas del apartado anterior en la columna % de exactitud, valores que se obtuvieron a través de la fórmula planteada en el capítulo 2:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Donde la interpretación de los términos en el reconocimiento facial sería:

TP: El algoritmo identifica positivamente la imagen de la persona en la base de datos.

FP El algoritmo identifica positivamente la imagen de la persona que no le corresponde en la base de datos.

TN: El algoritmo identifica negativamente la imagen de la persona porque no existe en la base de datos.

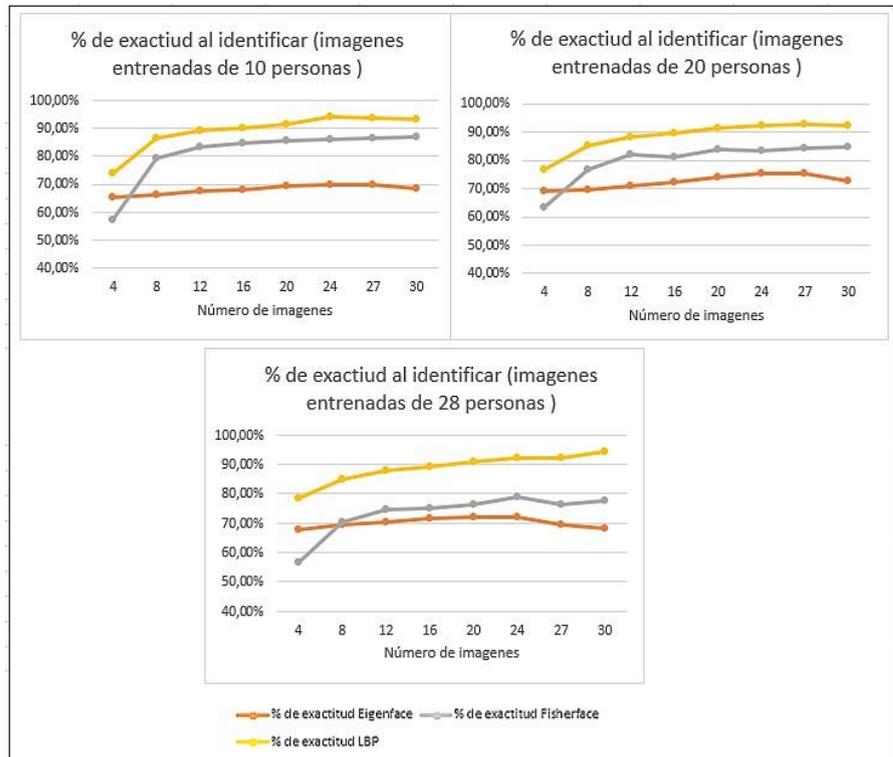
FN El algoritmo identifica negativamente la imagen de la persona a pesar que existe en la base de datos.

A continuación, se muestra el gráfico comparativo de los tres algoritmos Eigenface, Fisherface y LBP, para esta métrica, y en base a ello indicar aquel que mejor resultado muestra.

**Figura 46**

**Comparativa de las métricas de exactitud de los algoritmos**

**Eigenface, Fisherface y LBP.**



Nota: En la figura se muestra la comparativa de las métricas de exactitud para identificar personas en relación a un número de imágenes, Fuente: Elaboración propia

Se observa en la gráfica que en los tres segmentos del testeo (10, 20 y 28 personas) de las imágenes de estudiantes para el reconocimiento facial con los tres algoritmos programados: Eigenface, Fisherface y LBP; el que mejor ratio de exactitud muestra es el algoritmo LBP (trazo de color amarillo), con un valor máximo de exactitud del 94.37%, que indica que de cada 100 imágenes de rostros evaluados el algoritmo acierta en identificar positiva y negativamente a 94 de ellas, es decir aquellos que si están en la base de datos y aquellos que no existen en la base de datos. Por lo que se

puede concluir que con la evaluación de algoritmos de reconocimiento facial se logra determinar que el algoritmo que mejor resultado de exactitud arroja es el LBP (Local BinaryPattern)

#### 4.3.2. Hipótesis específica 2

La segunda hipótesis de investigación:

“Evaluando algoritmos de reconocimiento facial en aprendizaje automático se determinará aquella con mejor resultado de precisión en la identificación de personas”.

Para determinar aquel algoritmo que brinda mejor resultado de precisión se evaluó el rendimiento con la métrica de precisión para cada algoritmo, los resultados se muestran en las tablas del apartado anterior en la columna “% de precisión”, valores que se obtuvieron a través de la fórmula planteada en el capítulo 2:

$$precision = \frac{TP}{TP + FP}$$

Donde la interpretación de los términos en el reconocimiento facial sería:

TP: El algoritmo identifica positivamente la imagen de la persona en la base de datos.

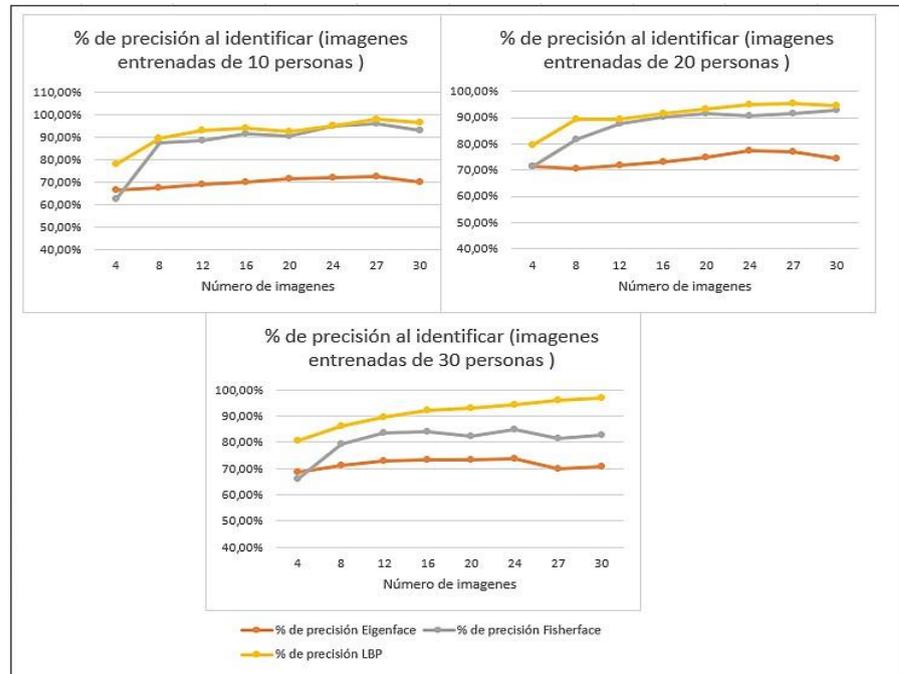
FP: El algoritmo identifica positivamente la imagen de la persona que no le corresponde en la base de datos.

A continuación, se muestra el gráfico comparativo de los tres algoritmos Eigenface, Fisherface y LBP, para esta métrica.

**Figura 47**

**Comparativa de las métricas de precisión de los algoritmos**

**Eigenface, Fisherface y LBP.**



Nota: En la figura se muestra la comparativa de las métricas de precisión para identificar personas en relación a un número de imágenes, Fuente: Elaboración propia

Se observa en la gráfica que en los tres segmentos del testeo (10, 20 y 28 personas) de las imágenes de estudiantes para el reconocimiento facial con los tres algoritmos programados: Eigenface, Fisherface y LBP; que la mejor ratio de precisión es del algoritmo LBP (trazo de color amarillo), sobre todo cuando el número de personas en entrenamiento no es muy amplio, este algoritmo logra un nivel de precisión máximo del 97.83%, es decir que de cada 100 imágenes que el algoritmo identifica 97 de ellas son identificaciones positivas y 3 son identificaciones negativas. Sin embargo, también es destacable mencionar que para un mayor número de imágenes

de entrenamiento el algoritmo Fisherface se acerca bastante al nivel porcentual de precisión alcanzado del algoritmo LBP. Por lo que se puede concluir que en la evaluación de algoritmos de reconocimiento facial se logra determinar que el algoritmo que mejor resultado en precisión arroja es el LBP, seguido a continuación por el algoritmo Fisherface.

#### **4.4. Discusión de resultados**

En resumidas cuentas, se puede observar que el algoritmo LBP es el que mejores resultados ha mostrado en la comparación hecha, por consiguiente, es el más confiable, tanto en exactitud como en precisión. Sin embargo, dentro del ratio de precisión el algoritmo Fisherface arroja resultados cercanos a los que brinda el algoritmo LBP, existiendo una diferencia de poco más de 2 puntos porcentuales. Resultado que es similar a la investigación hecha por Espinoza, D. y Jorquera, P. (2015) donde indica que los algoritmos LBP y Fisherface son los más eficientes.

En cuanto a la relación entre el porcentaje de exactitud y precisión en la identificación de personas versus la cantidad de fotos de entrenamiento empleadas se observa que a medida que más imágenes se entrenan se reducen los niveles porcentuales de las métricas de exactitud y precisión, sobre todo cuando se pasa de entrenar con 27 imágenes a 30 por persona, existiendo en varios casos sobreajuste, contrario a lo que manifiesta Castaño, D. y Alonso, J. (2019) que indica que se obtendrá un mejor reconocimiento facial mientras más imágenes se empleen en el entrenamiento.

## CONCLUSIONES

De la investigación se llega las siguientes conclusiones:

1. El trabajo se basó en la realización de un experimento, en el que se evaluó tres algoritmos de reconocimiento facial mediante aprendizaje automático: Eigenface, Fisherface y Local Binary Pattern (LBP), para determinar cuál de ellos ofrece una mejor métrica de exactitud y precisión al identificar personas, habiéndose entrenado previamente cada algoritmo con cantidades incrementales de imágenes; los resultados indicaron que en cuanto a exactitud el algoritmo Eigenface alcanzó un máximo de 75.32%, mientras que Fisherface llegó a 86.85% y por último el algoritmo LBP tomó un valor pico de 94.37%.
2. Estos porcentajes obtenidos indican que el algoritmo LBP es el que mejor resultado de exactitud logra en la identificación de persona, con una ventaja considerable sobre los otros dos algoritmos evaluados, en ese sentido se considera que se ha validado la primera hipótesis específica y cumplido con el primer objetivo específico de investigación.
3. En cuanto a la métrica de precisión el algoritmo Eigenface obtuvo un máximo de 77.35%, mientras que Fisherface llegó a 95.02% y por último el algoritmo LBP alcanzó un valor pico de 97.83%. Estos niveles porcentuales muestran que también el algoritmo LBP obtiene los mejores resultados en la identificación de persona, por lo que se considera que se ha validado la segunda hipótesis específica y cumplido con el segundo objetivo específico de investigación.
4. Se observa en algunas de las pruebas realizadas a los algoritmos de reconocimiento facial que cuando mayor número de imágenes de entrenamiento se emplean para ajustar el modelo, estas llegan a sufrir un sobreajuste, conocido también como overfitting, generalmente cuando se pasa de 27 a 30 imágenes de entrenamiento.

## RECOMENDACIONES

1. Se sugiere ampliar la investigación a analizar otros algoritmos como los relacionados con las redes neuronales, con el fin optimizar el proceso de reconocimiento facial que permita implementar soluciones alternas a la propuesta en esta tesis.
2. Se recomienda probar los algoritmos con un número mayor de imágenes de entrenamiento para observar cómo se comporta el sobreajuste en cada algoritmo de reconocimiento facial y establecer si los aparentes sobreajustes encontrados son solo distorsiones o picos anómalos en el comportamiento de la identificación de personas.
3. Hoy en día, se pueden emplear una gran variedad de herramientas de programación y de edición de código en lenguaje especializado, como Python para este tipo de temas, por lo que se sugiere implementar en la Curricula de la Escuela de Ingeniería de Sistemas y Computación para poder estar a la par con otras instituciones educativas que investigan sobre visión por computadora y en especial para el reconocimiento de imágenes de personas.

## REFERENCIAS BIBLIOGRAFICAS

### Físico

Balestrini, Miriam. (2006). Cómo se elabora el Proyecto de investigación.

Editorial Consultores Asociados.

Georghiades, A., Belhumeur, P. and Kriegman, D. (1999). Illumination-Based

Image Synthesis: Creating Novel Images of Human Faces under differing pose and lighting, Proc. IEEE Workshop Multi-View Modeling and Analysis of Visual Scenes, pp. 47-54, 1999

Hernández S. Roberto, Fernández C. Carlos y Baptista L. María del Pilar (2010).

Metodología de la Investigación. McGraw-Hill/Interamericana Editores, S.A. de C.V.

Kenneth E. Kendall. (2005). Análisis y diseño de sistemas (6.a ed.). México:

Pearson Education.

Myna A.N., Deepthi, K. y Prathibba, G. (2021). Classification of photographs

based on face recognition. International Journal Advanced Research Engineering a Technology. 12(3), 732-740. DOI: 10.34218/IJARET.12.3.2021.0

Li, Lixian & Mu, Xiaohui & Li, Siying & Haipeng, Peng. (2020). A Review of

Face Recognition Technology. IEEE Access. PP. 1-1. DOI: 10.1109/ACCESS.2020.3011028.

Viola, P. y Jones, M. (2004). Robust real-time face detection. International Journal

of Computer Vision, 57(2):137–154.

Yang, M., Kriegman D., Ahuja N. (2002). Detecting Faces in Images: A Survey.

Transactions on pattern analysis and Machine Intelligence, 24(1), 34-58.

## Electrónico

Aws. (s.f.). Amazon machine Learning. Guia para desarrolladores. Recuperado el

05 de marzo de 2022. [https://docs.aws.amazon.com/es\\_es/machine-learning/latest/dg/training-ml-models.html](https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/training-ml-models.html)

Castaño, D. y Alonso, J. (2019), Sistema de reconocimiento facial para controlar el acceso a viviendas. [Tesis de pregrado. Universidad Católica de Colombia].

Repositorio

Institucional.

<https://repository.ucatolica.edu.co/bitstream/10983/24032/1/Final%20Trabajo%20de%20grado.pdf>

Cadena, J., Flores, G., Villa, M., Chancusig, J. y Guaypatín, O. (2018).

Reconocimiento facial en sub-espacios: lineales y no-lineales, bases de datos de rostros y máquina de vectores de soporte.

<http://arje.bc.uc.edu.ve/arje22e/art41.pdf>

Cernánides, H. y Zapata, E. (2006). Identificación de personas mediante el reconocimiento dactilar y su aplicación a seguridad organizacional. [Tesis de pregrado. Universidad Mayor de San Marcos]. Repositorio Institucional.

[https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/3233/Cernandez\\_gh.pdf?sequence=1&isAllowed=y](https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/3233/Cernandez_gh.pdf?sequence=1&isAllowed=y)

Çarıkçı, M. y Özen, F. (2012). A Face Recognition System Based on Eigenface Method.

<https://www.sciencedirect.com/science/article/pii/S2212017312000242>

Dwidevi, D. (2018, abril). Reconocimiento facial para principiantes. Towards

Data Science. <https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2>

Eficiencia. (10 de octubre de 2021). En

Wikipedia.<https://es.wikipedia.org/wiki/Eficiencia>

El peruano. (2021). Ciberdelitos en el Perú: se elevan denuncias de fraude informático y suplantación de identidad.

<https://elperuano.pe/noticia/121876-ciberdelitos-en-el-peru-se-elevan-denuncias-de-fraude-informatico-y-suplantacion-de-identidad>

Espinoza, D. y Jorquera, P. (2015). Reconocimiento facial. [Tesis de pregrado.

Pontificia Universidad Católica de Valparaíso, Chile]. Repositorio

Institucional. [http://opac.pucv.cl/pucv\\_txt/txt-1000/UCD1453\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-1000/UCD1453_01.pdf)

Fotograma. (10 de octubre de 2021). En

Wikipedia.<https://es.wikipedia.org/wiki/Fotograma>

García, M. (2009). Diseño e implementación de una herramienta de detección facial.

[Tesis de maestría. Instituto Politécnico Nacional, México]. Repositorio

Institucional. <https://tesis.ipn.mx/jspui/handle/123456789/25106>

Garcial-Gil, D., Leungo, J., Garcia, S. y Herrera, F. (s.f.). Smart Data: Filtrado de

Ruido para Big Data.

<https://sci2s.ugr.es/sites/default/files/bbvasoftware/publications/CAEPIA-18-Ruido.pdf>

Gimeno, C. (2010). Estudio de técnicas de reconocimiento facial.

[https://upcommons.upc.edu/bitstream/handle/2099.1/9782/PFC\\_RogerGimeno.pdf](https://upcommons.upc.edu/bitstream/handle/2099.1/9782/PFC_RogerGimeno.pdf)

Juárez, C. y Serrano, J. (s.f.). Clasificación de

imágenes.<http://www.it.uc3m.es/jvillena/irc/practicas/09-10/05mem.pdf>

- IArtificial.net. (2020, 09 de octubre). Precision, Recall, F1, Accuracy en clasificación. <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>
- IBM. (2020, 15 de julio). Machine Learning. <https://www.ibm.com/cloud/learn/machine-learning>
- La Madrid, D. y Barriga, M. (2019). Modelo tecnológico de reconocimiento facial para la identificación de pacientes en el sector salud. [Tesis de pregrado. Universidad Peruana de Ciencias Aplicadas]. Repositorio Institucional. [https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/648832/La%20Madrid\\_ad.pdf?sequence=3&isAllowed=y](https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/648832/La%20Madrid_ad.pdf?sequence=3&isAllowed=y)
- Mery, D. (01 de abril de 2020). Características de intensidad, LBP (Local Binary Patterns) y reconocimiento facial. YouTube. <https://www.youtube.com/watch?v=7woR1nDc9r0>
- Morales, A. (2015). Uso de características no lineales para identificar llantos de recién nacidos con un conjunto clasificador. [Tesis de pregrado. Universidad de las Américas Puebla, México]. Repositorio Institucional. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lmt/morales\\_s\\_aa/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/morales_s_aa/capitulo3.pdf)
- Na8. (2017, diciembre). Qué es overfitting y underfitting y cómo solucionarlo. Aprende Machine Learning. <https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>
- Rahmad, C., Asmara, R., Putra, D., Gharma, I., Darmono, H. y Muhiqqin, I. (2020). Comparison of Viola-Jones Haar Cascade Classifier and Histogram of

Oriented Gradients (HOG) for face detection. IOP Conf. Ser.: Mater. Sci. Eng. <https://iopscience.iop.org/article/10.1088/1757-899X/732/1/012038>

Ramon, E (2011). Transformaciones basadas en el algoritmo Local BinaryPattern de imágenes capturadas con la Kinect para clasificación facial. [https://www.researchgate.net/publication/277155032\\_Transformaciones\\_basadas\\_en\\_el\\_algoritmo\\_Local\\_Binary\\_Pattern\\_de\\_imagenes\\_capturadas\\_con\\_la\\_Kinect\\_para\\_clasificacion\\_facial](https://www.researchgate.net/publication/277155032_Transformaciones_basadas_en_el_algoritmo_Local_Binary_Pattern_de_imagenes_capturadas_con_la_Kinect_para_clasificacion_facial)

Real Academia Española. (2020). Diccionario online. <https://dle.rae.es/venta>

Real Academia Española. (s.f.). Identidad. En Diccionario de la lengua española. Recuperado el 18 de octubre de 2021, de <https://dle.rae.es/identidad>

Rodrigo, Isaac & Cayetano, Pérez & Alejandro, José & Preciado, Jiménez & Antonio-Velázquez, Juan & González, Erika. (2021). Detección automática de rostros con cubre boca o sin cubre boca para restringir el acceso a institución educativa.

[https://www.researchgate.net/publication/351561791\\_Deteccion\\_automat\\_ica\\_de\\_rostros\\_con\\_cubre\\_boca\\_o\\_sin\\_cubre\\_boca\\_para\\_restringir\\_el\\_acceso\\_a\\_institucion\\_educativa](https://www.researchgate.net/publication/351561791_Deteccion_automat_ica_de_rostros_con_cubre_boca_o_sin_cubre_boca_para_restringir_el_acceso_a_institucion_educativa)

Rosebrock, A. (10 de mayo de 2021). OpenCV Eigenface para reconocimiento facial. <https://pyimagesearch.com/2021/05/10/opencv-Eigenface-for-face-recognition/>

Sady. (02 de julio de 2016). ¿Cuál es la diferencia entre reconocimiento, verificación e identificación? Tecnología biométrica. Blog. <http://tecbiometric.blogspot.com/2016/07/cual-es-la-diferencia-entre.html>

Sandipan, D. (2018). Hands-On Image Processing with

Python:

[https://subscription.packtpub.com/book/big\\_data\\_and\\_business\\_intelligence/9781789343731/1/ch01lv1sec11/what-is-image-processing-and-some-applications](https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789343731/1/ch01lv1sec11/what-is-image-processing-and-some-applications)

Bhattacharyya, S. y Rahul, K. (2013). Face recognition by linear discriminant análisis.

[https://www.researchgate.net/profile/Kumar-Rahul/4/publication/285683251\\_Face\\_recognition\\_by\\_linear\\_discriminant\\_analysis/links/5a966dafaca2721405696472/Face-recognition-by-linear-discriminant-analysis.pdf?origin=publication\\_detail](https://www.researchgate.net/profile/Kumar-Rahul/4/publication/285683251_Face_recognition_by_linear_discriminant_analysis/links/5a966dafaca2721405696472/Face-recognition-by-linear-discriminant-analysis.pdf?origin=publication_detail)

Szeliski, R. (2022). Computer Vision: Algorithms and Applications. <https://szeliski.org/Book>

Yashquero. (s.f.). Face Recognition using Fisherface.

Consultado el 10/03/2022. <https://iq.opengenus.org/face-recognition-using-Fisherface/>

Villegas, C. (2005). Reconocimiento de rostros utilizando análisis

de componentes principales: limitaciones del algoritmo. Tesis de maestría.

Universidad Iberoamericana, México]. Repositorio

Institucional. <http://ri.iberomex.mx/handle/iberomex/904>

# **ANEXOS**

**Anexo 1: Matriz de Consistència.**

**“Algoritmos de reconocimiento facial mediante aprendizaje automático para la identificación de personas en una institución educativa de Pasco - 2021”**

<b>PROBLEMA</b>	<b>OBJETIVOS</b>	<b>HIPÓTESIS</b>	<b>VARIABLE E INDICADORES</b>	<b>METODOLOGÍA</b>	<b>TÉCNICAS E INSTRUMENTOS</b>
<p><b><u>PROBLEMA</u></b></p> <p><b><u>GENERAL:</u></b></p> <p>¿Qué se consigue evaluando algoritmos de reconocimiento facial mediante aprendizaje automático en la identificación de personas de una institución educativa de</p>	<p><b><u>OBJETIVO</u></b></p> <p><b><u>GENERAL:</u></b></p> <p>Determinar el algoritmo de reconocimiento facial que mejor resulte en la identificación de personas de una institución educativa de Pasco - 2021.</p>	<p><b><u>HIPOTESIS</u></b></p> <p><b><u>GENERAL:</u></b></p> <p>Evaluando algoritmos de reconocimiento facial de aprendizaje automático se determinará aquella que mejor resulte en la identificación de personas de una</p>	<p>V.I.</p> <p>Algoritmos de Reconocimiento facial</p> <p><u>Indicador</u></p> <p>Algoritmos de reconocimiento facial implementado</p>	<p><b><u>Tipo y nivel de Investigación.</u></b></p> <p>Aplicada, descriptiva y explicativa</p> <p><b><u>Métodos de la Investigación</u></b> Hipotético deductivo <b><u>Población y Muestra</u></b></p> <p>Se toma 10 estudiantes de la Facultad de Ingeniería</p>	<p><b><u>Técnicas:</u></b></p> <p>La observación experimental para elaborar datos de las variables en estudio</p> <p>El Análisis Documental de las fuentes bibliográficas correspondientes.</p>

<p>Pasco- 2021?</p> <p><b><u>PROBLEMAS</u></b></p> <p><b><u>ESPECIFICO:</u></b></p> <p>¿Qué se consigue evaluando algoritmos de reconocimiento facial en aprendizaje automático en relación a la exactitud en la identificación de personas?</p> <p>¿Qué se consigue evaluando algoritmos de reconocimiento facial en aprendizaje automático en relación a la precisión en la identificación de</p>	<p><b><u>OBJETIVOS</u></b></p> <p><b><u>ESPECIFICOS:</u></b></p> <p>Determinar el algoritmo de reconocimiento facial en aprendizaje automático con mayor exactitud en la identificación de personas</p> <p>Determinar el algoritmo de reconocimiento facial en aprendizaje</p>	<p>institución educativa de Pasco - 2021.</p> <p><b><u>HIPOTESIS</u></b></p> <p><b><u>ESPECIFICAS:</u></b></p> <p>Evaluando algoritmos de reconocimiento facial en aprendizaje automático se determinará aquella con mejor resultado de exactitud en la identificación de personas.</p> <p>Evaluando algoritmos de reconocimiento facial en aprendizaje automático se</p>	<p>V.D.</p> <p>Identificación de personas</p> <p><b><u>Indicador</u></b></p> <p>Exactitud en la identificación de personas</p> <p>Precisión en la identificación de personas</p>	<p>UNDAC, al azar de quienes se obtuvo 31 imágenes faciales para la investigación. Siendo en total 310 imágenes para la muestra de estudio.</p>	<p>Captura de imágenes</p> <p>Métricas de reconocimiento facial</p> <p><b>Instrumentos:</b></p> <hr/> <p>En base a la lectura</p> <p>Textos</p> <p>Cámara fotográfica y de video digital</p> <p>Base de datos de imágenes</p> <p>Software de programación</p>
---	--	---	--	---	---

personas?	automático con mayor precisión en la identificación de personas.	determinará aquella con mejor resultado de precisión en la identificación de personas.			
-----------	--	--	--	--	--




Leyenda

TP: El algoritmo identifica positivamente la imagen de la persona en la base de datos.

FP El algoritmo identifica positivamente la imagen de la persona que no le corresponde en la base de datos.

TN: El algoritmo identifica negativamente la imagen de la persona porque no existe en la base de datos.

FN El algoritmo identifica negativamente la imagen de la persona a pesar que existe en la base de datos.

### Anexo 3:

#### Programa: Algoritmos de reconocimiento facial

```
import numpy as np
from PIL import Image
import cv2
import pickle
from time import time

# se inicia el temporizador
tiempo_inicial = time()

# se selecciona el directorio de la base de datos
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, "ExtendedYaleB")

# se escoge el detector facial de la carpeta "Cascades"
face_cascade =
cv2.CascadeClassifier('Cascades/data/haarcascade_frontalface_alt2.xml')
# se crea el algoritmo de reconocimiento facial
recognizer = cv2.face.EigenFaceRecognizer_create()

# se inician las variables
current_id = 0
label_ids = {} # se crea un diccionario vacio
y_labels = []
x_train = []

# bucle para todas las imágenes del directorio seleccionadas
for root, dirs, files in os.walk(image_dir):
    for file in files[1:5]:
        if file.endswith("pgm"):
            path = os.path.join(root, file)
            label = os.path.basename(root).replace(" ", "-").lower()
            print(label, path)
            if label not in label_ids:
                label_ids[label] = current_id
                current_id += 1
            id_ = label_ids[label]
            print(label_ids)
```

Continuación...

```
        faces = face_cascade.detectMultiScale(image_array, scaleFactor=1.3,
minNeighbors=4) # detección de caras

        for (x, y, w, h) in faces:
            region_of_interest_gray = cv2.resize((image_array[y: y+h, x:
x+w]), (260, 260)) # región de interés de la imagen
            x_train.append(region_of_interest_gray) # se crea un array con
las matrices de valores de cada imagen
            y_labels.append(id_) # se crea un array con las id de cada
imagen -> [0, 1, 2, etc]

# escribe la representación serializada de "labels_ids" en el archivo file (f)
que está abierto
with open("labels.pickle", 'wb') as f:
    pickle.dump(label_ids, f)

# training con los datos
recognizer.train(x_train, np.array(y_labels))
# se guardan los datos del training en el archivo correspondiente
recognizer.save("trainer_EigenFaces.yml")

# fin del temporizador
tiempo_final = time()
tiempo_ejecucion = tiempo_final - tiempo_inicial
print("El tiempo de ejecución es de: ", tiempo_ejecucion)

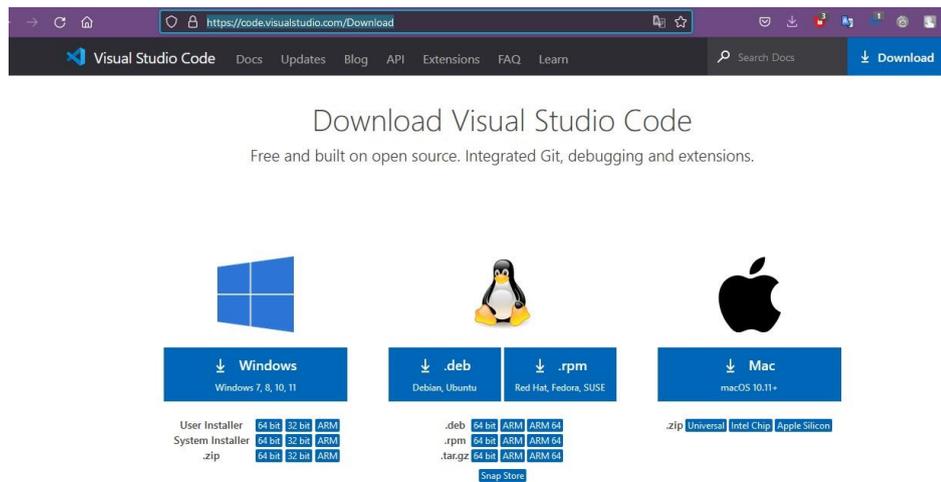
# bucle para todas las imágenes del directorio seleccionadas
for root, dirs, files in os.walk(image_dir):
    for file in files[0:1]:
        # se lee el archivo si acaba en .pgm (formato de las imágenes)
        if file.endswith("pgm"):
            path = os.path.join(root, file)
            image = Image.open(path).convert("L") # mode L: 8-bit pixels, black
and white -> una capa en blanco y negro
            image_array = np.array(image, "uint8") # convierte imágenes en
array de números, unsigned integer (0 a 255)
            faces = face_cascade.detectMultiScale(image_array, scaleFactor=1.2,
minNeighbors=4) # detección de caras
            for (x, y, w, h) in faces:
                region_of_interest_gray = cv2.resize(image_array[y:y+h, x:x+w],
(260, 260)) # región de interés de la imagen
                id_, conf = recognizer.predict(region_of_interest_gray) #
predicción de la identidad
                suma_conf = suma_conf + conf
                if conf > 0:
                    if (file[0:7].lower()) == labels[id_]:
                        aciertos += 1
                    else:
                        fallos += 1
                print("Imagen bajo test:", file, "Prediccion:",
labels[id ], "Confidence:", conf)

# se muestran los resultados en el terminal
print("El numero de aciertos es:", aciertos)
print("El numero de fallos es:", fallos)
print("El numero de imagenes predecidas es:", (aciertos+fallos))
print("El porcentaje de exitos es:", ((aciertos/(aciertos+fallos))*100), "%")
```

## Anexo 4:

### Instalar Microsoft Visual Studio Code

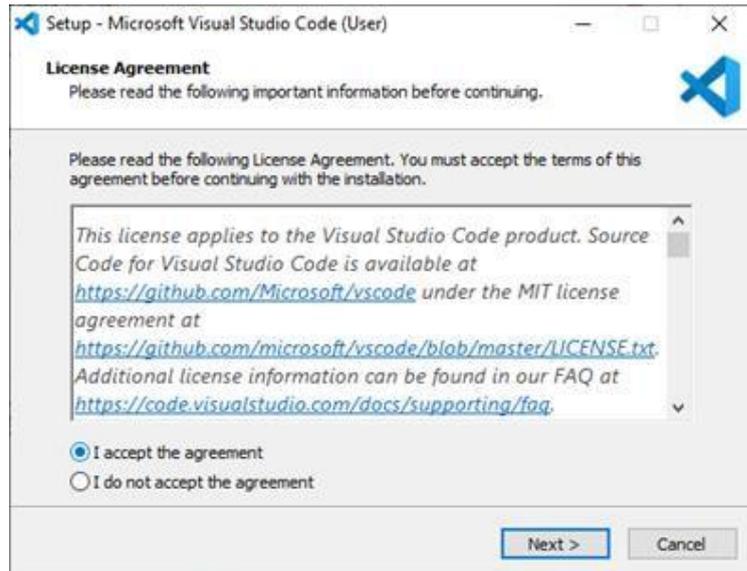
**Paso 1:** Ve a la página de descarga de Microsoft Visual Studio Code (<https://code.visualstudio.com/Download>) y haz clic en el botón correspondiente a Windows, selecciona la versión de 64 o 32 bits, en cualquiera de sus modalidades (instalador de usuario, sistema o zip).



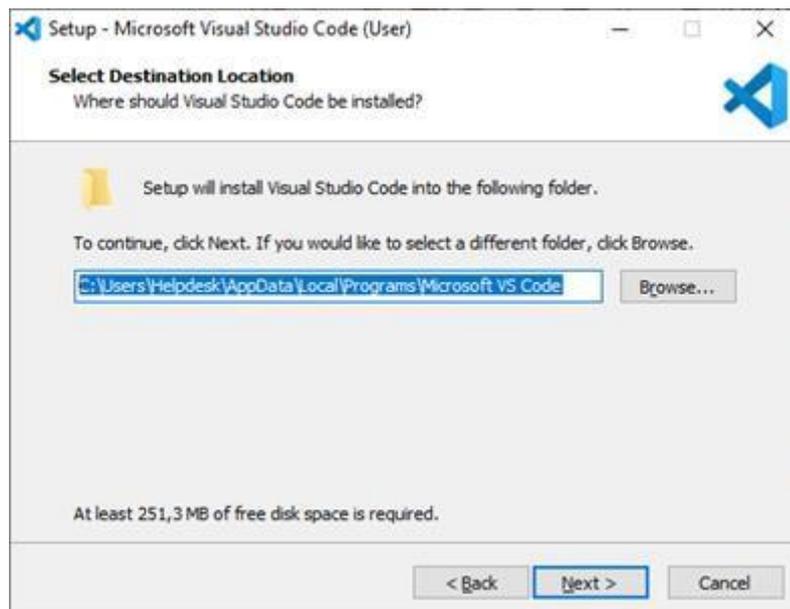
**Paso 2:** Abre el archivo de instalación .exe en tu carpeta de descargas para iniciar la instalación.

 **VSCoDeUserSetup-x64-1.63.2.exe**

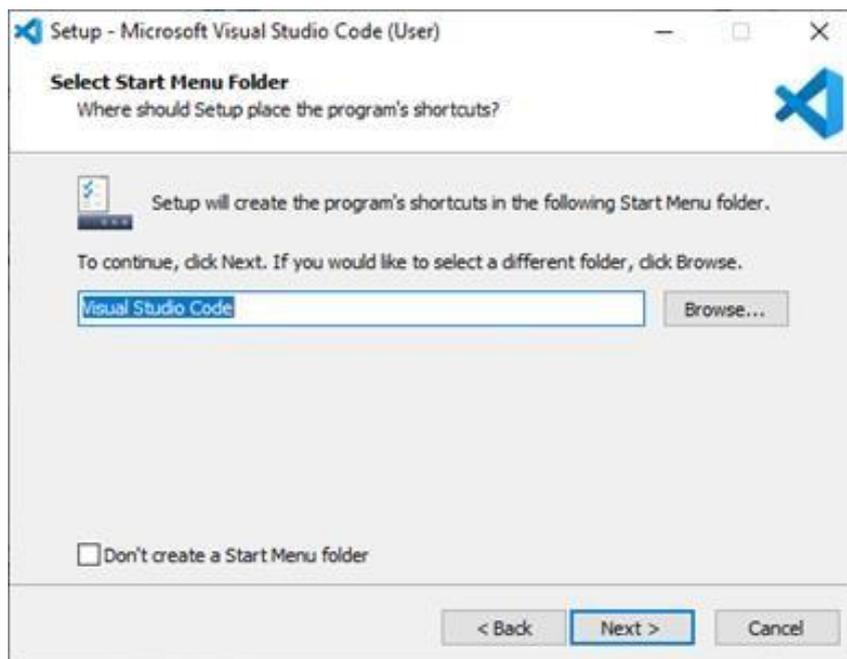
**Paso 3:** Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



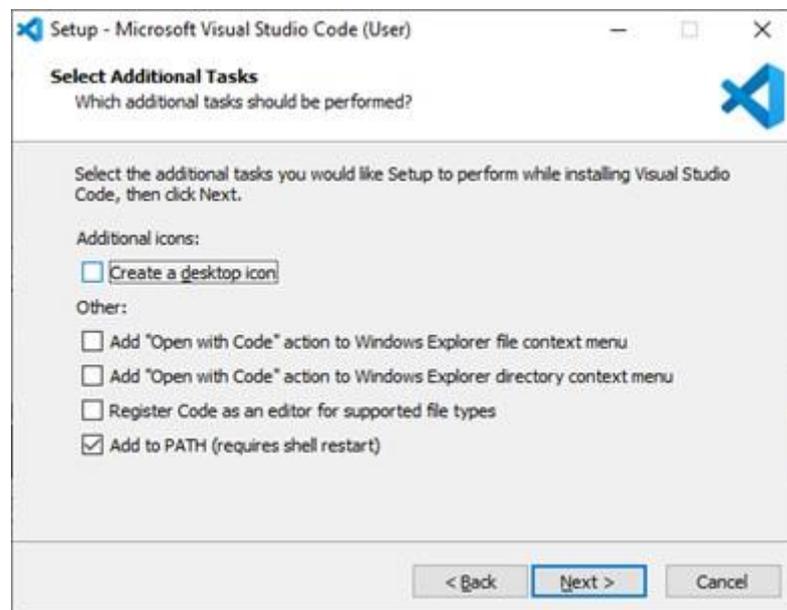
**Paso 4:** Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



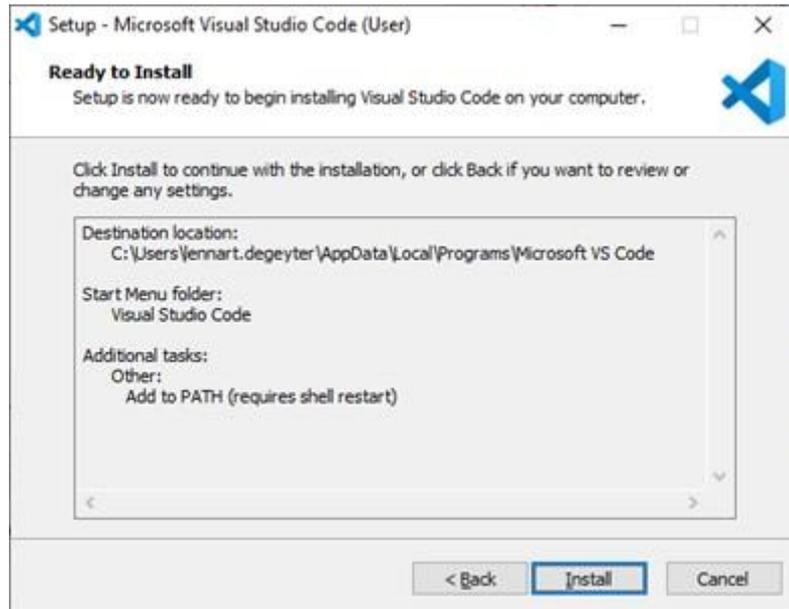
**Paso 5:** Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



**Paso 6:** Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.



**Paso 7:** Haz clic en Install para iniciar la instalación.



**Paso 8:** El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.

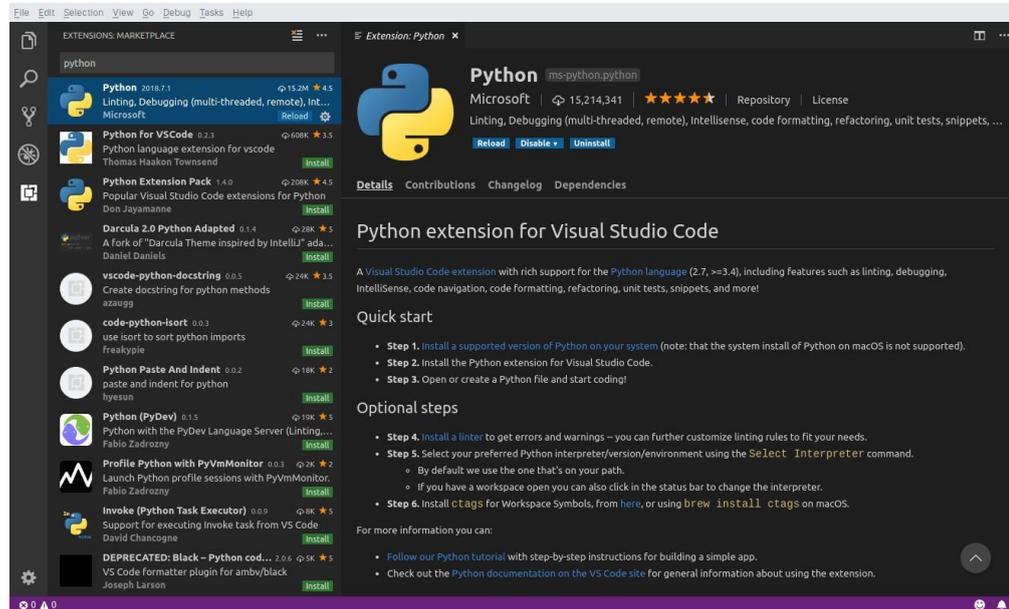


Se crea el siguiente acceso directo

### **Instalar Python desde el entorno de Visual Studio Code**



Lo primero es instalar la extensión de Python para Visual Studio Code. Para ello, hacemos click en el menú de extensiones a la izquierda y tecleamos 'Python'. Seleccionamos la de Microsoft y la instalamos.



## Configurar la extensión de Python en Visual Studio Code

Ahora que ya tenemos instalada la extensión de Python, podemos pasar a configurarla. Visual Studio Code gestiona las configuraciones mediante archivos JSON y se utilizan principalmente dos: un archivo para la configuración global y otro archivo específico para las configuraciones de proyectos.

Para abrir la configuración global podemos utilizar el atajo Control+, (coma) o navegar por '**Archivo>Preferencias>Ajustes**'. Vamos a añadir el siguiente fragmento:

```
{  
  
    "python.pythonPath": "/usr/bin/python3"  
  
}
```

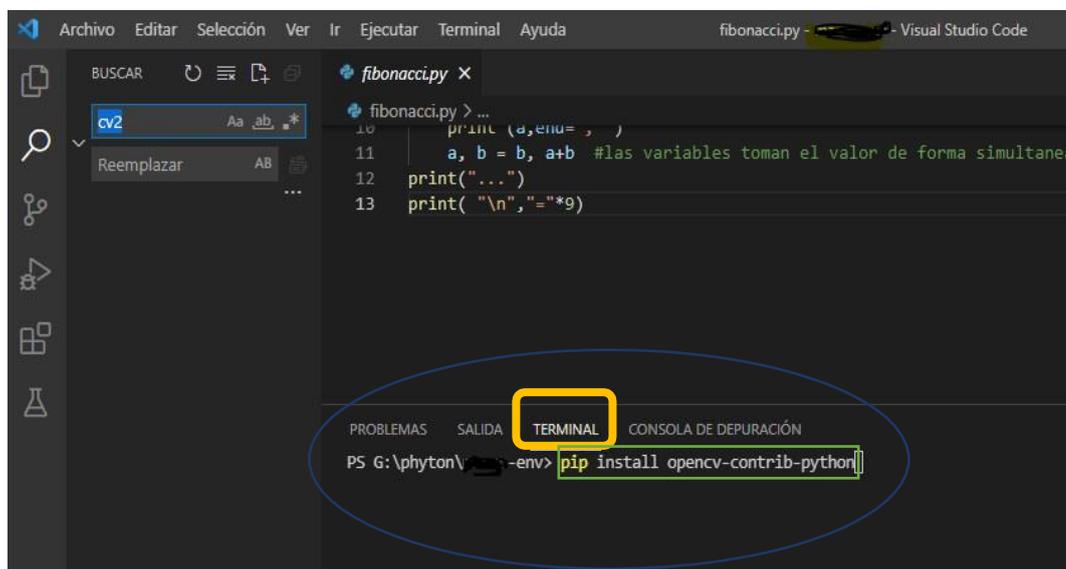
En este caso, he añadido /usr/bin/python3 porque es la ruta en la que se encuentra mi ejecutable de Python, pero ustedes tendrá que añadir la ruta en la que lo hayas instalado.

## Instalar OpenCV para Python desde el entorno de Visual Studio Code

Diríjase a la página web para seguir el proceso de instalación de OpenCV

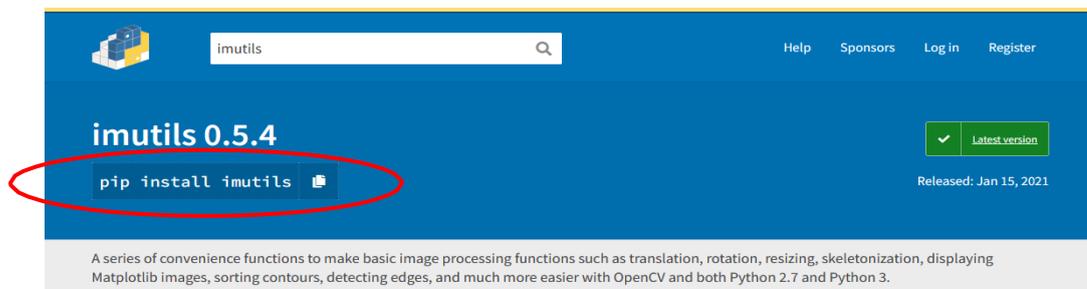
<https://pypi.org/project/opencv-python/>, esta se encuentra en inglés.

Una vez ahí vaya hasta **Installation and Usage** y ubíquese en la opción 2 del numeral 3, copie la línea de instrucción que permite una instalación completa de la librería OpenCV: “pip install opencv-contrib-python” sin las comillas, péguela en la ventana de TERMINAL del Visual Studio Code, como se aprecia en la imagen siguiente:



Pulse la tecla Enter y deje que se instale.

Terminado este paso, también debe instalar la extensión IMUTILS, se puede descargar desde la misma página de donde se descargó OpenCV (<https://pypi.org/project/opencv-python/>), escribiendo la palabra indicada “imutils” y luego la tecla Enter, vea la siguiente imagen:



Copie la línea “pip install imutils” y péguela en el Terminal de Visual Studio Code de forma similar a lo que hizo con OpenCV. Esto permitirá su instalación

**FACULTAD DE INGENIERIA  
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS Y  
COMPUTACION  
FICHA DE VALIDACIÓN DEL INSTRUMENTO**

**I. DATOS PERSONALES:**

- a) APELLIDOS Y NOMBRES DEL EXPERTO:  
PAUCAR PRADO LILIANA LUISA
- b) GRADO ACADÉMICO:  
UNIVERSITARIO
- c) CARGO E INSTITUCIÓN DONDE LABORA:  
ASISTENTE INFORMATICO -UDR PASCO
- d) TÍTULO DE LA INVESTIGACIÓN: Algoritmos de reconocimiento facial mediante aprendizaje automático para la identificación de personas en una institución educativa de Pasco - 2021
- e) AUTOR DEL INSTRUMENTO: Cristian Brayan MUÑOZ VILLANUEVA

**II. ASPECTOS DE EVALUAR: (Calificación Cuantitativa)**

INDICADORES	CRITERIO CUANTITATIVO/ CUALITATIVO	Deficiente 0-20 %	Regular 21-40 %	Bueno 41-60 %	Muy Bueno 61-80 %	Excelente 81-100 %
CLARIDAD	Esta formulado con lenguaje apropiado				X	
OBJETIVIDAD	Esta expresado con conductas observadas				X	
ACTUALIDAD	Adecuado al avance de la ciencia y calidad			X		
ORGANIZACION	Existe una organizacion logica del instrumento					X
SUFICIENCIA	Valora los aspectos en cantidad y calidad					X
INTENCIONALIDAD	Adecuado para cumplir con los objetivos				X	
CONSISTENCIA	Basado en el aspecto teorico cientifico del tema de estudios				X	
COHERENCIA	Entre las hipotesis, dimensiones Indicadores				X	
METODOLOGIA	Las estrategias responden al proposito del estudio				X	
CONVENIENCIA	Genera nuevas peutas para la investigación				X	

VALORACIÓN CUANTITATIVA: 80%

VALORACIÓN CUALITATIVA: Muy Bueno

OPINIÓN DE APLICABILIDAD: El instrumento mide adecuadamente la verificación de los algoritmos de reconocimiento facial

Pasco, enero de 2023

Firma y  
Posfirma del experto

DNI: 70248390  
Nº Telefónico: 963992129

  
**MINISTERIO DE SALUD  
SEGURO INTEGRAL DE SALUD  
UISP PASCO**  
 -----  
**Ing. Liliana L. Paucar Prado**  
 ASISTENTE INFORMATICO

**FACULTAD DE INGENIERIA  
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS Y  
COMPUTACION  
FICHA DE VALIDACIÓN DEL INSTRUMENTO**

**I. DATOS PERSONALES:**

- a) APELLIDOS Y NOMBRES DEL EXPERTO:  
ALANIA OSORIO LAURA VERENICE
- b) GRADO ACADEMICO:  
SUPERIOR
- c) CARGO E INSTITUCIÓN DONDE LABORA:  
ASISTENTE INFORMATICO – DIRESA PASCO
- d) TITULO DE LA INVESTIGACIÓN: Algoritmos de reconocimiento facial mediante aprendizaje automático para la identificación de personas en una institución educativa de Pasco - 2021
- e) AUTOR DEL INSTRUMENTO: Cristian Brayan MUÑOZ VILLANUEVA

**II. ASPECTOS DE EVALUAR: (Calificación Cuantitativa)**

INDICADORES	CRITERIO CUANTITATIVO/ CUALITATIVO	Deficiente 0-20 %	Regular 21-40 %	Bueno 41-60 %	Muy Bueno 61-80 %	Excelente 81-100 %
CLARIDAD	Está formulado con lenguaje apropiado				X	
OBJETIVIDAD	Esta expresado con conductas observadas				X	
ACTUALIDAD	Adecuado al avance de la ciencia y calidad				X	
ORGANIZACION	Existe una organización lógica del instrumento			X		
SUFICIENCIA	Valora los aspectos en cantidad y calidad					X
INTENCIONALIDAD	Adecuado para cumplir con los objetivos				X	
CONSISTENCIA	Basado en el aspecto teórico científico del tema de estudios				X	
COHERENCIA	Entre las hipótesis, dimensiones Indicadores			X		
METODOLOGIA	Las estrategias responden al propósito del estudio				X	
CONVENIENCIA	Genera nuevas pautas para la investigación				X	


VALORACIÓN CUANTITATIVA: 78%

VALORACIÓN CUALITATIVA: Muy Bueno

OPINIÓN DE APLICABILIDAD: El instrumento verifica el nivel de ayuda para conocer el algoritmo adecuado en una aplicación de reconocimiento facial

Pasco, enero de 2023

Firma y  
Posfirma del experto

DNI: 47442323

Nº Telefónico: 940161330

**FACULTAD DE INGENIERIA  
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS Y  
COMPUTACION  
FICHA DE VALIDACIÓN DEL INSTRUMENTO**

**I. DATOS PERSONALES:**

- a) APELLIDOS Y NOMBRES DEL EXPERTO:  
DORREGARAY COLCA HENRRY
- b) GRADO ACADEMICO:  
SUPERIOR
- c) CARGO E INSTITUCIÓN DONDE LABORA:  
INFORMatico – DIRESA PASCO
- d) TITULO DE LA INVESTIGACIÓN: Algoritmos de reconocimiento facial mediante aprendizaje automático para la identificación de personas en una institución educativa de Pasco - 2021
- e) AUTOR DEL INSTRUMENTO: Cristian Brayan MUÑOZ VILLANUEVA

**II. ASPECTOS DE EVALUAR: (Calificación Cuantitativa)**

INDICADORES	CRITERIO CUANTITATIVO/ CUALITATIVO	Deficiente 0-20 %	Regular 21-40 %	Bueno 41-60 %	Muy Bueno 61-80 %	Excelente 81-100 %
CLARIDAD	Esta formulado con lenguaje apropiado					X
OBJETIVIDAD	Esta expresado con conductas observadas			X		
ACTUALIDAD	Adecuado al avance de la ciencia y calidad				X	
ORGANIZACION	Existe una organización lógica del instrumento			X		
SUFICIENCIA	Valora los aspectos en cantidad y calidad					X
INTENCIONALIDAD	Adecuado para cumplir con los objetivos				X	
CONSISTENCIA	Basado en el aspecto teórico científico del tema de estudios				X	
COHERENCIA	Entre las hipótesis, dimensiones Indicadores					X
METODOLOGIA	Las estrategias responden al propósito del estudio				X	
CONVENIENCIA	Genera nuevas pautas para la investigación				X	

VALORACIÓN CUANTITATIVA: 79%

VALORACIÓN CUALITATIVA: Muy Bueno

OPINIÓN DE APLICABILIDAD: el instrumento verifica con objetividad la calidad de la investigación, ya que esto ayudaría a futuro a decidir qué tan buena sería una implementación de en el mundo real

Pasco, enero de 2023

Firma y  
Posfirma del experto

  
Ing. Henry E. DORREGARAY COLCA  
CIP 215079

DNI: 43342612

Nº Telefónico: 941861793