

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALAS**



CARRERA:

SISTEMAS Y COMPUTACIÓN

TEMA DE INVESTIGACIÓN:

EVALUACIÓN DE ALGORITMOS APLICADOS A LA EXTRACCIÓN DE
CARACTERÍSTICAS PARA EL RECONOCIMIENTO FACIAL EN BASE A LA
ISO/IEC 25010

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y
COMPUTACIÓN

LINEA DE INVESTIGACIÓN

PROGRAMACIÓN Y DESARROLLO DE SOFTWARE

AUTOR:

MENGHY RAÚL SÁNCHEZ MAILA

ASESOR:

M. Sc. JOSE LUIS CARVAJAL

Esmeraldas, mayo del 2021

TRIBUNAL DE GRADUACIÓN

Trabajo de tesis aceptado luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de Grado de la PUCESE previo a la obtención del título INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Presidente del tribunal de graduación

.....

M. Sc. José Luis Carvajal

Asesor de tesis

.....

M. Sc. Manuel Rogelio Nevarez

Lector 1

.....

PhD. Pablo Pico Valencia

Lector 2

.....

M. Sc. Susana Patiño

Coordinadora de la Escuela de Sistemas y Computación

Esmeraldas, de del 2021

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Menghy Raúl Sánchez Maila con cédula de identidad No. 0850160813 declaro que el presente trabajo de investigación y que los resultados que presento como tesis de grado, previo a la obtención del título de “Ingeniero en Sistemas y Computación” son totalmente de mi autoría, y legítimos.

Al mismo tiempo, declaro que todo el contenido incluyendo resultados, conclusiones y por otro lado los efectos académicos y legales que se desglosan de esta investigación son y serán de exclusiva responsabilidad académica y legal.

Menghy Raúl Sánchez Maila

Ci: 08501608130

DEDICATORIA

El presente trabajo de investigación se lo dedico principalmente a mí, que conociéndome con lo distraído que soy, y a pesar de mi gran habilidad para procrastinar las cosas, estoy aquí terminando quizás uno de los capítulos más importantes de mi vida profesional.

Y ¿cómo olvidarme de mis padres y mi hermana?, quienes han sido los responsables de mi educación desde que tengo memoria, y quienes se han sacrificado por lograr que mi carrera académica no se detenga por ningún motivo.

Menghy Raúl Sánchez Maila.

AGRADECIMIENTO

He llegado al punto en que me gustaría mencionar a muchas personas, sin embargo, creo que no alcanzarían las palabras para demostrarles mis agradecimientos.

Agradezco a mis padres y mi hermana, que siempre han velado por mi bienestar, sin muchos lujos, pero lo necesario para aprender a valorar las cosas que la vida me ha dado.

Agradezco a mi padrino, que siempre confió de alguna forma en mí, recordándome siempre que puedo hacer las cosas que me proponga y que estoy a la par de grandes profesionales y muy buenos estudiantes que él ha conocido.

Agradezco a ciertas personas que conforman la familia PUCESE, desde que los conocí se convirtieron en colegas y personas en las que de algún modo podía contar para un consejo o alguna duda que se me presentaba, profesor Marc Grob, profesora Susana Patiño, Erika Quintero, Alejandra Salazar, profesor Jacinto Fierro, profesor Juan Casierra, y como no mencionar a Jackeline Plaza que siempre estaba ahí como una mamá velando por cada estudiante de la carrera de Sistemas.

Agradezco también a mis amigos, aquellos con los que pasé horas y horas haciendo tareas, les estoy muy agradecido, porque gracias a que soy muy distraído siempre olvidaba que tenía tareas, y ellos eran conscientes de eso y me lo recordaban cada que me veían bien fresco como de costumbre.

Menghy Raúl Sánchez Maila

ÍNDICE DE CONTENIDO

TRIBUNAL DE GRADUACIÓN	2
DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD.....	3
DEDICATORIA	4
AGRADECIMIENTO	5
ÍNDICE DE CONTENIDO	6
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	10
ÍNDICE DE ALGORITMOS	11
RESUMEN	13
ABSTRACT.....	15
INTRODUCCIÓN	17
Planteamiento del problema.....	17
Justificación	19
Objetivos.....	20
Objetivo general.....	20
Objetivos específicos	20
CAPÍTULO I: MARCO TEÓRICO	21
1.1 Bases teóricas-científicas	21
1.1.1. Reconocimiento facial y detección de rostros	21
1.1.2. Biometría computacional.....	23
1.1.3. Reconocimientos de patrones	25
1.1.4. Herramientas para la implementación del reconocimiento de facial.....	25
1.1.5. Evaluación de calidad de software	28
1.1.6. Set de Datos.....	35

1.2	Antecedentes	37
1.3	Bases legales	40
CAPÍTULO II: MATERIALES Y MÉTODOS		42
2.1.	Delimitación de la investigación.....	42
2.2.	Tipo de investigación.....	42
2.3.	Métodos de investigación	43
2.4.	Técnicas e instrumentos de recolección de datos	44
2.5.	Técnicas de procesamientos de análisis de datos.....	44
2.6.	Materiales.....	44
2.7.	Matriz de variables e indicadores	47
CAPÍTULO III: IMPLEMENTACIÓN.....		51
3.1.	Selección de los algoritmos a evaluar	51
3.2.	Algoritmos para la extracción de características para reconocimiento facial ..	51
3.1.1	Algoritmo PCA (Principal componentes básicos)	52
3.1.2	Algoritmo EigenFace	53
3.1.3	Algoritmo FisherFace	56
3.1.4	Algoritmo Viola Jones	61
CAPÍTULO IV: RESULTADOS		65
4.1	Evaluación de adecuación funcional	66
4.2	Evaluación de eficiencia y desempeño	67
4.3	Evaluación de usabilidad.....	69
4.4	Evaluación de la fiabilidad.....	69
4.5	Evaluación de la mantenibilidad	70
4.6	Evaluación de la portabilidad.....	70
CAPÍTULO V: DISCUSIÓN		73
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES		75
6.1.	Conclusiones.....	75

6.2. Recomendaciones	76
REFERENCIAS.....	77
ANEXOS	82

ÍNDICE DE FIGURAS

Fig. 1: Almacenamiento de una imagen como matriz digital. Fuente: Elaboración propia.	22
Fig. 2: Familia de la Norma ISO/IEC 25000. Fuente: Elaboración propia.	29
Fig. 3: Características de calidad según la norma ISO/IEC 25010 [26].	33
Fig. 4: Criterio de Evaluación para los Algoritmos de Reconocimiento Facial en base a la norma ISO/IEC 25010. Fuente: Elaboración propia.....	45
Fig. 5: Valoración escalar para las características del marco de evaluación. Fuente: Elaboración propia.....	45
Fig. 6: Referencia del conjunto de datos yale face [58]	46
Fig. 7: Parte del dataset de imágenes a utilizar. Fuente: Elaboración propia.....	47
Fig. 8: Formato de presentación de la fotografía para entrenamiento de los algoritmos. Fuente: Elaboración propia.....	47
Fig. 9: Especificaciones de la imagen a utilizar. Fuente: Elaboración propia.....	47
Fig. 10: Descripción del entorno de trabajo para la implementación de los algoritmos. Fuente: Elaboración propia.....	53
Fig. 11: Funcionamiento del clasificador cascada Viola-Jones. Fuente Elaboración propia.	61
Fig. 12: Porcentaje de efectividad de los algoritmos. Fuente: Elaboración propia.	67
Fig. 13: Tiempo de ejecución de los Algoritmos. Fuente: Elaboración propia.....	68
Fig. 14: Resultado general de la evaluación de los algoritmos en base al marco de la ISO/IEC 25010. Fuente: Elaboración propia.	71

ÍNDICE DE TABLAS

Tabla 1: Divisiones de la familia de las ISO/IEC 25000.....	29
Tabla 2: Descripción de la familia de normas ISO/IEC 25000.	31
Tabla 3: Descripción de las 8 características de la norma ISO/IEC 25010 [26].....	33
Tabla 4: Set de Datos disponibles en la Web.....	36
Tabla 5: Variables e indicadores de evaluación.....	48
Tabla 6: Resultados Evaluación Según la ISO/IEC 25010.....	66
Tabla 7: Resultados generales de la puntuación obtenida de los algoritmos.....	72

ÍNDICE DE ALGORITMOS

Algoritmo 1: Definición del set de datos para el algoritmo de EigenFace. Fuente: Elaboración propia.....	54
Algoritmo 2: Lectura y almacenado de las imágenes de entrenamiento. Fuente: Elaboración propia.....	54
Algoritmo 3: Generación de la imagen media de los rostros. Fuente: Elaboración propia	55
Algoritmo 4: Normalización del vector principal de imágenes. Fuente: Elaboración propia.....	55
Algoritmo 5: Generación de la matriz de covarianza. Fuente: Elaboración propia.....	56
Algoritmo 6: Generación de los eigen vectores a partir de la matriz de covarianza - Algoritmo EigenFace. Fuente: Elaboración propia.....	56
Algoritmo 7: Generación las imágenes resultantes de las matrices con las características faciales extraídas del conjunto de imágenes. Fuente: Elaboración propia.....	56
Algoritmo 8: Definición del set de datos, lectura de las imágenes de entrenamiento. Fuente: Elaboración propia.....	57
Algoritmo 9: Generación de la imagen media del conjunto de imágenes. Fuente: Elaboración propia.....	58
Algoritmo 10: Generación de los vectores de imágenes y los vectores eigen. Fuente: Elaboración propia.....	58
Algoritmo 11: Generación de la matriz de proyección y distancia mínima de las características. Fuente: Elaboración propia	59
Algoritmo 12: Generación de las caras medias por clases. Fuente: Elaboración propia.....	59
Algoritmo 13: Selección de las 9 mejores clases. Fuente: Elaboración propia.....	60
Algoritmo 14: Amplitud de las características y proyección de los rostros seleccionados. Fuente: Elaboración propia.....	60
Algoritmo 15: Lectura y almacenado del set de datos - Algoritmo Viola – Jones. Fuente: Elaboración propia.....	62
Algoritmo 16: Separación del conjunto de datos de entrenamiento y prueba. Fuente: Elaboración propia.....	63
Algoritmo 17: Implementación de las características básicas para delimitación. Fuente: Elaboración propia.....	63

Algoritmo 18: Detección y extracción de las características de rostros. Fuente: Elaboración propia.	64
Algoritmo 19: Generación del archivo txt y almacenado de las características - Algoritmo Viola – Jones. Fuente: Elaboración propia.	64

RESUMEN

Cuando se menciona que un algoritmo es empleado para establecer una identificación o para un reconocimiento, se debe evitar confundir los verdaderos significados de estos términos, por ejemplo los algoritmos de reconocimiento facial se centran en extraer y reconocer las características que posee un rostro, sin embargo, los algoritmos de identificación facial, emplean algoritmos capaces de extraer las características de un rostro, compararlas e identificar con una etiqueta a quien pertenece el rostro detectado en la imagen.

La presente investigación se enfoca en la evaluación de los algoritmos más conocidos y empleados para la extracción de las características faciales en sistemas de reconocimiento facial. En la actualidad existen una variedad de algoritmos algunos de ellos y unos de los más usados son: Viola-Jones, EigenFace, FisherFace, Naive Bayes, Suport Vector Machine (SVM), los algoritmos pueden ser empleados solos, o en conjunto, dando a conocer a esta agrupación como “algoritmos híbridos”.

Evaluar cada uno de los algoritmos disponibles, puede llegar a ser una tarea agotadora, sin embargo, en esta investigación se realizó una evaluación en base a algunos parámetros que presenta la ISO/IEC 25010, los mismos que establecen el marco de calidad de un producto de software. El motivo de utilizar una ISO como un marco de referencia para una evaluación de distintos algoritmos es muy útil cuando reconocemos que los algoritmos son capaces de realizar su propósito, sin embargo, cada uno de ellos posee puntos fuertes y por ello no sería una evaluación muy justa, por ende, el analizar parámetros como: corrección funcional, modularidad, adaptabilidad, entre otros, establece una base uniforme para la evaluación de cada algoritmo.

En investigaciones posteriores los algoritmos ya mencionados han obtenido resultados considerados excelentes, referentes a su efectividad al momento de ser comparados con otros, uno de esos casos es el algoritmo FisherFace que en múltiples investigaciones se ha posicionado con un 98% de efectividad, frente a al algoritmo EigenFace el cual recurrentemente obtiene una valoración del 80 al 85%. Aunque estas investigaciones suelen ser enfocadas solo a los aspectos de efectividad, no siempre son del todo justas, dado que cada algoritmo tiene un punto fuerte según su estructura.

Existen varias formas en las que se puede implementar un algoritmo, usar un algoritmo en específico como el núcleo del funcionamiento es ideal para establecer una evaluación equilibrada; algoritmos basados en SVM (*support vector machine*), PCA (*análisis de componentes principales*) o en las famosas CNN (*red neuronal convolucional*), son buenas opciones a tener en cuenta. En esta investigación se implementaron los algoritmos Viola-Jones, EigenFace y FisherFace basados en el algoritmo PCA. La elección de este algoritmo como base, se establece debido a sus principios, el mismo algoritmo consiste en la extracción de los componentes principales que posee una imagen, para luego almacenar esos datos y poder generar un conjunto de datos locales, que le permitirán realizar el procedimiento de reconocimiento y extracción de características con mayor facilidad.

La implementación se inicia con la selección de los algoritmos, haciendo uso de los métodos inductivos y analíticos, con la finalidad de conocer, organizar y estructurar cada parte que compone a la investigación. Conocer los conceptos fundamentales del proceso de reconocimiento facial, las herramientas a disposición y las características a evaluar son indispensables para lograr generar un entorno de desarrollo y recopilación de información óptimo. Una vez completada esta primera etapa, se añaden dos métodos adicionales a la investigación como lo son: el método experimental y el método deductivo, los mismos que complementaron a la parte teórica, dado que permitieron establecer los resultados de las características a evaluar.

En cuanto a la evaluación realizada, los algoritmos fueron implementados de tal forma que su estructura sea totalmente visible, es decir, se evitó el utilizar bibliotecas o funciones que automatizaran todo el proceso de extracción de las características faciales, esto con la finalidad de generar una evaluación equilibrada y equitativa. La misma que generó como resultado una valoración de 56/70 para el algoritmo de FisherFace, seguido por EigenFace con un 54/70 y por último el algoritmo Viola-Jones con un 53/70, estos puntajes referentes a la evaluación en base al marco de la ISO/IEC 25010, sin embargo en lo que corresponde a efectividad al momento de realizar el reconocimiento y extracción de las características, el algoritmo mejor posicionado fue el Viola-Jones con un 95%, seguido de FisherFace con un 90% y por último el algoritmo EigenFace con un 86% de efectividad.

ABSTRACT

When it is mentioned that an algorithm is used to establish an identification or recognition, it is important to avoid confusing the true meanings of these terms. For example, facial recognition algorithms focus on extracting and recognizing the characteristics of a face; however, facial identification algorithms use algorithms capable of extracting the characteristics of a face, comparing them and identifying with a label to whom the face detected in the image belongs.

The present research focuses on the evaluation of the most well-known algorithms used for the extraction of facial features in face recognition systems. Currently there are a variety of algorithms some of them and some of the most used are: Viola-Jones, EigenFace, FisherFace, Naive Bayes, Support Vector Machine (SVM), the algorithms can be employed alone, or together, making this grouping known as "hybrid algorithms".

Evaluating each of the available algorithms can be an exhausting task, however, in this research an evaluation was performed based on some parameters presented by ISO/IEC 25010, which establishes the quality framework of a software product. The reason for using an ISO as a reference framework for an evaluation of different algorithms is very useful when we recognize that the algorithms are capable of performing their purpose, however, each one of them has strong points and therefore it would not be a very fair evaluation, therefore, analyzing parameters such as: functional correctness, modularity, adaptability, among others, establishes a uniform basis for the evaluation of each algorithm.

In subsequent research, the aforementioned algorithms have obtained results considered excellent, regarding their effectiveness when compared with others, one of these cases is the FisherFace algorithm that in multiple investigations has been positioned with 98% effectiveness, compared to the EigenFace algorithm which recurrently obtains a valuation of 80 to 85%. Although these investigations are usually focused only on the effectiveness aspects, they are not always fair, since each algorithm has a strong point according to its structure.

There are several ways in which an algorithm can be implemented, using a specific algorithm as the core of the performance is ideal to establish a balanced evaluation; algorithms based on SVM (support vector machine), PCA (principal component analysis)

or the famous CNN (convolutional neural network), are good options to consider. In this research, the Viola-Jones, EigenFace and FisherFace algorithms based on the PCA algorithm were implemented. The choice of this algorithm as a base, is established due to its principles, the same algorithm consists in the extraction of the main components that an image has, to then store these data and to be able to generate a set of local data, which will allow to perform the recognition procedure and extraction of features more easily.

The implementation begins with the selection of the algorithms, making use of inductive and analytical methods, in order to know, organize and structure each part that makes up the research. Knowing the fundamental concepts of the face recognition process, the tools available and the characteristics to be evaluated are essential to generate an optimal development and information gathering environment. Once this first stage was completed, two additional methods were added to the research: the experimental method and the deductive method, which complemented the theoretical part, since they allowed to establish the results of the characteristics to be evaluated.

Regarding the evaluation carried out, the algorithms were implemented in such a way that their structure is totally visible, that is to say, the use of libraries or functions that automate the whole process of extraction of the facial characteristics was avoided, this with the purpose of generating a balanced and equitable evaluation. As a result, the FisherFace algorithm was rated 56/70, followed by EigenFace with 54/70 and finally the Viola-Jones algorithm with 53/70, these scores referring to the evaluation based on the ISO/IEC 25010 framework, however, in terms of effectiveness at the time of recognition and extraction of features, the best positioned algorithm was Viola-Jones with 95%, followed by FisherFace with 90% and finally the EigenFace algorithm with 86% effectiveness.

INTRODUCCIÓN

Planteamiento del problema

La seguridad siempre ha sido un tema que se encuentra en constante análisis. El poder identificar qué personas ingresan en determinados lugares siempre tienen un margen presupuestario elevado dado que el medio normal, es mediante el reconocimiento visual entre personas; una práctica realizada desde tiempos remotos. Sin embargo, el uso de la tecnología puede abaratar costos y mejorar los niveles de seguridad que se implanten a dichos lugares.

El diseño de algoritmos que sean eficientes y capaces de cumplir su función en situaciones inesperadas es una de las principales tareas que tiene un desarrollador, motivo por el cual, tener métodos y marcos que permitan realizar una evaluación de estos antes de comercializarlos es una de las practicas que se usan muy a menudo. Por otra parte, existen métodos que solo se enfocan en medir el grado de precisión con el que el software realiza su tarea, sin tener presente que la forma en la que este se encuentra estructurado también influye en el uso que se le puede dar; solo hay que mirarlo del modo en el que, un código que puede ser entendido por cualquier desarrollador va a ser más usado que por aquellos que son de complejo entendimiento. Se debe tener presente que el hecho de que cumpla con su propósito no incluye el hecho de que sea comprensible.

Si bien se entiende que la identidad es el conjunto de rasgos característicos de un sujeto que lo diferencia de los demás miembros de una agrupación, de ahí parte que la misma es de gran importancia desde la antigüedad, en donde era utilizada con la finalidad de identificar miembros pertenecientes a grupos específicos de un entorno [1], [2].

La visión humana es considerada como un complejo sistema, dado que este es capaz de distinguir hasta 10 millones de colores, lo que permite diferenciar una gran cantidad de detalles, motivo por el cual se convierte en un tema de gran interés en la comunidad de investigadores de la visión por computadora [3].

En la actualidad el poder convivir en una sociedad en donde la seguridad es el enfoque central de un sentimiento reconfortante para cualquier persona [4], puesto que tener el control sobre el acceso de personal designado a áreas específicas se considera como asuntos de gran importancia; los procesos de identificación facial de las personas

ha atravesado por un sin número de experimentos dado que cada avance tecnológico permite la evolución de sistemas de video vigilancia e identificación de personas.

Para realizar un análisis completo y determinar la identidad de una persona se deben tener algunos aspectos para tener en cuenta; estas características pueden ser: características físicas (i.e., pupilas, forma del rostro, iris), o también pueden tomarse rasgos psicológicos tales como los gestos. Si bien todas las características deben ser evaluadas bajo un criterio [5].

- Generalidad: para saber qué tan común puede ser una de las características de la persona.
- Carácter individual: con la cual se determina que tan extraña o personal puede llegar a ser dicha característica.
- Persistencia: si poseen rasgos que pueden perderse con el tiempo, dentro de estos pueden ser marcas como: cicatrices, o cambio de tono de piel ya sea de forma natural o de alguna otra forma.
- Tergiversación: cuando el sistema intente ser engañado.

Aunque se presenten estos parámetros, el reconocimiento facial presenta otras limitaciones, las mismas que incluso ahora, al tomar una fotografía se logran denotar, ya sea por la calidad de la cámara o por condiciones del ambiente; estas limitaciones pueden ser: ruido de la fotografía, luminosidad, expresión facial, vello facial, decadencia y accesorios de uso continuo (i.e., gafas, aretes, incluso tatuajes).

Al tener como bases estas limitaciones, se considera que el punto central de la siguiente investigación consiste, en la elección de la técnica que mejor se adapte a las posibles limitaciones y logre generar un resultado óptimo además de cumplir con el estándar de calidad basado en la norma ISO/IEC 25010.

Justificación

Los sistemas de reconocimiento facial no necesariamente necesitan de la interacción con las personas a ser evaluadas, esto hace que se convierten en sistemas de gran importancia para las empresas que generan vigilancia como: aeropuerto, entidades bancarias, centros de reclusión, entre otras instalaciones que lo requieran.

En Ecuador es conocido que el manejo de este tipo de tecnologías no es común, motivo por el cual se convierte en un punto de partida principal para el desarrollo de sistemas de seguridad basados en reconocimiento facial, lo cual significaría una reducción notable en la brecha tecnológica que se tiene con otros países.

Existen muchas empresas que actualmente trabajan en desarrollar sistemas de reconocimiento facial, una de ellas, la gigante Amazon con su Amazon Rekognition [1], la cual asegura que mediante el uso de una imagen o video proporcionado a éste, la aplicación se encargará de detectar e identificar personas, objetos y entre otras actividades. Luego se tiene a los grandes de China, este país asiático en donde poseen un sistema de reconocimiento facial para mantener el orden y brindar acceso a servicios [6].

La importancia de lograr implementar un sistema capaz de realizar una correcta identificación de personal no solo dependerá de los dispositivos que se usen para determinada tarea. El algoritmo que controlará los sistemas debe ser eficiente, motivo por el cual esta investigación se enfoca en el análisis de los algoritmos usados para el reconocimiento facial, en base a la norma ISO/IEC 25010, la misma que encapsula los parámetros de calidad de software. Estos parámetros que han sido seleccionados se evaluaron en base a distintos marcos de evaluación presentados por distintos autores, en diversas investigaciones. Sin duda alguna se puede establecer la gran importancia que presentan la aplicación que hacen uso de los algoritmos enfocados al sector de la seguridad, desde smart phone, tablets, computadoras portátiles, hasta sistemas de video vigilancia en circuito cerrado o abierto. Se debe tener en cuenta que los mismos son creados por humanos, siendo este hecho un causal de posibles errores al momento de su desarrollo lo que conlleva a un defecto en el producto de software.

Por otra parte, la aplicación de algoritmos de reconocimiento facial en Esmeraldas es factible dado que se cuenta con el sistema de video vigilancia dado por el ECU 911, el mismo que puede ser empleado en conjunto con una base de datos poblacional, que permita reconocer a los infractores y o delincuentes de mejor manera, ahorrando una

cantidad considerable de tiempo y recursos económicos. En la actualidad se puede observar la aplicación de este sistema en el ECU 911 de la ciudad de Guayaquil, en donde ya se implementa el sistema de reconocimiento facial en vivo, mediante el mismo sistema de seguridad de video vigilancia.

Objetivos

Objetivo general

Evaluar los principales algoritmos utilizados para la extracción de las características empleadas en el reconocimiento facial, mediante la implementación y ejecución de cada algoritmo, con la finalidad establecer una conclusión acerca de la calidad que cada una posee como software en base a un marco de evaluación según la ISO/IEC 25010.

Objetivos específicos

- a) Examinar los distintos algoritmos de reconocimiento facial existentes y determinar aquellos que serán evaluados.
- b) Comparar e identificar las principales características de los distintos algoritmos de reconocimiento facial seleccionados.
- c) Valorar los algoritmos de reconocimiento facial con el marco de evaluación, en base al estándar ISO/IEC 25010.
- d) Establecer el mejor algoritmo en términos de la calidad de un producto de software, según la norma ISO/IEC 25010 y las ventajas que posee frente a los demás algoritmos de extracción de características faciales evaluados.

CAPÍTULO I: MARCO TEÓRICO

1.1 Bases teóricas-científicas

1.1.1. Reconocimiento facial y detección de rostros

Reconocimiento facial

Detección y reconocimiento facial, ¿es lo mismo? En muchas ocasiones estos términos han sido usado para referirse a lo mismo, cabe destacar que no es correcto la forma de emplearlo, si bien la detección facial consiste en el proceso para encontrar los rostros de un humano en una imagen, está muy lejos de poder identificar quien es la persona que está siendo detectada. Por el contrario, cuando se refiere a reconocimiento facial, se establece como principio el no solo realizar la búsqueda del rostro de una persona, sino el intentar establecer la identidad de la o las personas que se encuentran en la imagen.

Cuando se trabaja en el área de detección de los rostros humanos, los algoritmos deben ser capaces de lograr obtener la posición y la cantidad de rostros que se encuentran en las imágenes; frente a esto el mismo algoritmo puede enfrentarse a ciertos problemas que consisten en: dirección del rostro, expresiones faciales, niveles de iluminación, calidad de la imagen, dimensiones de los rostros y detección de componentes extras (i.e., cicatrices, lentes, bello facial).

Ya en el año 2002, el investigador Yang D.J. presentó su investigación [7], en donde menciona que existen cuatro modelos o métodos para el reconocimiento facial, aunque el primer método y el cuarto método pueden considerarse como un complemento de los mismos. Los métodos están basados en:

- **Conocimientos:** este método hace referencia a lo que hoy en día se conoce como modelos de aprendizaje supervisados, en los cuales se utiliza la experiencia y conocimiento humano para establecer modelos de rostros válidos.
- **Características únicas o invariantes:** en la que el algoritmo busca ciertas características que aun con diferentes condiciones de iluminación son presentes, un ejemplo de éstas puede ser el color de la piel o incluso la textura.

- Plantillas: este método al igual que el primero también hace referencia a un modelo de aprendizaje automático denominado, aprendizaje no supervisado, esto debido a que de un conjunto de imágenes se establecen conjuntos de patrones que permiten identificar los rostros de las personas.

Procesamiento de imágenes

El gobierno de las imágenes digitales es uno de los temas que ha recibido gran muestra de interés en las últimas décadas debido a las amplias áreas en las que puede ser utilizadas; desde las ciencias médicas, naturales, hasta las ciencias tecnológicas y muchas otras más. Junto a la necesidad de manipular las imágenes aparece el procesamiento de imágenes, con el objetivo de mejorar las imágenes y lograr hacer visibles pequeños detalles que contienen las mismas y así, lograr obtener un análisis que permita deducir y obtener decisiones a partir de la información generada. Las imágenes pueden ser capturadas por varios medios, los más usados: scanner y las cámaras digitales [2].

Para realizar el procesado de una imagen, en primera instancia se debe realizar el almacenamiento de la imagen. La unidad básica de una imagen en su formato digital es el píxel; la resolución de una imagen es dada por la multiplicación del número de filas por el número de columnas de pixeles que posee, siendo este el principal motivo por el cual una imagen se conoce como una matriz digital. En la misma se hallan valores en un rango de 0 (negro) a 255 (blanco), tal como se muestra en la **Fig. 1**.



Fig. 1: Almacenamiento de una imagen como matriz digital. Fuente: Elaboración propia

1.1.2. Biometría computacional

La biometría se remonta hace 31.000 años, al comprobar la existencia de cuevas con paredes llenas de ilustraciones a base de pintura, creadas por hombres prehistóricos que vivieron ahí y en ellas se encontraron impresiones de palmas de manos como firma del autor. La biometría computacional, hace referencia al uso de las matemáticas y estadísticas para lograr identidad a un ser vivo, en otras palabras, hace uso de técnicas matemáticas para verificar la identidad de un usuario basada en las características fisiológicas [8].

Propiedad de los rasgos biométricos

Si bien es conocido, un sistema biométrico debe poseer ciertas características importantes. Entre éstas se pueden incluir las siguientes:

- **Universalidad:** Características que todos los sujetos a usar el sistema deben poseer.
- **Singularidad:** Los patrones a identificar tienen que ser únicos e irrepetibles.
- **Usurpación:** Nivel que posee el sistema para contrarrestar la falsificación de identidad.
- **Cuantificable:** Los patrones o características de el / los sujetos, debe ser capaz de ser cuantificados.

Escáner de la cara o reconocimiento facial

Comúnmente las personas utilizan el método de reconocimiento de caras para lograr identificar a una persona; a pesar de que esta es una tecnología que ha sido usada hace varios años, es en la actualidad cuando se ha denotado un mayor impacto en la sociedad, con la implementación en bancos, aeropuertos, medios de migración, entre otros en donde el uso de software como el Rekognition, software de Amazon, son usados para la identificación de presuntos delincuentes [9]. El método del reconocimiento facial hace posible el poder identificar a una persona, ya sea por medio de una imagen o de un video. En la actualidad estos sistemas cuentan con algoritmos complejos y al mismo tiempo con un grado de efectividad alto.

Escáner de huella dactilar

Considerada como una de las técnicas más antiguas de la biometría, la identificación por medio de las huellas dactilares, se han utilizado en un sin número de aplicaciones debido a su singularidad, existen dos tipos de técnicas de identificación de huellas dactilares: Basadas en minucias, el mismo que basa su mecanismo de autenticación en la selección de patrones de fácil identificación, con lo que genera un modelo por cada usuario; no obstante se presenta un grado de error al ser utilizado debido a la complejidad de extraer los datos con precisión. Por otra parte, la segunda técnica, basadas en correlación, realiza un análisis del patrón global que posee la huella dactilar. A diferencia de la técnica anterior, el margen de error oscila entre la traslación y rotación de la imagen. Los sistemas de reconocimiento de huellas dactilares están compuestos por cuatro fases: adquisición de la huella, preprocesamiento de la imagen, segmentación de la imagen, extracción de los puntos característicos de la huella [5], [10].

Escáner de la retina o iris

Después de 25 años de haberse desarrollado la primera patente para el reconocimiento de individuos por medio de la identificación del iris, se considera a este método como uno de los más efectivos y fiables, colocándose por encima del reconocimiento facial, huella dactilar y de mano. Al igual que la huella dactilar posee una singularidad más estricta; los patrones que posee el iris están definidos desde el nacimiento, el cual no cambia durante el transcurso del tiempo. El procedimiento del escaneo de iris o retina se genera mediante, la captura de la retina con una cámara infrarroja especializada, la cual toma una fotografía del ojo en alta resolución que proporciona los detalles del iris para luego almacenarlos y registrarlos [10], [11].

Biometría de la voz

La autenticidad mediante el uso de la voz es considerada uno de los métodos de reconocimiento biométricos más complicado, esto debido a la gran cantidad de factores que intervienen en el proceso, uno de ellos, el ruido de fondo que genera un margen de error. Los algoritmos miden y estiman las similitudes entre las muestras previamente almacenadas para generar una lista de posibles usuarios identificados; debido a que este

método de reconocimiento solo necesita de un micrófono lo posiciona por debajo de los demás métodos de reconocimiento en cuanto a seguridad brindada [10], [12].

Funciones de un sistema biométrico

Los sistemas biométricos pueden ser utilizados de dos formas, ya sea como verificación o a su vez como un identificador. Mediante el uso de los distintos tipos de biometría el usuario puede verificar su identidad y a su vez la reclama como suya, esto se da cuando el usuario presenta un rasgo o patrón y este es comparado con el almacenado en la base de datos o dataset, por lo que se realiza una comparación de uno a uno; mientras que al usar el segundo modo (Identificación), el sujeto presenta un rasgo o patrón y esta información es comparada con todos los datos almacenados, es decir, realiza una comparación de uno a muchos [13].

1.1.3. Reconocimientos de patrones

Se lo puede definir como un sistema de clasificación, el cual está basado en datos obtenidos previamente. Este tipo de sistema puede ser usado en varios campos, ya sea para el reconocimiento de información multimedia, hasta el reconocimiento de individuos según sus características biométricas. El reconocimiento de patrones consiste en la clasificación y agrupación de patrones o características de un sujeto o entidad a estudiar; se asignan etiquetas a los patrones las cuales transmiten un significado a la agrupación a la que pertenecen, se considera que un patrón se representa mediante un vector en el cual se encuentran las características de un sujeto [14].

1.1.4. Herramientas para la implementación del reconocimiento de facial

Existe una variedad de herramientas que pueden ser escogidas para el desarrollo de la investigación, no obstante, es necesario la elección de un conjunto en base a las condiciones disponibles y o establecidas para un óptimo funcionamiento. Éstas van desde un conjunto de bibliotecas de programación hasta IDEs de desarrollo, pasando por sus respectivos lenguajes de programación.

Open CV

Es una librería creada en lenguaje c, por lo que su uso es totalmente libre ya sea para un uso tanto comercial como no comercial, por otra parte, cuenta con un aproximado de 300 funciones dirigidas fundamentalmente a la visión por computador en tiempo real. Entre las funciones básicas de la librería se encuentran: el procesamiento de imágenes, análisis de movimiento, reconstrucción de modelos 3D, entre otras [15].

Librería Dlib

La librería Dlib es un conjunto de herramientas diseñada en C++, esta biblioteca contiene algoritmos de aprendizaje automático y diversas herramientas para el dominio de la visión por computadora, además de poseer una licencia de código abierto, lo que facilita su uso y posible replica de ser necesario [16]. Algunas de las grandes ventajas que presenta la librería, radican en su documentación, la misma que incluye una gran lista de proyectos de código abierto, además de incluir documentación precisa acerca de las clases y funciones disponibles, también cuenta con diferentes modos de depuración disponibles enfocados a la mayoría de los errores.

C++

El lenguaje denominado como C++, nace de la evolución del tradicional lenguaje C, conservando su gran potencia; a pesar de poseer un gran tiempo en el mundo tecnológico sigue siendo uno de los lenguajes más demandados en el 2019. Este lenguaje es utilizado por lo general en proyectos de relación matemática, redes neuronales e incluso en implementaciones con bibliotecas de Python como TensorFlow [17], siendo esta una de las características por las cuales la aplicación de este lenguaje es recomendada en sistemas para reconocimiento facial mediante la visión por computadora, sin embargo el uso de este lenguaje ha contribuido mucho a otros lenguajes como Python e incluso Java mediante el desarrollo de conjuntos de bibliotecas enfocadas a tareas específicas, siendo una de ellas la visión por computadora y el procesamiento de imágenes [18].

Python

Python como el intérprete y entorno de programación que más se ha destacado en los últimos años, esto gracias a su sintaxis limpia, la misma que ofrece un código fácil de comprender, además de poder ser ejecutado en diversas plataformas; sin olvidar su facilidad para ser integrado con otros lenguajes de programación [19]. Gracias a la flexibilidad que posee y la gran adaptabilidad, Python es utilizado en investigaciones relacionadas ampliamente con el campo de la visión por computadora, siendo la gran cantidad de librerías disponibles una de las mejores características que éste posee [20].

Python se emplea bajo una licencia denominada “Python Licence”, la misma que ha obtenido una certificación por parte del grupo OPEN SOURCE, y a su vez es compatible con la (GNU Public Licence) de la fundación del Software Libre, principal motivo por el cual es ideal para realizar tanto software libre como software privado [21] .

Java

Java, es un lenguaje de programación con un gran alcance a nivel mundial, publicado por primera vez en 1995, su implementación se extiende desde aplicaciones de dispositivos móviles como celulares, laptops, smartwatch, hasta robustos sitios webs [22]. En la actualidad java es demandado gracias a su rapidez, fiabilidad y seguridad. No obstante, las aplicaciones desarrolladas con este lenguaje necesitan de forma obligatoria la base de plataforma de java para poder ser ejecutadas, siendo este su punto más desfavorable en ciertos campos frente a Python. Sin embargo, la compatibilidad que posee Java con otros lenguajes de programación, le posiciona como una buena herramienta para el procesamiento de imágenes y aplicaciones de algoritmos de inteligencia artificial.

GNU Octave

GNU Octave, es la alternativa gratuita frente a Matlab, al igual que su contraparte de paga, implementa un sistema de algebra computacional, lo que le permite ejecutar un procesamiento de imágenes igual de potente que Matlab; compartiendo otra de sus características, como ser multiplataforma y su compatibilidad directa con scripts realizados en Matlab, además de su posible comunicación con otros lenguajes de programación como Python, Java, Lenguaje M y C++ [23].

Matlab

Matrix Laboratory o más conocido como MatLab, es un famoso sistema de cómputo numérico, el mismo que cuenta con su propio lenguaje de programación denominado “Lenguaje M”. El entorno Matlab se enfoca en el análisis iterativo y procesos de diseño, en base a matrices y arrays matemáticos. De forma adicional el sistema permite la creación de interfaces de control que le permiten la comunicación con aplicaciones y o herramientas externas [24]. Una de las herramientas más usadas en el campo de la visión por computador en tiempo real es Simulink; por otra parte, Matlab es compatible con otros lenguajes de programación como Python, java e incluso C, además de tener una documentación practica sobre las funciones que posee [25].

1.1.5. Evaluación de calidad de software

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y sub – características [26].

Normas ISO/IEC

Las normas ISO son establecidas por el Organismo Internacional de Estandarización, con el objetivo de asegurar que los productos y/o servicios alcanzan un punto deseado. Para la sociedad las normas ISO representan una forma de acceder a mercados internacionales y nacionales; en la actualidad existen más de 19.500 normas [27].

Familia de las ISO/IEC 25000

La familia de normas ISO/IEC 25010, también conocidas como las normas SQuaRE o según la traducción de sus siglas (Requisitos de calidad y evaluación de sistemas y programas informáticos), tiene como objetivo principal, el desarrollo de un marco para la evaluación de calidad que puede llegar a obtener un producto de software [28].

El conjunto de normas que conforman a la gran familia de la ISO/IEC 25000, se dividen en cinco grupos, los mismos que han sido el resultado de una evolución de otras normas de calidad como: las ISO/IEC 9126 y las ISO/IEC 14598, siendo estas normas las que se enfocaban en aspectos como los modelos de calidad del producto de software y los procesos de evaluación de productos de software [29]–[31], la descripción de estos grupos los podemos observar en **Tabla 1**.

La división de los grupos se estructura como se muestra en la **Fig. 2**.

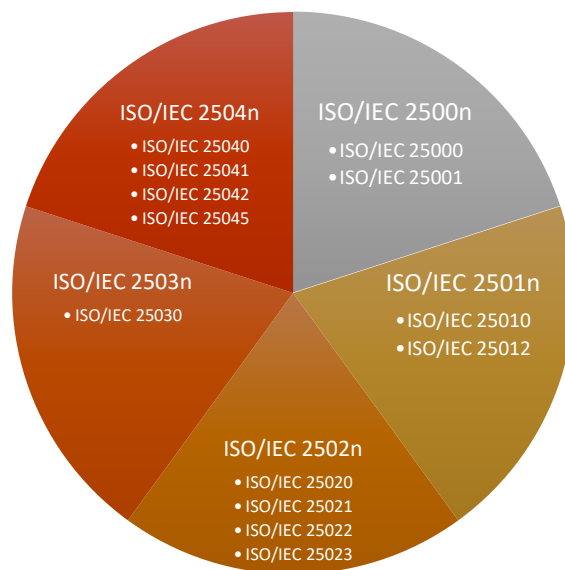


Fig. 2: Familia de la Norma ISO/IEC 25000. Fuente: Elaboración propia.

Tabla 1: Divisiones de la familia de las ISO/IEC 25000.

División	Descripción
ISO/IEC 2500n División de Gestión de Calidad	- El primer conjunto de normas está compuesto por aquellas que definen los modelos, términos, requisitos, e incluso definiciones que se hacen referencias en las demás normas del SQuaRE.
ISO/IEC 2501n División de Modelo de Calidad	- Presentan los modelos de calidad detallando las características, para la evaluación de la calidad interna, como externa además del uso del software.

ISO/IEC 2502n	-	Comprende el modelo que puede ser empleado como referencia, para ser utilizado la evaluar la calidad de los productos de softwares y o sistemas. En este conjunto se encuentran las métricas para la evaluación de calidad de uso, calidad de datos e incluso las métricas recomendadas para un servicio informático.
División de Medición de Calidad		
ISO/IEC 2503n	-	El apartado de las 2503n, comprenden los requisitos de calidad que se pueden ser utilizados en base a los modelos y medidas de calidad. En este conjunto solo se cuenta con la ISO/IEC 25030 denominada Requisitos de calidad
División de Requisitos de Calidad		
ISO/IEC 2504n	-	Este subconjunto de normas contiene directrices, requerimientos y algunas recomendaciones, para realizar la evaluación de un producto, las mismas que pueden ser aplicadas por evaluadores, usuarios finales e incluso por desarrolladores. Las normas incluidas en este grupo son: <ul style="list-style-type: none"> • ISO/IEC 25040 – Modelo y guía de referencia de evaluación. • ISO/IEC 25041 - Guía de evaluación para desarrolladores, adquirentes y evaluadores independientes. • ISO/IEC 25042 – Módulos de evaluación. • ISO/IEC 25045 – Módulos de evaluación para recuperación
División de Evaluación de Calidad		
ISO/IEC 25050	A	Este grupo de normas está enfocado a otras normas e informes técnicos que alcancen puntos específicos o en su defecto que puedan llegar a complementar otras normas que pertenezcan a la familia de normas SQuaRE.
ISO/IEC 25099		

Las normas de la familia ISO/IEC 25000, abarca un amplio campo para la evaluación de la calidad de un producto de software, en **Tabla 2**.

Tabla 2: Descripción de la familia de normas ISO/IEC 25000.

Norma	División	Descripción
ISO/IEC 25000	ISO/IEC 2500n - División de Gestión de Calidad	Conocido como la “Guide to SQuaRE”, proporciona la guía para el uso de los estándares internacionales para la evaluación de calidad que posee un software.
ISO/IEC 25001		La norma “Planning and Management”, indica los marcos necesarios para la orientación y gestión de la evaluación de los productos de software.
ISO/IEC 25010	ISO/IEC 2501n - División de Modelo de Calidad	Esta norma describe un conjunto de características de calidad con las cuales se evalúa la calidad del software.
ISO/IEC 25012		A diferencia de la anterior, esta norma define los modelos para evaluar la calidad de los datos que formar al sistema de información.
ISO/IEC 25020	ISO/IEC 2502n - División de Medición de Calidad	Orienta los modelos de referencias empleados para la medición en cuanto a la calidad del producto de software.
ISO/IEC 25021		La norma ISO/IEC 25021 a diferencia de la norma ISO/IEC 25010, menciona los elementos para la medición de la calidad que son recomendados para elaborar una medición de calidad a un software, ya sea una representación estática, de comportamiento o incluso medir los beneficios que posee el software al momento de ser utilizado.

ISO/IEC 25022		Presenta las métricas definidas específicas para realizar la medición de la calidad referente al uso del producto.
ISO/IEC 25023		Establece las métricas para realizar la evaluación en cuanto a la calidad de los productos y los sistemas software.
ISO/IEC 25024		Complementando a las anteriores, define los parámetros empleados para la evaluación de la calidad de los datos.
ISO/IEC 25030	ISO/IEC 2503n – División de Requerimientos de calidad.	La norma de requisitos de calidad contiene un conjunto de recomendaciones, que pueden ser aplicadas al momento de especificar los requisitos de calidad.
ISO/IEC 25040	ISO/IEC 2504n – División de Evaluación de Calidad.	Esta norma propone algunos modelos para realizar la evaluación, estos modelos consideran algunos aspectos como: restricciones y recursos, los cuales son considerados como datos de entrada para los procesos de la evaluación.
ISO/IEC 25041		En la guía de evaluación para desarrolladores, y evaluadores independientes, se establecen algunas recomendaciones y requisitos que pueden ser aplicados a la evaluación de los productos.
ISO/IEC 25042		De forma adicional a las anteriores, esta norma establece las directrices para la definición de un módulo de evaluación y documentación.
ISO/IEC 25045		Como una extensión de la norma anterior, se establece un módulo, con la finalidad de realizar una evaluación a una característica específica como lo es la “recuperabilidad”

Normativa a considerar estándar ISO/IEC 25010

Esta norma se centra en la evaluación de la usabilidad, la misma que aplica un marco de referencia centrado a la usabilidad, en la cual se determinan algunas características que se debe poseer al momento de evaluar los algoritmos o productos de software.

La norma ISO/IEC 25010 agrupa un conjunto de características que pueden ser aplicadas a distintos tipos de software según sus finalidades. El modelo de calidad que posee la ISO/IEC 25010 representa el punto central en la cual se establece el marco o conjunto de métricas para la evaluación de la calidad del producto. En el modelo se determinan los aspectos de calidad que se establecen a la hora de evaluar las propiedades de un producto software.

El modelo de calidad propuesto por la ISO/IEC 25010 se encuentra conformado por ocho características y sub-características de calidad que se muestran a continuación, en la **Fig. 3**.



Fig. 3: Características de calidad según la norma ISO/IEC 25010 [26].

Entrando en contexto de cada característica, estas son descritas brevemente en la **Tabla 3**.

Tabla 3: Descripción de las 8 características de la norma ISO/IEC 25010 [26].

Característica	Descripción
Adecuación funcional	Se enfoca directamente en calificar la capacidad que posee el software para satisfacer las necesidades en condiciones

	específicas, posee tres apartados de análisis: completitud, corrección y pertinencia funcional
Eficiencia de desempeño	Se enfoca en medir el desempeño del software en base a una cantidad de recursos utilizados, cumpliendo ciertos parámetros o condiciones determinadas. En el caso de esta investigación, si bien el procesamiento y puesta en marcha de los algoritmos requiere de una gran capacidad de cómputo, las condiciones dadas se ajustan a un computador con recursos medios para la ejecución de los distintos algoritmos.
Compatibilidad	Cuando se menciona el apartado de compatibilidad, rápidamente identificamos al hecho de intercambiar y o compartir información con otros softwares e incluso hardware, sin que sea afectado su rendimiento. Dentro de este apartado se evalúa la coexistencia y la interoperabilidad.
Usabilidad	Quizás uno de los parámetros más esenciales desde un punto de vista personal, dado que se enfoca en la evaluación de la capacidad que tiene el software para ser entendido, aprendido y usado por un usuario.
Fiabilidad	Es la capacidad de un sistema para desempeñar las funciones especificadas, en un periodo de tiempo determinados.
Seguridad	Sin duda alguna, mantener la integridad de los datos que son manejados y o generados por el sistema deben mantener un nivel de reserva según las siguientes metricas: confidencialidad, integridad, no repudio, responsabilidad y autenticidad.
Mantenibilidad	En este punto se analizará la capacidad que tienen los algoritmos para ser modificados de forma efectiva y eficientemente, esto considerando las evoluciones

correctivas o perfectivas; también se toma en cuenta las recomendaciones por parte de los desarrolladores de los componentes como el lenguaje (Python, Lenguaje M, C++ o Java) y las diferentes librerías que son usadas para la ejecución de los mismo.

Portabilidad

Cuando se hace referencia a la portabilidad, se trata de evaluar la capacidad que tiene el algoritmo para ser transferido de forma efectiva de un entorno a otro sin generar problemas.

1.1.6. Set de Datos

Los sets de datos o también conocidos como “conjuntos de datos”, o en su traducción al inglés “Data Set”, se definen como un contenedor que incluyen datos que combinan una serie de características disponibles para un estudio o sus relaciones con determinada entidad. También se puede definir al set de datos como esquemas, los cuales que constan de una dimensión denominada (“Clave”) y otra denominada métrica de importación [32].

Algunos de los repositorios de set de datos disponibles, para la búsqueda de todo tipo de información para vuestros proyectos, los tenemos en la **Tabla 4**.

Tabla 4: Set de Datos disponibles en la Web.

Repositorio de Datos	Descripción
DrivenData	<p>Este repositorio de datos de carácter open public, se especializa en proyectos con inserción de la ciencia de datos y el impacto social; analiza áreas en fase de desarrollo como: la salud, educación, investigación, conservación y el área de los servicios públicos [33].</p> <p>A pesar de estar abierto al público, en algunos países como Cuba, Irán, Irak, Corea del Norte, Sudán y Siria, siendo estos sancionados por la OFAC.</p>
Crowd ANALYTIX	<p>Esta plataforma se centra en el modelado y análisis estadístico, enfocados a problemas empresariales, técnicos, logísticos, entre otros [34], al igual que otras plataformas de data sets, permite la posibilidad de que los datos sean analizados tanto cualitativamente, como cuantitativamente.</p>
Signate	<p>Signate al igual que los mencionados anteriormente es un repositorio de datos para Japón, aunque al contrario de otros repositorios de datos, este no cuenta con set de imágenes, dado las políticas establecidas para el funcionamiento del sitio; sin embargo, la información almacenada en el solo será disponible previo registro [35].</p>
Kaggle	<p>En comparación a las otras plataformas de datos, Kaggle es quizás la mejor opción, con alrededor de 1 millón de dólares en premios, es considerada una las plataformas de acceso público más completas, en cuanto a estructura de datos, disponibilidad, formatos, soporte de datos. Kaggle dispone de alrededor de decenas de miles de conjuntos de datos disponibles de forma gratuita [36].</p>

Aunque existen diversas plataformas webs que brindan la posibilidad de emplear sus conjuntos de datos, existen repositorios pertenecientes a universidades, los cuales están enfocados a contribuir de forma gratuita con investigaciones. La Universidad de San Diego es una de estas universidades, la cual en su apartado web [37], presenta una gran variedad de data set públicos, destinados a contribuir con investigaciones académicas.

1.2 Antecedentes

En gran parte de investigaciones realizadas, se muestra el uso, la aplicación y en algunos casos una evaluación para su posterior implementación. Aunque el uso de sistemas de reconocimiento facial ha aumentado durante estos últimos años, no existe una gran variedad de investigaciones enfocadas en su comparación directa, sin embargo, las siguientes investigaciones mencionadas a continuación centraron su enfoque a este motivo.

En la actualidad existen diversas técnicas que son usadas para el reconocimiento facial, estas técnicas suelen enfocarse en la detección de los rostros, en Hernández et al. [38] se analiza el proceso de reconocimiento mediante las técnicas de PCA (*principal component analysis*), LAD (*Linear Discriminant Analysis*), LPP (*Locality Preserving Projections*) y DCT (*Discrete Cosine Transform*) en la cual cada uno de estos algoritmos se caracteriza por una forma de tratar los datos, a medida que cumplen con las etapas del reconocimiento facial, dando como resultados la observación de la dificultad al momento de analizar la comparación de los diferentes algoritmos debido a la forma en la que se presentaron los resultados.

La aplicación de LAD y PCA también la podemos observar en B. Anantha Barathi [39], en la cual se proponen combinaciones de técnicas efectivas como PCA y FisherFace utilizando LDA, a esta combinación se la puede denominar como un algoritmo híbrido, permitiendo que por una parte PCA realice la reducción de las dimensiones y que el algoritmo de FisherFace establezca un método más robusto para la clasificación de las clases sobre las diferentes condiciones de iluminación. Al emplear el algoritmo híbrido permite usar un modelo 3d de las imágenes y convertirlo en un modelo 2d y viceversa haciendo uso de métodos tradicionales obteniendo una valoración de hasta el 99.9 % de efectividad.

S. Rosero Vasquez [40], detalla el análisis cuantitativo de una evaluación realizada a los métodos tradicionales de reconocimiento facial, contra los métodos basados en algoritmos de aprendizaje profundo o también conocidos como “algoritmos basado en machine learning”. A diferencia de otras investigaciones, la investigación explica el entorno de desarrollo el mismo que según el autor catalogó como un entorno “hostil”, debido al uso limitado de recursos para la ejecución de los algoritmos, siendo la gran adaptabilidad una de las características de las técnicas de reconocimiento, esto es, desde los primeros algoritmos basados en modelos matemáticos, hasta los algoritmos conformados por redes neuronales convolucionales, las mismas que se basan en algoritmos de aprendizaje profundo. Yang Kriegman menciona que “la detección de un rostro consistía en determinar la verdadera existencia o no de un rostro humano y de ser así, lograr extraer la ubicación y principales características de éste”. En cuanto a técnicas tradicionales como FisherFace, LBPH (*Histogramas de patrones binarios*), Specific faces (*caras propias*) y por otra parte los algoritmos ANN (*artificial neural network*), CNN (redes neuronales convolucionales) y Open Face. Siendo el resultado del estudio teórico, el algoritmo de caras propias como el mejor, sin embargo, el algoritmo ANN implementado con Tensor Flow, obtuvo el mejor resultado en cuanto a su comparación práctica.

De forma adicional, y como lo mencionan en el estudio anterior, el uso de bibliotecas o también conocidas como API's es esencial en la aplicación y diseño de los algoritmos de reconocimiento facial. N. Boyko [41], detalla que los algoritmos destinados al reconocimiento facial, no siempre cumplen con todas las etapas conocidas para, en otras palabras, no existen métodos y/o tecnologías que sean uniformes, motivo por el cual, analizar los elementos con los cuales se implementan dichos algoritmos es fundamental para lograr una valoración en cuanto a sus aspectos positivos y negativos. A diferencia de otras investigaciones que suelen centrarse en la comparación de los algoritmos, este estudio menciona el uso de un solo algoritmo, KNN, el mismo que se caracteriza por identificar 128 rasgos faciales numéricos de manera única. El algoritmo fue implementado haciendo uso de las librerías más utilizadas para el manejo de visión por computadora empleando OpenCV y Dlib, de las cuales OpenCV se posicionó como una de las mejores opciones gracias a sus rendimiento y facilidad para ser enlazado con plataformas IoT.

Aunque se mencionó anteriormente que los algoritmos destinados al reconocimiento facial no son uniformes para una evaluación, M. Rajalakshmi [42] se comparan los principales algoritmos de machine learning: SVM (*support vector machine*), KNN (*k-vecino más cercanos*), random forest, árbol de decisión y el algoritmo de Naïve Bayes; aunque se tiene en cuenta que cada algoritmo obtendrá un resultado diferente debido a que los mismos son implementados con funciones o técnicas diferentes, motivo por el cual diversas investigaciones se enfocan en comparar a dichos algoritmos. En las pruebas realizadas, se mostró que los algoritmos mejor puntuados fueron SVM y KNN, estando ubicados sobre el 90% de efectividad. Normalmente estos algoritmos son implementados en los algoritmos de EigenFace, FisherFace y/o Viola-Jones, en algunas ocasiones con la intención de maximizar el rendimiento se establecen combinaciones de sub-algoritmos, pasando a ser considerados algoritmos híbridos.

Otros de los algoritmos más usados para el reconocimiento facial son EigenFace y FisherFace, durante la conferencia internacional de TI D. Wahyuningsih [43], menciona la comparación de estos dos algoritmos; siendo el algoritmo de EigenFace implementado con PCA uno de los menos puntuados con un 80% de efectividad al reconocer los rostros a diferencia de FisherFace, el cual cumplió con un 97% de efectividad, siendo una de las principales características de éste la velocidad con la que se ejecutó el proceso; el algoritmo de EigenFace emplea un sistema que se repite de forma constante al ingresar una nueva imagen al conjunto de datos, sin embargo, FisherFace solo ejecuta una vez el procesos de generación de vectores para luego hacer un barrido en cascada que le permite ahorrar tiempo de procesamiento de los patrones que han sido agregados.

Anteriormente se ha mencionado a FisherFace como un rival de EigenFace, sin embargo en B. W. Yohanes [44], se presenta al algoritmo como una mejora de EigenFace, al utilizar un discriminante lineal, lo que le permite mejorar el tiempo de maximización entre las clases; además menciona al algoritmo patrón binario local, que es empleado para obtener la forma y textura global de una imagen correspondiente a un rostro. En la investigación se valora la precisión, el tiempo de entrenamiento, los cuales son evaluados según la validación cruzada de K-fold, en la cual concluyeron que el algoritmo de FisherFace es una buena opción para aplicaciones de reconocimiento facial en tiempo real a diferencia del algoritmo de Patrón binario local.

El uso de los algoritmos de reconocimiento facial ha sido empleados en distintas áreas, sin embargo como podemos encontrar en A. Kartali [45], la aplicación de los algoritmos para el reconocimiento de emociones, las mismas que se agrupan en cuatro subgrupos: felicidad, tristeza, ira y miedo; siendo la principal tarea la extracción de las características faciales. En esta investigación emplean algoritmos de CNN (redes neuronales convolucionales), HOG (histogramas de gradientes orientados), AFI – CNN siendo este un algoritmo híbrido, y SVM (soporte de maquina vectorial); en la cual después de ser puestos a prueba el algoritmo SVM obtuvo un 96.77% de efectividad en comparación a las otras combinaciones.

Otra de las investigaciones que hacen referencia a la comparación de algoritmos enfocados a la extracción de patrones, en C. Rahmad [46], ponen a prueba a uno de los algoritmos más populares Viola-Jones Haar Cascade Classifier y al algoritmo HOG (Histograma de gradientes orientados), en el cual exponen que el algoritmo de Viola-Jones obtiene una puntuación del 75.33 % frente al 80.22 % que obtuvo HOG, esto debido a que Viola-Jones es capaz de reconocer los rostros en distintas posiciones, escalas, expresiones e incluso con distintas condiciones de iluminación, sin embargo, marcó un fallo al momento de detectar objetos adicionales, como gafas, máscaras u audífonos.

Sin duda alguna el poder establecer un nivel a la calidad que posee un software u algoritmo de implementación, es necesario el uso de un sistema de evaluación o en su defecto un marco de evaluación, el mismo que debe estar respaldado por una normativa sólida y fundamentada. La norma ISO/IEC 25010 establece parámetros fundamentados para medir la calidad de un producto de software, en F. A. S. Fierro [47], se establecen algunos argumentos y recomendaciones para el uso de la norma en esta línea de investigación.

1.3 Bases legales

El desarrollo de esta investigación estuvo basado en la implementación de algoritmos creados previamente por otros autores, motivo por el cual se tomó como aspecto fundamental el manejo de licencias disponibles para hacer uso de software que estos poseen.

Los algoritmos empleados en la investigación están protegidos bajos los lineamientos de Free software, lo que implica que están abiertos a mejoras, como lo indica la organización GNU.

“«Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software” [48], [49].

El banco de imágenes o set de datos empleado en la investigación para las respectivas pruebas se encuentra protegido mediante el licenciamiento del sitio web que aloja los datos, el mismo que es conocido como Visión perteneciente a la Universidad Católica de San Diego, siendo este un repositorio para una gran variedad de datasets, los mismos que son protegidos por las políticas internas del sitio y por un tipo de licencia de open source, la que permite su manipulación para proyectos personas, internos y no comerciales [50].

En el Ecuador tenemos el Art.1 del registro oficial N°-320 de la Ley de Propiedad Intelectual, el mismo que menciona lo siguiente: “El Estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las Decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes en el Ecuador [51], lo que permitió usar los algoritmos para su implementación y seguida evaluación.

CAPÍTULO II: MATERIALES Y MÉTODOS

2.1. Delimitación de la investigación

Esta investigación fue desarrollada en la ciudad de Esmeraldas, Ecuador, explícitamente para evaluar la calidad de los algoritmos Viola-Jones, EigenFace y FisherFace en base a algunos parámetros pertenecientes a la ISO/IEC 25010, haciendo referencia a la calidad del producto de software, mediante el desarrollo y elaboración de pruebas de laboratorio cuasiexperimentales por el autor; las mismas que fueron realizadas en un tiempo aproximado de 10 meses del año 2020.

2.2. Tipo de investigación

La siguiente investigación fue realizada usando diferentes técnicas de investigación, las cuales se mencionan a continuación: exploratoria, cualitativa, descriptiva y cuantitativa.

La investigación exploratoria, según Claire S. [52], "es una búsqueda de información la cual tiene el propósito de formular problemas e hipótesis para una investigación más profunda de carácter explicativo.". Este tipo de investigación nos permite familiarizarnos de forma más sencilla con la extensa documentación de los diversos algoritmos que se encuentran a disposición y así poder establecer las características, conceptos y variables de cada algoritmo a estudiar.

Por otro parte la investigación cualitativa, "tiene como objetivo la recopilación de información ya sea como experiencia personal, historia y/o textos observacionales" [53], [54], esta forma de investigación, se empleó como complemento a la investigación exploratoria, dado que su enfoque se centra en la comprensión de las técnicas, métodos y estructuras que poseen los algoritmos para poder ser implementados.

La investigación descriptiva, es considerada como una investigación de segundo nivel, dedicada a la recopilación de datos e información [52], según lo menciona Carlos Sabino [55], "la investigación descriptiva, se enfoca en la descripción de las características fundamentales, mediante el uso de criterios sistemáticos que permiten establecer una estructura o comportamiento y así obtener información comparable con otras fuentes de investigación". Esta técnica nos permite poder enfocarnos en aquellas

características o parámetros más relevantes de los algoritmos que son estudiados, y a su vez compararlos con la información obtenida previamente de estudios históricos.

La investigación cuantitativa, “se presenta como el análisis que se da a un conjunto de características esenciales para la medición e interpretación a partir de métodos estadísticos” [56], este tipo de investigación nos permite la evaluación de las características que son más relevantes para la medición en cuanto a la calidad del producto de software, las mismas que ya son establecidas por la ISO/IEC 25010.

2.3. Métodos de investigación

Durante la investigación se emplearon distintos métodos de investigación, los cuales han sido esenciales para la obtención de los resultados, y conclusiones sobre la misma.

El método inductivo, el cual “pasa de enunciados singulares como el resultado de observaciones, a enunciados universales como teorías, hipótesis o conclusiones”[57]. Siendo este método, el empleado al momento de la recolección de los datos esenciales de investigaciones anteriores referente a la evaluación de los algoritmos empleados a la extracción de características y reconocimiento faciales; siendo estos relevantes debido a la construcción de las bases para el marco de evaluación.

Por otra parte también fue empleado el método analítico, el mismo que “divide todo el proceso de investigación en partes, con el objetivo de estudiarlas, analizarlas por separado” [57]. Con este método se dividió la investigación de forma que, en la primera parte, se establecieron los trabajos previos, lo que ayudo a entender las características de los algoritmos a estudiar; para continuar con el estudio de la ISO/IEC 25010 y la selección de las características que conformarían el marco de evaluación, y para finalizar con el estudio y la selección de los diferentes algoritmos que cumplieran con la extracción de las características faciales.

Otro de los métodos empleados, es el método deductivo, el mismo que al contrario del método inductivo, va desde lo general a lo particular, lo que permitió llegar a las diferentes conclusiones al momento de realizar la evaluación de cada algoritmo y por último forma la conclusión de general.

No obstante, otro método fundamental para la investigación es el método experimental, esto debido a que para poder realizar la evaluación y obtener las

conclusiones de cada algoritmo se realizó un entorno de desarrollo virtual compuesto por el software necesario para la implementación de los algoritmos.

2.4. Técnicas e instrumentos de recolección de datos

Las técnicas empleadas en la investigación fueron la observación, la misma que nos permite observar los hechos, obtener la información de ellos y generar las conclusiones en base a la información obtenida. Por otra parte, también se empleó una ficha de evaluación siendo esta presentada en el **Anexo 1: Ficha de evaluación**, la misma que recopiló los datos de la observación al momento de desarrollar la implementación de los algoritmos.

2.5. Técnicas de procesamientos de análisis de datos

El análisis de los datos obtenidos en la investigación, se realizaron bajo la aplicación de la estadística descriptiva, con la que se estructuró una vista simple de los datos recopilados por cada uno de los aspectos evaluados presentados en la **Fig. 4**. Esta técnica permitió la graficación de los resultados en un diagrama de barras como se observa en la **Fig. 14**

2.6. Materiales

Marco de evaluación según la norma ISO/IEC 25010

El método de evaluación empleado en la investigación se basó en los parámetros de la ISO/IEC 25010, siendo las características seleccionadas las siguientes como podemos observar en la **Tabla 5**.

A continuación, se establece en forma resumida los parámetros a evaluar en base a la norma ISO/IEC 25010, estos parámetros establecerán el marco de evaluación para esta investigación **Fig. 4**.

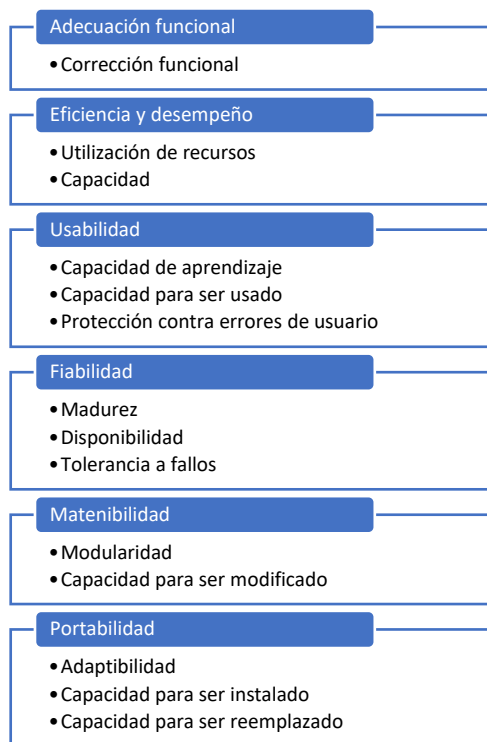


Fig. 4: Criterio de Evaluación para los Algoritmos de Reconocimiento Facial en base a la norma ISO/IEC 25010. Fuente: Elaboración propia

Con la finalidad de establecer una mejor interpretación de los resultados, se implementó una tabla de valoración para los algoritmos; la valoración del tipo escalar como se muestra en la **Tabla 5**.

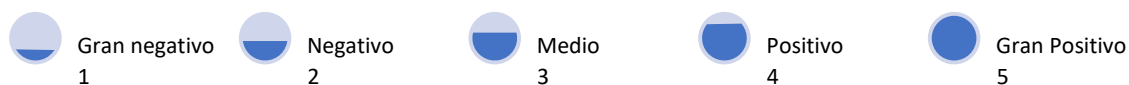


Fig. 5: Valoración escalar para las características del marco de evaluación. Fuente: Elaboración propia

Las pruebas realizadas a los distintos algoritmos se enfocan al ejecutar los módulos de forma independiente, sobre una misma plataforma, siendo esta una característica importante para la evaluación pareja con las mismas posibilidades; se puede denominar a este tipo de prueba como una prueba unitaria; al mismo tiempo se realiza una prueba de

sistema, siendo esta la que asegura la navegación en el algoritmo, el ingreso de datos y por consiguiente el procesamiento de la información. Por otra parte, las pruebas de estrés, volumen, tolerancia, y las pruebas de configuración [58], no lograrían establecerse dado al límite controlado del computador en el que se genera la investigación.

Set de datos de imágenes

El conjunto de datos empleado en la investigación tomado del sitio web de la Universidad de San Diego [37], yale face database o “La base de datos de rostros de Yale” [59], está compuesto por 165 imágenes configuradas a en escala de grises en formato gif y jpg, correspondientes a 11 personas como se aprecia en la **Fig. 6**, de las cuales 15 imágenes corresponden al mismo individuo con diferentes expresiones faciales, tipos de iluminación, además de incluir objetos extras como jafas, barba, etc. Una pequeña parte del conjunto de datos se muestra en **Fig. 7**, en la cual el tamaño difiere 1GB aproximado del conjunto de datos principal a 5.91MB, que también puede ser encontrado en el sitio web anteriormente mencionado.



Fig. 6: Referencia del conjunto de datos yale face [58]

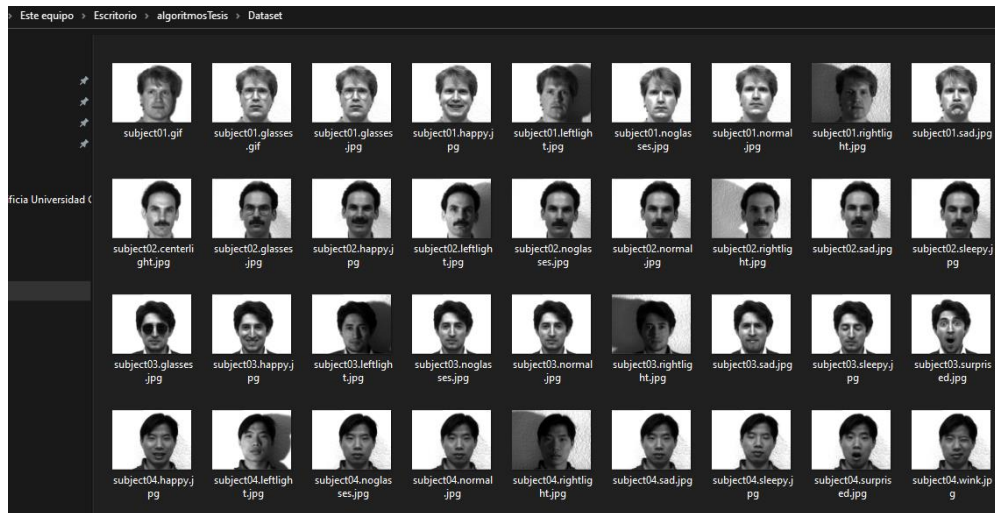


Fig. 7: Parte del dataset de imágenes a utilizar. Fuente: Elaboración propia

Las imágenes para el entrenamiento de los algoritmos contenidas en el conjunto se presentan de siguiente forma: **Fig. 8**, la estructura del nombre de la imagen se aprecia en la **Fig. 9**.

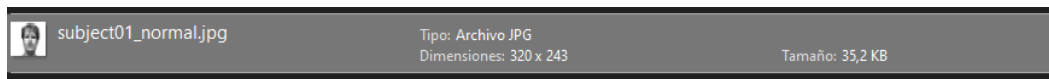


Fig. 8: Formato de presentación de la fotografía para entrenamiento de los algoritmos. Fuente: Elaboración propia

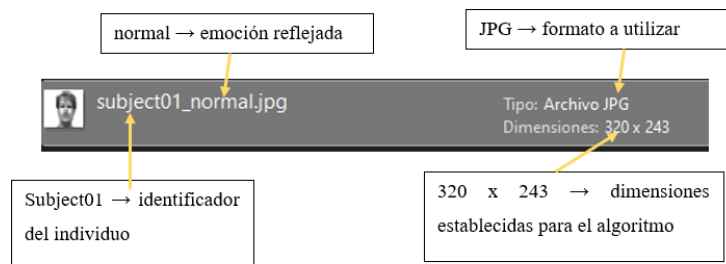


Fig. 9: Especificaciones de la imagen a utilizar. Fuente: Elaboración propia

2.7. Matriz de variables e indicadores

En la presente investigación se empleó un conjunto reducido de las variables e indicadores que conforman la ISO/IEC 25010, las mismas que se muestran a continuación en **Tabla 5**.

Tabla 5: Variables e indicadores de evaluación

<i>Variable</i>	<i>Descripción</i>	<i>Dimensiones</i>	<i>Indicadores</i>	<i>Descripción</i>	<i>Tipo de variable</i>
Calidad de un producto de software	Nivel de desempeño, que poseen las distintas características, que conforman el producto de software.	Adecuación Funcional	Corrección funcional	evalúa la capacidad del sistema para generar resultados correctos con un grado de precisión alto.	Cuantitativa
		Eficiencia de desempeño	Utilización de recursos	Evalúa la cantidad de recursos y tipos de recursos que son utilizados cuando el software está en funcionamiento bajo las condiciones establecidas.	Cuantitativa
			Capacidad	En este subapartado se evalúa que los límites máximos de un parámetro cumplen con los requisitos	Cuantitativa
		Usabilidad	Capacidad de aprendizaje	Evalúa la capacidad que tiene el software para que el usuario comprenda la aplicación.	Cuantitativa
			Capacidad para ser usado	Mide la capacidad que tiene el software que permite ser controlado por el usuario.	Cuantitativa
			Protección contra errores de usuario	Hace referencia a la capacidad del sistema para proteger a los usuarios de errores que pueden ser originados por ellos mismos.	Cuantitativa

	Fiabilidad	Madurez	Mide la capacidad del sistema para satisfacer la necesidad de fiabilidad bajo condiciones normales.	Cuantitativa
		Disponibilidad	Este parámetro evaluará la capacidad de estar operativo y accesible cuando se requiera.	Cuantitativa
		Tolerancia a fallos	Evaluar la capacidad del sistema para operar según lo previsto en presencia de fallos de hardware o software.	Cuantitativa
	Mantenibilidad	Modularidad	Mide la capacidad del sistema y los componentes que permiten realizar cambios o mejoras y que estos tengan un impacto mínimo sobre los demás componentes del sistema o en este caso los componentes del algoritmo.	Cuantitativa
		Capacidad para ser modificado	Este punto se encuentra muy ligado al modularidad, puesto que se evalúa la capacidad que tiene el algoritmo para ser modificado de forma efectiva y eficiente sin interferir en su desempeño o añadir defectos en el mismo, y por ende en su rendimiento y resultado.	Cuantitativa
	Portabilidad	Adaptabilidad	Es la capacidad que tiene el algoritmo para ser adaptado de forma efectiva y eficiente a diferentes entornos para su posterior uso.	Cuantitativa

		Capacidad para ser instalado	Mide la facilidad con la que el algoritmo puede ser instalado o desinstalado de forma exitosa en un determinado entorno.	Cuantitativa
		Capacidad para ser reemplazado	En este punto se establecerá la capacidad que tiene el algoritmo para ser utilizado en lugar de otro algoritmo determinado para el mismo propósito y en el mismo entorno; teniendo en cuenta que los al tener tres algoritmos dedicados para la misma tarea, es uno de los puntos más destacados de todos.	Cuantitativa

CAPÍTULO III: IMPLEMENTACIÓN

3.1. Selección de los algoritmos a evaluar

Al determinar cada uno de los diferentes algoritmos disponibles para ejercer la tarea de extracción de las características faciales, en la presente investigación se evaluaron los algoritmos Viola-Jones, FisherFace y EigenFace. Los algoritmos mencionados anteriormente podrían llegar a considerarse como los algoritmos tradicionales; esto debido que los algoritmos esta enfocados directamente a esa tarea, a diferencia de los otros estudiados anteriormente, dado que su uso puede ser adaptado a otras tareas que incluyan la extracción de patrones de imágenes.

La selección de estos algoritmos sobre SVM, PCA, LDA, KNN y otros se debe a que los mismos forman parte o sirven de base para el desarrollo de los algoritmos seleccionados, como se indica, el algoritmo PCA, es empleado en el algoritmo EigenFace, SVM en el Algoritmo FisherFace en conjunto con un gran proceso estadístico. Además, los algoritmos seleccionados han presentado excelentes resultados en investigaciones anteriores con porcentajes muy altos en efectividad, correspondiente al reconocimiento de los rostros.

Cabe destacar que los algoritmos como EigenFace pueden ser implementados con distintos sub - algoritmos, entre ellos KNN, SVM, LDA e incluso PCA [68].

3.2. Algoritmos para la extracción de características para reconocimiento facial

La mayor parte de las investigaciones realizadas enfocadas al reconocimiento facial, se enfocan directamente la evaluación del algoritmo en general, es decir al resultado del procesamiento de imagen, identificación, asignación, y reconocimiento del usuario a partir de una imagen, sin embargo, existen diversos algoritmos que se enfocan directamente a una fase del proceso, la cual es la extracción de características del rostro para posteriormente realizar el proceso de asignación y reconocimiento.

Con la finalidad de establecer un análisis equitativo, se implementarán los algoritmos EigenFace y FisherFace en sus formas que implementan el modelo de PCA, el mismo que

es utilizado por el algoritmo de Viola-jones al ser implementado con la biblioteca de OpenCV.

3.1.1 Algoritmo PCA (Principal componentes básicos)

El algoritmo PCA o también conocido por su traducción como análisis de componentes principales, es una técnica estadística que ordena los datos según su nivel de importancia; de igual forma elimina los datos menos importantes, reduciendo la cantidad de datos necesaria para el procesamiento de la información.

El uso de PCA, permite crear modelos que pueden representar comportamientos de las variables, esto por medio de la reducción de las dimensiones del conjunto de datos, en donde se eliminan características que poseen muy poco o nula información relevante [62].

El método PCA busca como principal objetivo extraer un subespacio en la cual se maximice la base de la varianza correspondiente al espacio de la imagen original, lo que da paso a la creación de un grupo de vectores que son denominados EigenFace, siendo estos correspondientes a la representación maximizada de la matriz de covarianza de la imagen analizada, consecuentemente el proceso de extracción establece métricas que serán utilizadas para la comparación de las semejanzas que poseen las imágenes. Este proceso de extracción de características es aplicado a otros algoritmos más estructurados como Viola-Jones, EigenFace y FisherFace, generando buenos resultados en comparación a KNN o Naive Bayes entre otros que pueden ser implementados para el funcionamiento de dichos algoritmos.

El entorno de implementación consiste en el lenguaje de programación Python, haciendo uso de la herramienta anaconda, con la opción de jupyter notebook, haciendo uso del comando watermark se muestran las especificaciones del equipo empleado **Fig. 10.**

```
The watermark extension is already loaded. To reload it, use:
%reload_ext watermark
Author: Menghy Raúl Sánchez

Last updated: Sat Feb 13 2021 09:19:59Hora est. Pacífico, Sudamérica

numpy      : 1.18.5
matplotlib: 3.2.2
cv2        : 4.0.1
skimage    : 0.16.2
math       : unknown
time       : unknown
os         : unknown

Last updated: 2021-02-13T09:20:00.043147-05:00

Python implementation: CPython
Python version        : 3.8.3
IPython version       : 7.16.1

Compiler   : MSC v.1916 64 bit (AMD64)
OS         : Windows
Release    : 10
Machine    : AMD64
Processor  : AMD64 Family 23 Model 17 Stepping 0, AuthenticAMD
CPU cores  : 8
Architecture: 64bit
```

Fig. 10: Descripción del entorno de trabajo para la implementación de los algoritmos. Fuente: Elaboración propia

La implementación de los algoritmos se describe a continuación:

3.1.2 Algoritmo EigenFace

El algoritmo EigenFace se encuentra basado en el algoritmo PCA, como lo indican Turk y Pendand [60], en donde se establece un entrenamiento usando el algoritmo PCA con un set de imágenes de rostros y con un set de imágenes sin rostros. Durante las diferentes ejecuciones se obtuvieron resultados muy favorables con valores del 96% en condiciones de iluminación variada, 85% de efectividad con diferentes direcciones de los rostros, y por último un 64% de efectividad en cuanto a diferencias de tamaños.

El funcionamiento del algoritmo, como se describe en [43], EigenFace, cuando es implementado con el sub algoritmo de PCA, busca reducir las dimensiones y al mismo tiempo encontrar vectores óptimos que permitan la distribución de las imágenes. Con el vector propio generado con la distribución de los vectores se establece la matriz de covarianza que se genera a partir de la imagen ingresada; dentro del proceso se ordenan los vectores propios, de acuerdo con los valores propios de mayor a menor para luego establecer al primer vector para generar el componente principal. Con esto se tiene que cada individuo es un conjunto de vectores generados a partir de una serie de fotografías, las cuales son capturadas con distintas configuraciones de iluminación; se considera que

el proceso de reconocimiento facial inicia con la generación de la matriz de un rostro que es incluido en la matriz, lo que generará un vector medio correspondiente a la imagen, que será incluido en la matriz principal y procederá a comparar los vectores entre imágenes.

Se destaca que el proceso de generación de vector medio se repite cada que se ingresa un nuevo rostro al conjunto de datos.

Para la selección del set de datos se definen las variables de ruta, este proceso se evidencia en el **Algoritmo 1**, esto permite que a futuro pueda entrenarse y testear el algoritmo con set de datos de imágenes diferentes. Este método se utiliza en todos los algoritmos. En el caso de EigenFace se implementan el set de datos de la siguiente forma: en donde la función *listdir* en la línea 2 de la sección del código, se extraerán los nombres y rutas de las imágenes que se encuentran en el directorio y por medio del método *flatten*, en la línea 40 del **Algoritmo 2**, se establece un array que contiene toda la información recolectada anteriormente.

Algoritmo 1: Definición del set de datos para el algoritmo de EigenFace. Fuente: Elaboración propia

```

1 conjunto_datos_path = 'conjunto_datos/'
2 conjunto_datos_dir = os.listdir(conjunto_datos_path)
3 width = 100, height = 100

```

Una vez seleccionadas las imágenes que conforman el conjunto de entrenamiento, se procede a leer dichas imágenes y generar el vector que contendrá la matriz de cada imagen, en el **Algoritmo 2**: Lectura y almacenado de las imágenes de entrenamiento. Fuente: Elaboración propia.. El mismo proceso se emplea con el conjunto de imágenes destinado para las pruebas.

Algoritmo 2: Lectura y almacenado de las imágenes de entrenamiento. Fuente: Elaboración propia.

```

35 training_tensor = np.ndarray(shape=(len(train_image_names),
36                               height*width), dtype=np.float64)
37
38 for i in range(len(train_image_names)):
39     img = plt.imread(conjunto_datos_path + train_image_names[i])
40     training_tensor[i,:] = np.array(img, dtype='float64').flatten()
41     plt.subplot(64,128,1+i)
42     plt.imshow(img, cmap='gray')
43     plt.tick_params(labelleft='off', labelbottom='off', bottom='off',
44                   top='off',right='off',left='off', which='both')
45     plt.show()

```

Una vez generados todos los vectores con las características de cada imagen, se genera una imagen media de todas las que han sido almacenadas en la lista por medio de la función *mean_face.reshape*, como se muestra en la línea 8 de la sección del código en el **Algoritmo 3**.

Algoritmo 3: Generación de la imagen media de los rostros. Fuente: Elaboración propia

```

1   mean_face = np.zeros((1,height*width))
3   for i in training_tensor:
4       mean_face = np.add(mean_face,i)
6       mean_face =
        np.divide(mean_face,float(len(train_image_names))).flatten()
8   plt.imshow(mean_face.reshape(height, width), cmap='gray')
9   plt.tick_params(labelleft='off', labelbottom='off', bottom='off',
10                  top='off',right='off',left='off', which='both')
11  plt.show()

```

Una vez obtenida la imagen media de todas las imágenes ingresadas para el entrenamiento, se procede a la normalización de todos los vectores, este procedimiento se evidencia en el **Algoritmo 4**.

Algoritmo 4: Normalización del vector principal de imágenes. Fuente: Elaboración propia.

```

1   normalised_training_tensor =
        np.ndarray(shape=(len(train_image_names), height*width))
3   for i in range(len(train_image_names)):
4       normalised_training_tensor[i] =
        np.subtract(training_tensor[i],mean_face)
5   for i in range(len(train_image_names)):
6       img = normalised_training_tensor[i].reshape(height,width)
7       plt.subplot(64,128,1+i)
8       plt.imshow(img, cmap='gray')
9       plt.tick_params(labelleft='off', labelbottom='off',
        bottom='off',top='off',
10                  right='off',left='off', which='both')
11  plt.show()

```

Después de generada la matriz media de todas las imágenes del conjunto de datos correspondientes al entrenamiento, se procede a la generación de la matriz de covarianza, en base a los vectores normalizados, esto se puede observar en el **Algoritmo 5**. Este proceso permite generar los vectores eigen, como se muestra en el **Algoritmo 6**.

Algoritmo 5: Generación de la matriz de covarianza. Fuente: Elaboración propia.

```
1 cov_matrix = np.cov(normalised_training_tensor)
2 cov_matrix = np.divide(cov_matrix,8.0)
3 print('impresión de la matriz de covarianza X: \n%s' %cov_matrix)
```

Algoritmo 6: Generación de los eigen vectores a partir de la matriz de covarianza - Algoritmo EigenFace. Fuente: Elaboración propia.

```
1 eigenvalues, eigenvectors, = np.linalg.eig(cov_matrix)
2 print('Eigenvectores de la Cov(X): \n%s' %eigenvectors)
3 print('\nValores Eigen de la Cov(X): \n%s' %eigenvalues)
```

Una vez ordenados los vectores, en forma descendente se muestran las matrices con las características faciales extraídas, las mismas se muestran en forma de imagen, para una mejor comprensión y visualización de los componentes en el **Algoritmo 7**.

Algoritmo 7: Generación de las imágenes resultantes de las matrices con las características faciales extraídas del conjunto de imágenes. Fuente: Elaboración propia.

```
1 for i in range(proj_data.shape[0]):
2     img = proj_data[i].reshape(height,width)
3     plt.subplot(4,8,1+i)
4     plt.imshow(img, cmap='jet')
5     plt.tick_params(labelleft='off', labelbottom='off', bottom='off'
6                     ,top='off',right='off',left='off', which='both')
7     plt.show()
```

3.1.3 Algoritmo FisherFace

El algoritmo de FisherFace es uno de los algoritmos más populares en cuanto a reconocimiento facial, algunas investigaciones han demostrado el grado elevado de efectividad que posee el algoritmo, a comparación de otras técnicas.

Algunas de las características de su efectividad se basan en su codificación, el algoritmo está compuesto por PCA, con el que realiza la reducción del espacio facial para luego aplicar un método denominado como “Discriminante lineal” (*FDL*) o también conocido como LDA (análisis discriminante lineal), lo que le permite obtener las características de toda la imagen [61],[43]. FisherFace a diferencia de EigenFace utiliza el análisis discriminante lineal, con la finalidad de establecer diferencias de las imágenes

de rostros en las clases, de esta forma puede maximizar la relación entre la variación de las clases; por otra parte, este algoritmo también incluye PCA, con el objetivo de reducir la dimensión de los patrones a utilizar y mediante LDA en las características de las dimensiones una vez ha terminado los procesos de reducción con PCA [44].

El proceso de la implementación del algoritmo FisherFace inicia, de una manera muy similar al algoritmo EigenFace, con la asignación del set de datos con las imágenes necesarias; este proceso se muestra en el **Algoritmo 8**, en la cual se observan las líneas 7 y 8 de la sección del código en las que se establecen las direcciones del conjunto de datos; posterior a esto en la sección que abarca desde la línea 11 a la 16, se realiza la lectura de las imágenes y el almacenado en una lista.

Algoritmo 8: Definición del set de datos, lectura de las imágenes de entrenamiento. Fuente: Elaboración propia

```
6      #definicion de rutas del set de datos
7      data_train_A = 'Face_Data/YaleA/facedatatrainsizedA/'
8      data_test_A = 'Face_Data/YaleA/facedataresizedtest/'
9
10     #lectura de los set de datos de entrenamiento
11     for i in os.listdir(data_train_A):
12         image = sk.imread(data_train_A+i,0).flatten()/255
13         im_list.append(image)
14         y_true.append(int(i[7:9]))
15         clwise.setdefault(i[7:9], []).append(image)
16     img_shape = cv2.imread(data_train_A+i,0).shape
```

Una de las particularidades del algoritmo FisherFace, es que el mismo es considerado como una actualización del algoritmo EigenFace, como se observara durante la implementación de este.

Al igual que en el algoritmo de EigenFace en la sección del **Algoritmo 3**, en el **Algoritmo 9** se genera una imagen media de todo el conjunto de imágenes empleado para el entrenamiento del algoritmo.

Algoritmo 9: Generación de la imagen media del conjunto de imágenes. Fuente: Elaboración propia.

```
1 #generamos la media de las imagenes ingresadas
2 avg_face = average_face(im_list,img_shape)
3 plt.imshow(avg_face.reshape(img_shape),cmap="gray")
4 plt.show()
```

Una vez generada la imagen media, se debe generar la matriz de varianza, este proceso inicia con la generación u obtención de los vectores de las imágenes, el fragmento del código en el **Algoritmo 10**, en las líneas de 2 a 6 se generan los vectores de las imágenes para luego en las líneas del 8 a 12 se generan los eigenvectores, siendo estos mismos vectores los que generaron la imagen media mediante el uso del proceso empleado por el algoritmo EigenFace, como se lo menciona anteriormente.

Algoritmo 10: Generación de los vectores de imágenes y los vectores eigen. Fuente: Elaboración propia

```
1 #Definimos las funciones importantes
2 def average_face(X, img_shape):
3     images = np.asarray(X)
4     image_sum = np.asarray([np.sum(i) for i in images.T])
5     avg_face = image_sum.reshape(img_shape)/len(X)
6     return avg_face.flatten()
7
8 def first_k_eigenfaces(A,k):
9     """retornamos el primer k eigenfaces en un vector"""
10    u,e,vh=np.linalg.svd(A,full_matrices=False)
11    u=u.T
12    return u[:k]
```

Una vez que se han generado los eigenvectores, se genera la matriz de proyección la misma que minimizara las matrices para luego maximizarlas. El proceso para generar la matriz de proyección se visualiza en las líneas 15 a 21 del **Algoritmo 11**, y la obtención de las distancias mínimas se evidencia en las líneas 24 a 30.

Algoritmo 11: Generación de la matriz de proyección y distancia mínima de las características.
Fuente: Elaboración propia

```

14     def first_k_eigenvectors(A,k):
15         """ retornamos el primer k fisherface en un vector"""
16         e , u = np.linalg.eig(A)
17         pairs = [(np.abs(e[i]),u[:,i]) for i in range(len(e))]
18         pairs = sorted(pairs,key = lambda x:x[0], reverse = True)
19         v = u[0: ,0:k].real
20         v = v.T
21         return v[:k]
22
23     def NNC(x):
24         proj_img = projection_image(W_opt,x,img_shape)
25         distance = []
26         for j in range(len(projections)):
27             distance.append(np.linalg.norm(x.flatten()-
28             projections[j].flatten()))
29         index = distance.index(min(distance))
30         pred = y_true[index]
31         return pred

```

Con estos valores se genera una clasificación de las clases (estas clases se generan a partir de las diferentes características que son extraídas de las imágenes del conjunto de datos), el proceso se evidencia en el fragmento de código en el **Algoritmo 12**.

Algoritmo 12: Generación de las caras medias por clases. Fuente: Elaboración propia.

```

1     #extraemos la Cara media de cada clase de Datos
2     cls_avg = {}
3     for i in clwise.keys():
4         avg_face = average_face(clwise[i],img_shape)
5         cls_avg[i] = avg_face
6         plt.imshow(avg_face.reshape(img_shape),cmap="gray")
7         plt.show()

```

El algoritmo de FisherFace, separa las características por clases, y entre ellas se selecciona a las características que estén más cerca las unas de las otras, el proceso se visualiza en la **Algoritmo 13**.

Algoritmo 13: Selección de las 9 mejores clases. Fuente: Elaboración propia

```
1 print("Trazado de las 9 mejores Eigenface")
2 fig,axes=plt.subplots(3,3)
3 for i in range(3):
4     for j in range (3):
5         axes[i,j].imshow(W_pca.T[j+i*3].
6             reshape(img_shape),cmap="gray")
7 plt.suptitle("Impresión de las 9 mejores eigenfaces")
8 plt.show()
```

de este modo el algoritmo realiza la búsqueda de la matriz de proyección para luego generar la amplitud de dichas características, esto se puede ver en el **Algoritmo 14**.

Algoritmo 14: Amplitud de las características y proyección de los rostros seleccionados. Fuente: Elaboración propia.

```
1 projections = []
2 for i in im_list:
3     projections.append(projection_image(W_opt,
4         i,img_shape))
5 print("trazar algunas imágenes y sus proyecciones")
6 fig,axes=plt.subplots(2,3)
7 for i in range(3):
8     img=im_list[10*i+24].reshape(img_shape)
9     axes[0,i].imshow(img,cmap="gray")
10    proj_img = projections[10*i+24]
11    axes[1,i].imshow(proj_img,cmap="gray")
12
13 plt.suptitle("fotos originales y sus proyecciones
14     en fisher face space")
15 plt.show()
```

Con esto los algoritmos terminarían sus procesos de extracción de características principales, completando el proceso de un algoritmo basado en PCA (análisis de componentes principales).

3.1.4 Algoritmo Viola Jones

El algoritmo de Viola y Jones es quizás el algoritmo más usado, el algoritmo permite la segmentación de una gran cantidad de rostros, con un muy bajo tiempo de procesamiento. [62]. Este algoritmo también es conocido como el algoritmo Cascada o como el algoritmo de Boosting. Viola y Jones en el 2001 [63], en su investigación reportan un 93.9% de efectividad al reconocer rostros de un conjunto de datos.

El funcionamiento se explica en [46][64], el algoritmo Viola-Jones escanea por primera ocasión en busca de características de un rostro humano, posterior a esto realiza el mismo procedimiento con otras imágenes en busca de los patrones, reduciendo la cantidad de patrones que debe escanear con cada interacción, de esta forma las características o patrones, se acumulan, siendo este el motivo de ser conocido como el algoritmo en cascada. El algoritmo V-J hace uso de clasificadores Haar, los cuales detectan las características de un rostro, por lo general los clasificadores Haar son empleados en sistemas de visión por ordenador con la finalidad de clasificar la intensidad que poseen los píxeles de una región seleccionada.

El algoritmo V-J se encuentra compuesto también por un “impulso adaptativo” o también conocido como AdaBoost, este consiste en una técnica que combina clasificadores de bajo nivel, con la finalidad de generar clasificaciones sólidas que permitan determinar las características faciales, el ciclo empleado para la colección de las características se aprecia de forma sencilla en la **Fig. 11**.

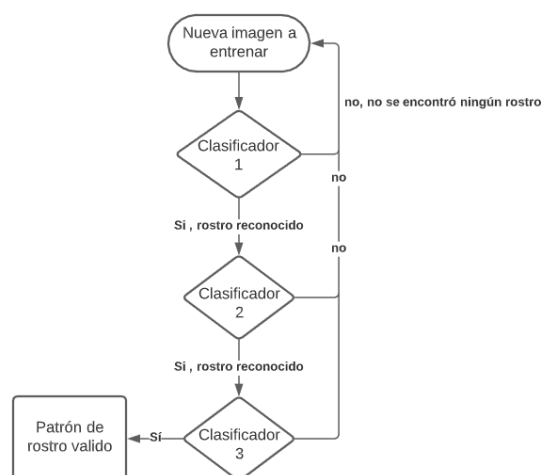


Fig. 11: Funcionamiento del clasificador cascada Viola-Jones. Fuente Elaboración propia.

En la implementación del algoritmo Viola – Jones, esta primera sección se realiza como se muestra en la **Algoritmo 15**. El proceso de la implementación de los ficheros inicia en la línea 5 el comando `os.makedirs()`, se emplea para generar un directorio de forma recursiva, sin embargo, en este caso el comando verifica la existencia de la ruta señalada para la carga del conjunto de datos. La función `glob.glob`, permite la obtención del nombre de los archivos en el directorio.

La sección del código **Algoritmo 15** comprendida desde la línea 7, hasta la línea 17, se encarga de leer las imágenes del conjunto de datos y posteriormente redimensionadas y guardadas en formatos png.

Algoritmo 15: Lectura y almacenado del set de datos - Algoritmo Viola – Jones. Fuente: Elaboración propia.

```
1 import glob
2 import cv2
3 import os
4
5 os.makedirs('dataset/yalefaces/', exist_ok = True)
6 for filepath in sorted(glob.glob('yalefaces/*')):
7     cap = cv2.VideoCapture(filepath)
8     i = 0
9     filepathlist = filepath.split('.')
10    ret, frame = cap.read()
11    filepathlist.append(str(i))
12    while ret != False:
13        filepathlist.pop(-1)
14        filepathlist.append(str(i))
15        i+=1
16        filepath = '.'.join(filepathlist) + '.png'
17        cv2.imwrite('dataset/' + filepath, frame)
18        ret, frame = cap.read()
```

En la siguiente sección de código en la **Algoritmo 16**, se muestra la separación de las imágenes entre un set de entrenamiento y un set de pruebas.

Algoritmo 16: Separación del conjunto de datos de entrenamiento y prueba. Fuente: Elaboración propia.

```
1 from sklearn.model_selection import train_test_split
2 trainList, testList = train_test_split(os.listdir('dataset/yalefaces/'),
3                                     test_size=0.2, shuffle=True)
4 os.makedirs('dataset/test/', exist_ok = True)
5 os.makedirs('dataset/train/', exist_ok = True)
6 for filepath in trainList:
7     filename = os.path.split(filepath)[-1]
8     os.rename('dataset/yalefaces/' + filename, 'dataset/train/'+
9             filename)
10 for filepath in testList:
11     filename = os.path.split(filepath)[-1]
12     os.rename('dataset/yalefaces/' + filename, 'dataset/test/'+
13             filename)
```

Existen diferentes formas para la implementación del algoritmo Viola – Jones, una de las más sencillas se reduce a la implementación con las características brindadas por la biblioteca de Open-CV, en donde se almacenan las características principales, sin embargo, en esta implementación se usó las características básicas para delimitar los cuadros para extraer las características de los rostros del set de datos, **Algoritmo 17**.

Algoritmo 17: Implementación de las características básicas para delimitación. Fuente: Elaboración propia.

```
1 face_cascade = cv2.CascadeClassifier
2 ('haarcascade_frontalface_default.xml')
3 os.makedirs('yaleface/test/', exist_ok = True)
4 os.makedirs('yaleface/train/', exist_ok = True)
```

El algoritmo de Viola – Jones, utiliza la función CascadeClassifier para generar una serie de iteraciones, en donde las características encontradas se van agrupando continuamente, siendo este uno de los procesos más largos que posee el algoritmo, en este proceso inicia con la lectura de las imágenes y por medio de la función *face_cascade.detectMultiScale* se realiza la búsqueda de los rostros u ojos. El proceso es realizado por sectores en los que la imagen es dividida, siendo este el motivo del extenso tiempo de ejecución, esto se aprecia en el **Algoritmo 18**.

Algoritmo 18: Detección y extracción de las características de rostros. Fuente: Elaboración propia.

```
1 for path in sorted(glob.glob('dataset/*/')):
2     img = cv2.imread(path)
3     faces = face_cascade.detectMultiScale(img, 1.3, 5)
4     for (x,y,w,h) in faces:
5         face = img[y-20:y+h+20, x-10:x+w+10]
6         writepath = 'cropped' + path
7         cv2.imwrite(writepath, cv2.resize(face, (50, 50)))
```

Los procesos que poseen EigenFace y FisherFace son muy parecidos, a diferencia de Viola -Jones el cual, una vez delimitados los cuadros de las características a buscar, se procede con la detección, extracción de las características de los rostros, este proceso se puede observar en el **Algoritmo 17** y en el **Algoritmo 18**, finalmente generar un archivo del tipo texto requerido por la librería OpenCV para poder ejecutar el reconocimiento de rostros, proceso que se muestra en el **Algoritmo 19**.

Algoritmo 19: Generación del archivo txt y almacenado de las características - Algoritmo Viola – Jones. Fuente: Elaboración propia.

```
1 from shutil import copyfile
2 os.makedirs('positivesTrain/', exist_ok = True)
3 positiveCount = 0
4 f = open('positives.txt', 'w')
5 for filepath in glob.glob('yaleface/train/*'):
6     copyfile(filepath, 'positivesTrain/'
7         + str(positiveCount) + '.jpg' )
8     f.write('positivesTrain/' + str(positiveCount)
9         + '.jpg ' + '1' + ' 0 0 50 50\n')
10    positiveCount += 1
11    f.close()
```

CAPÍTULO IV: RESULTADOS

Al finalizar las diferentes implementaciones de los algoritmos en un computador portátil con las siguientes características: (procesador: Ryzen 5 2500U, memoria RAM: 16 GB, memoria gráfica: Raedon Vega 8 de 1GB), las cuales son ideales para el manejo de procesamiento de imágenes y video.

Se recalca que cada algoritmo fue implementado usando el algoritmo PCA como su “núcleo”, a pesar de tener disponibles algoritmos con un “núcleo” del tipo SVG o CNN, sin embargo, no todos los algoritmos hallados en la investigación brindaban la posibilidad de poder ser implementados usando dichos códigos, esto inicialmente determinado por el hecho de evaluar algoritmos con una misma estructura, y por consiguiente permitiendo una evaluación equilibrada.

Frecuentemente se encuentra en investigaciones previas, evaluaciones entre algoritmos con diferente implementación, por ejemplo, evaluaciones en donde se muestran una versión de Eigen Face con SVG contra una versión de Fisher Face con PCA [64], y aunque se considera que Fisher Face es la actualización de Eigen Face, la misma estructura no permitía visibilizar los procesos de extracción de las características faciales, siendo este el motivo principal por el cual se decidió emplear PCA como la estructura base de los tres algoritmos evaluados.

La investigación desarrollada por el autor se concertó de la investigación, desarrollo e implantación de los algoritmos de forma individual, con esto se forjaron resultados que se pueden observar en la **Tabla 6**.

Tabla 6: Resultados Evaluación Según la ISO/IEC 25010

Aspectos Para Evaluar	Algoritmos														
	EigenFace					FisherFace					Viola-Jones				
	Valoración					Valoración					Valoración				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
a.- Adecuación Funcional															
Corrección funcional			●												
b.- Eficiencia y Desempeño															
Utilización de recursos				●					●				●		
Capacidad				●					●						●
c.- Usabilidad															
Capacidad de aprendizaje.				●					●				●		
Capacidad para ser usado				●					●						●
Protección contra errores de usuario			●						●				●		
d.- Fiabilidad															
Madurez.				●						●					●
Disponibilidad			●						●					●	
Tolerancia a fallos		●					●					●			
e.- Mantenibilidad															
Modularidad				●						●					●
Capacidad para ser modificado.				●					●					●	
f.- Portabilidad															
Adaptabilidad					●					●					●
Capacidad para ser instalado					●					●				●	
Capacidad para ser reemplazado					●					●					●

4.1 Evaluación de adecuación funcional

Una vez presentados los resultados de la evaluación en base a la observación e implementación de cada algoritmo se puede establecer el motivo de los resultados mostrados. En la primera sección Adecuación funcional, en donde se evaluó el apartado de Corrección funcional, estableciendo puntajes de 3, 4 y 5 respectivamente al orden de cada algoritmo debido a los diferentes algoritmos no lograron obtener resultados equivalentes al 99% de efectividad, siendo estos equivalentes a: EigenFace 86.60%, FisherFace 90% y Viola-Jones 95%, tal como se muestra en la **Fig. 12**.

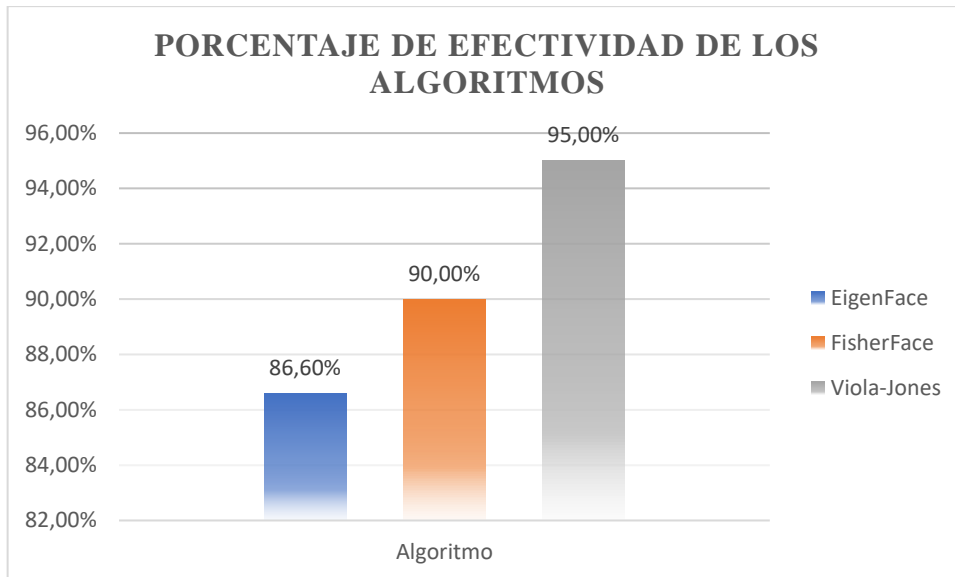


Fig. 12: Porcentaje de efectividad de los algoritmos. Fuente: Elaboración propia.

Es importante señalar que estos valores pueden variar significativamente según algunos factores como: la cantidad de fotografías empleadas para el entrenamiento o generación de las características básicas, además del tipo de algoritmo base o “núcleo” del algoritmo que sea utilizado para su implementación. De forma particular el algoritmo base para la implementación de los algoritmos fue el algoritmo PCA, pese a esto se tiene conocimiento que en otras investigaciones en donde se ha utilizado como base el algoritmo SVM ha generado excelentes resultados.

4.2 Evaluación de eficiencia y desempeño

En la segunda sección correspondiente a Eficiencia y Desempeño, se evaluaron dos parámetros: Utilización de recursos y Capacidad. Con respecto al primer apartado se obtuvieron resultados de EigenFace 4 puntos, FisherFace 3 y por último Viola-Jones 2 puntos; los resultados de utilización de recursos en este caso fueron obtenidos al comparar con el tiempo de ejecución de cada algoritmo, siendo EigenFace 7 minutos, FisherFace 9 minutos y Viola-Jones 25 minutos. Estos resultados se describen en la **Fig. 13**.

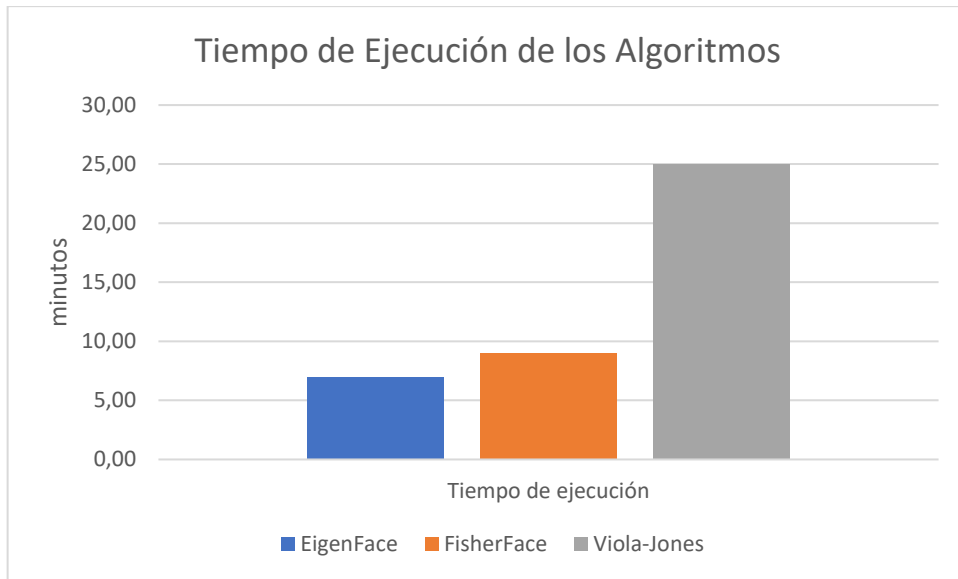


Fig. 13: Tiempo de ejecución de los Algoritmos. Fuente: Elaboración propia.

Se considera que los mismos procesos que se emplean para la obtención de las características es el principal factor para el tiempo de ejecución. El algoritmo EigenFace extrae los vectores de las imágenes, para luego reducir los valores y obtener un vector único con las características del rostro, a diferencia del algoritmo de FisherFace que se extiende a una amplitud de los vectores obtenidos con la técnica de EigenFace, esto con la finalidad de maximizar las características de forma eficiente, motivo por el cual es considerado como una mejora del algoritmo EigenFace. Por último, el algoritmo Viola-Jones se encarga de obtener las características por medio del método cascada, en el cual las imágenes son revisadas cuadro a cuadro identificando las características para luego guardarlas e ir creando una especie de cascada en donde se va optimizando con cada iteración las características extraídas.

Correspondiente al segundo parámetro evaluado la Capacidad, los tres algoritmos llegaron a un empate, con el puntaje de 4, debido a que entre más recursos posea el ordenador en el que se estén ejecutando los algoritmos, se obtendrá una mejor respuesta por parte de los algoritmos. En este apartado también se destaca la premisa propuesta por M. Rajalakshmi [42], en donde menciona que las evaluaciones de distintos algoritmos no son eficientes dado que cada algoritmo tiene una estructura diferente y esta puede influenciar en la cantidad de recursos empleados por el ordenador en el que se estén ejecutando. Siendo también esta premisa de la conclusión del puntaje que obtuvieron los algoritmos en este apartado.

4.3 Evaluación de usabilidad

En la tercera sección de Usabilidad, se evaluaron los parámetros: capacidad de aprendizaje, capacidad para ser usado y protección contra errores. En donde los algoritmos EigenFace y FisherFace obtuvieron un puntaje de 4 a diferencia del algoritmo Viola-Jones el cual obtuvo un puntaje de 2, estos resultados se obtuvieron después de analizar la estructura de los algoritmos siendo el algoritmo EigenFace y FisherFace muy similares, a diferencia del algoritmo Viola - Jones debido a que su proceso de extracción de características se divide en una variedad de subprocesos en cascada.

En el parámetro de Capacidad para ser usado, los algoritmos obtuvieron un valor de 4 puntos, esto debido a que en su mayoría dependiendo de la forma de su implementación, el usuario podrá modificar los sets de datos de entrenamiento y de prueba, en algunos casos como lo indican investigaciones anteriores, también se permite la modificación de las iteraciones para la extracción de las características.

El tercer parámetro de la sección de Usabilidad, se evaluó la protección contra errores del usuario; en este parámetro los algoritmos obtuvieron 3 puntos, debido a que, al cometer un error en alguna sección de los algoritmos, se generara de forma inmediata un error, el mismo que no permitirá que el algoritmo continúe ejecutándose de forma normal.

4.4 Evaluación de la fiabilidad

En lo que respecta a la Fiabilidad, se evalúan los parámetros madurez, disponibilidad y tolerancia a fallos. En el primer parámetro, madurez, en este apartado se ha tomado en cuenta la calidad de las imágenes empleadas para la obtención de las características faciales, y en donde el algoritmo EigenFace se puntúa con 3 a diferencia de FisherFace y Viola - Jones los cuales obtuvieron una puntuación de 5, teniendo muy buenos resultados a en condiciones normales de iluminación.

El segundo parámetro, Disponibilidad, los tres algoritmos empatan al obtener 3 puntos, esto debido a que para su uso deben implementarse ciertas bibliotecas complementarias como: OpenCV, numpy, matplotlib, skimage, math y time, que en su mayoría de veces no se encuentran instaladas en el entorno de Anaconda Python.

Como tercer parámetro a evaluar, la Tolerancia a Fallos, los tres algoritmos obtienen 2 puntos, debido a que el mínimo error en el algoritmo provoca que el mismo emita el mensaje de error y se termine todo tipo de proceso.

4.5 Evaluación de la mantenibilidad

En lo relacionado a la Mantenibilidad, se evaluó el parámetro de modularidad, y la Capacidad para ser modificado. En el primer parámetro el algoritmo EigenFace obtiene 4 puntos, frente a los 5 puntos que obtuvieron los algoritmos FisherFace y Viola – Jones, esto debido a que el algoritmo EigenFace consiste en un solo proceso para la extracción de las características faciales, mientras que el algoritmo FisherFace al igual que el algoritmo Viola – Jones pueden mejorar su rendimiento al adaptarse otros algoritmos de forma complementaria. En otras palabras, estos algoritmos pueden ser modificados hasta el punto de llamarse algoritmos híbridos, siendo conformados por más de dos algoritmos con la finalidad de mejorar su rendimiento.

Al mismo tiempo también se evaluó la Capacidad para ser modificado, en donde todos obtuvieron una valoración de 4 puntos, debido a que su algoritmo base puede ser cambiado, es decir, se puede implementar un algoritmo EigenFace, en base a SVM o PCA, y se obtendrán resultados similares.

4.6 Evaluación de la portabilidad

En torno a la Portabilidad, se evaluaron los parámetros Adaptabilidad, Capacidad para ser instalado y Capacidad para ser reemplazado. En estos parámetros todos han obtenido una puntuación de 5 puntos, dado que cumplen con esos parámetros a la perfección, sin embargo, el algoritmo Viola – Jones en el parámetro Capacidad para ser instalado, obtiene un valor de 4 puntos, esto debido a que el algoritmo es implementado usando por completo la librería de OpenCV, es muy fácil, sin embargo, cuando se implementa de forma que se genere el conjunto de características en el momento se convierte en un proceso muy largo y en ocasiones difícil de implementar.

De forma general se puede ver los resultados de los tres algoritmos en una sola gráfica **Fig. 14**, en este se detalla la puntuación de cada uno de ellos, en la misma, el algoritmo Viola Jones, es considerado como el mejor algoritmo para reconocimiento facial, esto

debido que reúne las características que son exigidas por la ISO/IEC 25010, sin embargo estos resultados son producidos por los componentes que forman los algoritmos, es decir, la disponibilidad de documentación, bibliotecas y otras fuentes de datos que permiten que se tenga la capacidad para lograr manipularlos.

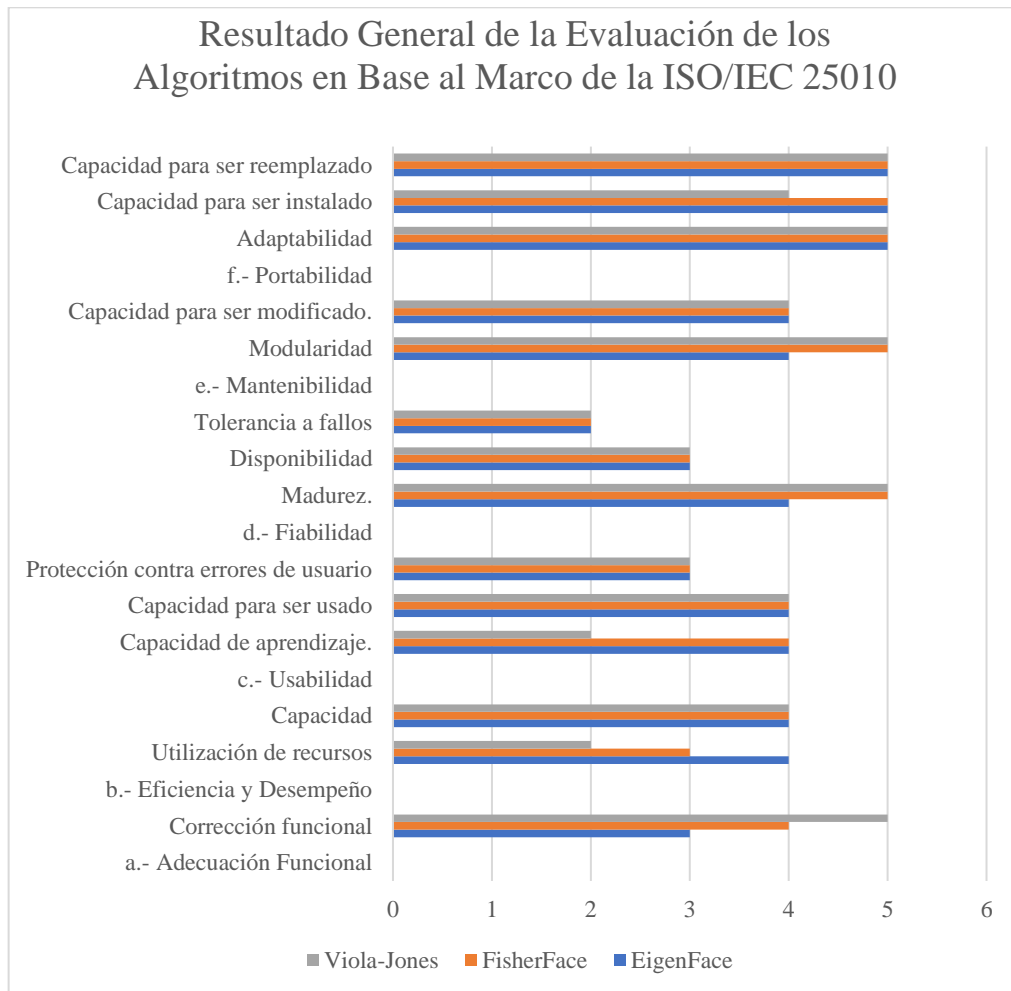


Fig. 14: Resultado general de la evaluación de los algoritmos en base al marco de la ISO/IEC 25010. Fuente: Elaboración propia.

En la **Tabla 7**, se detalla la puntuación de cada algoritmo, en donde se establece al algoritmo FisherFace como el mejor valorado en aspectos de calidad, seguido del algoritmo EigenFace y por último el algoritmo Viola – Jones, sin embargo, el algoritmo Viola – Jones se ubica como el mejor algoritmo en ámbito de funcionalidad en donde obtuvo un 95% de exactitud en el reconocimiento de rostros, seguido por el algoritmo FisherFace con un 90% y en último lugar el algoritmo EigenFace con un 86.6%, estos últimos valores se aprecian en la **Fig. 12**.

Tabla 7: Resultados generales de la puntuación obtenida de los algoritmos

Aspectos Para Evaluar	EigenFace	FisherFace	Viola-Jones
a.- Adecuación Funcional			
Corrección funcional	3	4	5
b.- Eficiencia y Desempeño			
Utilización de recursos	4	3	2
Capacidad	4	4	4
c.- Usabilidad			
Capacidad de aprendizaje.	4	4	2
Capacidad para ser usado	4	4	4
Protección contra errores de usuario	3	3	3
d.- Fiabilidad			
Madurez.	4	5	5
Disponibilidad	3	3	3
Tolerancia a fallos	2	2	2
e.- Mantenibilidad			
Modularidad	4	5	5
Capacidad para ser modificado.	4	4	4
f.- Portabilidad			
Adaptabilidad	5	5	5
Capacidad para ser instalado	5	5	4
Capacidad para ser reemplazado	5	5	5
Puntaje Total	54	56	53

CAPÍTULO V: DISCUSIÓN

En el marco de los resultados que fueron obtenidos en la presente investigación, se observa la implementación y el análisis de la evaluación realizada a los tres algoritmos: EigenFace, FisherFace y Viola-Jones, en donde el algoritmo FisherFace se posicionó como el mejor algoritmo con respecto a su calidad como software obteniendo un puntaje de 56 puntos sobre 70, en la aplicación del marco de la ISO/IEC 25010, sin embargo, cuando se analizó el apartado de funcionalidad el algoritmo que se posicionó como el mejor, fue el algoritmo Viola-Jones al obtener los mejores resultados en cuanto a la extracción de las características con un 95% de efectividad y dejando en segundo lugar al algoritmo FisherFace con un 90% de efectividad. En concordancia con los resultados de investigaciones anteriores en donde los algoritmos obtienen puntajes muy similares.

En la investigación se ha buscado implementar que todos los algoritmos cumplan un mismo marco de implementación, a diferencia de investigaciones anteriores en las que se empleaban algoritmos en sus versiones genéricas, como [38]–[40], siendo estos evaluados solo en su apartado de efectividad. Cumpliendo con esto, la implementación de los tres algoritmos evaluados cumple parcialmente con la propuesta dada por S. Rosero Vasquez en [40], en la cual menciona la evaluación de algoritmos en entornos controlados, en el caso de esta investigación se construyó un entorno virtual con la herramienta anaconda para la implementación de los algoritmos, a su vez el entorno computacional fue el mismo para todos, teniendo la misma cantidad de recursos disponibles.

De forma adicional, otro concepto que se fue aplicado lo expone N. Boyko en [41], en la que menciona que no es correcto realizar una evaluación de varios algoritmos debido a que los mismos no hacen uso de las mismas funciones, siendo este un motivo por el cual se pueden omitir procesos y generar una evaluación desequilibrada; motivo por el cual, se decidió realizar la implementación de los algoritmos empleando el mismo principio del análisis de componente principales o más conocido como PCA; a diferencia del autor que evaluó al mismo algoritmo con diferentes implementaciones.

Cuando se analizó el apartado de la evaluación se tomó muy en cuenta lo mencionado por M. Rajalakshmi en [42], donde mencionaba al igual que otros autores que las evaluaciones siempre generaran un resultado diferente cuando solo se centra en el apartado de la eficiencia, motivo por el cual, el marco de evaluación implementado fue

en base a la ISO/IEC 25010 siendo este un marco estable y analizado para la evaluación de la calidad de todo tipo de producto de software. En este apartado se puede mencionar que el mismo marco de evaluación permitió evaluar características que en investigaciones anteriores no se tomaban en cuenta, como el apartado de usabilidad que poseen los algoritmos e incluso el apartado de modularidad. Al comparar esta investigación se logra demostrar que las evaluaciones de los algoritmos pueden llegar a ser mejor estructuradas cuando se incluyen características fundamentales para un producto de software, en este caso las comprendidas en la ISO/IEC 25010.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Después de haber estudiado e implementado los algoritmos sujetos a estudio, se obtuvo como mejor puntuado al algoritmo de FisherFace en cuanto a calidad, llegando a tener un puntaje de 56/70, seguido por el algoritmo EigenFace con 54/70 y por último el algoritmo Viola-Jones con un puntaje de 53/70, no obstante al momento de evaluar su efectividad, el algoritmo Viola – Jones se puntúa como el mejor de los tres algoritmos con un 95 % de efectividad al reconocer rostros humanos, seguido por el algoritmo de FisherFace con un 90 % y por último el algoritmos de EigenFace con un 86.6 %. Aunque estos valores pueden variar según la cantidad de imágenes usadas para entrenar los algoritmos, un ejemplo que puede considerarse de mucha relevancia en la implementación del algoritmo de Viola-Jones se puede emplear los archivos brindados por la librería OpenCV para su entrenamiento, lo que aumentaría de forma exponencial su puntaje con respecto a la calidad, dado que el nivel de complejidad para ser implementado se reduciría, además de que parámetros como la capacidad de aprendizaje, capacidad de ser usado, protección contra errores de usuario, recursos, madurez, disponibilidad, modularidad y su capacidad para ser modificado, llegaría a cumplir con las expectativas con un posible puntaje de 5 puntos en cada uno de los parámetros. no obstante, en esta investigación se implementó el algoritmo en la forma tradicional, usando como estructura básica el algoritmo PCA, para establecer un equilibrio en los tres algoritmos.

En investigaciones anteriores, ya habían comparado la efectividad de los algoritmos, aunque no directamente entre los tres algoritmos evaluados, en donde FisherFace obtuvo resultados similares en cuanto a efectividad, al igual que EigenFace; en este punto referente a Viola-Jones, el algoritmo en su mayoría de tiempo fue implementado de forma híbrida, en donde los algoritmo como SVM o PCA e incluso el algoritmo de EigenFace complementaban las funciones, con la finalidad de maximizar la eficacia del algoritmo.

Se tiene presente que existen una gran variedad de algoritmos que pueden ser utilizados al momento de extraer las características faciales de una imagen, sin embargo, algoritmos como SVM, PCA, naive bales, randon forest e incluso CNN, son algoritmos básicos que conforman a los algoritmos denominados algoritmos especializados como EigenFace, FisherFace e incluso Viola-Jones. Algunos autores y desarrolladores incluso

combinan más de un algoritmo para generar un algoritmo híbrido, el mismo que busca superar la eficiencia al momento de reconocer el rostro de una persona en una fotografía.

Sin duda, en esta investigación se resalta el algoritmo FisherFace como el mejor puntuado en base a calidad, debido a que, a comparación del mejor puntuado en efectividad, el algoritmo Viola- Jones, posee mayor cantidad de información al implementarse con diferentes métodos, mientras que el algoritmo de Viola-jones es implementado con la guía de la biblioteca de OpenCV en donde ya se poseen conjuntos de datos en formato xml con las características eficientes de los rostros humanos.

6.2. Recomendaciones

Dentro de este tipo de investigación siempre se busca que exista una mejora continua del mismo, por lo tanto, se recomienda a futuros estudiantes, el uso de metodologías con marcos preestablecidos, en el campo del software, no obstante, si se desea desarrollar su propio marco de evaluación no se debe olvidar incluir la calidad del software, esto debido a que, se puede tener un buen código, pero si este no es entendible para otros programadores, no habrá un margen de calidad para que sea implementado en otros proyectos.

Otra recomendación, sería incluir un análisis más profundo para determinar la factibilidad de los algoritmos, implementar escenarios con diferentes circunstancias al momento de capturar las imágenes que se usan como datos para el entrenamiento, esto debido a que el tener control sobre los datos que se ingresan, dará un control más estable sobre los parámetros que se estén analizando.

Académicamente, es importante este tipo de investigaciones, esto debido al uso que se pueden realizar con estos algoritmos, como lo mencione al inicio de la investigación, la aplicación en áreas de seguridad es un tema que nunca dejará de ser importante, y tener herramientas que evolucionen al mismo ritmo que lo hace la tecnología es mucha importancia.

REFERENCIAS

- [1] “Amazon Rekognition – Videos e imágenes – AWS.” <https://aws.amazon.com/es/rekognition/> (accessed Nov. 13, 2019).
- [2] “Universidad Autónoma de Puebla Procesamiento Digital de Imágenes.”
- [3] “Digital Image,” in *Digital Image Interpolation in MATLAB®*, Singapore: John Wiley & Sons Singapore Pte. Ltd, 2018, pp. 17–70.
- [4] Collantes, “Reconocimiento facial de personas basado en el análisis de componentes principales PCA,” *Collantes Arana*, 2013.
- [5] E. M. Cherrat, R. Alaoui, and H. Bouzahir, “Improving of fingerprint segmentation images based on K-means and DBSCAN clustering,” *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 2425–2432, 2019, doi: 10.11591/ijece.v9i4.pp2425-2432.
- [6] “China, el Estado que todo lo ve: así es la red de videovigilancia más grande y más sofisticada del mundo - BBC News Mundo.” <https://www.bbc.com/mundo/noticias-internacional-42398920> (accessed Feb. 28, 2021).
- [7] y N. A. M.H. Yang, D.J. Kriegman, “Detecting faces in images: A survey. Pattern Analysis and Machine Intelligence,” pp. 24:34-58, 2002.
- [8] L. Fabiana, A. Sánchez, D. Dana Urribarri, D. S. Castro, and J. Bosco, “Técnicas biométricas: análisis de las técnicas actuales y nuevas tendencias,” 2019.
- [9] Q. García, C. López, J. Universidad, A. X. El Sabio, C. Quesada García, and J. López Palafox, “Historia de la identificación personal: Desde el reconocimiento facial hasta el ADN dental,” *Biociencias*, vol. 14, no. 1, pp. 1–19, 2019, [Online]. Available: <https://revistas.uax.es/index.php/biociencia/article/view/1268>.
- [10] INCIBE_PTE, “Tecnologías biométricas aplicadas a la ciberseguridad.” Accessed: Jan. 06, 2020. [Online]. Available: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_tecnologias_biometricas_aplicadas_ciberseguridad_metad.pdf.
- [11] M. I. Chacon Murguía and J. B. Jordan, “Fuzzy neural network for document region classification using human visual perception features,” in *Proceedings of the International Joint Conference on Neural Networks*, 1999, vol. 4, no. 3, pp. 2908–2913, doi: 10.1109/ijcnn.1999.833547.
- [12] S. Rakshit, “User Identification and Authentication Through Voice Samples,” in *Advances in Intelligent Systems and Computing*, 2020, vol. 999, pp. 247–254, doi: 10.1007/978-981-13-9042-5_21.
- [13] J. Augusto Cadena Moreano, R. Humberto Montaluisa Pulloquina, G. Alfredo Flores Lagla, J. Carlos Chancúsig Chisag, and O. Alejandro Guaypatín Pico, “RECONOCIMIENTO FACIAL CON BASE EN IMÁGENES.”
- [14] M. N. Murty and V. S. Devi, *Pattern Recognition*, vol. 0. London: Springer London, 2011.

- [15] Bradski Gary and Kaehler Adrian, *Learning OpenCV. Computer vision with OpenCV library*. O'Reilly Media, Inc., 2008.
- [16] “dlib C++ Library.” <http://dlib.net/> (accessed Jul. 23, 2020).
- [17] D. Gupta, A. Khanna, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds., *International Conference on Innovative Computing and Communications*, vol. 1166. Singapore: Springer Singapore, 2021.
- [18] A. Williams and P. Multithreading, “M A N N I N G,” 2012.
- [19] R. González Duque, “Python para todos.”
- [20] “Welcome to Python.org.” <https://www.python.org/> (accessed Jul. 23, 2020).
- [21] C. Holguín, Y. Díaz-Ricardo, and R. Antonio Becerra-García, “Ciencias Holguín, Revista trimestral, Año XX, abril-junio 2014 El lenguaje de programación Python/The programming language Python Ivet Challenger-Pérez,” Accessed: Dec. 05, 2020. [Online]. Available: <http://www.linuxjournal.com/article/2959>.
- [22] “¿Qué es Java y para qué es necesario?” https://java.com/es/download/help/whatis_java.html (accessed Dec. 05, 2020).
- [23] “GNU Octave.” <https://www.gnu.org/software/octave/index> (accessed Dec. 05, 2020).
- [24] “MATLAB Documentation - MathWorks América Latina.” https://la.mathworks.com/help/index.html?s_tid=CRUX_lftnav (accessed Feb. 13, 2020).
- [25] “MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink.” <https://la.mathworks.com/products/matlab.html> (accessed Dec. 05, 2020).
- [26] “ISO 25010.” <https://iso25000.com/index.php/normas-iso-25000/iso-25010> (accessed Jun. 30, 2020).
- [27] “Normas ISO.” <https://www.isotools.org/normas/> (accessed Jun. 30, 2020).
- [28] “NORMAS ISO 25000.” <https://iso25000.com/index.php/normas-iso-25000?limit=4&limitstart=0> (accessed Nov. 30, 2020).
- [29] “ISO - ISO/IEC 9126-1:2001 - Software engineering — Product quality — Part 1: Quality model.” <https://www.iso.org/standard/22749.html> (accessed Nov. 30, 2020).
- [30] “ISO - ISO/IEC 14598-6:2001 - Software engineering — Product evaluation — Part 6: Documentation of evaluation modules.” <https://www.iso.org/standard/24907.html> (accessed Nov. 30, 2020).
- [31] “ISO - ISO/IEC 25000:2014 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE.” <https://www.iso.org/standard/64764.html> (accessed Nov. 30, 2020).
- [32] “Descripción de los conjuntos de datos - Ayuda de Analytics.” <https://support.google.com/analytics/answer/6014980?hl=es> (accessed Dec. 02, 2020).
- [33] “DrivenData.” <https://www.drivendata.org/> (accessed Dec. 05, 2020).

- [34] “CrowdANALYTIX:Solvers Agreement.” <https://www.crowdanalytix.com/jq/solversAgreement.html> (accessed Dec. 05, 2020).
- [35] “IA / Ciencia de datos en formato práctico| SIGNATE Quest.” <https://quest.signate.jp/about> (accessed Dec. 05, 2020).
- [36] “Kaggle: Your Machine Learning and Data Science Community.” <https://www.kaggle.com/> (accessed Dec. 05, 2020).
- [37] “Home | vision.ucsd.edu.” <http://vision.ucsd.edu/> (accessed Jan. 31, 2021).
- [38] R. Gimeno Hernández, J. R. Morros, and R. Barcelona, “ESTUDIO DE TÉCNICAS DE RECONOCIMIENTO FACIAL,” 2010.
- [39] B. Anantha Barathi, “Face Recognition System Simultaneous Using PCA & LDA Method with 3d Morphable Model,” *Middle-East J. Sci. Res.*, vol. 19, no. 6, pp. 869–874, 2014, doi: 10.5829/idosi.mejsr.2014.19.6.1461.
- [40] S. Rosero Vasquez, “Facial recognition: Traditional methods vs. methods based on deep learning,” in *Advances in Intelligent Systems and Computing*, 2020, vol. 1137 AISC, pp. 615–625, doi: 10.1007/978-3-030-40690-5_59.
- [41] N. Boyko, O. Basystiuk, and N. Shakhovska, “Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library,” in *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, Oct. 2018, pp. 478–482, doi: 10.1109/DSMP.2018.8478556.
- [42] M. Rajalakshmi and S. Srividhya, “A comparative analysis of machine learning algorithms used for training in face recognition,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 5, pp. 7351–7356, Sep. 2020, doi: 10.30534/ijatcse/2020/67952020.
- [43] D. Wahyuningsih, C. Kirana, R. Sulaiman, Hamidah, and Triwanto, “Comparison Of The Performance Of Eigenface And Fisherface Algorithm In The Face Recognition Process,” in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Nov. 2019, pp. 1–5, doi: 10.1109/CITSM47753.2019.8965345.
- [44] B. W. Yohanes, R. Diaz Airlangga, and I. Setyawan, “Real Time Face Recognition Comparison Using Fisherfaces and Local Binary Pattern,” in *2018 4th International Conference on Science and Technology (ICST)*, Aug. 2018, pp. 1–5, doi: 10.1109/ICSTC.2018.8528608.
- [45] A. Kartali, M. Roglic, M. Barjaktarovic, M. Duric-Jovicic, and M. M. Jankovic, “Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches,” in *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, Nov. 2018, pp. 1–4, doi: 10.1109/NEUREL.2018.8587011.
- [46] C. Rahmad, R. A. Asmara, D. R. H. Putra, I. Dharma, H. Darmono, and I. Muhiqqin, “Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 732, no. 1, p. 012038, Jan. 2020, doi: 10.1088/1757-899X/732/1/012038.

- [47] F. A. S. Fierro, C. A. P. Manosalvas, N. N. C. Rodríguez, and P. Landeta, "Performance efficiency analysis in augmented reality applications using the ISO / IEC / 25010 standard," *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*, vol. 2019, no. E22, pp. 256–267, 2019.
- [48] "¿Qué es el software libre? - Proyecto GNU - Free Software Foundation." <https://www.gnu.org/philosophy/free-sw.html> (accessed Feb. 12, 2020).
- [49] "gnu.org."
- [50] "Terms of Use | Kaggle." <https://www.kaggle.com/terms> (accessed Jul. 23, 2020).
- [51] "Registro Oficial No 320 Ley de Propiedad Intelectual."
- [52] N. Teodoro and E. Nieto, "TIPOS DE INVESTIGACIÓN."
- [53] Vasilachis de Gialdino Irene *et al.*, *Estrategias de investigación cualitativa*, Primera edición. Barcelona: Editorial Gedisa, S.A., 2006.
- [54] Dr. C. José Ignacio Herrera Rodríguez, Dra. C. Geyce Emma Guevara, and Dr. C. Harold Munster de la Rosa, "Los diseños y estrategias para los estudios cualitativos. Un acercamiento teórico-metodológico," Cuenca, Jul. 15, 2015.
- [55] "Vista de Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción) | RECIMUNDO." <https://www.recimundo.com/index.php/es/article/view/860/1560> (accessed Mar. 01, 2021).
- [56] A. O. Ortega, "ENFOQUES DE INVESTIGACIÓN." Accessed: Mar. 01, 2021. [Online]. Available: <https://www.researchgate.net/publication/326905435>.
- [57] G. E. Barchini, "Métodos 'I + D' de la Informática."
- [58] "Vista de Análisis sistemático de información de la Norma ISO 25010 como base para la implementación en un laboratorio de Testing de software en la Universidad Cooperativa de Colombia Sede Popayán | Memorias de Congresos UTP." <https://revistas.utp.ac.pa/index.php/memoutp/article/view/1483/html> (accessed Dec. 16, 2020).
- [59] "Yale Face Database | vision.ucsd.edu." <http://vision.ucsd.edu/content/yale-face-database> (accessed Jan. 31, 2021).
- [60] M. T. y A. Pentland., "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- [61] M. Anggo and La Arapu, "Face Recognition Using Fisherface Method," in *Journal of Physics: Conference Series*, Jun. 2018, vol. 1028, no. 1, doi: 10.1088/1742-6596/1028/1/012119.
- [62] "Reconocimiento De Expresiones Faciales Prototipo Usando Ica," *Sci. Tech.*, vol. XV, no. 41, pp. 81–86, 2009, doi: 10.22517/23447214.2859.
- [63] P. V. y M. Jones., "Rapid object detection using boosted cascade simple features," *Proceedings of the Computer Vision and Pattern Recognition*. pp. 511–518.
- [64] R. Kvyetnyy *et al.*, "Object detection in images with low light condition," in *Photonics Applications in Astronomy, Communications, Industry, and High*

Energy Physics Experiments 2017, Aug. 2017, vol. 10445, p. 104450W, doi:
10.1117/12.2281001.

ANEXOS

Anexo 1: Ficha de evaluación

Aspectos Para Evaluar	Algoritmos														
	EigenFace					FisherFace					Viola-Jones				
	Valoración					Valoración					Valoración				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
a.- Adecuación Funcional															
Corrección funcional															
b.- Eficiencia y Desempeño															
Utilización de recursos															
Capacidad															
c.- Usabilidad															
Capacidad de aprendizaje.															
Capacidad para ser usado															
Protección contra errores de usuario															
d.- Fiabilidad															
Madurez.															
Disponibilidad															
Tolerancia a fallos															
e.- Mantenibilidad															
Modularidad															
Capacidad para ser modificado.															
f.- Portabilidad															
Adaptabilidad															
Capacidad para ser instalado															
Capacidad para ser reemplazado															

Nota: Se debe marcar con un “●” o una “X” en la casilla que corresponda a la puntuación dada, correspondiente a cada algoritmo. Se tiene en cuenta que 1 corresponde a un “gran negativo” y que 5 corresponde a un “gran positivo”.