TESIS USS

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

ANÁLISIS COMPARATIVO DE MÉTODOS DE RECONOCIMIENTO FACIAL EN PLATAFORMA ANDROID PARA RESOLVER PROBLEMAS DE SEGURIDAD E ILUMINACION EN AMBIENTES NO CONTROLADOS

PARA OPTAR TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor
Br. Luis German Culquichicón Vílchez

Asesor
Mg. Víctor Tuesta Monteza

Ciencias de la Computación

Pimentel – Perú
2018

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Aprobación de la Tesis

Ing. Jaime Bravo Ruiz

Presidente del jurado de tesis

Ing. Heber Ivan Mejia Cabrera

Secretario del jurado de tesis

Ing. Víctor Tuesta Monteza

Vocal del jurado de tesis

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

DEDICATORIA

A mis padres, Jorge Luis Culquichicón Tufinio y Maritza Vílchez Vásquez, a mis hermanos Marcos, Jorge y Yadhira quienes son mi fortaleza y motivación para seguir cumpliendo mis metas.

A los ingenieros que me enseñaron a lo largo de mi vida universitaria de los cuales aprendí y adquirí muchos conocimientos.

A mis compañeros de la Universidad Señor de Sipán que me acompañaron a los largo de mi formación académica profesional y formación personal.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

AGRADECIMIENTOS

En primer lugar y con todo mi corazón a Dios por regalarme bendiciones día a día, guiando mi caminar y regalándome una vida llena de paz, amor, generosidad y humildad.

A mis padres Jorge Culquichicón Vílchez y Maritza Vílchez Vásquez, porque me apoyaron en todo momento y nunca me dejaron solo.

A todos los ingenieros que me enseñaron a lo largo de mi vida universitaria de quienes aprendí mucho, muchas gracias por compartir sus conocimientos y mostrar la calidad y exigencia característica de todo egresado de la USS.

Gracias.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Contenido

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ÍNDICE DE TABLAS

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ÍNDICE DE GRÁFICOS

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

RESUMEN

El presente trabajo de investigación tiene como objetivo principal realizar el análisis comparativos de métodos de reconocimiento facial en sistemas operativos Android para resolver problemas de seguridad e iluminación en ambientes no controlados, esta investigación surge debido al estudio realizado en las bases teóricas científicas, estado del arte y los diversos problemas que existen en trabajos anteriores con la implementación de métodos de reconocimiento facial donde existen métodos que son muy efectivos y procesan el resultado rápidamente pero solo en ambientes controlados, es por esta razón que se plantea realizar un análisis comparativo de dos métodos de reconocimiento facial para saber que método funciona mejor en tiempo real y teniendo en cuenta la iluminación y la seguridad.

Para realizar el trabajo de esta investigación se utilizaron 450 fotos faciales de personas adquiridas de manera propia como muestra para la puesta a prueba de los 2 métodos seleccionados los cuales son el FisherFaces y el Patrón de Binario Local (LBP), además se creó un propio protocolo de adquisición de imágenes, para luego almacenarlas en una base de datos para que de esta manera sean procesadas por los métodos seleccionados.

Para la emisión de resultados se implementó una aplicación móvil en Android Studio con lenguaje Java y con ayuda de los módulos de OpenCv en donde se mide el tiempo de respuesta de cada prueba realizada con ambos métodos y además el porcentaje de éxito y error de cada método.

Al término de los experimentos se concluyó que para evitar problemas de seguridad es más conveniente usar el método FisherFaces debido a que su porcentaje de error es de 40% por lo que es más bajo que el método LBP que emitió un porcentaje de 60%. En cuanto a la iluminación se concluyó que el método LBP es más eficiente y preciso para detectar caras con luz de día debido a que los experimentos arrojaron un 100% como tasa de éxito, mientras que con luz de tarde y luz nocturna un porcentaje de 90%.

Palabras claves: Reconocimiento Facial, ambientes no controlados, FisherFaces, Patrón Binario Local, seguridad, iluminación.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ABSTRACT

The main objective of this research work is to perform a comparative analysis of facial recognition methods in Android operating systems to solve security and lighting problems in uncontrolled environments. This research arises due to the study carried out in the scientific theoretical bases, state of the art and the various problems that exist in previous works with the implementation of facial recognition methods where there are methods that are very effective and process the result quickly but only in controlled environments. For this reason, it is proposed to carry out a comparative analysis of two methods of facial recognition to know which method works best in real time and taking into account lighting and security.

To carry out the work of this research, 450 facial photos of people were used as samples for the testing of the 2 selected methods, which are the FisherFaces and the Local Binary Pattern (LBP), and an own protocol for the acquisition of images, to then store them in a database so that in this way they are processed by the selected methods.

For the emission of results, a mobile application was implemented in Android Studio with Java language and with the help of the OpenCv modules where the response time of each test performed with both methods is measured and also the percentage of success and error of each method. .

At the end of the experiments it was concluded that to avoid safety problems it is more convenient to use the FisherFaces method because its error percentage is 40%, which is why it is lower than the LBP method that issued a 60% percentage. Regarding lighting, it was concluded that the LBP method is more efficient and accurate for detecting faces with daylight, given that the experiments showed a 100% success rate, while in the evening and nightlight a percentage of 90 %.

Keywords: Facial Recognition, uncontrolled environments, FisherFaces, Local Binary Pattern, security, lighting.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación problemática:

Durante los últimos años el uso de “Tecnologías de la Información (TI)” se ha convertido en uno de los campos que más interacción tiene con los seres humanos debido a sus grandes avances tecnológicos. El desarrollo de nuevos Software y Hardware informáticos para campos como la seguridad ha experimentado un gran impulso, tal es el caso de los sistemas biométricos de reconocimiento por huella digital, voz, iris y facial; entre estos, el más prometedor es el reconocimiento facial debido a la idea innovadora que un computador procese imágenes faciales, identifique rasgos y detecte la identidad de una persona. (Akariman, Jati, & Novianty, 2015).

Se sabe que la tecnología del procesamiento de imágenes faciales es un área llena de retos aún sin resolver, más aun cuando se quiere implementar en distintos dispositivos de escritorio o móviles, cabe recalcar que la implementación en este último se ha desarrollado recientemente. (Bertok & Fazekas, 2016)

La importancia del desarrollo de esta tecnología del reconocimiento de imágenes faciales en dispositivos móviles radica principalmente en que la mayoría de personas posee por lo menos un celular en donde pueden llevar a donde quieran su información personal; es por ello que de esta manera los usuario podrán interactuar de manera más continua con esta nueva tecnología del reconocimiento facial, ya sea en sus actividades diarias o en sus actividades empresariales y les sirve de ayuda para automatizar sus procesos.(Bertok & Fazekas, 2016)

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Los autores, Akariman et al., (2015) aportan conceptos en el campo empresarial, planteando que el reconocimiento facial a través de móviles se ha desarrollado ampliamente para el apoyo y el crecimiento de las empresas, por ejemplo en gestionar sus procesos para mejorar la atención al cliente y que la tecnología biométrica es, en la actualidad, un importante recurso que le da a la empresa soluciones de seguridad, monitoreo y control.

Se sabe que los dispositivos de biometría son una tecnología de seguridad usada por sistemas computarizados, primordialmente para reconocer rasgos físicos como atributos faciales, patrones oculares, huellas dactilares, patrones de la voz y patrones de escritura, que están implementados para estándares de seguridad muy altos y múltiples aplicaciones.(Gofman & Mitra, 2016)

La tecnología que más se ha venido desarrollando en la actualidad dentro del área de tecnología biométrica es el reconocimiento facial debido a que es importante en los campos de asistencia, seguridad y vigilancia, es por esto que se necesita obtener imágenes faciales en tiempo real ya que los rasgos faciales brindan información específica de la identidad personal; por estas razones es que la tecnología de reconocimiento facial debería plasmarse en dispositivos prácticos, flexibles, portátiles y de fácil manejo.

A partir de aquí es que los dispositivos móviles se convirtieron en un importante motor para la tecnología biométrica, en especial para el reconocimiento facial, pero antes de implementarla en un dispositivo móvil tendríamos que escoger una plataforma que sea flexible y de fácil manejo para los usuarios, por estas razones la mayoría de investigaciones de reconocimiento facial en dispositivos móviles vienen siendo implementadas bajo una plataforma Android; según Oka Sudana existen 3 razones para que el reconocimiento facial sea implementado en Android: La interfaz

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

de Android es de fácil manejo por parte de los usuarios, Android posee una base de datos interna y gracias a esto no se necesita de otros servidores para almacenar la información, las API'S de Android vienen implementadas con métodos y funciones que ayudan al procesamiento de imágenes y detección de rostros.(Oka Sudana, Darma Putra, & Arismandika, 2014)

Sin embargo, las recientes investigaciones han encontrado que para implementar este tipo de tecnologías existen muchas formas que están siendo actualmente exploradas y diferentes entornos donde se hacen pruebas para medir la efectividad del proceso de reconocimiento facial.

Por ejemplo, Wang Yuanzhi y Luo Xiao (2012) implementaron 3 tecnologías para reconocimiento facial las cuales son: Transformada Wavelet, Análisis de Componentes Principales (PCA) y redes neuronales artificiales (ANN) para entorno controlados y como resultado se encontró un 95.8% de tasa de efectividad del reconocimiento facial.

Pero, por otro lado, se han hecho investigaciones para entornos no controlados y a partir de ello surgen distintos problemas en cuanto a la tasa de efectividad y la seguridad. Por ejemplo, Jie Ni y Rama Chellappa (2010) realizaron una investigación en la cual combinaron 2 métodos: el Algoritmo de línea base y el algoritmo basado en la representación escasa, los cuales están basados en imágenes con entornos no controlados y llegaron a la conclusión que mientras más lejos estaba la persona a la cámara la tasa de efectividad bajaba.

Cada vez que se hacen investigaciones de reconocimiento facial el tema se complica por varios aspectos como la luz de día o noche y luz artificial, esto sin duda afecta a la seguridad y peligro de vulnerabilidad, pero Ravibabu y Krishnan en el año 2015

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

implementaron pruebas para entornos no controlados y se basaron en mejorar la seguridad y problemas de iluminación comparando 3 métodos lo cuales son: Patrón Binario Local (LBP), PCA y Análisis de Discriminante Lineal (LDA), estas pruebas llegaron a la conclusión que el método LBP era más eficiente en cuando a seguridad y a iluminación con luz nocturna. (Ravibabu & Krishnan, 2015). Sin embargo, los investigadores anteriores hicieron pruebas teniendo en cuenta ángulos horizontales del giro de la cara, pero no tomaron en cuenta ángulos verticales que también son importantes.

Como se puede apreciar, hay algoritmos que han tenido resultados aceptables y con una buena tasa de efectividad; sin embargo, otro problema al que también se enfrentan es el problema de rendimientos de software y hardware. Es bien sabido que el rendimiento de software es muy importante debido al consumo de recursos y al tiempo de espera de ejecución en estos dispositivos, por otro lado, el rendimiento de hardware también es importante por el consumo de batería que es consecuente a los procesos del software. Ante todo, ello, los investigadores Yong Hwan Lee, Cheong Ghil Kim, Youngseop Kim, Taeg Keun Whangbo trabajaron en mejorar un método para disminuir el rendimiento de tiempo de ejecución del reconocimiento facial en Android, el cual es Modelo activo de forma (ASM); al momento de modificarlo y mejorarlo obtuvieron resultados exitosos debido al tiempo de ejecución bajo. (Lee, Kim, Kim, & Whangbo, 2013)

Es por todo lo planteado anteriormente que en este proyecto de investigación se planeó realizar un análisis de métodos de reconocimientos facial en entornos no controlados en condiciones de iluminación en tiempo real y aspectos de seguridad para que funcionen bajo la plataforma Android.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1.2. Formulación del problema:

¿Qué método de reconocimiento facial se desempeña mejor en aspectos de seguridad y condiciones de iluminación en tiempo real, para su implementación en dispositivos móviles con sistema operativo Android?

1.3. Delimitación de la investigación:

La investigación se delimita a analizar comparativamente los métodos con mejor desempeño en condiciones de seguridad e iluminación y no discute en las características de orientación de ángulos faciales ni rendimiento del dispositivo.

Otra delimitación que afronta la investigación es que se realizarán las pruebas en equipos con sistema operativo Android y la investigación se limita para la realización de las pruebas desde Android 4.4.

El presente proyecto de delimita temporalmente entre los meses de abril a diciembre.

1.4. Justificación e importancia:

Esta investigación acerca del reconocimiento facial es muy importante y pertinente debido a sus avances tecnológicos y porque pertenece al área de reconocimiento de imágenes e inteligencia artificial, siendo estas dos áreas de mucho interés para la escuela académica e institución donde realizo este proyecto de investigación.

Según el estudio de Ravibabu & Krishnan (2015) el porcentaje de efectividad en cuanto a las pruebas de algoritmos de reconocimiento facial en dispositivos alámbricos e inalámbricos en

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

entornos no controlados expulsan resultados deficientes en cuanto a efectividad, debido a problemas de iluminación y seguridad; todo esto porque un sistema se puede vulnerar alterando la luz y poniendo una fotografía delante del dispositivos que realiza el reconocimiento facial; y añadido a esto resultan aún más vulnerables en dispositivos Android. Es por ello que resulta muy relevante realizar esta investigación basada en una comparación de métodos, para obtener nuevos porcentajes de efectividad al momento de implementar estos algoritmos y ponerlos a prueba.

Con la comparación de métodos en cuanto a seguridad e iluminación se pretende encontrar un método que logre aumentar los niveles de eficiencia en el reconocimiento facial y disminuir el grado de complejidad para la extracción de características o rasgos faciales, clasificación y verificación, que permitan identificar a cada una de las personas durante el proceso de reconocimiento.

1.5. Limitaciones de la investigación:

Entre las limitaciones que se han tenido en el presente proyecto están la captura de imágenes en ambientes no controlados debido a diferentes factores de ambiente, todo esto está ligado a problemas de recolección de datos.

1.6. Objetivos:

1.6.1. Objetivo General:

Analizar comparativamente métodos de reconocimiento facial bajo plataforma Android para resolver problemas de seguridad e iluminación en ambientes no controlados.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1.6.2. Objetivos Específicos:

a) Diseñar un protocolo de captura de adquisición de imágenes.
b) Pre procesar las imágenes capturadas
c) Seleccionar los métodos de reconocimiento facial.
d) Realizar la implementación de los métodos en un lenguaje de programación.
e) Implementar un clasificador para el reconocimiento facial.
f) Realizar pruebas y experimentos de reconocimiento facial.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de estudios:

En el 2004, Ana Belén Moreno Díaz en la Universidad Politécnica de Madrid (España) realizó su tesis doctoral llamada "Reconocimiento Facial automático mediante técnicas de visión tridimensional", en la cual plantea una innovadora solución al problema de vulnerabilidad de la seguridad del Reconocimiento Facial Automático enmarcada dentro de la Visión Tridimensional para que de esa forma se evite engañar al sistema colocando una foto 2D de una persona frente al dispositivo que realiza el reconocimiento facial. En este caso, se trata de reconocer la cara humana, usando para ello información 3D.

Así, se ha profundizado en metodologías que brindan una tasa de efectividad alta en cuanto al reconocimiento facial, y al control de condiciones de adquisición de las imágenes (iluminación, maquillaje, sombras, transformaciones geométricas, expresiones faciales, etc.) que normalmente suelen ser un problema para toda investigación. Los métodos que utilizó son: El Análisis de Componentes Principales (PCA) y Support Vector Machines (SVM), ambos basados en ver la profundidad de la cara y las sombras que causan los efectos 3D. Al momento de hacer experimentos, la autora concluyó que estos dos métodos funcionan a la par para evitar problemas de falsificación y vulnerabilidad del sistema que implemente la tecnología del reconocimiento facial obteniendo un 94% de efectividad en todas sus pruebas de experimento.

En el 2015, Javier Calli Olvea realizó una tesis de bachiller en la Universidad Andina Néstor Cáceres Velásquez de Juliaca con el tema: "Reconocimiento facial basado en el algoritmo eigenface", en

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

la cual el principal problema que encontró era la deficiencia de algoritmos de reconocimiento facial al momento de hacer experimentos en ambientes o entornos reales y la falsificación que se lograba en estos entornos.

El autor de esta tesis presentó un algoritmo de localización y reconocimiento de rostros en imágenes digitales de frente con variación en escala, generando un protocolo de adquisición de imágenes para generar una base de datos de fotografías que permitan ser comparadas contra imágenes con cambio de escala. Se implementó el algoritmo Eigenface el cual permite el reconocimiento de rostros en ambientes con condiciones totalmente realistas y es el más recomendable en cuanto a tiempo de respuesta adecuado en el momento de ejecución. De esta forma, después de experimentar y comparar resultados, el autor llegó a la conclusión que el algoritmo es capaz de reconocer cualquier rostro en una sola imagen la cual es usada posteriormente para reconocer a la persona bajo diferentes condiciones en el ambiente.

En el 2015, Yulian André Cama Castillo realizó una tesis de pregrado en la Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería con el tema: “Prototipo computacional para la detección y clasificación de expresiones faciales mediante la extracción de patrones binarios locales”, en la cual su principal problema era la baja efectividad de algoritmos al momento de detectar distintas expresiones faciales cuando se identificaba la imagen facial. Planteó que existen distintos factores que no ayudan al crecimiento de esta tecnología como la iluminación de la imagen, la cercanía o lejanía del rostro en la imagen, o incluso que el ángulo del rostro (oclusión) afectaba la correcta extracción de las características. Por todo ellos su tesis se centró en el estudio del

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

método Patrón binario local (LBP) para hacerlo más efectivo en cuanto al reconocimiento de expresiones faciales, debido a que este método se basa en la apariencia y textura de la imagen digital.

De esta manera llego a describir las expresiones en el rostro y así pudo clasificarlas entre las emociones básicas mediante el uso de técnicas Boosting de aprendizaje de máquina. Con ayuda OpenCV y su clase “CvBoost”, la cual maneja métodos para la generación de modelos de predicción binarios a través de la especificación de ciertos parámetros, logró el entrenamiento y aprendizaje de máquina mediante la técnica Boosting la cual almacenaba la característica y rasgos más comunes de cada expresión.

En su experimento con 350 imágenes para el entrenamiento del modelo y 35 imágenes para las pruebas llegó a la conclusión que uso del método desarrollado permitió que la primera fase fuera posible en el prototipo al brindar una tasa alta de detección del rostro sobre la base de datos usada, sin embargo por observación se llegó a notar que el método no tolera rotaciones (15 grados como máximo) por lo que para lograr mejorar un sistema de reconocimiento de este tipo es necesario encontrar o desarrollar un método de detección más robusto que lo contemple. Además, el autor concluyó que se pudo confirmar la efectividad del método LBP para tareas de reconocimiento como la realizada en esta tesis, ya que permite representar y acentuar las líneas marcadas por las expresiones en el rostro.

No existen investigaciones en el entorno local donde se está desarrollando esta investigación.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.2. Estado del Arte:

En esta parte se muestran los siguientes artículos de investigación que están relacionados con el tema de reconocimiento facial, y brindan un gran aporte en cuanto a nuevas formas de mejora en este campo tecnológico:

En el año 2010, Jie Ni y Rama Chellappa en la conferencia "Proceedings - International Conference on Image Processing, ICIP" expusieron su artículo investigativo con el tema "Evaluación de la técnica de algoritmos para el reconocimiento facial remoto", en dicho artículo plasmaron que la mayoría de algoritmos de reconocimiento facial se aplican en base a datos con una corta distancia y bajo ambientes controlados, sin embargo al momento de ser usado por usuarios en entornos de la vida real el nivel de efectividad de los algoritmos baja debido a que las imágenes sufren mala iluminación, desenfoque, oclusiones, etc.

A partir de esta problemática desarrollaron y modificaron dos algoritmos para que funcionen de maneras más robusta en el área de reconocimiento facial en entornos no controlados, para ello realizaron un protocolo de adquisición de imágenes bajo diferentes escenarios y ambientes no controlados; a partir de aquí se pre procesan y procesan las imágenes con 2 algoritmos llamados: Algoritmos de línea base y Algoritmos de representación escasa. Estos dos métodos se modificaron para que funcionaran en nuevos ambientes de la vida real. Una vez echo los experimentos comprobaron claramente que la degradación de las imágenes disminuye el rendimiento del sistema, especialmente cuando las caras están ocluidas y muy borrosas, y así concluyeron que la tasa de reconocimiento facial disminuye a medida que las imágenes adquiridas en ambientes no controlados se degradan debido al desenfoque o a la distancia en la que se adquieran las imágenes.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En el año 2012, Wang Yuanzhi y Luo Xioa en el “Journal of Digital Information Management” llevaron a cabo la publicación de artículo científico con el tema “Investigación sobre el Reconocimiento de Rostro Basado en la Transformación Wavelet y PCA-ANN”, ellos plasmaron su problemática diciendo que resultaba complicado distinguir el fondo de la imagen, rasgos faciales que salen con el cambio de la edad, cosas extras que tapen la cara como barba, sombrero, vidrios; aparte de ello las imágenes pueden contener iluminaciones y ángulos distintos. Todo esto afectaba a la tasa de efectividad del reconocimiento facial.

Para esto, diseñaron una nueva tecnología combinando 3 métodos: Transformada Wavelet, PCA y red neuronal ANN. Describieron su método diciendo que las imágenes pasan por un pre procesamiento de normalización de tamaño y balance de grises, luego aplicar Transformada Wavelet para eliminar las partes de baja frecuencia; a partir de aquí se usa PCA y por último se entra la red neuronal ANN para ser enviado a un clasificador, todo esto con ayuda de la BD ORLFACE. Entre sus resultados obtenidos se dieron cuenta que la combinación de estos 3 métodos dio un porcentaje de 95.8% en cuando a tasa de reconocimiento facial, concluyendo que tienen la ventaja de mayor velocidad de identificación, y mayor eficiencia de reconocimiento.

En el año 2014, Oka Sudana, Darma Putra y Alan Arismandika en el instituto “Department of Information Technology, Udayana University” de Indonesia desarrollaron el tema “Sistema de reconocimiento facial en Android utilizando el método Eigenface” para lo cual tuvieron como problemática que la tecnología del reconocimiento facial debería implementarse en un dispositivo portátil, flexible y de fácil manejo para los usuarios, para lo cual hicieron un sistema de reconocimiento facial con entornos no

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

controlados en Android, aplicando el método de EigenFace, un algoritmo basado en la apariencia de la imagen y enfocado en los pixeles, dividiendo el proceso en dos fases, identificación y detección de rostros, ambos con el mismo protocolo de adquisición de imágenes. Sus respectivos resultados con 50 imágenes de prueba y 2500 veces de experimentos, obtuvieron una tasa de exactitud y precisión de 97%, midiendo dos indicadores los cuales son FMR con 2,52% y FNMR con 3%.

En el año 2015, Ravibabu y Krishnan en "IEEE International Conference on Computational Intelligence and Computing Research" mostraron un artículo de investigación llamado "Un enfoque variado de reconocimiento facial con mecanismos verdaderos para Android Mobile contra la falsificación" en el cual plasmaron como problema principal que la tasa de efectividad del reconocimiento facial venía siendo afectada por la vulnerabilidad de la seguridad al momento de mostrar una foto, o un video a la cámara, aparte de problemas con la iluminación de la luz al momento de detectar la cara. Para lo cual hicieron la comparación de 3 métodos para ver cuál funcionaba mejor en distintos aspectos de iluminación y evitar la falsificación de imágenes; esto consistía en la combinación de métodos los cuales son LBP, PCA y LDA , estos métodos están basados en las características faciales de la cara, realizando además su propio protocolo de adquisición de imágenes tomadas en diferentes ángulos de la misma persona, a partir de aquí implementaron los 3 métodos en diferentes aspectos de iluminación con ayuda del detector ORB y BRIEF para confirmar rasgos de vivacidad, luego de hacer los respectivos experimentos se dieron cuenta que los 3 métodos funcionan perfectamente en un ambiente libre, pero que en un ambiente nocturno el método LBP es el más apropiado, y con iluminación de luz interior de día y de noche los métodos LBP y LDA son mejores. De esta manera

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

concluyeron que LBP es un algoritmo que proporciona mejores resultados en cuando a reconocimiento facial en todos los ángulos y con diversas iluminaciones que PCA y LDA.

En el año 2016, Mikhail Gofman y Sinjini Mitra en la revista "Communications of the ACM" presentaron una investigación llamada "Biometría multimodal para mejorar la seguridad de los dispositivos móviles", plasmando un problema complicado en la actualidad sobre la seguridad de información en dispositivos móviles debido a que muchas personas burlaban el bloqueo de patrón, bloqueo facial, bloqueo dactilar, etc y es por eso que necesitaban nuevos mecanismos que no fueran vulnerables ante el engaño de los usuarios. A partir de esta problemática diseñaron e implementaron de un sistema llamado Proteus, basado en un biométrico multimodal de reconocimiento facial y de voz. El reconocimiento facial se desarrolló tomando el método FisherFace basado en la apariencia de rasgo. Formaron su propio modelo de adquisición de imágenes, y para la extracción de características usaron PCA; con la implementación del sistema obtuvieron resultados favorables en cuando al tiempo de reconocimiento con unos 0.108 segundos y por otro lado el indicador EER dio un 2.14 %. De esta manera concluyeron que el sistema Proteus aprovecha las capacidades existentes de hardware de dispositivos móviles y mejora el rendimiento de ejecución.

2.3. Bases teóricas científicas:

Para continuar con este informe de investigación, es necesario tener como base ciertos conceptos relacionados al tema tratado, para que proporcionen un marco de referencia conceptual para la puesta en marcha del presente proyecto.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Es por esta razón que se definirá puntos relacionados al ámbito de la inteligencia artificial, reconocimiento de imágenes, reconocimiento facial, tecnologías y plataformas comunes para desarrollar el reconocimiento facial, algoritmos y métodos que son usados para el desarrollo de esta tecnología, entre otros conceptos.

2.3.1. Inteligencia Artificial:

Según Marvin Minsky en el libro “Inteligencia artificial: modelos, técnicas y áreas de aplicación” (2003) define a la inteligencia artificial como lo siguiente: “Es la ciencia de construir máquinas para que hagan cosas que, si las hicieran los humanos requerirían inteligencia”.

De esto se puede deducir que la inteligencia artificial es la facultad que tienen las máquinas de conocer, entender, comprender y aprender algo.

2.3.2. Imágenes digitales:

Según Hugo Cruzado (2009) en su libro “Imagen Digital, Conceptos básicos” explica que “Una imagen digital es aquella representación bidimensional construida a partir de una matriz binaria, la cual está formada por un conjunto definido de bits; un bit es la mínima unidad de información computada por dígitos binarios.”

Según el tipo de calidad o también según el grado de detalle (resolución) existen 2 tipos de imágenes digitales, las cuales son:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Las imágenes vectoriales; estas son imágenes creadas por programas de diseño por lo tanto tienen una resolución estática y cuando se amplían no sufren efectos de distorsión. Los formatos que poseen este tipo de imágenes son:

AI (Adobe Illustrator)
CDR (Corel Draw)
DXF (Autodesk)
EMF, EPS, ODG (Open Office Draw)
SVG (Inkscape)
SWF (Adobe flash)
WMF (Microsoft)

Las imágenes de mapa de bits; estas imágenes están formadas por un conjunto de pixeles, cada pixel de la imagen contiene información de su tono o luminosidad, pero en formato binario debido a que es el único que entienden los ordenadores. Cabe resaltar que este tipo de imágenes es la más usada en la informática. Los formatos que poseen este tipo de imágenes son:

- BMP (Formato muy antiguo de Microsoft)
- GIF (Admite solo 256 colores por lo que no es adecuado para imágenes fotográficas)
- JPEG (Formato ubicado entre los más conocido y utilizado para fotografías digitales ya que tolera millones de colores)
- PNG (Admite más colores que GIF es posible crear imágenes transparentes con mayor detalle)
- PSD (Formato por defecto del editor de imágenes Adobe Photoshop)
- TIFF(Formato utilizado por escáneres, millones de colores, capas y canales alfa)

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.3.3. Procesamiento de imágenes digitales:

Según José Esqueda y Enrique Palafox (2015) en su libro "Fundamentos para el procesamiento de imágenes" detallan claramente la definición de este término, diciendo que es una tecnología que potencia el aspecto de las imágenes y hace más resaltadas determinadas características que se desea que sobresalgan de las mismas, todo esto con ayuda de métodos ópticos o métodos digitales en un computador.

Es el proceso por el cual las imágenes son sometidas a un procesamiento con el fin de extraer determinadas características o parámetros, o para crear o generar nuevas imágenes procesadas con el material de salida del procesamiento. (Visual Interaction & Communication Technologies, 2009)

2.3.4. Biometría:

"Es una tecnología de seguridad basada en el reconocimiento de una característica de seguridad y en el reconocimiento de una característica física e intransferible de las personas, como por ejemplo la huella digital." (Homini, 2004)

Según Purificación Aguilera (2010), en su libro "Seguridad Informática", plantea que "Los dispositivos biométricos son usados en sistemas computarizados de seguridad, principalmente para identificar atributos físicos. Estos han sido diseñados para máximos estándares de seguridad y múltiples aplicaciones."

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

De estas definiciones se deduce que el término “Biometría” es una tecnología que mide e identifica alguna característica propia de la persona, ya sea de manera ocular, dactilar o facial.

2.3.5. Reconocimiento facial:

Este término es definido como; “Una solución biométrica que emplea un algoritmo automático para verificar o reconocer la identidad de una persona en función de sus características faciales.” (Smowltech.com, 2017)

Según Li & Jain, (2005) en su libro “Handbook of Face Recognition” da a entender que el reconocimiento facial es una tecnología que en los últimos años ha surgido muy activamente, innovando en los campos de seguridad, verificación de personas y entretenimiento informático, pero aún existe desafíos sin resolver y vienen siendo estudiados por varios especialistas en la materia.

En el sitio web Es.mathworks.com (2013) comentan que el reconocimiento facial es “El proceso de identificación de una o varias personas en imágenes o vídeos mediante análisis y comparación de patrones. Los algoritmos de reconocimiento facial normalmente extraen las características faciales y las comparan con una base de datos para obtener la mejor coincidencia. De manera breve deduzco que el reconocimiento facial es una parte importante de muchos sistemas biométricos, de seguridad y de vigilancia, así como de sistemas de indexación de imágenes y vídeos.”

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.3.6. Principales métodos de reconocimiento facial:

a) EigenFaces:

Eigenfaces es construido sobre técnicas de Análisis de Componentes Principales (PCA) introducido por M.A. Turk & A.P. Pentland (1991) en el "Journal Cognitive Neuroscience", quienes describen que es un método holístico basado en la apariencia de la imagen, es decir un método enfocado en los pixeles y que se manipula vectorialmente para sacar el resultado del reconocimiento facial.

b) FisherFaces:

Fisherfaces es un método propuesto por Belhumeur, Hespanha & Kriegman (1996) en su libro "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", quienes describen que se utiliza PCA y el Discriminante Lineal de Fisher (FLD) o también llamado LDA. Este método tiene relación con el método EigenFaces debido a que también trabaja con la apariencia de todo el rostro, pero la diferencia se da cuando se maximiza la relación entre la distribución de clases y la distribución de interclases, porque de esa manera reducen dimensiones de rasgos.

c) Análisis de Componentes Principales (PCA):

Probablemente es la técnica más usada en la rama de procesamiento de imágenes; es una tecnología tradicional de detección de características programadas

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

por el especialista, sin embargo, dentro un espacio para reconocimiento de caras, es probablemente la más utilizada también, cuya función principal es adquirir, mediante fórmulas matemáticas y valores vectoriales, los rasgos faciales más notorios y separarlos de la señal ruido que pueda tener una imagen digital. (Eleyan & Demirel, 2007)

d) Transformada de wavelet:

Es un método matemático que tiene múltiples aplicaciones en el procesamiento de señales y procesamiento de imágenes. Es usada para el control de procesos y detección de patrones. (Lang, 2014)

Según Lang (2014) afirma que: “Es interesante la utilización de las Wavelets en el análisis de imágenes, pues los cambios de regiones o bordes pueden ser detectados más fácilmente.”

La trasformada Wavelet está dada de la siguiente manera:

$$(\text{[illegible]})(\text{[illegible]}) = |\text{[illegible]}|^{-1/2} \int \delta\text{[illegible]}(\text{[illegible]})\psi(\frac{\text{[illegible]}}{\text{[illegible]}})$$

La familia de wavelets se puede construir dilatando y trasladando:

$$\psi^{\text{[illegible]}}(\text{[illegible]}) = |\text{[illegible]}|^{-\frac{1}{2}}\psi(\frac{\text{[illegible]}}{\text{[illegible]}})$$

Entonces la transformada wavelet continua respecto a esta familia de wavelets es:

$$(\text{[illegible]})(\text{[illegible]}) = \langle \text{[illegible]}\psi^{\text{[illegible]}} \rangle$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$(\text{illegible})(\text{illegible}) = \int \delta(\text{illegible}) |\text{illegible}|^{-1/2} \psi$$

La función f puede ser recuperada de su transformada wavelet como sigue:

$$\text{illegible} = \text{illegible}_{\psi}^{-1} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\delta(\text{illegible})}{\text{illegible}^2} \psi(\text{illegible})(\text{illegible}) \psi^{\text{illegible}}$$

e) Análisis de Discriminante Local (LDA):

Este método es una herramienta muy poderosa que es usada para la reducción de las dimensiones y extracción de características en las aplicaciones de reconocimiento de patrones. (Sahoolizadeh, Heidari & Dehghani, 2008)

LDA se basa principalmente en utilizar la información de las muestras de las clases que están etiquetadas para reducir la dimensionalidad del espacio de las características. (Belhumeur, Hespanha and Kriegman, 1997)

Según Poveda & Merchán (2015) explican que lo que busca el método LDA es maximizar la siguiente función:

$$J(w) = \frac{W \text{illegible}_B W}{W^T \text{illegible}_W W}$$

Dónde: SB es la matriz de dispersión inter-clase.

SW es la matriz de dispersión intra-clase.

Siendo más precisos:

$$\text{illegible}_{\text{illegible}} = \sum_{\text{illegible}} \text{illegible}(\mu_{\text{illegible}} - \mu)(\mu_{\text{illegible}} - \mu)^{\text{illegible}}$$

$$\text{illegible}_W = \sum_{\text{illegible}} \sum_{\text{illegible}} (\text{illegible} - \mu_{\text{illegible}})(\text{illegible} - \mu_{\text{illegible}})^{\text{illegible}}$$

Siendo: $\mu_{\text{illegible}}$ La media de cada clase.

μ La media de todos los datos.

$\text{illegible}_{\text{illegible}}$ La cantidad de patrones de la clase c.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.3.7. Redes Neuronales:

Existen muchas maneras de conceptualizar a las redes neuronales, empezando por las más genéricas y finalizando con los conceptos más detallados. Por ejemplo:

Es un sistema de computación compuesto de nodos informáticos usado para clasificar resultados de un determinado procesamiento, esta información es procesada nuevamente por medio de su estado dinámico como respuesta a entradas externas.(Khashman, 2007) Según Eleyan & Demirel (2007) en su libro “PCA and LDA based Neural Networks for Human Face Recognition” definen a las redes neuronales como “Herramientas poderosas que pueden ser entrenadas para realizar una compleja y diversas funciones en aplicaciones de visión artificial, tales como pre procesamiento (extracción de límites, restauración de imágenes, filtrado de imágenes), extracción de características (extracción de características de dominio transformadas), memoria asociativa (almacenamiento y recuperación de información), y reconocimiento de patrones”; de esto se deduce que las redes neuronales aportan en el campos de toma de decisiones al momento de clasificar una imagen facial del resto.

a) Clasificación de Redes Neuronales:

Según el sitio web Redes-neuronales.com.es del 2017, se dan a conocer la siguiente calificación de las redes neuronales:

a.1) Según la topología o estructura de la red neuronal, se distinguen según el tipo de capas (ocultas

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

o visibles), esto es una característica muy importante porque la agrupación de neuronas forma capas, y la agrupación de capas forman a la red neuronal. Existen 2 tipos:

a1.1) Redes monocapas; compuestas por solamente una capa, la cual es formada cuando las neuronas implementan conexiones laterales y así se conectan con otras de su misma capa. Usado para implementar un aprendizaje auto asociativo y recurrente, una de las redes más usadas y famosas es la Red Hopfield la cual fue propuesta por John Hopfield en el año 1980 y existen otras, tales como "Brain State In The Box" o también llamado Asociador lineal, introducido por James Anderson.

a1.2) Redes multicapas; están computas por 2 a más capas y se generan cuando una capa recibe un conjunto de señales de entrada de una capa anterior y a su vez emite señales de salida para una capa posterior, este tipo de redes usados especialmente para el reconocimiento o clasificación de patrones. Se distinguen por el tipo de conexiones que se tienen, por ejemplo, las conexiones hacia atrás o también llamada Feedback (la información puede regresar a la capa anterior) y las conexiones hacia delante o también llamado Feedforward (la información circula hacia delante sin regresar).

Ejemplo de redes con conexiones hacia atrás son:

- Art, Memoria Bidireccional Asociativa (BAM)
- Cognitron

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Cabe resaltar que este tipo de conexiones son mayormente usadas en aprendizaje bicapa.

Ejemplo de redes con conexiones hacia delante son:

- **Perceptrón;** compuesta por una capa de entrada (las cuales reciben los patrones de entrada de la red), una capa de salida (constituida por neuronas que emiten un valor de salida la cual es la salida de toda la red) y una o más capas ocultas (Formada por neuronas que reciben señales de patrones de una capa anterior y realizan el procesamiento de las mismas para enviar información de salida hacia capas posteriores); fue introducido en el año 1957 por Frank Rosenblatt.
- **Neurona Lineal Adaptiva (Adaline);** es un modelo creado en 1960 por Bernard Widrow y es más poderoso en cuanto aprendizaje que el Perceptrón, debido a que minimiza el ruido en la fase de entrenamiento y procesamiento de datos.

Otro ejemplo de red con conexión hacia delante y atrás está el Backpropagation entre otros ejemplos como modelos de Cuantizacion del vector de aprendizaje (LQV) de Kohonen.

a.2) Según su algoritmo de aprendizaje, las redes neuronales se clasifican según el tipo de aprendizaje que se les da a las neuronas, optimizando así el aprendizaje de patrones que pasan por la red neuronal, existen dos tipos:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

a.2.1) Aprendizaje no supervisado; es aquella red cuya salida representa el grado de parentesco o igualdad entre la información de entrada y la información que se presenta en la salida. Está compuesta por reglas que brindan a la red habilidades de aprender patrones de una manera asociativa, una vez que se ha aprendido de manera asociada, hace que la red posea la habilidad de reconocimiento de esos patrones y la habilidad recordar. Habitualmente son redes monocapas.

a.2.2) Aprendizaje supervisado; es aquel que se caracteriza porque su modo de aprendizaje es realizado por un agente externo por medio de un entrenamiento controlado. El agente externo es aquel que controla los datos de salida de la red y si por alguna razón estos datos no coinciden con los datos deseados, se volverá a procesar los pesos de conexión con la finalidad de obtener una salida que sea similar a la deseada, un ejemplo de redes con aprendizaje supervisado es el Perceptrón Multicapa.

2.3.8.Tecnologías:

a) Lenguajes de programación:

Para la implementación de métodos y algoritmos en sistemas de reconocimiento facial se debe seleccionar lenguajes de programación óptimos para esta rama tecnológica; entre los más usados están:

- Python
- Java
- C y C++

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Python es un lenguaje de programación orientado a objetos muy utilizado en el desarrollo de algoritmos de inteligencia artificial, por lo que existen diferentes librerías y herramientas que pueden ser muy útiles para conseguir los objetivos de una manera más sencilla. (R. Cazorla, 2015)

Por otro lado, **Java** es un lenguaje de programación que es usado muy a menudo porque varias de sus librerías aportan facilismo al procesamiento de imágenes y redes neuronales. La razón principal de usar Java es que puede ser fácilmente adaptado para Android. (Stoimenov, Tsenov, Mladenov, & Member, 2016).

Los lenguajes de programación **C y C++** están oficialmente patentados para el uso de reconocimiento facial en entornos como en Matlab, OpenCv, Qt, etc, debido a su fácil manejo y ahorro de recursos en procesamiento de código.

b) Entornos de programación:

b.1) OpenCV:

Según el sitio web Docs.opencv.org (2016); es un entorno de visión muy popular iniciado por Intel en 1999. Este entorno con muchas bibliotecas de multiplataforma establece su enfoque en el procesamiento de imágenes en tiempo real e incluye implementaciones de métodos de reconocimiento facial de manera automática.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Aparte de ellos viene con una interfaz de programación en C, C ++, Python y Android

Entre los algoritmos disponibles más usados en la actualidad son:

- Eigenfaces (createEigenFaceRecognizer())
- Fisherfaces (createFisherFaceRecognizer())
- LBPH(createLBPHFaceRecognizer())

b.2) Matlab:

Según la Web Mathworks.com (2013), la web oficial de Matlab plantea que es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio, el cual es lenguaje Microsoft (M), aportando de manera significativa al procesamiento de imágenes y sobre todo al reconocimiento facial, esto debido a que posee librerías de apoyo eficiente, una interfaz de fácil manejo y muy entendible para el usuario; normalmente es usado para hacer pruebas de laboratorio.

b.3) Android Studio:

Android Studio es un entorno de desarrollo implementado con java que proporciona las herramientas más rápidas para crear apps en dispositivos móviles con Sistema Operativo (SO) Android.

“La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento,

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

sistema de compilación flexible y sistema instantáneo de compilación e implementación que permiten concentrarte en la creación de aplicaciones únicas y de alta calidad.” (Developer.android.com, 2014), son características importantes acerca de este entorno de desarrollo móvil.

Según Oka Sudana et al., (2014) plantean que Android Studio es un entorno muy recomendado para implementar algoritmos de reconocimiento facial debido a que posee librerías como “FaceDetector”, el cual es usado para el recorte de una cara en una imagen digital.

b.4) QT Creator:

Según la web oficial de QT.io es un entorno de programación integrado para Qt; este ID integra un SDK multiplataforma, manejando de primera instancia el lenguaje C++, es de fácil uso, muy rápido y manejable, está además activo para crear aplicaciones móviles en Android.

2.4. Definición de términos básicos:

a) Seguridad:

Es aquella que se encarga de proteger que el sistema de reconocimiento facial no sea vulnerado por alguna fotografía, esto asegura la integridad y la efectividad en la tasa de reconocimiento porcentual.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

b) Iluminación:

Es el parámetro que variará con mucha frecuencia para determinar si el método de reconocimiento facial funciona en distintos ambientes, ya sean con poca luz, baja luz, luz nocturna, etc.

c) S.O Android:

Las versiones de Android más recientes son:

- Android 6.0 (MarshMallow, Api 23)
- Android 7.0 (Nougat, Api 24 y Api 25)

d) Ambientes Controlados:

Se le llama ambiente controlado a aquel entorno o espacio cerrado, cuyas características o parámetros de luz, temperatura, humedad, presión, etc. están manipuladas por el hombre para hacer experimentos. (Gofman & Mitra, 2016)

e) Ambientes no controlados:

Son aquellos ambientes donde el entorno es aquel que se muestra tal cual es la vida real, con parámetros de luz, temperatura, humedad, presión, son variados de acuerdo a la hora del día. (Gofman & Mitra, 2016)

f) Dispositivos Móviles:

Son aparatos de uso portátil y de tamaño disminuido, con capacidades de procesamiento, conexión a internet, con memoria limitada, el cual ha sido creado y diseñado específicamente para realizar función y facilitar las actividades de cada persona. (Gofman & Mitra, 2016)

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo y diseño de la investigación:

3.1.1. Tipo de investigación:

Investigación Cuantitativa, debido a que se usaron indicadores que brindaron información para medir los resultados de tipo numéricos con una unidad de medida, de esta forma se pudo emitir estadísticas de porcentaje.

3.1.2. Diseño de la investigación:

Diseño Experimental, debido a que fue un proceso sistemático con aproximación científica en la cual se hicieron experimentos donde se manipularon las variables dependientes, este diseño es del subtipo Cuasi experimental y con diseño de pruebas Post test.

3.2. Población y muestra

Según los artículos científicos analizados en el marco teórico del proyecto, se tomó como referencia el promedio de población que se usaron para los experimentos en aquellas investigación, por lo tanto, se planteó hacer 15 tomas fotográficas a 30 personas, realizando un conteo de 450 fotos en la población en total que fueron almacenadas en la base de datos para que fuera entrenada. Para proponer la cantidad de experimentos realizados, fue necesario establecer la cantidad promedio de 10 experimentos en cada etapa de iluminación del día, es decir 10 experimentos por la mañana, 10 experimentos por la tarde y 10 experimentos por la noche, debido a que los experimentos medirán resultados en distinto ambientes de iluminación, todo esto según el protocolo experimental del artículo científico denominado “Evaluación de

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

métodos de reconocimiento facial en ambientes no controlados" de Jie Ni, Rama Chellappa en el año 2010.

La muestra fue de tipo censal, debido a que se usó toda la población para los experimentos, es decir que se utilizaron como fuente primaria toda la población para los experimentos y la emisión de resultados cuantitativos.

3.3. Hipótesis:

El resultado del análisis concluyó que el método de reconocimiento facial llamado FisherFaces es el que mejor se desempeña en cuanto a los aspectos de seguridad, mientras que en aspectos de iluminación el método de reconocimiento facial llamado LBP es el que mejor se desempeña.

3.4. Variables:

a) Variables independientes:

La variable independiente del presente proyecto es:

- Métodos de reconocimiento facial

b) Variables dependientes:

En este proyecto se tiene dos variables dependientes:

- Aspectos de Seguridad
- Aspectos de Iluminación

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.5. Operacionalización:

Tipo	Variables	Dimensiones	Indicadores	Técnicas o instrumentos de recolección de datos	FÓRMULA
Independiente	Métodos de Reconocimiento facial	Efectividad	Porcentaje de Éxito	Ficha de Observación	PE=[illegible]00
		Eficiencia	Tiempo de Respuesta en milisegundos	Ficha de Observación	TR=Tiempo inicio – Tiempo final
Dependiente	Seguridad	Error	Porcentaje de Error	Ficha de Observación	PXP=[illegible] PXN=[illegible]
	Iluminación	Reconocimiento	Porcentaje de precisión de	Ficha de Observación	PP=[illegible]00

3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos:

3.6.1. Abordaje metodológico:

Este proyecto buscó encontrar una solución al problema planteado haciendo varios experimentos a toda la muestra, tomándolo como un solo grupo. Por tal razón esta investigación es de tipo Cuasi experimental, ya que como se fundamentó antes todos los experimentos de esta investigación se hicieron a toda la población (Se tomaron como solo un grupo experimental).

3.6.2. Técnicas de recolección de datos:

- **Ficha de Observación,** estudio y análisis de fichas de datos estadísticos generadas de la emisión de resultados (Tabla y gráficos estadísticos).

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.6.3. Instrumentos de recolección de datos:

- Documentos de investigación
- Reportes estadísticos de resultados

3.7. Procedimientos para la recolección de datos:

Análisis Documental, a través de documentos o reportes que describan resultados de experimentos de reconocimiento facial, se extrajeron los métodos más eficientes para ser puestos a prueba y ver si también son eficientes en ambientes de iluminación y seguridad, teniendo en cuenta sus resultados y comparándolos con los resultados de este proyecto.

3.8. Plan de análisis estadístico de datos:

Una vez que se obtuvieron los datos, se procedió a hacer cuadros y diagramas estadísticos en Microsoft Excel debido a que es una potente herramienta en análisis de datos, así de esa manera se obtuvieron resultados porcentuales y se sacaron conclusiones estadísticas.

3.9. Principios éticos:

a) Confidencialidad:

Los códigos de ética hacen énfasis ante la seguridad y la protección de los participantes como la identidad y como también los informantes de la investigación. También expresa como un anonimato en la identidad de los participantes detallando en un seudónimo entre los participantes.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

b) Derechos de Autor:

Cada definición y conceptos usados para el llevar a cabo el tema de tesis serán citados y referenciados, así mismo con sus respectivos autores como participantes en dar aportes al trabajo.

3.10. Criterios de rigor científico:

a) Fiabilidad:

El proyecto cumplirá con las expectativas expresadas en su contenido y su implementación con el estándar y políticas para su desarrollo.

b) Validez:

Los datos obtenidos por el estudio o proyecto serán evaluados y analizados por los ingenieros especializados en el tema para decretar su veracidad.

c) Consistencia:

La investigación representará material consistente y certificado por la comunidad científica.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Resultados

4.1.1 Pruebas Realizadas

En este capítulo se muestran las pruebas realizadas con los métodos propuestos los cuales está basados en procesamiento digital para la detección de rostro en tiempo real y en ambientes no controlados, teniendo en cuenta la iluminación y seguridad.

Aplicación Desarrollada

En este proyecto de investigación se desarrolló una aplicación móvil con el lenguaje de programación Java, utilizando como entorno de desarrollo a Android Studio 2.3.3, además se usaron las librerías de OpenCv 2.4.9 y las librerías de JavaCv y JavaCpp, para de esta forma se pueda implementar los métodos propuestos en el capítulo 5.

Puesto que la finalidad del presente trabajo de investigación es analizar métodos de reconocimiento fácil bajo plataforma Android, se establecieron 4 indicadores en la operacionalización del capítulo 3, en la cual los resultados están basados métricamente dentro de las siguientes características:

Características de Hardware que se usó para las pruebas del método de reconocimiento facial:

Las características del celular donde se realizaron las pruebas se detallan en breve:

Sistema Operativo: S.O Android

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Memoria RAM: 1GB

Versión de Android: 6.1

Almacenamiento: 16GB

Indicadores:

El método propuesto fue puesto a prueba para poder medir y comparar los siguientes indicadores:

Tabla 1: Lista de indicadores de investigación

N°	INDICADOR
1	Porcentaje de pruebas con éxito
2	Tiempo de respuesta en milisegundos
3	Porcentaje de pruebas con Errores (Falsos positivos y falsos negativos)
4	Porcentaje de precisión de las pruebas con éxito

Fuente: Elaboración propia

4.1.2. Resultados de las pruebas:

a. Porcentaje de pruebas con éxito (PE):

Este indicador mide el porcentaje de éxito que se obtuvo de acuerdo a las pruebas que se hicieron en el experimento.

Su fórmula consta en dividir la cantidad de personas que se reconocieron con éxito entre la cantidad total de personas que se quisieron reconocer, todo esto multiplicado por 100 para obtener el resultado en porcentaje.

$$PE=\frac{\square}{\square}\square 100$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Experimento:

En 450 fotos entrenadas con ambos métodos propuestos se trató de identificar a 30 personas, para lo cual se tomó una foto a cada persona para determinar si era reconocido por el sistema de reconocimiento facial; el resultado obtenido fue que con el método FisherFaces un número de 23 de esas fotos tomadas sí fueron reconocidas con éxito, mientras que con el método LBP un numero de 27 de esas fotos tomadas si fueron reconocidas con éxito. De esta manera se obtiene los siguientes datos:

Método FisherFaces:

- N° Muestra= 30
- N° Aciertos= 23

Método LBP:

- N° Muestra= 30
- N° Aciertos= 27

A partir de los resultados se reemplazó en la fórmula para obtener el porcentaje de pruebas con éxito, obteniendo como resultado el siguiente:

Método FisherFaces:

PE= 76.7%

Método LBP:

PE= 90.0%

Cabe resaltar que la tasa de porcentaje de error es la diferencia para que el experimento alcance el 100% de efectividad.

Los resultados se muestran en la tabla 2.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

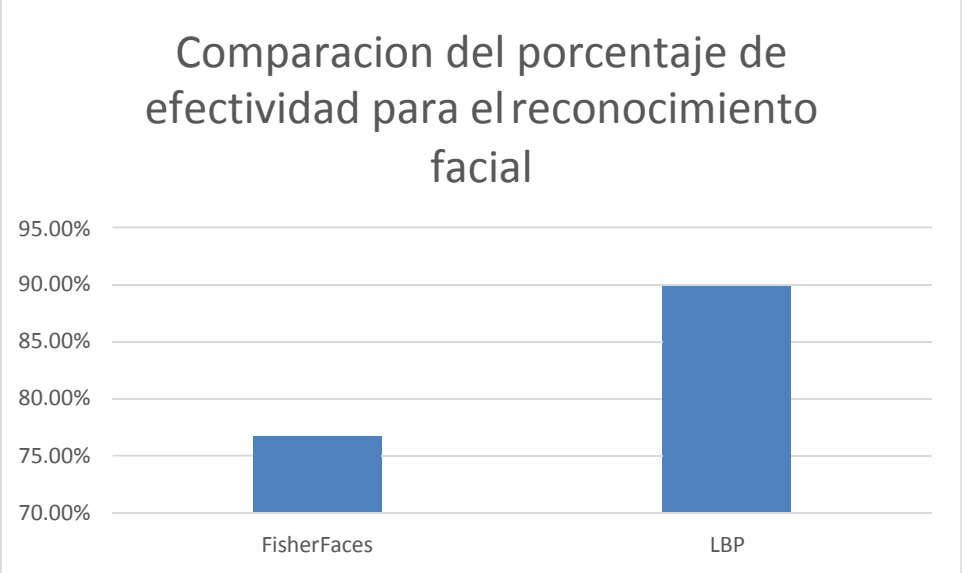
Tabla 2: Resultado del porcentaje de efectividad y éxito para los métodos seleccionados

Pruebas	Método	N° Aciertos	N° No Aciertos	% Efectividad
Todas	FisherFaces	23	7	76.70%
Todas	LBP	27	3	90.00%

Fuente: Elaboración Propia

Ahora, en el gráfico 1 se muestran la comparación entre los porcentajes de acierto de los resultados, para que de esta forma se visualice mejor la efectividad de los métodos seleccionados.

Gráfico 1: Comparación del porcentaje de efectividad para el reconocimiento facial



Fuente: Elaboración propia

Como se puede apreciar en el gráfico 1, el método LBP para el reconocimiento facial usado en esta investigación tiene una tasa de porcentaje de éxito superior en comparación a la tasa de porcentaje de éxito del método LBP.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Entonces de esta manera se demostró que el método LBP es más efectivo para el reconocimiento facial en ambientes no controlados y en una plataforma Android que el método FisherFaces.

b. Tiempo de respuesta en milisegundos (TR):

Este indicador mide en milisegundos el tiempo de respuesta que tarda el sistema en reconocer un rostro, de esta manera se puede medir el tiempo de respuesta de analisis del sistema operativo Android 4.4 para el reconocimiento facial, considerando que las pruebas del experimento se realizaron en la mañana, tarde y noche.

Su fórmula consiste en hallar la diferencia entre el tiempo en que se terminó la prueba de reconocimiento y el tiempo en que se inició la prueba de reconocimiento, todo esto medido en milisegundos y tomando la hora actual del sistema operativo Android.

TR= (Tiempo Final – Tiempo Inicial) milisegundos

Experimento:

En 450 fotos entrenadas por los métodos de reconocimiento facial seleccionados se quisieron reconocer a 30 personas, dividiendo en partes iguales para la mañana, tarde y noche. De esta manera se hizo el experimento queriendo reconocer a 10 personas en la mañana, 10 personas en la tarde y 10 personas en la noche, todo esto para calcular en milisegundos el tiempo que el sistema demoraba para obtener una respuesta por cada prueba.

De esta forma se halló la sumatoria de todos los tiempos de respuesta de cada prueba y se dividió entre el número total de pruebas, calculando un tiempo promedio (TP) para cada situación planteada, de la siguiente manera:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$\bar{P} = \frac{\sum \text{[illegible]}}{\text{[illegible]}}$$

Los resultados que se obtuvieron de sacar el promedio del tiempo de respuesta con el método de FisherFaces son los siguientes:
Promedio de Tiempo de respuesta en la mañana:
956 milisegundos
Promedio de Tiempo de respuesta en la tarde:
878 milisegundos
Promedio de Tiempo de respuesta en la noche:
1045 milisegundos

Los resultados que se obtuvieron de sacar el promedio del tiempo de respuesta con el método de LBP son los siguientes:

Promedio de Tiempo de respuesta en la mañana:
734 milisegundos
Promedio de Tiempo de respuesta en la tarde:
755 milisegundos
Promedio de Tiempo de respuesta en la noche:
879 milisegundos

Los resultados se muestran en la tabla 3.

Tabla 3: Tiempo Promedio de Respuesta con los métodos seleccionados para el reconocimiento facial

	Pruebas	Métodos	Promedio de Tiempo de Respuesta
Mañana	10	FisherFaces	956
	10	LBP	734
Tarde	10	FisherFaces	878
	10	LBP	755
Noche	10	FisherFaces	1045
	10	LBP	879

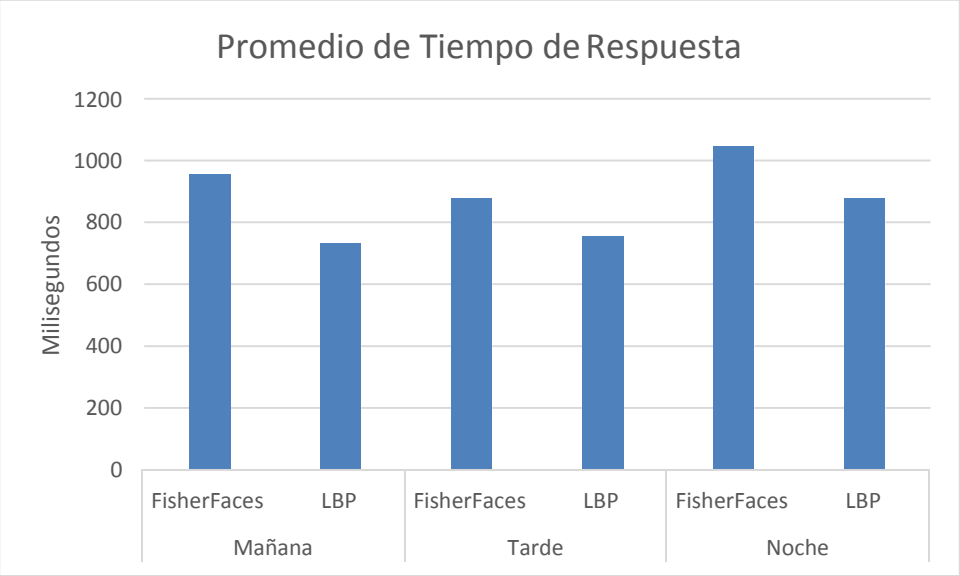
Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Ahora en el gráfico 2 se muestran la comparación entre el tiempo promedio de respuesta de la mañana, tarde y noche de los dos métodos seleccionados, para que de esta forma se visualice mejor la diferencia entre el tiempo promedio de respuesta que se obtuvo con la implementación de ambos métodos.

Como se puede apreciar en el gráfico 2, el método seleccionado para el reconocimiento facial de esta investigación llamado FisherFaces tiene un promedio de tiempo de respuesta alta en la noche, es decir que el proceso de reconocimiento facial es más lento por la noche.

Gráfico 2: Promedio de tiempo de respuesta con el método seleccionado



Fuente: Elaboración propia

Por otro lado, se puede observar que el promedio de tiempo de respuesta es menor por las mañanas, tardes y noches con el método LBP.

De esta manera se demostró que el método para el reconocimiento facial LBP procesa más rápido el resultado en las mañanas, tardes y noches en ambientes no controlados y en una plataforma Android.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

c. Porcentaje de pruebas con errores (PX):

Este indicador se encarga de medir la seguridad que existe entre los 2 método seleccionados, a la vez posee dos sub indicadores los cuales miden y comparan la tasa de porcentaje de error que se obtuvo de acuerdo a las pruebas que se hicieron en el experimento. El sub indicador 1 se encarga de medir el porcentaje de falsos positivos, y el sub indicador 2 se encarga de medir el porcentaje de falsos negativos.

Porcentaje de Error para falsos positivos (PXP):

Se define como falsos positivos o pruebas erróneas positivas aquellas pruebas que identifican a una persona de manera errónea, es decir que el sistema detecta a otra persona que no es la persona a la cual se le hace la prueba del experimento.

Por lo tanto, este indicador se encarga de medir el porcentaje de los resultados de errores positivos.

Su fórmula consta en dividir el número total de pruebas erróneas positivas y el número total de la muestra, todo esto multiplicado por 100 para obtener el porcentaje.

$$P\text{\textdiamond}P = \frac{\text{\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond}}{\text{\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond\textdiamond}} \text{\textdiamond} 100$$

Porcentaje de Error para falsos negativos (PXN):

Se define como falsos negativos o pruebas erróneas negativas aquellas pruebas que no identifican a ninguna persona, es decir que el sistema no detecta a ninguna persona con las cuales se entrenó el método seleccionado.

Por lo tanto, este indicador se encarga de medir el porcentaje de los resultados de errores negativos.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Su fórmula consta en dividir el número total de pruebas erróneas negativas y el número total de la muestra, todo esto

$$P\square\square = \frac{\square\square\square\square\square\square\square}{\square\square\square\square\square\square\square} \square 100$$

Experimento:

En 450 fotos entrenadas con los dos métodos propuestos, se trató de identificar a 30 personas, poniendo a 15 personas atrás del celular para tomar una foto y poniendo 15 fotografías en papel atrás del celular para tomar una foto, de esta forma se trató de medir el porcentaje de error tanto para el método FisherFaces y el método LBP, todas estas fotos se pusieron a pruebas para ver si eran reconocidas por el sistema de reconocimiento facial.

El resultado obtenido del reconocimiento facial con FisherFaces fue que de las 30 fotos que se pusieron a prueba 12 de ellas fueron erróneas, mientras que el resultado que se obtuvo con el reconocimiento facial con LBP fueron 18 resultados erróneos.

Método FisherFaces:

Entre las pruebas que se obtuvieron con resultado erróneo 10 de ellas se hicieron a personas y 2 de ellas se hicieron a fotografías de papel. De esta manera se obtiene los siguientes datos:

- N° Muestra= 30
- N° Errores= 12

A partir de los resultados se reemplazó en la fórmula para obtener el porcentaje de pruebas con error con el método FisherFaces, obteniendo como resultado el siguiente:

PX= 40%

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Método LBP:

Entre las pruebas que se obtuvieron con resultado erróneo 4 de ellas se hicieron a personas y 14 de ellas se hicieron a fotografías de papel. De esta manera se obtiene los siguientes datos:

- N° Muestra= 30
- N° Errores= 18

A partir de los resultados se reemplazó en la fórmula para obtener el porcentaje de pruebas con error con el método LBP, obteniendo como resultado el siguiente:

PX= 60%

Cabe resaltar que la tasa de porcentaje de efectividad es la diferencia para que el experimento alcance el 100% de efectividad. El resultado del porcentaje de error se muestra en la tabla 4.

Tabla 4: Valores del porcentaje de Error para el reconocimiento facial con el método seleccionado

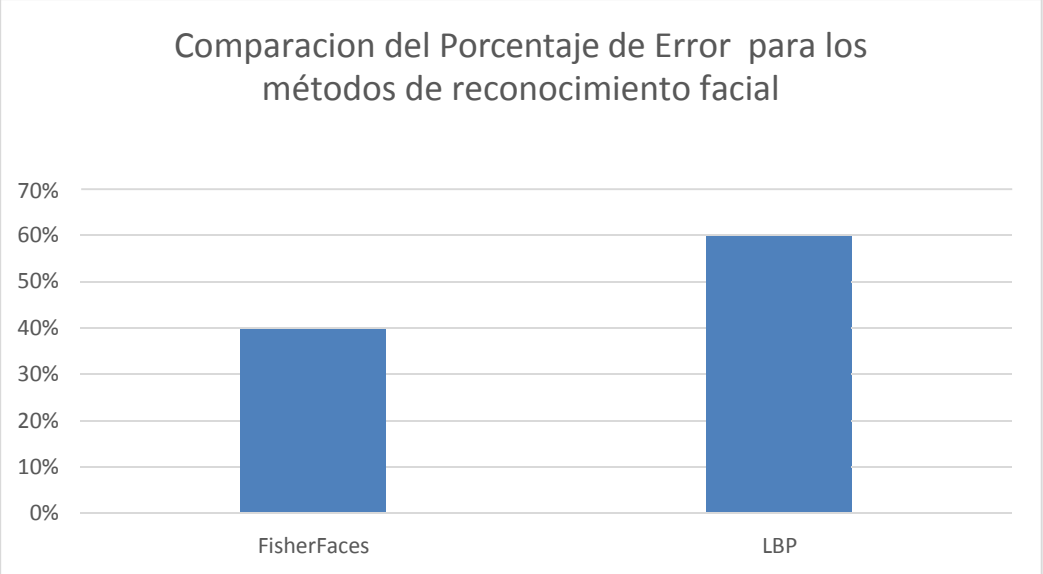
Pruebas	Métodos	N° No Aciertos	N° Aciertos	% Error	% Efectividad
Todas	FisherFaces	12	18	40%	60%
Todas	LBP	18	12	60%	40%

Fuente: Elaboración propia

Se muestra ahora en el gráfico 3, elaborado en Excel, la comparación entre el porcentaje de error entre el método FisherFaces y el método LBP, para que de esta forma se visualice mejor la tasa de error de ambos métodos analizados.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Gráfico 3: Comparación del porcentaje de error para el método de reconocimiento facial seleccionado



Fuente: Elaboración propia

Porcentaje de Error para falsos positivos (PXP):

- **Método FisherFaces:**

De las 12 pruebas con resultado erróneo se detectaron de manera positiva a 5 personas, de esta manera se obtiene los siguientes datos:

- N° Muestra = 12
- N° Errores positivos = 5

A partir de este resultado se obtuvo el porcentaje de pruebas con errores positivos en base a la fórmula planteada anteriormente, obteniendo como resultado el siguiente:

PXP = 41,7%

- **Método LBP:**

De las 18 pruebas con resultado erróneo se detectaron de manera positiva a 14 personas, de esta manera se obtiene los siguientes datos:

- N° Muestra = 18
- N° Errores positivos = 14

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

A partir de este resultado se obtuvo el porcentaje de pruebas con errores positivos en base a la fórmula planteada anteriormente, obteniendo como resultado el siguiente:

PXP = 77,78%

Porcentaje de Error para falsos negativos (PXN):

- **Método FisherFaces:**

De las 12 pruebas con resultado erróneo no se detectó a ninguna persona en 7 ocasiones, de esta manera se obtiene los siguientes datos:

- N° Muestra= 12
- N° Errores positivos= 7

A partir de este resultado se obtuvo el porcentaje de pruebas con errores negativos en base a la fórmula planteada anteriormente, obteniendo como resultado el siguiente:

PXP= 58,3%

- **Método LBP:**

De las 18 pruebas con resultado erróneo no se detectó a ninguna persona en 4 ocasiones, de esta manera se obtiene los siguientes datos:

- N° Muestra= 18
- N° Errores positivos= 4

A partir de este resultado se obtuvo el porcentaje de pruebas con errores negativos en base a la fórmula planteada anteriormente, obteniendo como resultado el siguiente:

PXP= 22,23%

Cabe resaltar que el porcentaje de errores para falsos positivos y para falsos negativos se deben complementar y la sumatoria de

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ambos porcentajes deben llegar a 100%, el cual es el porcentaje total de errores.

Los resultados de ambos sub indicadores se muestran en la tabla 5.

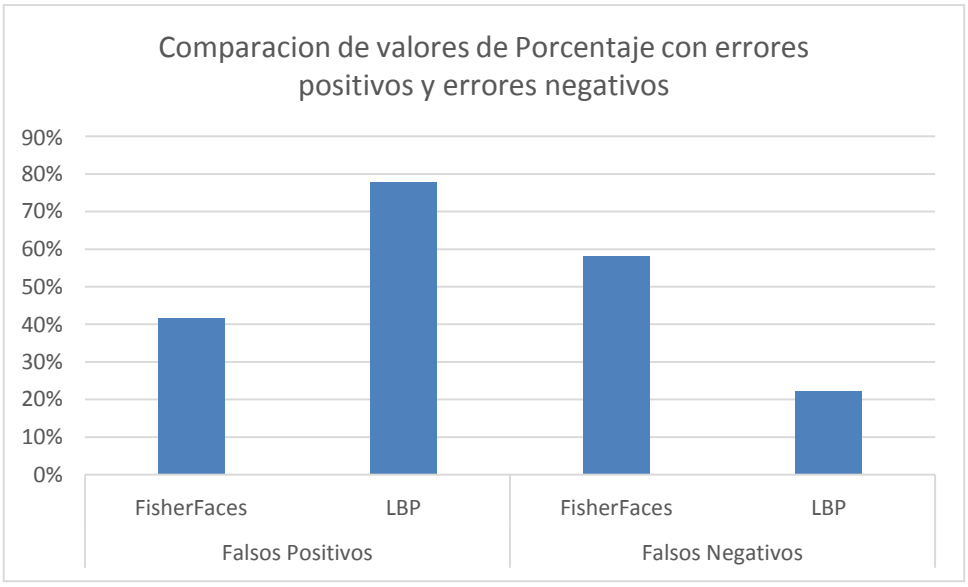
Tabla 5: Valores de porcentajes para las pruebas con errores positivos y errores negativos

	Métodos	N° Pruebas Erróneas	% Error
Falsos Positivos	FisherFaces	5	41.70%
	LBP	14	77.78%
Falsos Negativos	FisherFaces	7	58.30%
	LBP	4	22.23%

Fuente: Elaboración propia

Para una mejor visualización se muestra en el gráfico 4, elaborado en Excel, la comparación entre el porcentaje de errores con falsos positivos y el porcentaje de errores con falsos negativos.

Gráfico 4: Comparación de valores de Porcentaje con errores positivos y errores negativos



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Como se puede apreciar en el gráfico 4, el método de reconocimiento facial llamado FisherFaces tiene una tasa de porcentaje de errores con falsos positivos muy inferior con 41.7%, mientras que con el método LBP tiene un porcentaje de errores con falsos positivos de 77.78%.

Por otro lado la tasa de porcentaje de errores con falsos negativos del método FisherFaces es de 58.3%, esto es muy superior en comparación al 22.23% del porcentaje que arrojo el experimento del reconocimiento facial con LBP.

Sin embargo, como se apreció en el grafico 3, el porcentaje de error general que se calculó de todas las pruebas realizadas en el experimento con ambos métodos se observa que la tasa de porcentaje de error con el método FisherFaces es de 40%, mientras que con el método LBP es de 60%. De esta manera se demostró que el método FisherFaces tiene una tasa baja de error para el reconocimiento facial en ambientes no controlados y en una plataforma Android.

d. Porcentaje de precisión de las pruebas con éxito(PP):

Este indicador es el encargado de medir el porcentaje de precisión que se obtuvo de acuerdo a las pruebas del experimento que se hicieron con diferentes iluminaciones, como por ejemplo en la mañana, tarde y noche.

Su fórmula consiste en dividir el número de personas que se reconocieron exitosamente entre el número total de personas que se quisieron reconocer, todo esto multiplicado por 100 para obtener el resultado en porcentaje.

$$PP = \frac{\square}{\square} \square 100$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Experimento:

En 450 fotos entrenadas con los 2 métodos propuestos se trató de identificar a 30 personas, para ello se tuvo que tomar una foto a cada persona, se dividió en partes iguales para sacar los resultados en base a las diferentes iluminaciones, de esta manera se hicieron para cada método de reconocimiento facial 10 pruebas con la iluminación de la mañana, 10 pruebas con la iluminación de la tarde y 10 pruebas con iluminación nocturna, es así como se realizara el cálculo del porcentaje de precisión con las pruebas que se reconocieron de manera exitosa reconocido por el sistema de reconocimiento facial.

- **Método FisherFaces:**

El resultado obtenido con los diferentes tipos de iluminaciones fue el siguiente:

Luz de día: Se quisieron detectar a 10 personas de las cuales 8 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 8

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

$$PP=80\%$$

Luz de tarde: Se quisieron detectar a 10 personas de las cuales 7 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 7

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

PP=70%

Luz nocturna: Se quisieron detectar a 10 personas de las cuales 5 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 5

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

PP=50%

- **Método LBP:**

El resultado obtenido con los diferentes tipos de iluminaciones fue el siguiente:

Luz de día: Se quisieron detectar a 10 personas de las cuales 10 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 10

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

PP=100%

Luz de tarde: Se quisieron detectar a 10 personas de las cuales 9 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 9

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

PP=90%

Luz nocturna: Se quisieron detectar a 10 personas de las cuales 9 de ellas fueron reconocidas de manera precisa. Los datos son los siguientes:

- N° Muestra= 10
- N° Aciertos= 9

Con estos datos se desarrolla la fórmula para obtener el porcentaje de precisión, el cual es el siguiente:

PP=90%

Los resultados se muestran en la tabla 6.

Tabla 6: Valores de Porcentaje de Precisión con diferentes iluminaciones

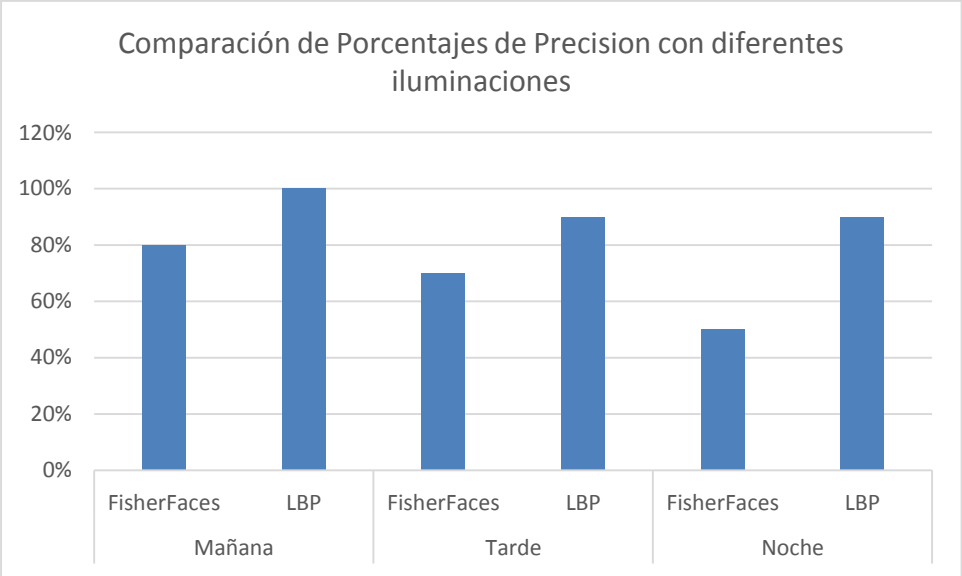
	Métodos	**% Precisión**
Mañana	FisherFaces	80%
	LBP	100%
Tarde	FisherFaces	70%
	LBP	90%
Noche	FisherFaces	50%
	LBP	90%

Fuente: Elaboración propia

Ahora se muestra en el gráfico 5, elaborado en Excel, la comparación entre el porcentaje de precisión de la iluminación de día, iluminación de tarde e iluminación nocturna para comparar los resultados obtenidos de ambos métodos.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Gráfico 5: Comparación de Porcentajes de precisión con diferentes iluminaciones



Fuente: Elaboración propia

Como se puede apreciar en el gráfico 5, el método LBP tiene un porcentaje de precisión alto en las mañanas, es decir que el proceso de reconocimiento facial es más efectivo y preciso por las mañanas en comparación a los resultados del método FisherFaces.

Por otro lado, se puede apreciar en el gráfico 5 que el porcentaje de precisión es mucho menor con iluminación nocturna con el método FisherFaces, es decir que el proceso de reconocimiento facial con este método no es tan efectivo ni preciso por las noche.

De esta manera se demostró que el método de reconocimiento facial LBP es más preciso con iluminación de día, iluminación de tarde e iluminación nocturna en ambientes no controlados y en una plataforma Android.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

La propuesta que se plantea en esta investigación, comienza con el diseño de un protocolo de captura de adquisición de imágenes con el propósito de determinar una forma estándar de cómo capturar las imágenes y las características del pre procesamiento de la imagen; posteriormente se plantearon las características mínimas que debe tener el dispositivo móvil para poder capturar dichas imágenes, para que de esta manera cada foto capturada sea fuente primaria del pre procesamiento, es decir, la detección del rostro, detección de ojos y el recorte del rostro y de esa forma almacenarlas en una base de datos.

Posteriormente se realizó un análisis de métodos y algoritmos de reconocimiento facial mediante una matriz de comparación para determinar qué métodos se seleccionarían para la evaluación de rendimiento, efectividad y eficiencia en cuando a seguridad e iluminación mediante una matriz de comparación.

En base a estos análisis se seleccionaron dos métodos de reconocimiento facial, los cuales son FisherFaces y Patrón Binario Local (LBP) para el análisis comparativo en base a resultados, teniendo en cuenta la seguridad e iluminación.

A partir de aquí se empieza el reconocimiento facial, seleccionando como entorno de programación a Android Studio con OpenCv en el lenguaje de programación Java. De esta manera se diseñaron las interfaces de entrenamiento de la base de datos, procesamiento y resultados de la aplicación. Seguidamente se codificaron los métodos seleccionados para el reconocimiento facial en la fase de entrenamiento y procesamientos del software.

Una vez realizado el entrenamiento de la base de datos con los métodos FisherFaces y LBP, se prosiguió a implementar un clasificador basado en el entrenamiento de la base de datos para así finalmente realizar las pruebas y experimentos de reconocimiento facial para analizar el rendimiento, eficiencia,

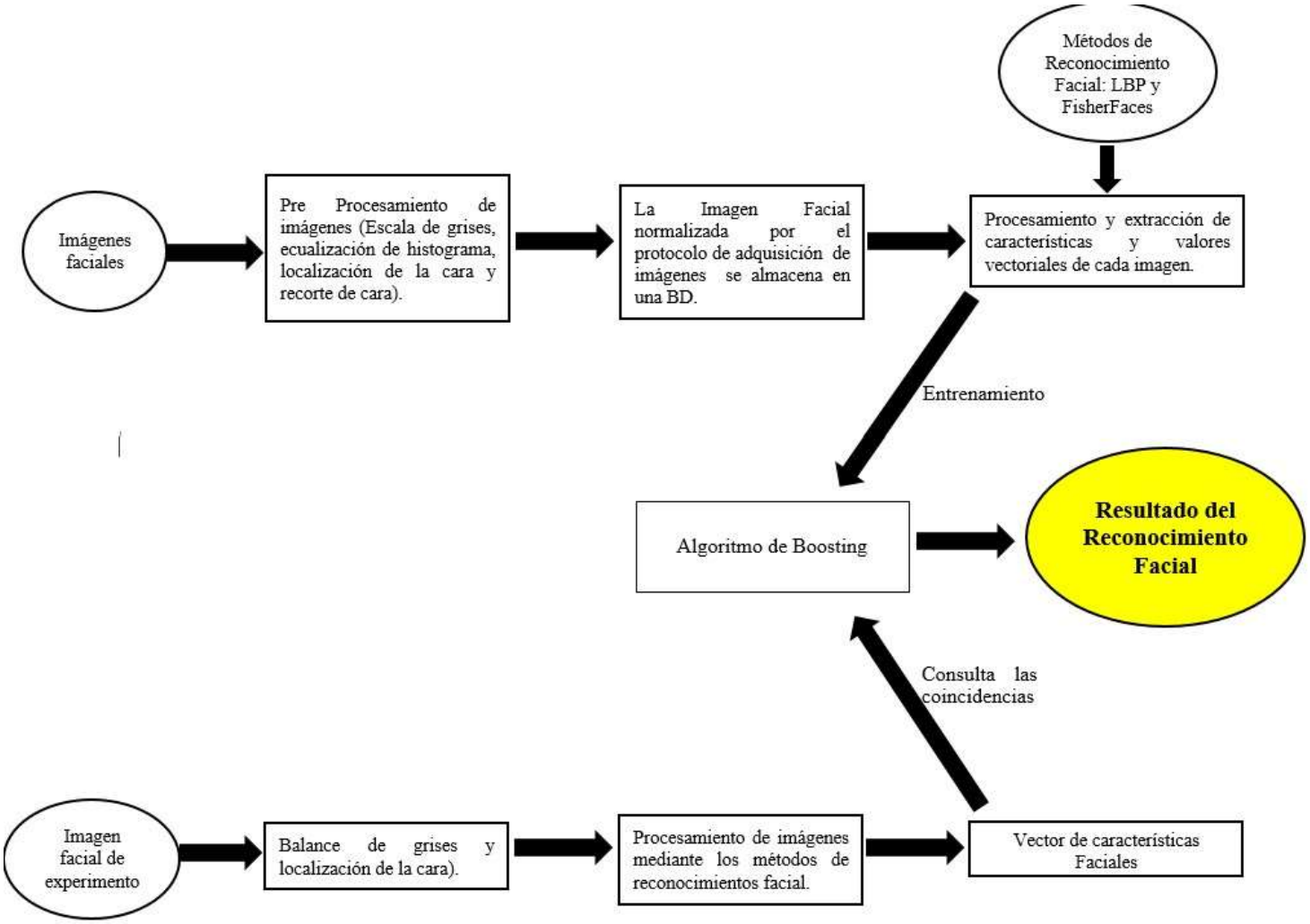
Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

efectividad en función a la seguridad e iluminación de los ambientes no controlados.

Para llevar a cabo esta propuesta de investigación fue necesario configurar el entorno de programación de Android Studio combinado con los módulos de OpenCv. Todo esto se detalla en el anexo 01.

A partir de aquí se empezó el desarrollo de la propuesta planteada, el diagrama de la figura 1 muestra el proceso de la propuesta planteada.

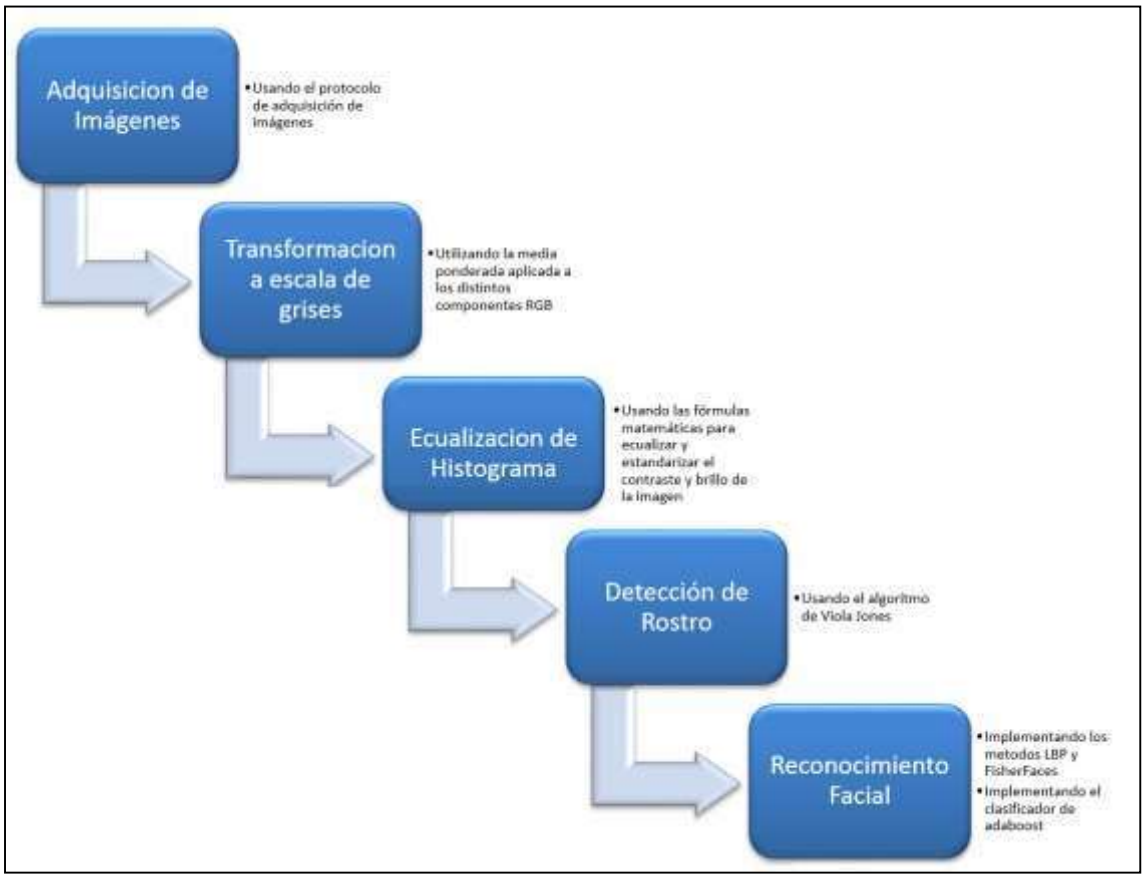
Figura 1: DIAGRAMA DEL PROCESO DE LA PROPUESTA PLANTEADA.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 2: DIAGRAMA DEL DETALLE DE LA PROPUESTA PLANTEADA.



Fuente: Elaboración propia

5.1 Protocolo de Adquisición de imágenes:

Esta investigación aborda el tratamiento de imágenes tomadas en ambientes no controlados. Un ambiente no controlado es un escenario de la realidad donde no se manipulan los siguientes parámetros: Iluminación y fondo del objeto de estudio principalmente. (Jie Ni & R. Chellappa, 2010)

Debido a que las imágenes de los rostros deben adquirirse en condiciones no controladas, fue necesario establecer algunos parámetros de guía para la adquisición, sin manipular la iluminación ni el fondo del objeto.

Por esa razón es que el protocolo de adquisición de imágenes presentado a continuación plantea diversos parámetros que no afectan ni manipulan la iluminación del ambiente.

Primero se establecieron requerimientos que debería tener la foto estos son:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

El dispositivo celular toma las fotos a cada persona con una resolución por defecto de 720 x 720 pixeles, este tamaño establecido por ser el tamaño promedio que no afecta el concepto de ambiente no controlado. Luego se estableció un parámetro de distancia para la captura de la foto el cual fue de 50 cm, el cual no afecta el concepto de ambiente no controlado. (Jie Ni & R. Chellappa, 2010).

Se tuvo en cuenta además que las fotos se tomaron con los patrones anteriores en diferentes ambientes del día con diferentes iluminaciones, por ejemplo: Luz en la mañana, luz de mediodía, luz de noche, luz interna de noche, etc. Debido a que dentro del concepto de ambientes no controlados existen diversas iluminaciones que surgen durante el día.

Las imágenes se almacenaron en el dispositivo con la siguiente nomenclatura:

CodigodelIdentificado-Nombre_Apellido_NumerodeFoto.Extension

De esta manera se tomaron 15 fotografías a cada persona que participó en el experimento, cada toma se hizo con diferentes gestos hacia la cámara: Cara Neutral, cara sonriendo, cara alegre, cara triste, cara enojada, ojos mirando hacia arriba, ojos mirando hacia abajo, ojos mirando a los costados, cejas levantadas, boca abierta, etc.

Segundo, la implementación del protocolo de adquisición de imágenes se realizó con un dispositivo móvil con las siguientes características:

Cámara de 5MP, debido a que es la resolución promedio que una cámara de celular de gama estándar tiene, así mismo un almacenamiento mínimo de 8GB para almacenar todas las fotos capturadas y una Memoria RAM de 1 GB. Todo esto bajo el sistema operativo Android 4.4. Se recalca que todos los resultados de los experimentos están siendo analizados y comparados baja las características Android anteriormente mencionadas. Estableciendo de esta manera que no se tomara en cuenta el análisis en plataforma Android distintas. (Ravibabu & Krishnan, 2015).

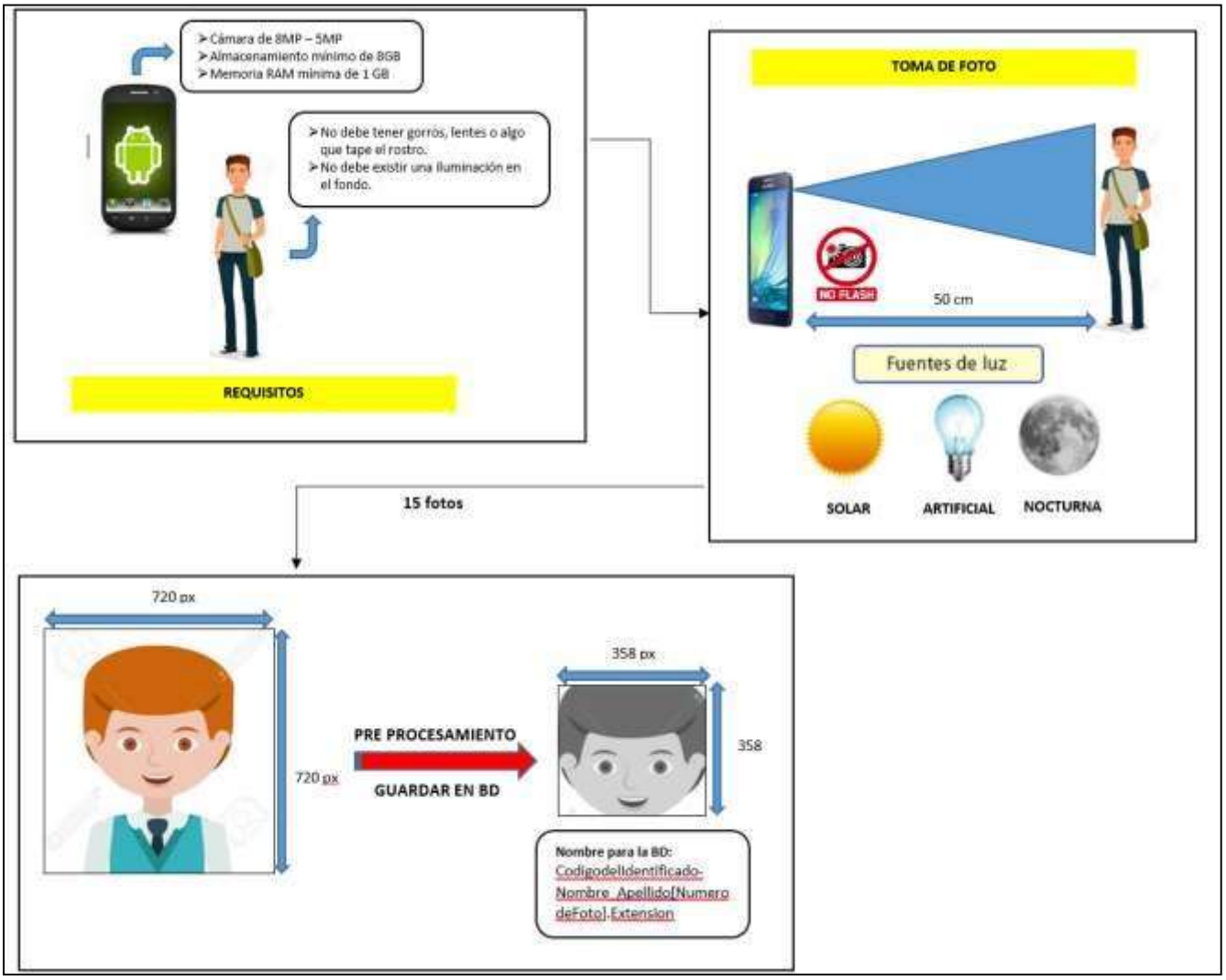
Por último se establecieron reglar de captura de foto, muy importantes para no discrepar con el concepto de ambientes no controlados, dichas reglas con las siguientes:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1. Las personas a la que se les tomaron las fotos respectivas no usaba gorros, lentes o algo que les tapara el rostro.
2. No existió una iluminación en el fondo.
3. No se tomaron con flash.
4. La posición del dispositivo a la hora de capturas las imágenes, fue vertical.

La figura 3 muestra en forma gráfica el protocolo de adquisición de imágenes, expuesto anteriormente.

Figura 3: DISEÑO DE PROTOCOLO DE ADQUISICIÓN DE IMAGENES.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Una vez terminado el diseño del protocolo de captura adquisición de imágenes, cada imagen capturada se pasó a una fase de pre procesamiento y, como se dijo anteriormente, esta fase se encarga de la detección de rostro y del recorte del mismo.

5.2. Pre procesar las imágenes capturas:

Esta fase es muy importante en la presente investigación debido a que aquí se prepararon las fotos para que luego sean procesadas adecuadamente.

Primero se tuvo que pasar la imagen a una imagen en escala de grises, de esta forma se disminuyen los recursos computacionales al momento de hacer el procesamiento, luego esa imagen en escala de grises a una imagen con ecualización de histogramas, seguidamente se hace la detección un rostro, a partir de aquí se aisló dicho rostro para ser recortado y de esa forma almacenarlo en una base de datos facial.

a) Pasar una imagen a escala de grises:

Este paso consiste en transformar la imagen con intensidades de Rojo, Verde y Azul (RGB) por pixel que serían de 3 canales, convirtiendo a escala de grises de 1 canal que varían de 0 a 255 intensidades de grises. Siendo entonces un paso importante para el pre procesamiento de la imagen con la finalidad de centrar la información en una dimensión y mostrar las diversas texturas de la imagen. (M.Sc Jimy Alexander Cortés Osorio, 2011).

Para hacer el proceso de pasar a una imagen en escala de grises se estudia la luminancia de la imagen para derivar los valores RGB de cada pixel de la imagen a valores en escala de gris.

Por esa razón el cálculo del equivalente de cada pixel de color a un pixel en escala de grises se debe realizar como una media ponderada aplicada a los distintos componentes RGB de cada pixel, esta media

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ponderada se realiza debido a que se aproxima a la dimensión del ojo humano. (M.Sc Jimy Alexander Cortés Osorio, 2011).

La ecuación matemática que hace posible esta conversión de pixeles de color a pixeles en escala de grises es la siguiente:

$$\text{Y} = [(\text{R} * 0.3) + (\text{G} * 0.59) + (\text{B} * 0.11)]$$

Donde:

Y: Es el nuevo valor de los componentes RGB ya convertido a escala de grises

R: Es el valor componente Rojo del pixel de la imagen de colores

G: Es el valor componente Verde del pixel de la imagen de colores

B: Es el valor componente Azul del pixel de la imagen de colores

Por tanto, para realizar la conversión a escala de grises se debe aplicar esta ecuación a cada pixel de la imagen de color. De esta forma aparecerá una nueva matriz de un byte por pixel con el valor de Y que daría la información de luminancia o grises.

Ejemplo de la ecuación matemática:

Color ingresado: 

Transformación de RGB a escala de gris

$$\text{Y} = 143 * 0.3 + 156 * 0.59 + 250 * 0.11$$

$$\text{Y} = 42.9 + 92.04 + 27.5$$

$$\boldsymbol{Y = 162}$$

Conversión en escala de gris: 

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

a.1) Algoritmo de conversión:

El pseudocódigo de la implementación de la conversión de una imagen a escala de grises se verifica en la figura 4.

Figura 4: ***Pseudocódigo de cómo pasar imagen a escala de grises.***

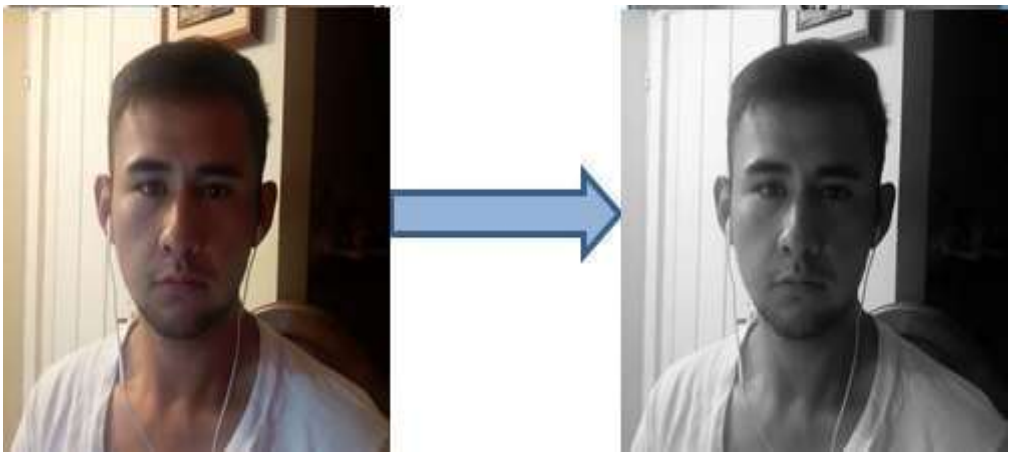
```
Inicio
        Leer MatrizPixel_imagen[R,G,B]
        Leer grayscaleImage[][]
        Para(entero i=0; i< MatrizPixel_imagen.tamañoHorizontal;i++)
                Para(entero j=0; j< MatrizPixel_imagen.tamañoVertical;j++)
                        R ← Leer MatrizPixel_imagen[i][j]
                        G ← Leer MatrizPixel_imagen[i][j]
                        B ← Leer MatrizPixel_imagen[i][j]
                        Y=(R*0.3)+(G*0.59)+(B*0.11)
                        grayscaleImagen[i][j]=Y
                Fin Para
        Fin Para
        Retornar grayscaleImage[][]
Fin
```

Fuente: Elaboración Propia

Para este método se ingresa la imagen a color y de salida se almacenará en la variable grayscaleImage que es una matriz llena del valor de cada pixel y también se envía el tipo de conversión que se desea hacer en este caso se realizó la conversión a escala de grises de 8 bits que es el medio de conversión más usado. (Oka Sudana, 2014)

Por ejemplo, en la Figura 5 se muestra el antes y el después de la conversión de una imagen normal a una escala de grises.

Figura 5: ***Antes y después de la conversión de una imagen normal a una escala de grises.***



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

b) Pasar una imagen de escala de grises a ecualización de histograma:

A continuación, se aplicó la ecualización de histograma a la imagen en grises para estandarizar el contraste y brillo de la imagen, esto para que distintas condiciones de iluminación no afecten la detección del rosto en la imagen, (paso importante ya que la investigación trata de resolver problemas de iluminación en ambientes no controlados) de este modo el algoritmo de detección de rostro sea más eficaz en la imagen.

El objetivo de transformar una imagen a ecualización de histograma es que la imagen tenga una distribución uniforme sobre toda la escala de grises, es decir, que exista el mismo número de pixeles para cada nivel de gris del histograma de una imagen monocroma.

El proceso de hace de la siguiente manera:

Lo primero es calcular el histograma de la imagen, que es la suma total de la misma intensidad del pixel.

$$\text{[illegible]} = h[\text{[illegible]}]$$

Donde:

h = histograma

k = 0 – 255

n = Número total de pixeles de la imagen

En la Figura 6 se visualiza un ejemplo de intensidades de pixeles para calcular la fórmula:

Figura 6: Ejemplo de intensidades.

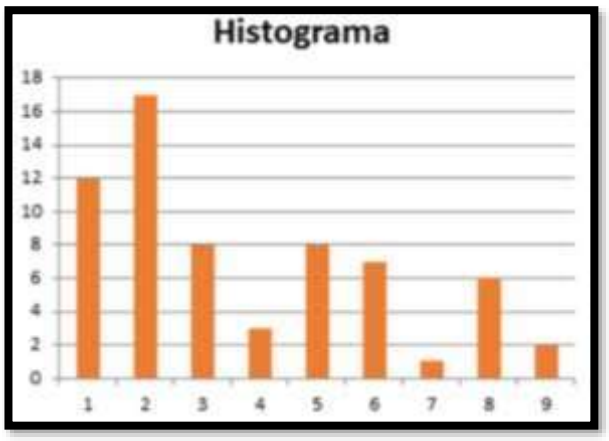
2	1	5	5	8	9	5	5
8	2	3	3	2	1	6	6
6	2	2	2	4	2	1	1
5	8	4	2	1	2	4	2
3	2	2	1	8	1	2	1
6	3	5	3	1	2	1	2
8	3	6	3	1	1	2	5
9	6	2	6	8	3	5	7

Fuente: Elaboración Propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Una vez que se tiene las intensidades de pixeles de una imagen en una matriz se realizó la sumatoria de todos los pixeles con la misma intensidad, para ello se obtiene cuantas veces se repite la misma intensidad, por ejemplo, a continuación, se sumó todos los pixeles de la misma intensidad y se obtuvo un histograma como se visualiza en la Figura 7.

Figura 7: Histograma de ejemplo.



Fuente: Elaboración Propia

El algoritmo para obtener el histograma de una imagen se muestra en la Figura 8.

Figura 8: Algoritmo de obtención del histograma de una imagen.

```
// Calcular Histograma
Vector hist[256];
Matriz input[][];
Entero indice;
Para (entero i=0; i <256; i++)
        hist[i] = 0;
Fin para
Para(entero r=0; r < input.rows ; r++)
        Para(entero c=0; c< input.cols ; c++)
                indice= input[r][c] ;
                hist [indice] = hist [indice]+1;
        Fin para
Fin para
Retornar hist[]
```

Fuente: Elaboración Propia

Una vez obtenido el histograma de la imagen se calculará el histograma acumulado con la siguiente formula:

$$\sum_{i=1}^{n} h(i-1) + h(i)$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Donde:

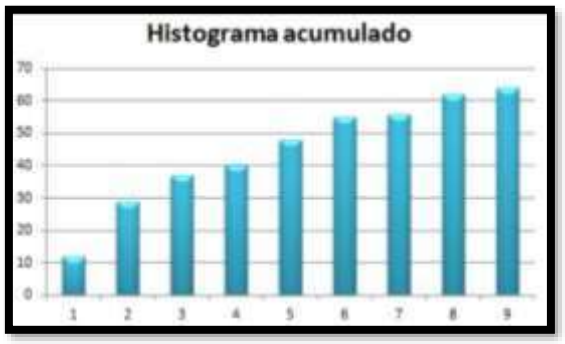
MN = Total de pixeles

H = Histograma

i = intensidad

El histograma acumulado se calcula con el total de la intensidad de los pixeles más la sumatoria del pixel anterior. Por ejemplo, en la Figura 9 se visualiza el histograma acumulado del histograma generado anteriormente.

Figura 9: Histograma Acumulado.



Fuente: Elaboración Propia

El algoritmo para obtener el histograma acumulado de una imagen se muestra en la Figura 10.

Figura 10: Algoritmo para obtener un Histograma Acumulado.

```
Obtener hist[];
Entero accum_hist [256];
accum_hist [0] = hist [0];
Para (entero i=1; i <256; i++)
        accum_hist [i] = hist [i]+ accum_hist [i -1];
fin para
Retornar accum_hist[]
```

Fuente: Elaboración Propia

Ahora se continúa con las operaciones para lograr la ecualización del histograma, el cual se desarrolló con la siguiente formula:

$$\text{???}(\text{?}) = \text{?????\_??????}(\text{?}(\text{?}). \quad \frac{\text{?} - 1}{\text{?} \cdot \text{?}})$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Una vez efectuada la operación matemática del histograma acumulado por k -1 entre el total de pixeles de la imagen se obtiene una nueva matriz de la imagen.

El algoritmo para obtener el histograma ecualizado se muestra en la Figura 11.

Figura 11: Algoritmo para obtener un Histograma Ecualizado.

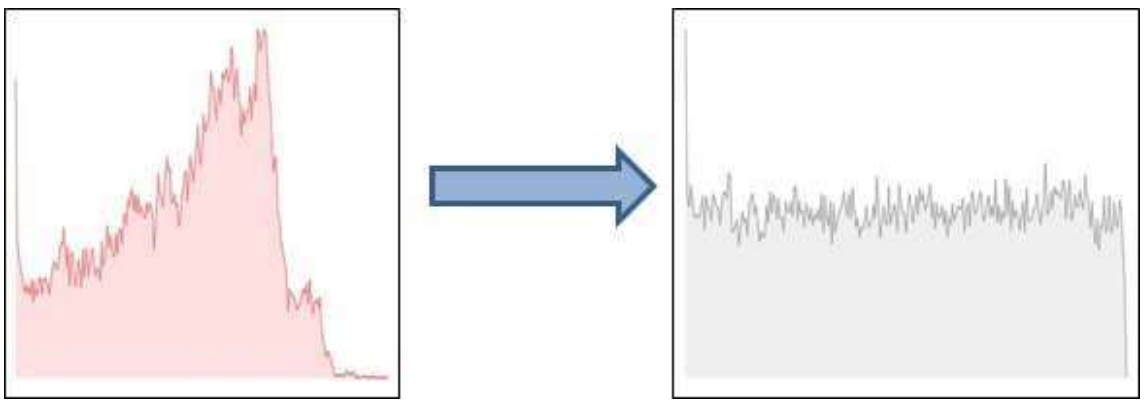
```
Leer heq[256]
Obtener accum_hist[i]
Leer cantidadcolumnas,cantidadfilas
Para (entero i = 0; i <256; i++)
        heq[i] = Parte Entera[(accum_hist [i] - 1) /( cantidadcolumnas * cantidadfilas)];
Fin Para
Retornar heq[]
```

Fuente: Elaboración Propia

Una vez terminado este proceso la ecualización de histogramas crea una nueva imagen donde los niveles de intensidades se normalizar en el rango de 0 a 255. El resultado de la ecualización del histograma mejorando las intensidades de brillo excesivo y maximiza el contraste de una imagen sin perder la estructura de la imagen.

Los resultados del histograma ecualizado del antes y después se aprecian en la figura 12.

Figura 12: Antes y después de la ecualización de histogramas.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Método usado en entorno Android:

En el entorno de programación de Android se usa el método de denominado equalicehist el cual contiene todo lo descrito anteriormente y viene implementado dentro de Opencv.

En la figura 13 se muestra en síntesis el código de programación del método equalicehist.

Figura 13: Código de programación del método equalizehist.

```
Public Mat equalizehist (const Mat& image, OutputArray _hist, const int histSize, const float* _range){
                    vx_int32 offset = (vx_int32)(_range[0]);
                    vx_uint32 range = (vx_uint32)(_range[1] - _range[0]);
                    if (float(offset) != _range[0] || float(range) != (_range[1] - _range[0]))
                       return false;
                    size_t total_size = image.total();
                    int rows = image.dims > 1 ? image.size[0] : 1, cols = rows ? (int)(total_size / rows) : 0;
                    if (image.dims > 2 && !(image.isContinuous() && cols > 0 && (size_t)rows*cols == total_size))
                       return false;
                    ivx::Context ctx = ovx::getOpenVXContext();
                    if (ctx.vendorID() == VX_ID_KHRONOS && (range % histSize))
                          return false;
                    ivx::Image
                    img = ivx::Image::createFromHandle(ctx, VX_DF_IMAGE_U8,
                    ivx::Image::createAddressing(cols, rows, 1, (vx_int32)(image.step[0])), image.data);
                    ivx::Distribution vxHist = ivx::Distribution::create(ctx, histSize, offset, range);
                    ivx::IVX_CHECK_STATUS(vxuHistogram(ctx, img, vxHist));
                    _hist.create(1, &histSize, CV_32F);
                    Mat hist = _hist.getMat(), ihist = hist;
                    ihist.flags = (ihist.flags & ~CV_MAT_TYPE_MASK) | CV_32S;
                    vxHist.copyTo(ihist);
                    ihist.convertTo(hist, CV_32F);
                    return hist;
}
```

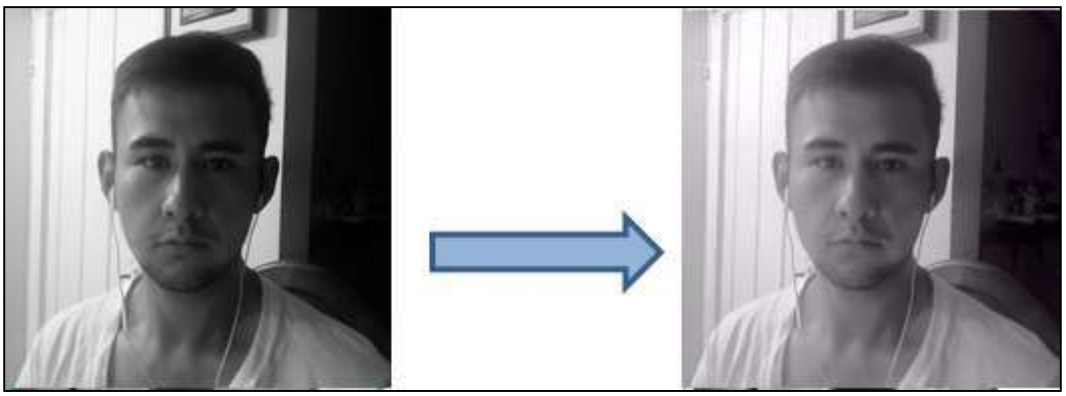
Fuente: Elaboración Propia

Los resultados de las imágenes del antes y después de la ecualización se aprecian en la figura 14.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 14: Antes y después de la ecualización de una imagen.



Fuente: Elaboración Propia

c) Detección de rostro:

Con el resultado de la imagen en ecualización de histograma se pasó a detectar el rostro con la finalidad de obtener solo el fragmento que más no interesa de la imagen el cual es la cara y asi dejar de lado cualquier otro enfoque que no sea el rostro.

Para ello contamos con las bibliotecas de OpenCv que poseen algoritmos para la detección de objetos entre ellos la detección de rostro. Los cuales están en formato XML llamados “haarcascade.xml”, dicho archivo contiene las características principales de tipo haar el cual hace efectivo la detección de objetos.

Las características de tipo haar que sirven para detectar rostros en una imagen son las siguientes:

Zona que detecta	Característica Haar
Ojo izquierdo	
Ojo Derecho	
Frente	
Mejilla	

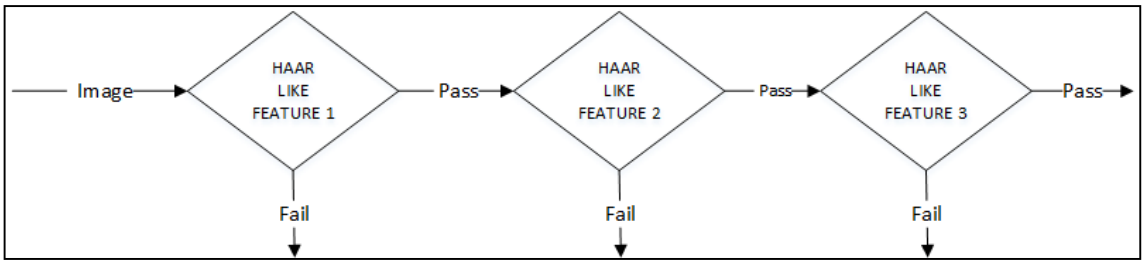
Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Nariz	
Boca	

En la figura 15 se muestra de manera sintetizada como se utiliza este archivo xml, en la cual una imagen pasa por todos los filtros de características de tipo haar y cada filtro va determinando si contiene o no un rostro.

Figura 15: Proceso de funcionamiento del archivo haarcascade



Fuente: Elaboración Propia

Papageorgiou et al. (2002) en su artículo “A general framework for object detection” plasmó que la detección de objetos en el área del procesamiento digital utilizaba clasificadores de cascada de tipo Haar, el cual es un método muy efectivo para la detección de objetos dentro de una imagen digital, este algoritmo fue propuesto en 2001 por Paul Viola y Michael Jones en su artículo “Detección Rápida de Objetos utilizando una Cascada de Características Simples”, por esta razón es también llamado “Algoritmo Viola Jones”.

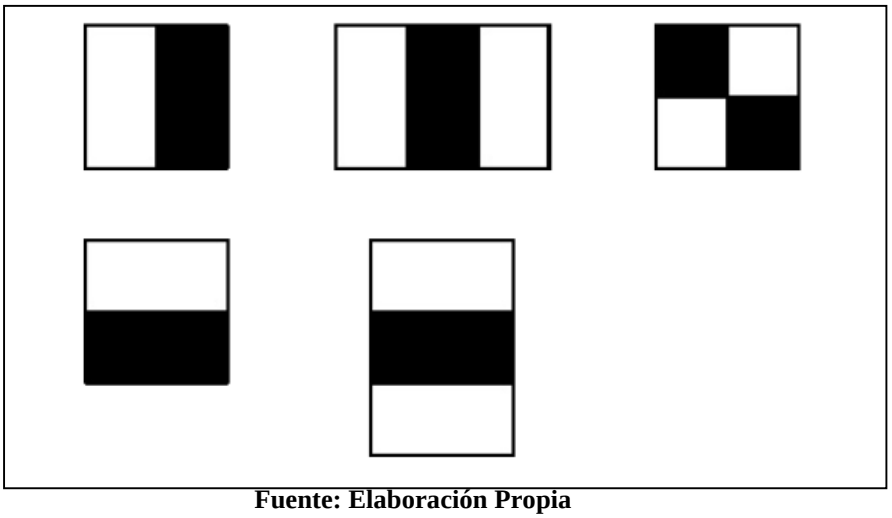
La lógica del algoritmo muestra un enfoque basado en un aprendizaje automático, en donde la función de cascada se entrena de muchas imágenes positivas y negativas para que después sea utilizado para detectar objetos (rostro) en las demás imágenes. Paro ello, el algoritmo necesita muchas imágenes positivas, es decir imágenes de rostro, y muchas imágenes negativas, es decir imágenes sin rostro, para que el clasificador en cascada sea entrenado.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Según los autores de este algoritmo, para este entrenamiento del clasificador en cascada se usan las características de filtro de tipo Haar básico, las cuales se implementan sobre regiones rectangulares de una imagen normalizada a escala de grises. Una característica Haar está compuesto por un número limitado de rectángulos y su valor escalar consiste en la suma de los pixeles de cada rectángulo sumados aplicando un valor especifico como factor de peso.

Estos valores de características Haar se generan de los modelos básicos de filtro de tipo Haar, los cuales son los siguientes:

Figura 16: Modelos básicos de filtros de tipo Haar.



Fuente: Elaboración Propia

Para calcular el valor de una característica de tipo Haar de cada pixel se usa la siguiente fórmula en base a la combinación de los modelos mostrados anteriormente:

$$\text{característica} = \sum_{1 \le i \le n} \text{peso}_i \times \text{pixeles}_i$$

Donde:

$\{r_1, \ldots, r_n\}$ = Conjunto de rectángulos que forman la característica.

w_i = Peso de cada uno.

Estas características de tipo Haar son como el núcleo convolucional. Cada una resulta de un único valor que se genera

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

restando la suma de píxeles del rectángulo blanco de la suma de píxeles del rectángulo negro.

La fórmula matemática es la siguiente:

$$R = \sum_{(x,y) \in N} f(x,y) - \sum_{(x,y) \in B} f(x,y)$$

Donde:

N: La zona de contribución positiva del filtro. (Color negro)

B: La zona de contribución positiva del filtro. (Color Blanco)

x,y: Posición de los pixeles de la imagen.

R: Resultado del filtro.

Un ejemplo de lo explicado se muestra a continuación en la imagen 17, donde se muestra la intensidad de los pixeles de una imagen:

Figura 17: Modelos básicos de filtros de tipo Haar.

200	200	100	100	200	200	100	100
250	250	50	50	250	250	50	50
255	255	255	255	100	100	100	100
255	255	255	255	100	100	100	100
200	200	100	100	200	200	100	100
250	250	50	50	250	250	50	50
255	255	255	255	100	100	200	200
255	255	255	255	100	100	250	250

Fuente: Elaboración Propia

Entonces se calcula la zona de distribución positiva:

N=200+200+250+250

N=900

Y la zona de distribución negativa:

B=100+100+50+50

B=300

El resultado del filtro haar es el siguiente:

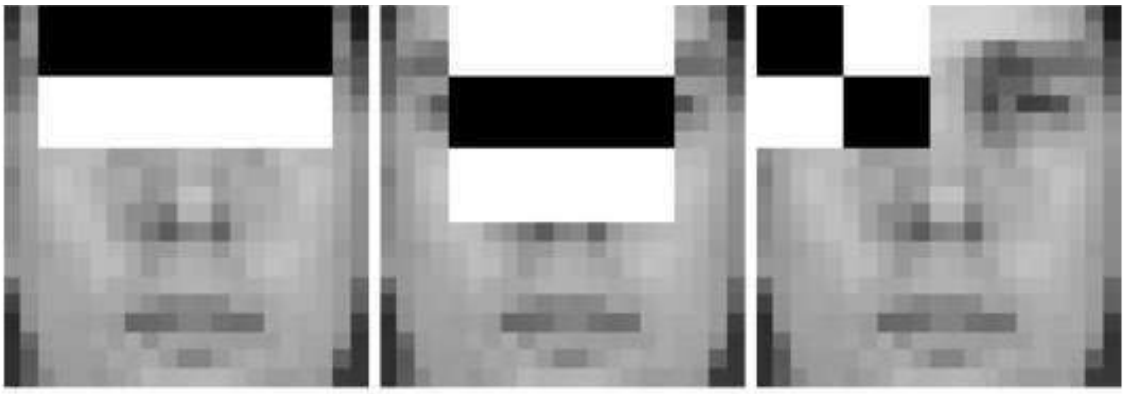
R=900-300

R=600

Cabe resaltar que las combinaciones que se hacen de los modelos de filtros básicos de tipo Haar varían de acuerdo al tamaño de la imagen.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 18: Ejemplos de filtros Haar aplicados en una cara.



Fuente: Elaboración Propia

Luego que se sacan los resultados de cada pixel con la fórmula anterior se obtiene un nuevo clasificador final, que es nada más y nada menos que la suma ponderada de estos clasificadores débiles. Se le llama débil por la razón de que solo no es capaz de clasificar la imagen, pero en conjunto con otros forman un clasificador fuerte.

El algoritmo que demuestra la fórmula matemática y obtiene el resultado del filtro haar se muestra en la figura 19.

Figura 19: Pseudocódigo de la obtención del resultado del filtro haar

```
Inicio
Leer MatrizNegra[][];
Leer MatrizBlanca[][];
Entero N=0,B=0;
Para (Entero i=0; i<MatrizNegra.tamañoHorizontal, i++)
        Para(Entero j=0; j<MatrizNegra.tamañoVertical,j++)
                N=N+MatrizNegra[i][j];
        Fin Para
Fin Para
Para (Entero i=0; i< MatrizBlanca.tamañoHorizontal, i++)
        Para(Entero j=0; j< MatrizBlanca.tamañoVertical,j++)
                B=B+ MatrizBlanca [i][j];
        Fin Para
Fin Para
R=N-B;
Retornar R
Fin
```

Fuente: Elaboración Propia

El artículo de investigación de Viola y Jones dice que incluso 200 características proporcionan detección con 95% de precisión. De esta manera su configuración final abarca alrededor de 6000 características.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Así que a partir de aquí el algoritmo interactúa con la imagen en función a los pixeles, aplicando alrededor de 6000 características a la misma.

La mayor parte de la región de la imagen es el área que no contiene el rostro, por lo tanto, el algoritmo posee un método sencillo para comprobar si una imagen no es una región de rostro. Si no lo es, entonces la desecha y no vuelva a procesarlo.

De esta manera se agiliza el proceso y se ahorra tiempo de ejecución para comprobar una posible región de rostro.
Ahora en lugar de aplicar todas las 6000 funciones en la imagen, lo que hará es agrupar las características en diferentes etapas de los clasificadores y las aplicará una por una.
Entonces si el proceso falla en la primera etapa, la desecha y no considera las características restantes en él. Si pasa la primera etapa, entonces aplica la segunda etapa de funciones y continúa el proceso del algoritmo. Para el final la imagen que pasa por todas las etapas es una región que posee un rostro.

Y así de manera sencilla e intuitiva es cómo funciona el algoritmo de la detección de rostros y ojos de Viola-Jones.

- **Algoritmo de detección:**

  Como habíamos dicho anteriormente OpenCv contiene al algoritmo Viola Jones en un archivo XML listo para ser implementado en el entorno de desarrollo Android Studio.

  Para detectar rostros es: **haarcascade_frontalface_default.xml**
  El código de la implementación del algoritmo Viola Jones se muestra en la figura 20 y figura 21.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 20: Código del algoritmo Viola Jones.

```
public bool detectMultiScale (IntegralImage2 image, Rectangle rectangle, 1.1, 3,
CASCADE_FIND_BIGGEST_OBJECT | CASCADE_DO_ROUGH_SEARCH){
  int x = rectangle.X;
  int y = rectangle.Y;
  int w = rectangle.Width;
  int h = rectangle.Height;
  double mean = image.GetSum(x, y, w, h) * invArea;
  double factor = image.GetSum2(x, y, w, h) * invArea - (mean * mean);
  factor = (factor >= 0) ? Math.Sqrt(factor) : 1;
  // Para cada etapa de clasificacion en la cascada
  foreach (HaarCascadeStage stage in cascade.Stages) {
    // Comprueba si el filtro ha rechazado la imagen
    if (stage.Classify(image, x, y, factor) == false)      {
      return false; // La imagen ha sido rechazada
    }
  }
  // Si el objeto ha pasado todos los filtros y no ha
  //  sido rechazado, el objeto ha sido detectado
  return true; // La imagen facial a sido detectada.
}
```

Fuente: Elaboración Propia

El código mostrado en la figura 21 muestra el funcionamiento del algoritmo de Viola Jones que funciona para reconocer rostros en una imagen.

Figura 21: Código del algoritmo Viola Jones.

```
public bool Classify(IntegralImage2 image, int x, int y, double factor){
  double value = 0;
  // Se recorre cada característica de tipo haar en la etapa actual
  foreach (HaarFeatureNode[] tree in Trees)  {
    int current = 0;
    do {
      // Se obtiene el nodo con la característica del filtro
      HaarFeatureNode node = tree[current];
      // Evalua la característica del nodo
     //GetSum es el método descrito en la Figura 20
      double sum = node.Feature.GetSum(image, x, y);
      // Aumenta el acumulado del valor
      if (sum < node.Threshold * factor) {
        value += node.LeftValue;
        current = node.LeftNodeIndex;
      }
      else{
        value += node.RightValue;
        current = node.RightNodeIndex;
      }
    } while (current > 0);
  }
  // Después de que se ha evaluado la salida para
  //  la etapa actual, se comprueba si el valor
  //  es aún menor que el umbral de la etapa.
  if (value < this.Threshold) {
    // Si es así, la etapa ha rechazado los filtros
    // de la imagen y no contiene el objeto facial.
    return false;
  }
  else{
    // El escenario ha aceptado la imagen actual.
    return true;
  }
}
```

Fuente: Elaboración Propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Para que sea implementado se debe crear un objeto de la clase CascadeClassifier, que es una clase netamente creada para clasificar archivos XML.

A continuación, se explica cómo se realizó la etapa de la detección de rostro:

Primero, se inicializó el detector de la clase CascadeClassifier con el nombre faceDetector para que luego cargue y lea con el archivo indicado, se asume que el archivo se encuentra en la carpeta del proyecto.

```
CascadeClassifier faceDetector;
faceDetector.load ("haarcascade_frontalface_default.xml"))
```

A continuación, se llamó al método “detectMultiScale” para enviarle los parámetros respectivos entre ellos la imagen ecualizada, de esta forma se pasó a detectar el rostro presente en la imagen, las coordenadas se guardaron en un vector llamado Coordenadas.

```
Vector<Rect> Coordenadas;
faceDetector.detectMultiScale (histograma, Coordenadas, 1.1, 3,
                               CASCADE_FIND_BIGGEST_OBJECT |
                               CASCADE_DO_ROUGH_SEARCH);
```

Luego se recorrieron las coordenadas encontradas por el algoritmo de detección de rostro y se pintaron en la imagen original para que de esa forma quede visible.

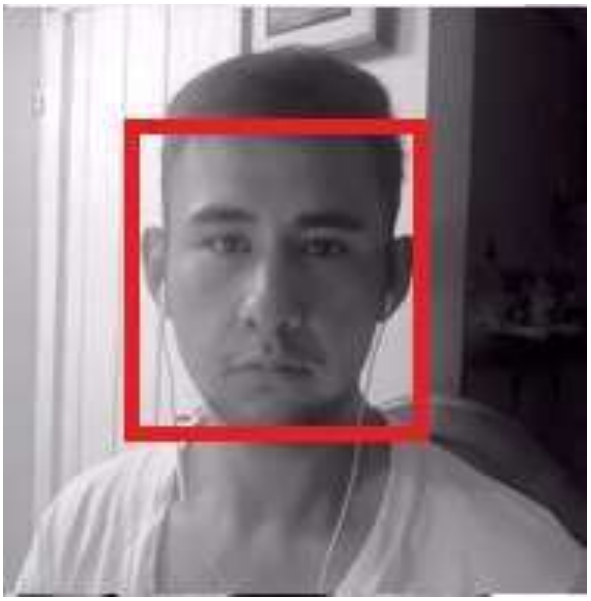
```
For (Rect r: Coordenadas)
{
Rectangle (imagen,
Point (rc.x, rc.y),
Point (rc.x + rc.width, rc.y +
rc.height),
CV_RGB (0, 255, 0), 2);
}
```

El resultado de la detección del rostro de muestra en la figura 22.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Luego que se terminó de detectar la cara, la imagen ya está lista para pasar a la fase de recorte de cara.

Figura 22: Resultado de la Detección de Rostro.



Fuente: Elaboración propia

d) Recorte de cara:

El siguiente paso es el recorte de cara, el cual consiste en extraer la parte de la región de la cara detectada en las fases anteriores, todo esto con ayuda de las coordenadas válidas del rostro obtenidas anteriormente para recortarlo y alinearlo.

El proceso de recorte de la cara consiste en aplicar transformaciones de rotación, transformaciones de escala y luego el recorte, todo esto en base a la imagen en ecualización de histogramas.

La figura 23 muestra un algoritmo que devuelve el rostro alineado y recortado en función a los parámetros que se le envían:

Este resultado es el que se almacenara en la base de datos junto con todas las imágenes pre procesadas para de esta manera cumplir con el protocolo de adquisición de imágenes.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 23: Código de recorte de cara.

```
      Point left = Point(lEye.x + lEye.width/2, lEye.y +lEye.height/2);
       Point right = Point(rEye.x + rEye.width/2, rEye.y +rEye.height/2);
       Point2f eyesCenter = Point2f( (left.x + right.x) * 0.5f, (left.y + right.y) * 0.5f );
       // Obtengo el angulo entre los 2 ojos
       double dy = (right.y - left.y);
       double dx = (right.x - left.x);
       double len = sqrt(dx*dx + dy*dy);
       double angle = atan2(dy, dx) * 180.0 / CV_PI;
       // Las mediciones manuales mostraron que el centro del ojo izquierdo idealmente debería estar
       // a aproximadamente 0,19, 0,14 de una imagen de la cara escalada.
       const double DESIRED_RIGHT_EYE_X = (1.0f - DESIRED_LEFT_EYE_X);
       // Obtengo la cantidad que necesitamos para escalar la imagen para que sea el tamaño fijo que deseamos.
       double desiredLen = (DESIRED_RIGHT_EYE_X - DESIRED_LEFT_EYE_X) * FaceWidth;
       double scale = desiredLen / len;
       // Obtengo la matriz de transformación para girar y escalar la cara al ángulo y tamaño deseados.
       Mat rot_mat = getRotationMatrix2D(eyesCenter, angle, scale);
       // Cambio el centro de los ojos para que sea el centro deseado entre los ojos.
       rot_mat.at<double>(0, 2) += FaceWidth * 0.5f - eyesCenter.x;
       rot_mat.at<double>(1, 2) += FaceHeight * DESIRED_LEFT_EYE_Y - eyesCenter.y;
       //Preparamos el frame que contendra la cara recortada
       warped = Mat(FaceHeight, FaceWidth, CV_8U, Scalar(128));
       //Obtenemos la region de la cara de la imagen ecualizada anteriormente
      face=gray(rect[0]);
       //Hacemos el recorte de la cara deacuerdo al frame que lo contiene
       warpAffine(face, warped, rot_mat, warped.size());
        //
      }
}
 return warped;
```

Fuente: Elaboración propia

El resultado del recorte se aprecia en la figura 24.

Figura 24: Resultado del recorte de la cara.

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

5.3. Selección del Método de Reconocimiento Facial:

Una vez terminado la fase de recorte, se da por culminado la fase de pre procesamiento de imágenes, a continuación, se pasó a seleccionar que métodos de reconocimiento facial eran más factibles para comparar y medir el rendimiento, eficacia en cuando a seguridad e iluminación mediante una matriz de comparación.

La matriz de comparación de métodos de reconocimiento facial en base a los resultados de los experimentos en los papers del estado de arte de la presente investigación se aprecia en la tabla 9.

Tabla 7: Matriz de comparación de métodos de reconocimiento facial

Método	Paper	Muestra	Resultados
Eigenfaces	En 2014, Oka Sudana Darma Putra y Alan Arismandiza publicaron un paper en "Department of Information Technology, Udayana University, Indonesia" mostrando la implementación del método Eigenfaces con el objetivo de implementar el reconocimiento facial en celulares Android	Se hicieron 250 experimentos con 50 imágenes	La mejor tasa de precisión es de 94,48%. El Porcentaje de error es igual a 3%. El sistema tiene una precisión rechazada de aproximadamente el 97,48% y la tasa de exactitud aceptada de alrededor del 97%.
PCA	En 2012, Wang Tuanzhi y Luo Xioa publicaron un paper en "Journal of Digital Information Management" implementaron el método PCA para reconocimiento facial en entornos controlados.	Usaron 200 imágenes, de 40 persona sacando 5 fotos por persona	La tasa de efectividad del reconocimiento facial dio un porcentaje de 95.8%
FisherFaces	En 2016, Mikhail Gofman y Sinjini Mitra publicaron un paper en "Communications of the ACM" en donde implementaron un sistema de reconocimiento facial y voz, para lo cual usaron el método FisherFace en el reconocimiento facial.	Usaron 480 imágenes, de 54 personas	El promedio del tiempo de respuesta en cuando al reconocimiento fue un 0.108 segundo, por otro lado el indicador error dio un 2.14 %
LBP	En 2015, Ravibabu y Krishnan publicaron un paper en "IEEE International Conference on Computational Intelligence and Computing Research" en el cual mostraron un sistema de reconocimiento facial en contra de la falsificación para lo cual el método más destacado y efectivo fue LBP.	Usaron 15 imágenes por persona, 3 imágenes por 5 aspectos diferentes de iluminación	El porcentaje de efectividad se divide de acuerdo al ambiente de iluminación: En ambientes al aire libre con luz del día obtuvo un 100%, en ambientes de modo noche al aire libre obtuvo un 75%, en iluminación de luz del día cubierta un 100% y en ambiente de noche interior un 75%

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

De acuerdo al análisis hecho en base a la matriz de comparación de resultados de los métodos de reconocimiento facial descrita anteriormente, se concluyó que los métodos más efectivos en ambientes no controlados en cuanto a seguridad e iluminación son:

- Método FisherFaces
- Método LBP

5.4. Implementación de los métodos seleccionados en un lenguaje de programación:

a) FisherFaces:

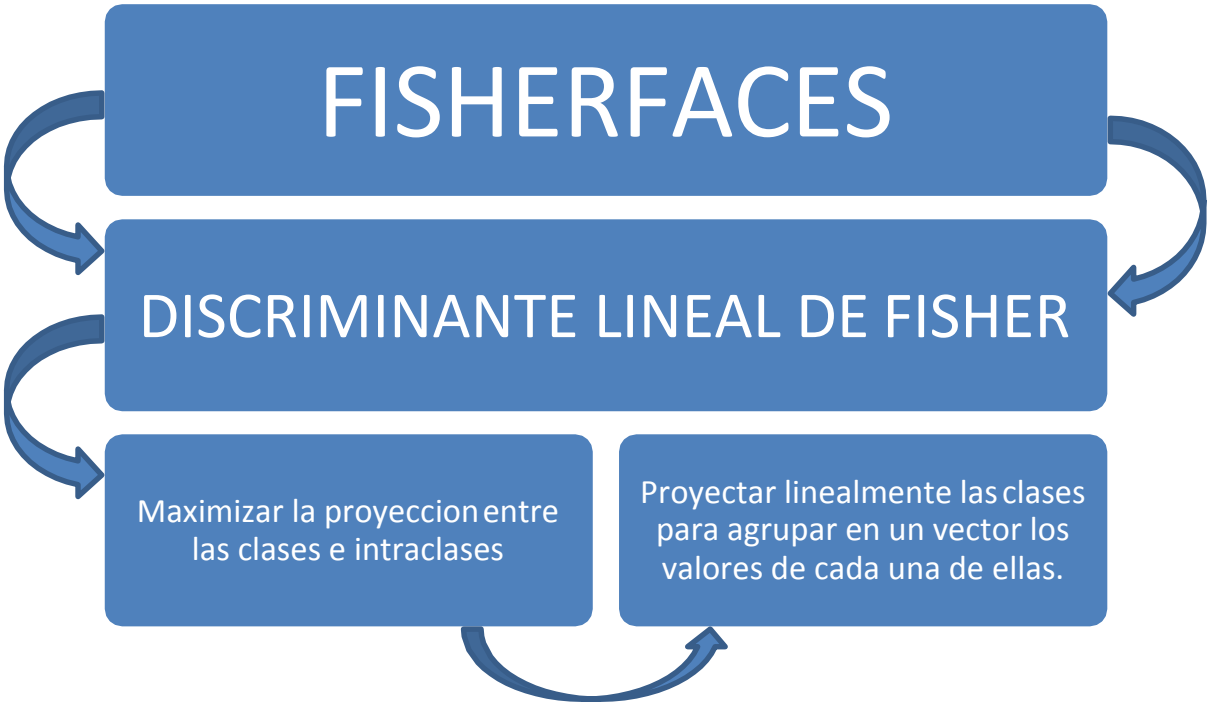
El FisherFaces es un método utilizado para el reconocimiento facial, trabaja teniendo en cuenta el reflejo de la luz y las expresiones faciales tales como gestos, es por eso que se recomienda para mejorar la seguridad de este proceso de reconocimiento facial. (Ravibabu & Krishnan, 2015)

Según Belhumeur, Hespanha & Kriegman (1996) en su libro “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection”, plantea que este algoritmo maximiza la relación entre la distribución de las clases (proyección que maximiza la separación entre imágenes de diferentes personas) y la distribución intra-clases (proyección que minimiza la distancia entre imágenes de una misma clase), FisherFaces clasifica y reduce la dimensión de las caras utilizando el método Discriminante Lineal de Fisher (FLD). Éste método crea una proyección lineal que maximiza las diferentes imágenes de caras proyectadas.

En la Figura 25 se muestra en síntesis cómo funciona el método Fisherfaces.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 25: Esquema de síntesis del método FisherFaces.

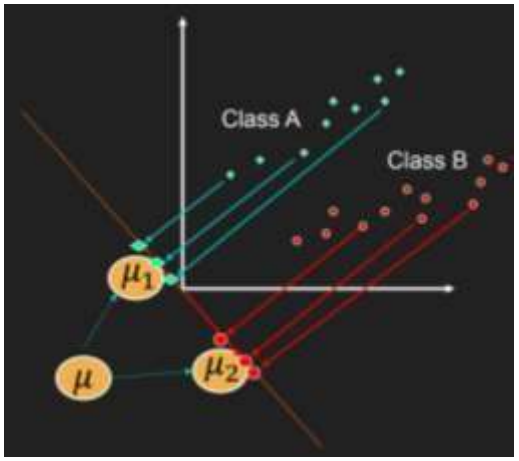


Fuente: Elaboración propia

Como se dijo el método FisherFaces utiliza el Discriminante Lineal de Fisher (FLD), para la reducción de dimensión, maximizando la distribución

Figura 26: Esquema de la distribución entre clases e intra-clases.

clases), tal como muestra la Figura 26.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Para esto se define la matriz S_B de distribución entre clases como:

$$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

Y la matriz S_W de distribución intra-clases:

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu)^T$$

Donde:

c **=** Es la cantidad de personas.

x **=** Vector con muestras extraídas de C clases.

μ_i **=** Es la media del vector de clase X_i.

μ **=** Es la media total de todas las caras.

N_i**=** Es el número de imágenes de la clase X_i

Para calcular μ se usa la siguiente fórmula:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

El algoritmo que sintetiza el funcionamiento de cómo obtener la media total de las caras se muestra en la Figura 27.

Figura 27: Pseudocódigo para obtener la media total de los rostros

```
Inicio
Leer N//Cantidad de imágenes
Leer X [N] // Vector con N muestras por cada persona
Declarar  M = 0 // Media de la clase
Para (entero i=1, i <= N, i++)
        M  = M+ X[i]
Fin Para
M= (1/N)*M
Retornar M
Fin
```

Fuente: Elaboración propia

A partir de aquí se obtiene la distribución entre clases (S_B) con el algoritmo de la Figura 28.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 28: Pseudocódigo para obtener la distribución entre clases

```
Inicio
Leer N//Cantidad de imágenes
Leer C //Cantidad de personas
Leer M [ ] // Vector con la media de cada persona
Leer Me // Media total de todas las imágenes
Declarar Sb = 0 // Distribución entre las clases
Para (entero i=1, i <= C, i++)
        Sb = Sb + (N  * (M[i] – Me))
Fin Para
Retornar Sb
Fin
```

Fuente: Elaboración propia

Para obtener la distribución intra-clases se utiliza el algoritmo de la Figura 29.

Figura 29: Pseudocódigo para obtener la distribución intra- clases

```
Inicio
Leer N//Cantidad de imágenes
Leer C //Cantidad de personas
Leer M [ ] // Vector con la media de cada persona
Leer X[N] // Vector de muestras por cada persona
Leer Me // Media total de todas las imágenes
Declarar Sw = 0 // Distribución entre las clases
Para (entero i=1, i <= C, i++)
        Para (entero k=1, k <= N, k++)
                Sw = Sw + (N  * (X[k] – M[i]))
        Fin Para
Fin Para
Retornar Sw
Fin
```

Fuente: Elaboración propia

De manera que el cociente entre la distribución entre clases y la distribución intra-clases sea máxima.

$$W_{opt} = \arg\max \frac{|W^T S_B W|}{|W^T S_W W|} \qquad (1)$$

$$W = [w_1, w_2, w_3, \dots, w_m]$$

Donde $\{w_i \mid i = 1,2,3, \dots, m\}$ es el conjunto de valores propios de S_B y S_W correspondiente a los m mayores valores propios $\{\lambda_i \mid i = 1,2,3, \dots, m\}$ esto es:

$$S_B w_i = \lambda_i S_W w_i \qquad i = 1,2,3, \dots, m$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Se observa entonces, que a lo sumo se tienen c−1 valores propios distintos de cero, y por lo tanto el límite superior de m es c−1, donde c es el número de clases. Para el problema de reconocimiento de caras, se tiene que la matriz $S_W \in \Re^{n \times n}$ es siempre singular, dado que el rango de S_W es a lo sumo N − c, y en general, el número de imágenes de entrenamiento: N, es mucho más chico que el número de píxeles de cada imagen: n. Por lo tanto, puede ser posible elegir una matriz W tal que la distribución intra-clases de las imágenes proyectadas pueda ser exactamente cero. Como alternativa entonces, al criterio establecido en la ecuación (1), se proyecta el conjunto de imágenes a un espacio de menor dimensión, de manera que la matriz resultante de la distribución intra-clases S_W es no singular.

El problema de optimización puede reescribirse como:

$$W_{fld} = \arg\max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}$$

La matriz W, es la que proyecta una muestra en el c−1 espacio dimensional y viene dada por: $W = W_{fld}^T$

En la Figura 30 se muestra el algoritmo de cómo obtener la proyección máxima y mínima de las clases e intra-clases.

Figura 30: Pseudocódigo para obtener las proyección de las clases e intra-clases

```
Inicio
Leer Sb,Sw
Leer m // Cantidad de proyeccion esperadas
Declarar Wopt [m] // Vector con las distintas proyección
Declarar W=0
Para (entero i=1, i <= m, i++)
        Wopt[i]=Sb/Sw
Fin Para
Para (entero i=1, i <= m, i++)
        W=W+Wopt[i]
Fin Para
W=W/m
Retornar W
Fin
```

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En la Figura 31 se muestra el algoritmo sintetizado de cómo funciona el discriminante lineal de Fisher.

Figura 31: Código de discriminante lineal de Fisher.

```
// Obtener matrices de datos
Mat W = _W.getMat();
Mat mean = _mean.getMat();
Mat src = _src.getMat();
// obtener el número de muestras y la dimensión
int n = src.rows;
int d = src.cols;
// asegúrese de que la media sea correcta si no está vacía
if(!mean.empty() && (mean.total() != (size_t) d)) {
        String error_message = format("Wrong mean shape for the given data matrix. Expected %d, but was %d.", d,
mean.total()); CV_Error(Error::StsBadArg, error_message);
}
// crear matrices temporales
Mat X, Y;
// asegúrese de operar con el tipo correcto
src.convertTo(X, W.type());
// seguro de hacer, debido a la afirmación anterior
if(!mean.empty()) {
        for(int i=0; i<n; i++) {
                Mat r_i = X.row(i);
                subtract(r_i, mean.reshape(1,1), r_i);
        }
}
//finalmente calcule la proyección como Y = (X - media) * W
gemm(X, W, 1.0, Mat(), 0.0, Y);
return Y;
```

Fuente: Elaboración propia

De esta manera todas las fórmulas matemáticas aplicada para la discriminante lineal de Fisher fueron implementadas tal como se muestra en la figura 31, el cual recibe la imagen que fue almacenada en la base de datos de entrenamiento y también el nombre o etiqueta de la persona, el algoritmo codificado del método general de FisherFaces se describe paso a paso en la figura 32.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 32: Código del entrenamiento del método FisherFaces.

```
void Fisherfaces::Recognizer(InputArrayOfArrays _src, InputArray _local_labels) {
        if(_src.total() == 0) {
                String error_message = format("Empty training data was given. You'll need more than one sample to
        learn a model.");
                 CV_Error(Error::StsBadArg, error_message);
        } else if(_local_labels.getMat().type() != CV_32SC1) {
                String error_message = format("Labels must be given as integer (CV_32SC1). Expected %d, but was
                %d.", CV_32SC1, _local_labels.type());
        }
        if(_src.total() > 1) { // Asegúrese de que los datos tengan el tamaño correcto
                for(int i = 1; i < static_cast(_src.total()); i++) {
                        if(_src.getMat(i-1).total() != _src.getMat(i).total()) {
                                String error_message = format("¡En el método Fisherfaces todas las muestras de
                        entrada (imágenes de entrenamiento) deben ser del mismo tamaño! Expected %d pixels, but
                        was %d pixels.", _src.getMat(i-1).total(), _src.getMat(i).total());
                        }
                }
        }
        Mat labels = _local_labels.getMat();// Obtener etiquetas
        Mat data = asRowMatrix(_src, CV_64FC1); // observaciones en fila
        int n = data.rows; // número de muestras
        _labels.release();// borrar los datos del modelo existente
        _projections.clear();
        if((_num_components <= 0) || (_num_components > n)) // cantidad de clips de componentes para ser válida
                _num_components = n;
        fdl(data, Mat(), FLD::DATA_AS_ROW, _num_components); // Realizar el FLD
        // Copiar los resultados de FLD
        _mean = fld.mean.reshape(1,1); // Almacenar el vector malo
        _fishervalues = fld.fishervalues.clone(); // Valores propios por fila
        transpose(fld.fishervectors, _fishervectors); // fishervectores por columna
        // Almacenar etiquetas para predicción
        _labels = labels.clone();
        for(int sampleIdx = 0; sampleIdx < data.rows; sampleIdx++) { // guardar proyecciones
                Mat p = LDA::subspaceProject(_eigenvectors, _mean, data.row(sampleIdx));
                _projections.push_back(p);
        }
}
```

Fuente: Elaboración propia

b) Patrón Binario Local:

Este método se concentra en extraer las características locales de las imágenes. La idea principal y la lógica de los algoritmos de este método es no mirar toda la imagen como un vector de alta dimensión, sino describir solo las características locales de un objeto. De esta manera las características extraídas tendrán un tipo de dimensionalidad baja de forma implícita.

Según la web oficial de OpenCv se dice que la idea básica de los patrones binarios locales es resumir la estructura local en una imagen comparando

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

cada píxel con su vecindad. Tome un píxel como centro y limite a sus vecinos. Si la intensidad del píxel central es mayor-igual a su vecino, entonces denote con 1 y 0 si no es así. Obtendrá un número binario para cada píxel, al igual que 11001111. Por lo tanto, con 8 píxeles circundantes, obtendrá 2 ^ 8 combinaciones posibles, denominadas *"Patrones Binarios Locales"* o, a veces, denominados "*códigos LBP* ".

Una descripción más formal del operador de LBP se puede dar como:

$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

Dónde:

(x_c, y_c) = Es el pixel central

i_c = Intensidad del pixel central en escala de grises

i_p = Intensidad del pixel vecino

s = Función

Con (x_c, y_c) un píxel central con intensidad i_c e i_p siendo la intensidad del píxel vecino. S es la función del signo definida como:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

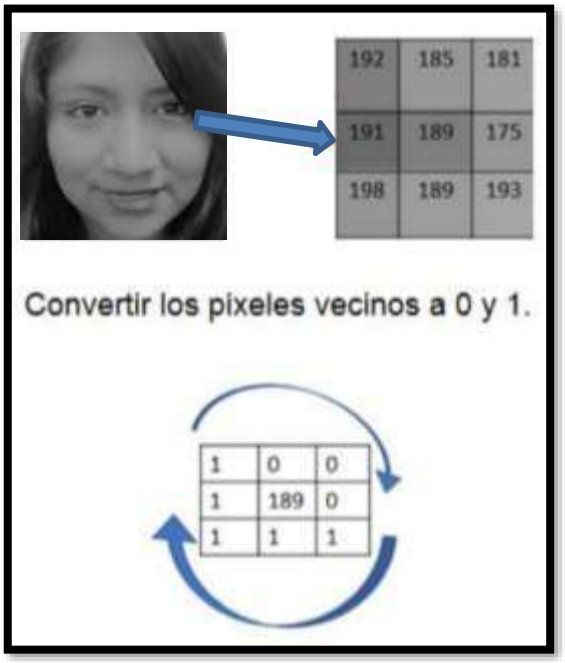
Donde el algoritmo para cada pixel de la imagen reconoce 8 pixeles vecinos alrededor del pixel central, teniendo una dimensión de 3x3. Por lo tanto, se determinó que cada pixel vecino se verifico la intensidad por medio de su valor: Si el pixel vecino es mayor o igual que el pixel central es 1; si el pixel vecino es menor que el pixel central es 0.

Por ejemplo, como se muestra en la Figura 33 el valor del pixel central es 189 entonces el pixel vecino 192 es mayor por lo que el valor es 1; como se ve en el segundo caso el pixel vecino es 185 por lo tanto el valor es 0 remplazando la misma posición del valor del pixel. De manera que se

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

codifico en forma horaria en una cadena binaria formada de 8 números (Yulian André Cama Castillo, 2015).

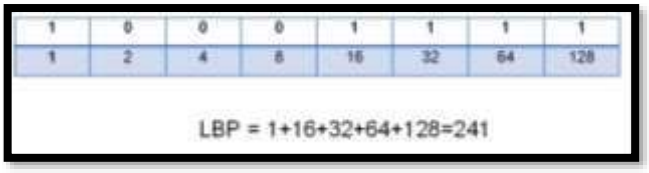
Figura 33: *Proceso para la creación del operador Patrón Binario Local con una vecindad de 3x3***.**



Fuente: Elaboración propia

Una vez realizado la transformación a binario se calculó el decimal como se visualiza en la Figura 34 teniendo como resultado del ejemplo un decimal de 241 para el mejor desarrollo del algoritmo.

Figura 34: *Conversión de los vecinos 3x3 a decimales***.**



Fuente: Elaboración propia

Para la implementación de la fórmula matemática se realizó el pseudocódigo mostrado en la figura 35.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 35: Pseudocódigo para la generación de los patrones binarios locales.

```
Inicio
        Entrada matriz;
        Generar matriz_LBP[fila_matriz-2, columna_matriz-2];
        Para(i=1; i< filas_matriz-1; aumentar i) hacer
                Para(j=1; j< columnas_matriz; aumentar i) hacer
                        pixel_centro = elemento_matriz[f,c];
                        código_LBP = 0;
                        código_LBP |= (elemento_matriz(i-1,j-1)>= pixel_centro) <<7;
                        código_LBP |= (elemento_matriz(i-1,j)>= pixel_centro) <<6;
                        código_LBP |= (elemento_matriz(i-1,j+1)>= pixel_centro) <<5;
                        código_LBP |= (elemento_matriz(i,j+1)>= pixel_centro) <<4;
                        código_LBP |= (elemento_matriz(i+1,j+1)>= pixel_centro) <<3;
                        código_LBP |= (elemento_matriz(i+1,j)>= pixel_centro) <<2;
                        código_LBP |= (elemento_matriz(i+1,j-1)>= pixel_centro) <<1;
                        código_LBP |= (elemento_matriz(i,j-1)>= pixel_centro) <<0;
                        elemento_matriz_LBP[i,j] = código_LBP;
                Fin Para
        Fin Para
Fin
```

Fuente: Elaboración propia

El algoritmo codificado descrito anteriormente esta implementado en la figura 36, en donde se puede visualizar la implementación de las fórmulas matemáticas.

Figura 36: Implementación de las fórmulas matemáticas del método.

```
Template <typename_Tp> static void olbp_(InputArray_src, OutputArray_dst){
        Mat src = _src.getMat();// Obtener matrices
        _dst.create(src.rows-2,src.cols-2, CV_8UC1); // Asignar memoria para resultado
        Mat dst = _dst.getMat();
        // Cero la matriz de resultados
        dst.setTo(0);
        // Calcular patrones
        For(int i=1; i<src.rows-1; i++){
                For(int j=1; j<src.cols-1; j++){
                        _Tp center = src.at<_Tp>(i,j);
                        Unsigned char code = 0;
                        Code |=(src.at<_Tp>(i-1,j-1) >= center) << 7;
                        Code |=(src.at<_Tp>(i-1,j) >= center) << 6;
                        Code |=(src.at<_Tp>(i-1,j+1) >= center) << 5;
                        Code |=(src.at<_Tp>(i,j+1) >= center) << 4:
                        Code |=(src.at<_Tp>(i+1,j+1) >= center) << 3;
                        Code |=(src.at<_Tp>(i+1,1) >= center) << 2;
                        Code |=(src.at<_Tp>(i+1,j-1) >= center) << 1;
                        Code |=(src.at<_Tp>(i,j-1) >= center) << 0;
                        dst.at<unsigned chat>(i-1,j-1) = code;
                }
        }
}
```

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Luego la imagen o fotos del rostro se divide en diferentes ventanas, la cual cada ventana genero diferentes histogramas, indicando que cada histograma de las ventanas es único, que se va concatenando en secuencia, con el fin de obtener un vector de histogramas, por lo que guardara la información espacial. Donde el proceso se muestra a continuación:

En la Figura 37 por ejemplo: es la imagen de entrada de tamaño 128*128. Donde la imagen completa es representada M * N dicha imagen es procesada de manera que se empieza a iterar a través de los pixeles para generar ventanas más pequeñas de toda la imagen que contiene información relevante del rostro.

Figura 37: Imagen pre procesada de entrada.

Fuente: Elaboración propia

Donde:

R = Imagen completa de 16 384 pixeles

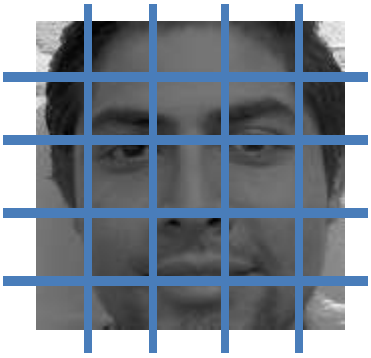
M = Alto de la imagen

N = Ancho de la imagen

En este caso se generaron 8 x 8 ventanas con un total de 64 ventanas generadas de toda la imagen ingresada de M * N = 16 384 pixeles, cada ventana tiene un rango de 16 pixeles en la fila y 16 pixeles en la columna con un total de 256 pixeles por ventana, entonces al multiplicar 64 ventanas por 256 pixeles se obtiene R que es el total de pixeles.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 38: Imagen divido en ventanas



Fuente: Elaboración propia

El cual la imagen está representada en la siguiente fórmula matemática.

$$\text{V} = (\text{X} * \text{Y}) * K$$

Dónde:

V = Ventana generada

X = Ancho de la ventana de la imagen

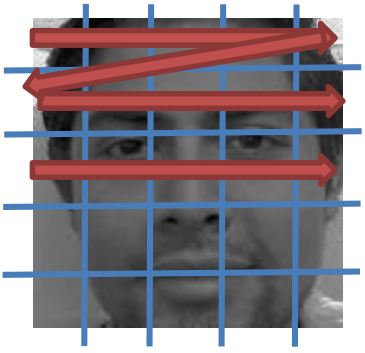
Y = Alto de la ventana de la imagen

K = Numero de Ventanas

R = Foto completa de 128 * 128 pixeles

Las ventanas son generadas como una matriz de izquierda a derecha como se muestra en la Figura 39.

Figura 39: Secuencia de Generación de ventanas



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

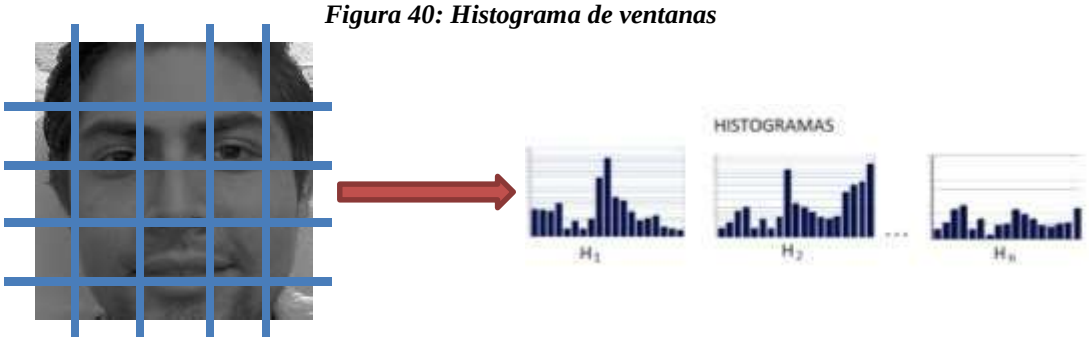
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Esto está reflejado en el código que se visualiza a continuación se aplicó dos for para recorrer toda la imagen pixel a pixel.

```
for(int i = 0; i < grid_y; i++) {
        for(int j = 0; j < grid_x; j++) {
                Mat src_cell = Mat(src, Range(i*height,(i+1)*height), Range(j*width,(j+1)*width));
        }
}
```

Esto con la finalidad de tener un histograma de 8 bits por cada uno de las ventanas donde tiene descripción importante del rostro.

Figura 40: Histograma de ventanas



Fuente: Elaboración propia

Donde la Figura 40 se representa en la siguiente función: la ventana 1 es igual al histograma 1, la ventana 2 es igual al histograma 2 y así 95 sucesivamente dependiendo el número de ventanas será igual el número de histogramas.

$$\text{ventana}_1 \text{ventana}_2 \text{ventana}_3 \ldots \text{ventana}_n = H_1 H_2 H_3 \ldots H_n$$

Entonces el descriptor de toda la imagen es concatenar todos los histogramas, por lo tanto, si tenemos n la dimensión de los histogramas será $59n$ El cual el vector de histogramas sirve para comparar los vectores tanto de la imagen de entrada como las ya entrenadas y poder clasificarlos. Este proceso se realiza tanto para la clasificación como para el entrenamiento

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Para ello se tuvo que implementar el método LBPH donde realiza el procedimiento descrito anteriormente. Esto se hizo con la finalidad de que los histogramas espaciales o las ventanas generadas guarden la información resaltante de la textura o puntos importantes del rostro. Para eso la figura 41 se comenta las principales acciones.

Figura 41: Código del método LBPH.

```
static Mat spatial_histogram(InputArray _src, int numPatterns, int grid_x, int
grid_y, bool) {
        Mat src = _src.getMat();
        // calcular el tamaño del parche LBP
        int width = src.cols/grid_x;
        int height = src.rows/grid_y;
        // asignar memoria para el histograma espacial
        Mat result = Mat::zeros(grid_x * grid_y, numPatterns, CV_32FC1);
        // devolver la matriz con ceros si no se dieron datos
        if(src.empty())
                return result.reshape(1,1);
        // fila de resultado inicial
        int resultRowIdx = 0;
        // iterar a través de la grilla
        for(int i = 0; i < grid_y; i++) {
                for(int j = 0; j < grid_x; j++) {
                        //genera las Ventana
                        Mat src_cell = Mat(src,
                        Range(i*height,(i+1)*height),
                        Range(j*width,(j+1)*width));
                        //Calcula histograma de la ventana
                        Mat cell_hist = histc(src_cell, 0, (numPatterns-1),
                        true);
                        // copiar a la matriz de resultados
                        Mat result_row = result.row(resultRowIdx);
                        cell_hist.reshape(1,1).convertTo(result_row
                        , CV_32FC1);
                        // aumentar el recuento de filas en la matriz de
                        resultados
                        resultRowIdx++;
                }
        }
         // Resultado de devolución como vector de características
        reformado
        return result.reshape(1,1);
}
```

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

c) Implementación en Android Studio:

Primero se adjuntó las librerías “jacpp.jar” y “javacv.jar” para usar las clases que nos ayudaran a implementar los métodos FisherFaces y LBP, con ayuda de los módulos de OpenCv agregados al proyecto de Android Studio se empezó a instancia un objeto de la clase “FaceRecognizer”, esta clase se usa para iniciar la instancia del método FisherFaces y la instancia para el método LBP, estos contienen algoritmos que se describieron anteriormente.

El código de las instancias es el siguiente:

```
FaceRecognizer facerecognizer= createFisherFaceRecognizer (0, Double.MAX_VALUE);
```

```
FaceRecognizer facerecognizer= createLBPHFaceRecognizer ();
```

Para empezar el procesamiento de todas las imágenes de la base de datos con los métodos seleccionados se implementó lo siguiente:
Un vector de tipo Mat, el cual contiene todas las imágenes de la base de datos con su respectivo identificador, para llenar este vector de imágenes de usó el método “put” el cual se encarga de adicional la imagen con su identificador.

```
public void llenarvectordeimagenes(List<Persona> lista){
    imagenes=new MatVector(34);
    int ban=0;
    for (Persona p: lista) {
      Bitmap b=p.getFoto();
      Mat m   = new Mat(b.getWidth(), b.getHeight(), CvType.CV_8UC1);
      Utils.bitmapToMat(b,m);
      Imgproc.cvtColor(m, m, Imgproc.COLOR_BGR2GRAY);
      Imgproc.equalizeHist(m,m);
      IplImage img=MatToIplImage(m,m.width(),m.height());
      imagenes.put(ban,img);
      ban++;
    }
}
```


Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Un vector de tipo entero, el cual contiene todos los índices de cada persona. (Esto fue muy importante ya que con este índice se identificó el nombre de la persona).

```
public void llenarvectordeindices(List<Persona> lista){
    indices=new int[34];

    for (int i = 0; i < indices.length ; i++) {
        indices[i]=lista.get(i).getCodigo()
                          ;
    }

}
```

A partir de aquí se empezó a entrenar todas las imágenes faciales del vector de imágenes con la siguiente sentencia:

```
public void entrenandocaras(){
    facerecognizer.train(imagenes, indices);
}
```

El método donde se implementó todos los procedimientos descritos se llama train donde su finalidad es entrenar a todas las imágenes que se ingresen con su respectiva etiqueta y generar un histograma global por foto para posteriormente realizar una clasificación y mostrar el nombre de la persona que se desea identificar en el sistema. En la figura 42 se muestra de manera comentada los pasos importantes que realiza el método.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 42: Código del método train (Entrenamiento).

```
void METODO::train(InputArrayOfArrays _in_src, InputArray _in_labels, bool preserveData) {
        if(_in_src.kind() != _InputArray::STD_VECTOR_MAT && _in_src.kind() != _InputArray ::
        STD_VECTOR_VECTOR) {
                String error_message = " Las imágenes se esperan como InputArray::
                STD_VECTOR_MAT (a std::vector) or _InputArray::STD_VECTOR_VECTOR (a
                std::vector< std::vector >).";
                CV_Error(Error::StsBadArg, error_message);
        }
        if(_in_src.total() == 0) {
                String error_message = format("Se entregaron datos de entrenamiento vacíos. Necesitarás
                más de una muestra para aprender un modelo.");
                CV_Error(Error::StsUnsupportedFormat, error_message);
        } else if(_in_labels.getMat().type() != CV_32SC1) {
                String error_message = format("Las etiquetas se deben dar como un entero (CV_32SC1).
                Expected %d, but was %d.", CV_32SC1, _in_labels.type());
                CV_Error(Error::StsUnsupportedFormat, error_message);
        }
        // obtener el vector de matrices
         std::vector src;
        _in_src.getMatVector(src);
        // obtener la etiqueta de la matriz
        Mat labels = _in_labels.getMat();
        // verificar si los datos están bien alineados
        if(labels.total() != src.size()) {
                String error_message = format("El número de muestras (src) debe ser igual al número de
                etiquetas (etiquetas). Was len(samples)=%d, len(labels)=%d.", src.size(), _labels.total());
                CV_Error(Error::StsBadArg, error_message);
        }
         // si este modelo debe ser entrenado sin preservar los datos antiguos, elimine los datos del
        modelo anterior
         if(!preserveData) {
         _labels.release();
         _histograms.clear();
         }
         // añadir etiquetas a la matriz de etiquetas
        for(size_t labelIdx = 0; labelIdx < labels.total(); labelIdx++) {
                _labels.push_back(labels.at((int)labelIdx));
         }
         // almacenar los histogramas espaciales de los datos originales
         for(size_t sampleIdx = 0; sampleIdx < src.size(); sampleIdx++) {
                // calcular la imagen
                Mat lbp_image = olbp(src[sampleIdx], _radius, _neighbors);
                // obtener histograma espacial de esta imagen
                Mat p = spatial_histogram( lbp_image, /* lbp_image */ static_cast(std::pow(2.0,
         static_cast(_neighbors))), _grid_x, /* grid size x */ _grid_y, /* grid size y */ true);
                // agregar a las plantillas XML
                _histograms.push_back(p);
         }
}
```

Fuente: Elaboración propia

Una vez que se realizó todo este procedimiento se da por finalizado la implementación de los métodos de reconocimiento facial.

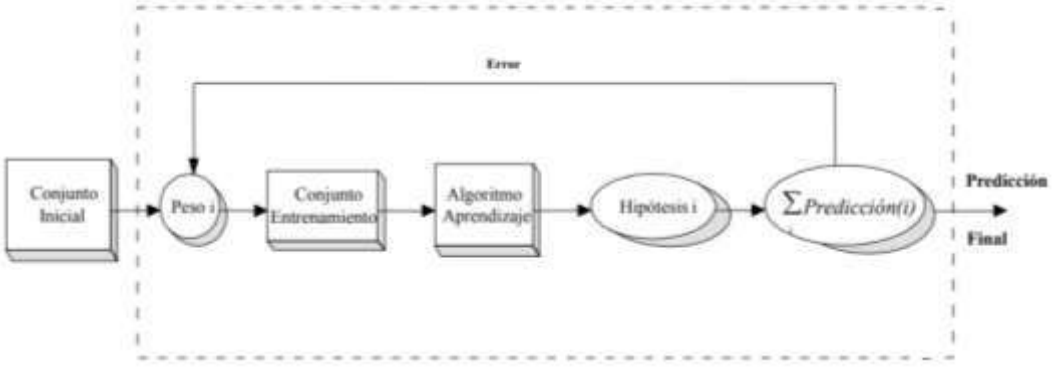
Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

5.5. Implementar un clasificador para el reconocimiento facial:

La importancia de implementar un clasificador de reconocimiento facial radica en que los valores de resultado de cada método antes estudiado tienen que ir ordenados y apareados con el umbral de una imagen que se puso a pruebas.

La elección del clasificador que se usó en esta investigación se basa en el análisis realizado en el capítulo II dentro del estado del arte, en la cual los autores V. Ravibabu y N.Krishnan en su artículo de investigación llamado “Enfoque variado para el reconocimiento facial con mecanismo verdaderos para Android Mobile contra la falsificación” publicado en el año 2015 en la revista de la IEEE “International Conferece on Computacional Intelligence and Computing Research”, en el cual concluyen que el algoritmo de Adaboost es el más apropiado para la clasificación de los valores vectoriales resultantes de los métodos de reconocimiento facial debido a su manera iterativa de entrenar sub clasificadores débiles y sub clasificadores fuertes llegando a una alta efectividad al momento de hacer la comparación del umbral con el vector entrenado y además por motivo que este clasificador probabilístico no consume muchos recursos y realiza un cálculo muy rápido por lo que unido con LBP o FisherFaces se convierte en un método atractivo para un sistema en tiempo real. (Ravibabu & Krishnan, 2015)

Figura 43: Esquema de funcionamiento del algoritmo de Adaboost.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En la Figura 43 se muestra el esquema de funcionamiento del algoritmo de adaboost en la cual combina múltiples hipótesis generadas con el mismo algoritmo de aprendizaje, de esta manera asigna un peso a las hipótesis según su calidad y asi emitir una hipotesis ponderada

a) Algoritmo de Adaboost:

Según Freund y Sachapire en su paper "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting" (1997) el algoritmo Adaboost, cuyo nombre proviene de "adaptative boosting", es un nuevo y mejorado algoritmo de boosting, este algoritmo consiste en la produccion de predicciones muy exactas a partir de combinar clasificadores débiles.

Freund y Sachapire relatan que el algoritmo Adaboost consiste en entrenar de manera iterativa un conjunto de clasificadores base, para que de esta manera cada nuevo clasificador centre su atención a los datos clasificados de forma errónea por los anteriores clasificadores, a partir de aquí se combinan para que se obtenga un nuevo clasificador con elevadas prestaciones y alta efectividad.

El algoritmo consiste en dar ciertos datos de entrada los cuales son: Secuencia de N etiquetas:

$$< (\text{□}_1, \text{□}_1), \dots, (\text{□}_\text{□}, \text{□}_\text{□}) >$$

Una distribución "D" que interactúe con las N etiquetas, de aquí sales lo vectores de pesos.

Inicialización del vector de pesos:

$$\text{□}_\text{□}^1 = \text{□}(i) \text{ , para } i = 1,2 \dots, \text{□}$$

El peso está definido por el número de interacción de la distribución D.

La fórmula para calcular el valor de cada distribución es la siguiente:

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$\text{[illegible]} = \frac{\text{[illegible]}}{\sum_{\text{[illegible]}=1} \text{[illegible]}}$$

El algoritmo que sirve para obtener la distribución de cada valor de los pesos y etiquetas se visualiza en la Figura 44.

Figura 44: Pseudocódigo para la generación de los patrones binarios locales.

```
Inicio
Leer n //Cantidad de etiquetas
Leer w[n] // Vector de pesos
Declarar D[ ], P=0
Para (entero i=0, i<n, i++)
        P=P+w[i]
Fin Para
P=P/n
Para (entero i=0, i<n, i++)
        D[i]=P/w[i]
Fin Para
Retornar D[i]
Fin
```

Fuente: Elaboración propia

La meta del algoritmo es encontrar la hipótesis final con el menor error relativo, dada la distribución D sobre los ejemplos de entrenamiento y la instancia controlada por el clasificador.

El algoritmo mantiene un conjunto de pesos [illegible] sobre el conjunto de entrenamiento. En cada iteración [illegible] la distribución [illegible] es calculada para normalizar esos pesos. Con esta distribución, se alimenta al clasificador débil, que genera una hipótesis $h_{\text{[illegible]}}$ que se espera tenga poco error sobre esa distribución.

La fórmula para calcular la hipótesis $h_{\text{[illegible]}}$ es la siguiente:

$$h_{\text{[illegible]}} = \sum_{\text{[illegible]}=1}^{\text{[illegible]}} \text{[illegible]} \left| h_{\text{[illegible]}}(\text{[illegible]}) - \text{[illegible]} \right|$$

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Con esta nueva hipótesis h_t, el algoritmo genera el siguiente vector de pesos D^{t+1} y el proceso se repite. Luego de T iteraciones, se obtiene la hipótesis final h_f. Esta salida h_f combina las salidas de los T clasificadores débiles utilizando voto pesado por mayoría.

En la figura 45 se muestra las fórmulas matemáticas implementadas dentro del algoritmo de Adabost.

Figura 45: Pseudocódigo de las fórmulas matemáticas del algoritmo Adaboost.

Inicio

Inicializar $D=\{x_1, y_1, x_2, y_2 \ldots, x_m, y_m\}$ $D_1(i) = \frac{1}{m}$, $i = 1, \ldots,$

$t \leftarrow 0$

Hacer $t \leftarrow t + 1$

Entrenar con un aprendizaje débil h_t usando D que muestra datos según la distribución $D_t(i)$.

$\varepsilon_t \leftarrow$ Entrenar el error de h_t dentro de D usando $D_t(i)$.

$\alpha_t \leftarrow \frac{1}{2} \ln[(1 - \varepsilon_t)/\varepsilon_t]$

Si $h_t(x_i) = y_i$ ***Entonces*** (actualizar distribución)

$D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} *$

Si no Entonces (actualizar distribución)

$D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} *$

Fin si

Hasta $K = K_{max}$

Retornar h_t y α_t para k=1 hasta K_{max} (Conjuto de clasificadores con pesos)

Fin

Fuente: Elaboración propia

a.1) Implementación en Android Studio:

Dentro del entorno de programación de Android Studio combinado con los módulos de OpenCv y librerías .jar descritas anteriormente existe el algoritmo de Adaboost implementado dentro del método “predict” en la clase FaceRecognizer.

Este método al momento de ingresar la nueva imagen se genera el histograma para realiza una comparación con el histograma de la imagen ya entrenada y predice cuál de las imágenes se parecen más para posteriormente devolver la etiqueta con el nombre registrado.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Para implementarlo solo hace falta pasarle la imagen procesada de la base de datos como parámetros, un vector de tipo entero para que almacene los resultados de los clasificadores y un vector de tipo doublé para que almacené el valor del umbral o valor de efectividad del reconocimiento.

El código de la implementación se muestra a continuación:

```
public int predict(Bitmap bm) {

    int indicador[] = new int[1];
    double umbral[] = new double[1];
    opencv_core.IplImage ipl =
BitmapToIplImage(bm,bm.getWidth(), bm.getHeight());
    faceRecognizer.predict(ipl, indicador, umbral);
    if (indicador[0]!=-1)
        mProb=(int)umbral[0];
    else
        mProb=-1;
    if (indicador[0] != -1)
        return indicador[0];
    else
        return -1;
}
```

Se muestra la parte en que primeramente la imagen nueva que ingresa a ser identificada se convierte a una imagen de tipo IplImage y recortando del mismo tamaño de las imágenes entrenadas. Para posteriormente enviar la imagen a predecir y también las variables n que es la que almacenara la etiqueta de la persona reconocida y p la probabilidad de reconocimiento.

Una vez que se envía la imagen al método predict, primeramente, verifica si ya se entrenó las imágenes de entrenamiento, es importante entrenar por lo que se comparó los histogramas de las imágenes entrenadas con la nueva imagen, por eso el código implementado primeramente verifica que se haya realizado el entrenamiento y si no lo es retornara error o -1. Después se obtiene la imagen espacial y realiza una comparación con los histogramas entrenados.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

VI. CONCLUSIONES Y RECOMENDACIONES

- **Conclusiones:**

Los métodos de reconocimiento facial seleccionados para esta investigación llamados FisherFaces y LBP fueron elegidos por poseer antecedentes muy buenos en investigaciones pasadas teniendo en cuenta aspectos de seguridad e iluminación, en base a eso se hicieron los experimentos para medir y comparar sus resultados con diferentes iluminaciones y diferentes ambientes de seguridad.

El procesamiento de imágenes facial con el método LBP posee un porcentaje de efectividad de 90%, el cual es elevando en comparación al porcentaje de efectividad del método FisherFaces de 76.7%, es decir que el método LBP es mejor para el reconocimiento facial en ambientes no controlados, todo esto fue demostrado en el experimento de medición de efectividad del indicador 1.

Los métodos FisherFaces y LBP fueron puestos a prueba para medir el tiempo de respuesta promedio en distintos ambientes de día, tarde y noche para el reconocimiento facial, en donde se pudo concluir que: En las mañana el método LBP procesa y reconocer imágenes faciales más rápido que el método FisherFaces, por las tardes el método LBP procesa y reconoce imágenes faciales en menos tiempo de lo que lo hace el método FisherFaces, y por las noches el método LBP arroja un tiempo promedio de procesamiento facial de 879 milisegundos, mientras que por otro lado el método FisherFaces emite un promedio de tiempo de 1045 milisegundos, esto nos quiere decir que el método LBP es más rápido procesando imágenes faciales que el método FisherFaces.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Los métodos FisherFaces y LBP fueron puesto a prueba para clasificar el porcentaje de error que emite a la hora de procesar imágenes reales e imágenes de papel, por lo que se concluye que el método FisherFaces es mucho mejor que el método LBP en aspectos de seguridad, debido a que tuvo menos margen de error al momento de reconocer imágenes de papel, con esto se puede decir que el método FisherFaces es más seguro reconociendo imágenes fáciles en ambientes no controlados.

Al momento de medir los porcentajes de precisión con ambos métodos teniendo en cuenta la iluminación se concluyó que: con la iluminación de la mañana el método de reconocimiento facial LBP es más preciso con un 100% de éxito en todas las pruebas en comparación al método FisherFaces que arrojo un 80% de éxito con todas las pruebas realizadas, por otro lado con iluminación de la tarde el método LBP sigue siendo más preciso al método FisherFaces y de igual forma con la iluminación nocturna el método LBP es mucho más preciso que el método FisherFaces, todo esto se concluyó en base a los resultados obtenido de la investigación.

- **Recomendaciones:**

A continuación, se presentan algunas recomendaciones, con la finalidad de ampliar los objetivos de esta investigación.

1. Establecer la comparación entre los rendimientos de tiempos computacionales en distintos dispositivos con sistema operativo Android, debido a que cada celular no posee la misma rapidez para procesar tareas de esta magnitud.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2. Hacer experimentos con el método seleccionado llamado FisherFaces, para procesar imágenes faciales con distintos ángulos de orientación al momento de hacer las pruebas. De esta manera se medirá con mejor precisión la efectividad del método FisherFaces.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

REFERENCIAS

Akariman, Q., Jati, A. N., & Novianty, A. (2015). Face recognition based on the Android device using LBP algorithm. *ICCEREC 2015 - International Conference on Control, Electronics, Renewable Energy and Communications*, 166–170. https://doi.org/10.1109/ICCEREC.2015.7337037

Bertok, K., & Fazekas, A. (2016). Face recognition on mobile platforms. *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, (CogInfoCom), 000037–000042. https://doi.org/10.1109/CogInfoCom.2016.7804521

Eleyan, A., & Demirel, H. (2007). PCA and LDA based Neural Networks for Human Face Recognition. *Eastern Mediterranean University*.

Gofman, M. I., & Mitra, S. (2016). Multimodal biometrics for enhanced mobile device security. *Communications of the ACM*, *59*(4), 58–65. https://doi.org/10.1145/2818990

Khashman, A. (2007). Intelligent Global Face Recognition. *Near East University*. https://doi.org/10.5772/60142

Lee, Y. H., Kim, C. G., Kim, Y., & Whangbo, T. K. (2013). Facial landmarks detection using improved active shape model on android platform. *Multimedia Tools and Applications*, 1–10. https://doi.org/10.1007/s11042-013-1565-y

Li, S. Z., & Jain, A. K. (2005). *Handbook of Face Recognition*. *Handbook of face recognition*. https://doi.org/10.1017/CBO9781107415324.004

Oka Sudana, A. A. K., Darma Putra, I. K. G., & Arismandika, A. (2014). Face recognition system on android using Eigenface method. *Journal of Theoretical and Applied Information Technology*, *61*(1), 128–134.

Ravibabu, V., & Krishnan, N. (2015). A vary approach to face recognition veritable mechanisms for Android mobile against

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

spoofing. *2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2014*. https://doi.org/10.1109/ICCIC.2014.7238290

Stoimenov, S., Tsenov, G. T., Mladenov, V. M., & Member, S. (2016). Face Recognition system in Android Using Neural Networks. *Symposium on Neural Networks and Applications (NEUREL)*, 13–16.

Smowltech.com. (2017). Smowltech. [online] Available at: http://smowltech.com/es/technology [Accessed 15 May 2017].

Carlosjulioph's Blog. (2017). Análisis de Componentes Principales (PCA). [online] Available at: https://carlosjulioph.wordpress.com/analisis-de-componentes-principales-pca/ [Accessed 15 May 2017].

Docs.opencv.org. (2017). Face Recognition with OpenCV — OpenCV 2.4.13.2 documentations. [online] Available at: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html [Accessed 15 May 2017].

Es.mathworks.com. (2017). Reconocimiento facial - MATLAB & Simulink. [online] Available at: https://es.mathworks.com/discovery/reconocimiento-facial.html [Accessed 15 May 2017]

Mathworks.com. (2017). MATLAB - MathWorks. [online] Available at: https://www.mathworks.com/products/matlab.html [Accessed 15 May 2017].

Developer.android.com. (2017). Cómo descargar Android Studio y SDK Tools | Android Studio. [online] Available at: https://developer.android.com/studio/index.html?hl=es-419 [Accessed 15 May 2017].

Redes-neuronales.com.es. (2017). Redes Neuronales Clasificación respecto al Aprendizaje. [online] Available at: http://www.redes-neuronales.com.es/tutorial-redes-neuronales/clasificacion-de-redes-neuronales-respecto-al-aprendizaje.htm [Accessed 8 Jun. 2017].

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Qt. (2017). The future is written with Qt: Cross-platform software development for embedded & desktop. [Online] Available at: https://www.qt.io/es/ [Accessed 7 Sep. 2017].

C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. Computer Vision, 1998. Sixth International Conference.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

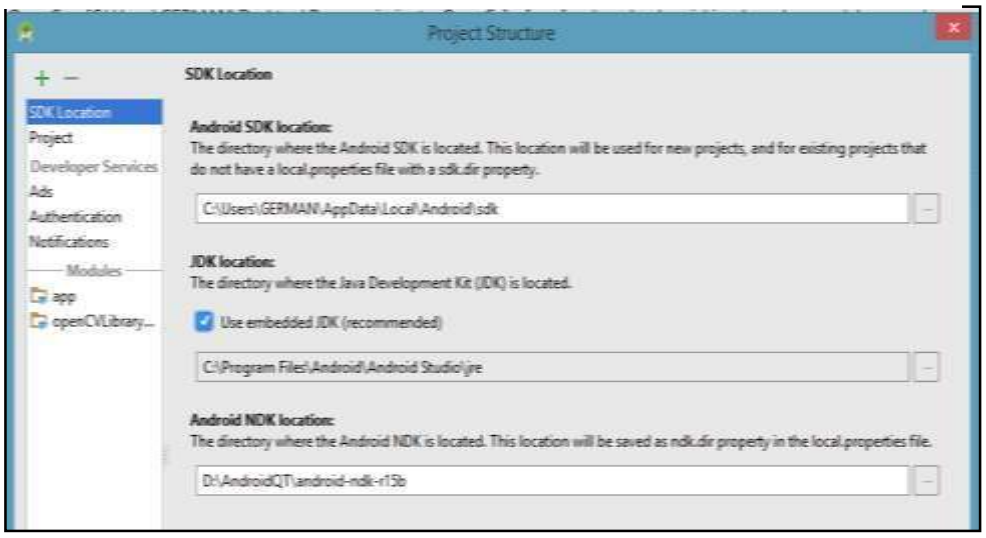
ANEXOS

Anexo 01:

Se descargó los módulos de OpenCv para Android en su sitio web oficinal de Opencv el cual es https://opencv.org/platforms/

El proyecto se trabajó bajo la plataforma Android. Allí se configuraron las rutas del Kit de desarrollo de software (SDK), Kit de Desarrollo de Java (JDK) y el Kit de Desarrollo Nativo (NDK), como se aprecia en la figura 41.

Figura 41: Configuración de SDK, JDK y NDK en Android Studio



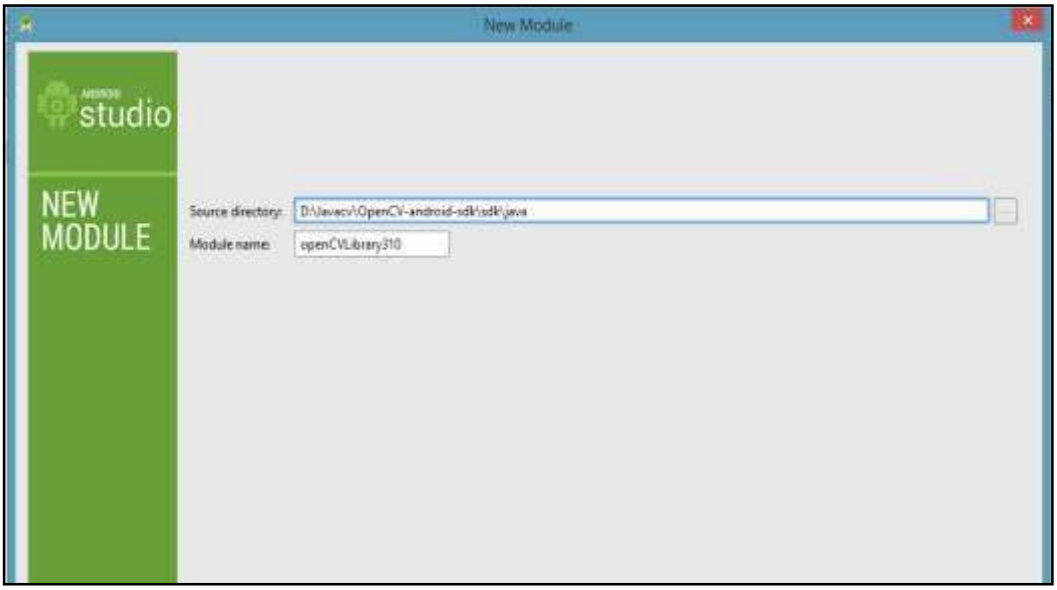
Fuente: Elaboración propia

A continuación, se importó el módulo de OpenCv para Android descargado de su sitio oficial, pero seleccionamos la carpeta JAVA ubicado dentro de la carpeta SDK (Ver figura 2).

Luego se configuró el módulo de OpenCv agregado para que dependa del proyecto donde se trabajó, después de esto se configura el Grandle del módulo para que se compatible con el Grandle del proyecto. La dependencia del módulo de OpenCv se muestra en la figura 3.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

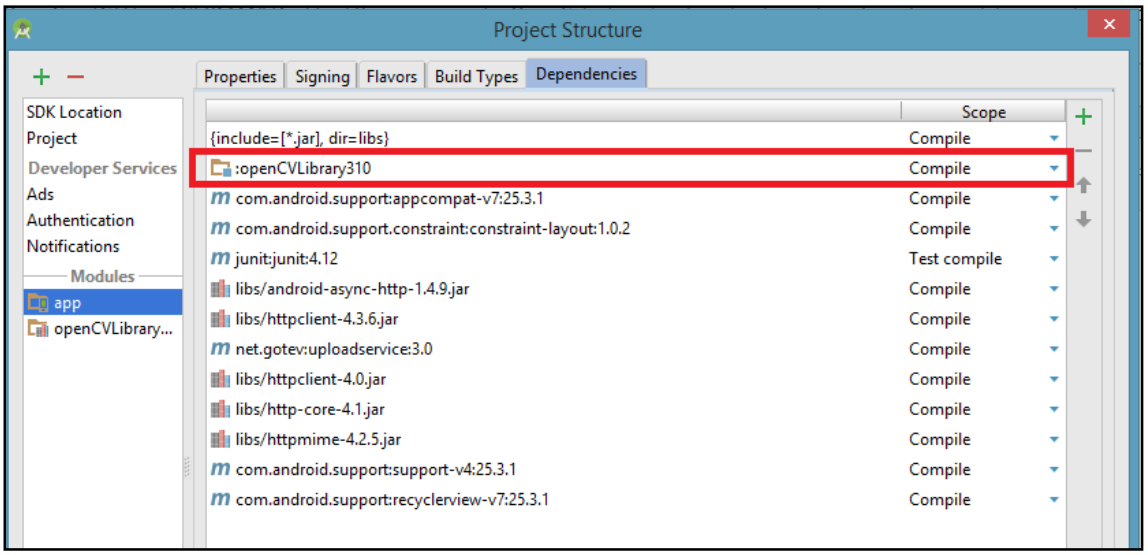
Figura 42: Importación del módulo OpenCv para Android.



Fuente: Elaboración propia

Por último, se copiaron las librerías de la carpeta Native de NDK del módulo OpenCv dentro del proyecto Android para usarlas en el desarrollo y la implementación del reconocimiento facial.

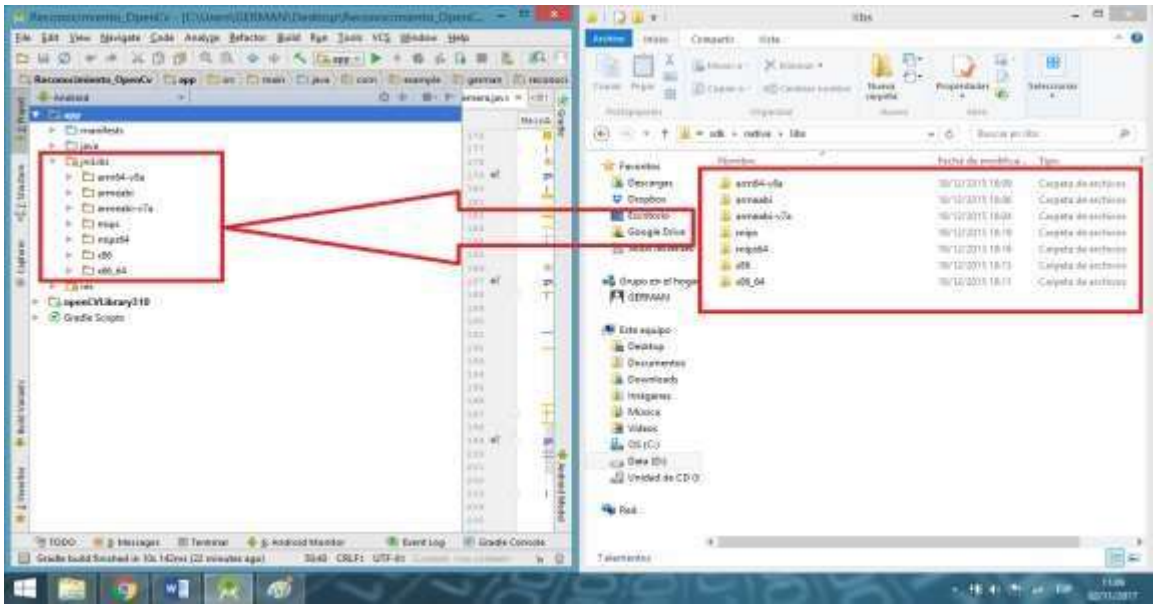
Figura 43: Dependencia del módulo OpenCv en Android Studio.



Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 44: Copiar Librerías NDK de OpenCv para el proyecto Android.



Fuente: Elaboración propia

Anexo 02:

Matriz de Resultado de experimento para el indicador 1:

Porcentaje de Pruebas con Éxito:

A continuación de muestra las matrices de resultados que se obtuvo de todas las pruebas hechos para el experimento 1 del indicador 1 de la presente investigación.

a) Método FisherFaces:

Tabla 8: Matriz de Resultado - Experimento 1 Método FisherFaces

Experimento N° 01: Método FisherFaces				
N°	**Cod_Fotografia**	**Nombre de Persona**	**Acierto**	**No acierto**
1	F001	Ítalo Almanza	X	
2	F002	David Heredia	X	
3	F003	Ricardo Gil	X	
4	F004	Edwin Salamanca	X	X
5	F005	Martin Wilant	X	X
6	F006	Nicolás Siderides	X	

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

7	F007	Danilo Sesejl	X	
8	F008	Gonzalo Montoya		X
9	F009	Paola Lince	X	
10	F010	Camila Gomina	X	
11	F011	Michel Centurión	X	
12	F012	Milagros Fernández	X	
13	F013	Belén Ortellano	X	
14	F014	Aike Lima	X	
15	F015	Marvy Rincón	X	
16	F016	Mariano Valentini	X	
17	F017	Fernanda Gallegos		X
18	F018	Karen Fuquene		X
19	F019	Manuela Dotty	X	
20	F020	Kenia Burgos	X	
21	F021	Vania Ortiz		X
22	F022	Karla Korona	X	
23	F023	Belén Ferrando	X	
24	F024	Agustina Roland	X	
25	F025	Hillary Santander	X	
26	F026	Florencia Villareal	X	
27	F027	Nathaly Diaz	X	
28	F028	Eduardo Clavijo	X	
29	F029	Jesu Colombo		X
30	F030	Elsa Aguirre	X	

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

b) Método LBP:

Tabla 9: Matriz de Resultado - Experimento 1 Método LBP

Experimento N° 01: Método LBP				
N°	**Cod_Fotografia**	**Nombre de Persona**	**Acierto**	**No acierto**
1	F001	Ítalo Almanza	X	
2	F002	David Heredia	X	
3	F003	Ricardo Gil	X	
4	F004	Edwin Salamanca		X
5	F005	Martin Wilant	X	
6	F006	Nicolás Siderides	X	
7	F007	Danilo Sesejl	X	
8	F008	Gonzalo Montoya		X
9	F009	Paola Lince	X	
10	F010	Camila Gomina	X	
11	F011	Michel Centurión	X	
12	F012	Milagros Fernández	X	
13	F013	Belén Ortellano	X	
14	F014	Aike Lima	X	
15	F015	Marvy Rincón	X	
16	F016	Mariano Valentini	X	
17	F017	Fernanda Gallegos	X	
18	F018	Karen Fuquene		X
19	F019	Manuela Dotty	X	
20	F020	Kenia Burgos	X	
21	F021	Vania Ortiz	X	
22	F022	Karla Korona	X	
23	F023	Belén Ferrando	X	
24	F024	Agustina Roland	X	
25	F025	Hillary Santander	X	
26	F026	Florencia Villareal	X	

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

27	F027	Nathaly Diaz	X	
28	F028	Eduardo Clavijo	X	
29	F029	Jesu Colombo	X	
30	F030	Elsa Aguirre	X	

Fuente: Elaboración propia

Anexo 03:

Matriz de Resultado de experimento para el indicador 2:

Tiempo de Respuesta en Milisegundos:

A continuación de muestra las matrices de resultados que se obtuvo de todas las pruebas hechos para el experimento 2 del indicador 2 de la presente investigación.

a) Método FisherFaces:

Tabla 10: Matriz de Resultado - Experimento 2 Método FisherFaces

Experimento N° 01: Método FisherFaces				
N°	**Cod_Fotografia**	**Nombre de Persona**	**Iluminación**	**Tiempo**
1	F001	Ítalo Almanza	Mañana	931
2	F002	David Heredia	Mañana	989
3	F003	Ricardo Gil	Mañana	954
4	F004	Edwin Salamanca	Mañana	957
5	F005	Martin Wilant	Mañana	952
6	F006	Nicolás Siderides	Mañana	963
7	F007	Danilo Sesejl	Mañana	956
8	F008	Gonzalo Montoya	Mañana	933
9	F009	Paola Lince	Mañana	971
10	F010	Camila Gomina	Mañana	954
11	F011	Michel Centurión	Tarde	879
12	F012	Milagros Fernández	Tarde	897
13	F013	Belén Ortellano	Tarde	879
14	F014	Aike Lima	Tarde	876
15	F015	Marvy Rincón	Tarde	859
16	F016	Mariano Valentini	Tarde	878

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

17	F017	Fernanda Gallegos	Tarde	869
18	F018	Karen Fuquene	Tarde	889
19	F019	Manuela Dotty	Tarde	895
20	F020	Kenia Burgos	Tarde	859
21	F021	Vania Ortiz	Noche	1008
22	F022	Karla Korona	Noche	1082
23	F023	Belén Ferrando	Noche	1045
24	F024	Agustina Roland	Noche	1109
25	F025	Hillary Santander	Noche	1017
26	F026	Florencia Villareal	Noche	1015
27	F027	Nathaly Diaz	Noche	1025
28	F028	Eduardo Clavijo	Noche	1081
29	F029	Jesu Colombo	Noche	1045
30	F030	Elsa Aguirre	Noche	1023

Fuente: Elaboración propia

b) Método LBP:

Tabla 11: Matriz de Resultado - Experimento 2 Método LBP

Experimento N° 01: Método FisherFaces				
N°	**Cod_Fotografia**	**Nombre de Persona**	**Iluminación**	**Tiempo**
1	F001	Ítalo Almanza	Mañana	768
2	F002	David Heredia	Mañana	757
3	F003	Ricardo Gil	Mañana	758
4	F004	Edwin Salamanca	Mañana	742
5	F005	Martin Wilant	Mañana	658
6	F006	Nicolás Siderides	Mañana	733
7	F007	Danilo Sesejl	Mañana	738
8	F008	Gonzalo Montoya	Mañana	742
9	F009	Paola Lince	Mañana	716
10	F010	Camila Gomina	Mañana	728
11	F011	Michel Centurión	Tarde	783
12	F012	Milagros Fernández	Tarde	755
13	F013	Belén Ortellano	Tarde	781
14	F014	Aike Lima	Tarde	752
15	F015	Marvy Rincón	Tarde	747
16	F016	Mariano Valentini	Tarde	742

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

17	F017	Fernanda Gallegos	Tarde	774
18	F018	Karen Fuquene	Tarde	724
19	F019	Manuela Dotty	Tarde	719
20	F020	Kenia Burgos	Tarde	773
21	F021	Vania Ortiz	Noche	866
22	F022	Karla Korona	Noche	879
23	F023	Belén Ferrando	Noche	875
24	F024	Agustina Roland	Noche	882
25	F025	Hillary Santander	Noche	905
26	F026	Florencia Villareal	Noche	848
27	F027	Nathaly Diaz	Noche	898
28	F028	Eduardo Clavijo	Noche	878
29	F029	Jesu Colombo	Noche	863
30	F030	Elsa Aguirre	Noche	896

Fuente: Elaboración propia

Anexo 04:

Matriz de Resultado de experimento para el indicador 3: Porcentaje de Pruebas con errores:

A continuación de muestra la matriz de confusión de resultados que se obtuvo de todas las pruebas hechos para el experimento 3 del indicador 3 de la presente investigación.

a) Método FisherFaces:

Tabla 12: Matriz de Confusión para el método FisherFaces

Matriz de Confusión		**RESPUESTA DEL SISTEMA**	
		POSITIVO	**NEGATIVO**
LO QUE SE MUESTRÓ AL SISTEMA	**POSITIVO**	12	5
	NEGATIVO	6	7

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

b) Método LBP:

Tabla 13: Matriz de Confusión para el método LBP

Matriz de Confusión		RESPUESTA DEL SISTEMA	
		POSITIVO	NEGATIVO
LO QUE SE MUESTRÓ AL SISTEMA	POSITIVO	10	14
	NEGATIVO	2	4

Fuente: Elaboración propia

Anexo 05:

Matriz de Resultado de experimento para el indicador 4:

Porcentaje de Precisión de las pruebas con éxito:

A continuación de muestra la matriz de resultados que se obtuvo de todas las pruebas hechos para el experimento 4 del indicador 4 de la presente investigación.

a) Método FisherFaces:

Tabla 14: Matriz de Resultado - Experimento 4 Método FisherFaces

Experimento N° 01: Método LBP					
N°	Cod_Fotografia	Nombre de Persona	Iluminación	Acierto	No acierto
1	F001	Ítalo Almanza	Mañana	X	
2	F002	David Heredia	Mañana	X	
3	F003	Ricardo Gil	Mañana	X	
4	F004	Edwin Salamanca	Mañana	X	
5	F005	Martin Wilant	Mañana	X	
6	F006	Nicolás Siderides	Mañana	X	
7	F007	Danilo Sesejl	Mañana	X	
8	F008	Gonzalo Montoya	Mañana		X
9	F009	Paola Lince	Mañana		X

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

10	F010	Camila Gomina	Mañana	X	
11	F011	Michel Centurión	Tarde	X	
12	F012	Milagros Fernández	Tarde		X
13	F013	Belén Ortellano	Tarde	X	
14	F014	Aike Lima	Tarde		X
15	F015	Marvy Rincón	Tarde	X	
16	F016	Mariano Valentini	Tarde	X	
17	F017	Fernanda Gallegos	Tarde	X	
18	F018	Karen Fuquene	Tarde		X
19	F019	Manuela Dotty	Tarde	X	
20	F020	Kenia Burgos	Tarde	X	
21	F021	Vania Ortiz	Noche		X
22	F022	Karla Korona	Noche		X
23	F023	Belén Ferrando	Noche	X	
24	F024	Agustina Roland	Noche	X	
25	F025	Hillary Santander	Noche	X	
26	F026	Florencia Villareal	Noche		X
27	F027	Nathaly Diaz	Noche		X
28	F028	Eduardo Clavijo	Noche	X	
29	F029	Jesu Colombo	Noche	X	
30	F030	Elsa Aguirre	Noche		X

Fuente: Elaboración propia

b) Método LBP:

Tabla 15: Matriz de Resultado - Experimento 4 Método LBP

Experimento N° 01: Método LBP					
N°	**Cod_Fotografia**	**Nombre de Persona**	**Iluminación**	**Acierto**	**No acierto**
1	F001	Ítalo Almanza	Mañana	X	
2	F002	David Heredia	Mañana	X	
3	F003	Ricardo Gil	Mañana	X	

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

4	F004	Edwin Salamanca	Mañana	X	
5	F005	Martin Wilant	Mañana	X	
6	F006	Nicolás Siderides	Mañana	X	
7	F007	Danilo Sesejl	Mañana	X	
8	F008	Gonzalo Montoya	Mañana	X	
9	F009	Paola Lince	Mañana	X	
10	F010	Camila Gomina	Mañana	X	
11	F011	Michel Centurión	Tarde	X	
12	F012	Milagros Fernández	Tarde	X	
13	F013	Belén Ortellano	Tarde	X	
14	F014	Aike Lima	Tarde	X	
15	F015	Marvy Rincón	Tarde		X
16	F016	Mariano Valentini	Tarde	X	
17	F017	Fernanda Gallegos	Tarde	X	
18	F018	Karen Fuquene	Tarde	X	
19	F019	Manuela Dotty	Tarde	X	
20	F020	Kenia Burgos	Tarde	X	
21	F021	Vania Ortiz	Noche	X	
22	F022	Karla Korona	Noche	X	
23	F023	Belén Ferrando	Noche	X	
24	F024	Agustina Roland	Noche	X	
25	F025	Hillary Santander	Noche	X	
26	F026	Florencia Villareal	Noche	X	
27	F027	Nathaly Diaz	Noche	X	
28	F028	Eduardo Clavijo	Noche	X	
29	F029	Jesu Colombo	Noche		X
30	F030	Elsa Aguirre	Noche	X	

Fuente: Elaboración propia

Anexo 06:

Aplicación generada para la implementación de métodos y emisión de resultados:

La aplicación cuenta con un menú principal que muestra diferentes opciones.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

La primera opción ayuda a gestionar y crear una base de datos facial propia, que consta de almacenar 15 fotografías por persona.

La segunda opción muestra la base de datos que ya se ha almacenado dentro de la aplicación.

La tercera opción sirve para entrenar toda la base de datos con el método de reconocimiento facial que escojamos.

La cuarta opción sirve para poner a prueba el reconocimiento facial.

La pantalla con el menú principal de la aplicación se muestra en la figura 45.

Figura 45: Interfaz principal de la aplicación.

Fuente: Elaboración propia

En la figura 46 se muestra la interfaz de entrenamiento de la base de datos, cabe resaltar que el usuario es el que elije con que método de reconocimiento facial quiere que sea entrenada su base de datos, de esta forma se podrá comparar los resultado de uno y otro método.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 46: Interfaz de entrenamiento de la base de datos.

Fuente: Elaboración propia

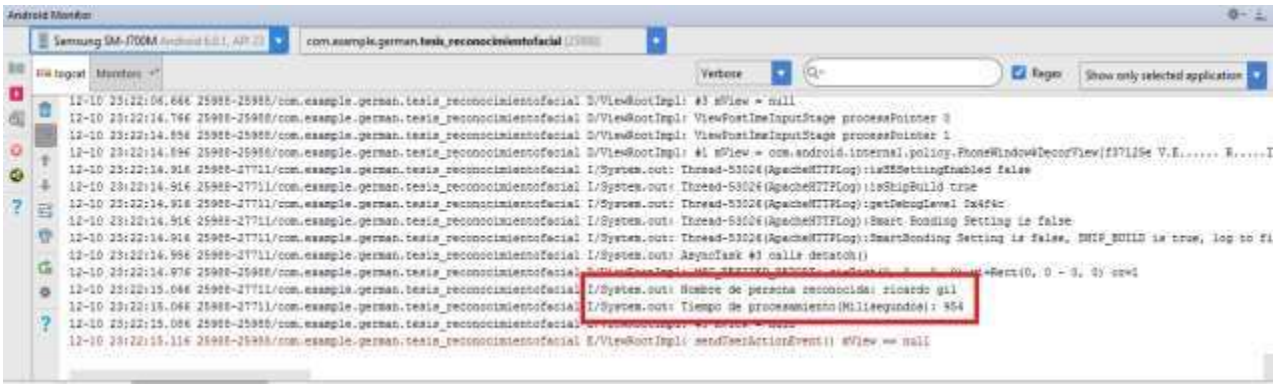
En la figura 47 se muestra la interfaz de reconocimiento facial de la aplicación, en esta interfaz se capturaron los resultado de los experimentos.

Figura 47: Interfaz de reconocimiento de la aplicación móvil.

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Telef: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 48: Muestra de resultado y cálculo de tiempo computacional del reconocimiento facial



Fuente: Elaboración propia

Figura 49: Base de datos facial

Fuente: Elaboración propia

Anexo 07:

Código fuente del proyecto:

A continuación se muestra la clase de Reconocimiento facial la cual se encarga de enviar las imágenes de entrenamiento y recibir los datos de predicción.

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 50: Clase principal del proyecto de reconocimiento facial

```
public final static int MAXIMG = 100;
FaceRecognizer faceRecognizer; String
mPath;
int count=0;
RegistroPersonas registroPersonas;
static final int WIDTH= 128;
static final int HEIGHT= 128;
private int mProb=999;

ReconocimientoFacial(String path) {
        faceRecognizer = createFisherFaceRecognizer (0,Double.MA X_VALUE);
        mPath=path;
        registroPersonas = new RegistroPersonas(mPath);
}
public void add(Mat m, String description) {
        Bitmap bmp= Bitmap.createBitmap(m.width(),
        m.height(), Bitmap.Config.ARGB_8888);
        Utils.matToBitmap(m,bmp);
        bmp= Bitmap.createScaledBitmap(bmp, WIDTH, HEIGHT, false);
        FileOutputStream f;
        try {
                String nombres[] = description.toString().split(" ");
                StringBuilder s = new StringBuilder();
                for (String nom : nombres) {
                        s.append(nom); s.append("_");
                }
                f = new FileOutputStream(mPath + s + "" + count + ".jpg",true);
                count++;
                bmp.compress(Bitmap.CompressFormat.JPEG, 100, f);
                f.close();
        } catch (Exception e) {
                Log.e("error",e.getCause()+" "+e.getMessage());
                e.printStackTrace();
        }
}
public boolean train() {
        File root = new File(mPath);
        FilenameFilter pngFilter = new FilenameFilter() {
                public boolean accept(File dir, String name) {
                        return name.toLowerCase().endsWith(".jpg");
                };
        };
        File[] imageFiles = root.listFiles(pngFilter);
        MatVector images = new MatVector(imageFiles.length);
        int[] labels = new int[imageFiles.length];
        int counter = 0;
        int label;
        IplImage img=null;
        IplImage grayImg;
        int i1=mPath.length();
        for (File image : imageFiles) {
                String p = image.getAbsolutePath();
                img = cvLoadImage(p);
```


Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
                    if (img==null)
                            Log.e("Error","Error al cargar imagen");
                    Log.i("image",p);
                    int i2=p.lastIndexOf("_");
                    int i3=p.lastIndexOf(".");
                    int icount=Integer.parseInt(p.substring(i2+1,i3));
                    if (count<icount) count++;
                     String description=p.substring(i1,i2);
                     if (registroPersonas.get(description)<0)
                     registroPersonas.add(description, registroPersonas.max()+1);
                     label = registroPersonas.get(description);
                     grayImg = IplImage.create(img.width(), img.height(), IPL_DEPTH_8U,
                    1);
                     cvCvtColor(img, grayImg, CV_BGR2GRAY);
                     images.put(counter, grayImg);
                     labels[counter] = label;
                     counter++;
            }
            if (counter>0)
                    if (registroPersonas.max()>1) {
                            faceRecognizer.train(images, labels);
                    }
            registroPersonas.Save();
            return true;
}
public boolean canPredict() {
            if (registroPersonas.max()>1)
                    return true;
            else
                    return false;
}
public String predict(Mat m) {
            if (!canPredict())
                    return "";
            int n[] = new int[1];
            double p[] = new double[1];
            IplImage ipl = MatToIplImage(m,WIDTH, HEIGHT);

            faceRecognizer.predict(ipl, n, p);

            if (n[0]!=-1)
                    mProb=(int)p[0];
            else
                    mProb=-1;
            if (n[0] != -1)
                    return registroPersonas.get(n[0]);
            else
                    return "Cara desconocida";
}
IplImage MatToIplImage(Mat m,int width,int heigth) {
            Bitmap bmp=Bitmap.createBitmap(m.width(), m.height(),
            Bitmap.Config.ARGB_8888);
            Utils.matToBitmap(m, bmp);
            return BitmapToIplImage(bmp,width, heigth);
}
```


Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
IplImage BitmapToIplImage(Bitmap bmp, int width, int height) {
        if ((width != -1) || (height != -1)) {
                Bitmap bmp2 = Bitmap.createScaledBitmap(bmp, width, height, false);
                bmp = bmp2;
        }
        IplImage image = IplImage.create(bmp.getWidth(), bmp.getHeight(),
        IPL_DEPTH_8U, 4);
        bmp.copyPixelsToBuffer(image.getByteBuffer());
        IplImage grayImg = IplImage.create(image.width(), image.height(),
        IPL_DEPTH_8U, 1);
        CvtColor(image, grayImg, opencv_imgproc.CV_BGR2GRAY);
        return grayImg;
}
```

Fuente: Elaboración propia

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tesis publicada con autorización del autor
Algunos Derechos Reservados. No olvide citar esta tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe