



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
INGENIERÍA EN SISTEMAS DE INFORMACIÓN
PROYECTO DE INVESTIGACIÓN

**TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL
UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.**

Proyecto de Investigación presentado previo a la obtención del Título de Ingenieros en
Sistemas de Información.

Autores:

Larco Borja Pablo Damián

Chacon Gavilanes Luis Fernando

Tutor Académico

Ing. Mg. Cadena Moreano José Augusto

Latacunga – Ecuador

Marzo 2022

DECLARACIÓN DE LA AUTORÍA

Nosotros, **Luis Fernando Chacon Gavilanes** con C.I.: 0504151440 y **Pablo Damián Larco Borja** con C.I.: 1721851838, ser los autores del presente proyecto de Investigación: "TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.", siendo el Ing. Mg. José Augusto Cadena Moreano, tutor del presente trabajo, eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Atentamente,



.....

Chacon Gavilanes Luis Fernando

C.I.: 0504151440



.....

Larco Borja Pablo Damián

C.I.: 1721851838

AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación con el título:” **TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.**”, de los estudiantes, **Luis Fernando Chacon Gavilanes** con C.I.: **050415144-0** y **Pablo Damián Larco Borja** con C.I.: **172185183-8**, de la carrera de Ingeniería en Sistemas de Información ,considero que dicho proyecto investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Febrero del 2022



TUTOR ACADÉMICO

Ing. Mg. Cadena Moreano José Augusto

C.I.:0501552798

AVAL DE JUICIO DE EXPERTO

Yo, **Mayra Albán** con cedula de identidad N° **050231198-8**, en calidad de Ingeniera de la carrera de Sistemas de Información, **CERTIFICO** que el prototipo para el reconocimiento facial global es funcional, de acuerdo al tema de proyecto de investigación:” **TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.**”, realizado por los estudiantes, **Luis Fernando Chacon Gavilanes** con C.I.: **050415144-0** y **Pablo Damián Larco Borja** con C.I.: **172185183-8**, de la carrera de Ingeniería en Sistemas de Información de la Universidad Técnica de Cotopaxi, siendo el **Ing. Mg. Cadena Moreano José Augusto** tutor del presente trabajo.

Es todo a cuanto puedo certificar en honor a la verdad y autorizo hacer uso del presente certificado de la manera más ética que estimaren conveniente.

Latacunga, Marzo del 2022

Atentamente,



Ing. Mg. Mayra Albán

C.I.: 050231198-8

AVAL DE JUICIO DE EXPERTO

Yo, **Manuel Villa** con cedula de identidad N° **180338695-0**, en calidad de Ingeniero de la carrera de Sistemas de Información, **CERTIFICO** que el prototipo para el reconocimiento facial global es funcional, de acuerdo al tema de proyecto de investigación:” **TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.**”, realizado por los estudiantes, **Luis Fernando Chacon Gavilanes** con C.I.: **050415144-0** y **Pablo Damián Larco Borja** con C.I.: **172185183-8**, de la carrera de Ingeniería en Sistemas de Información de la Universidad Técnica de Cotopaxi, siendo el **Ing. Mg. Cadena Moreano José Augusto** tutor del presente trabajo.

Es todo a cuanto puedo certificar en honor a la verdad y autorizo hacer uso del presente certificado de la manera más ética que estimaren conveniente.

Latacunga, Marzo del 2022

Atentamente,



.....
Ing. Mg Manuel Villa
C.I.: 180338695-0

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de **CIENCIAS DE LA INGENIERÍA Y APLICADAS**; por cuanto, los postulantes : **LUIS FERNANDO CHACON GAVILANES Y PABLO DAMIÁN LARCO BORJA** , con el título de proyecto de investigación: :” **TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.**”, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación del proyecto.

Por los antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Marzo del 2022


Para constancia firman:



.....
Lector 1 (Presidente)
MG. CANTUÑA KARLA
C.I.: 050230511-3



.....
Lector 2
MG. QUISAGUANO RENÉ
C.I.: 172189518-1



.....
Lector 3
DR. CHANCUSIG JUAN CARLOS
C.I.: 050227577-9

AGRADECIMIENTO

Quiero agradecer primeramente a mis padres: Zoila Gavilanes y Milton Chacon por haberme apoyado en mis estudios, por los consejos que me ayudaron a no rendirme y principios que me han inculcado.

También quiero agradecer a mi tutor .Ing. Mg. José Cadena quien me guio en la realización de esta investigación con su paciencia y rectitud como docente.

Luis Chacon

DEDICATORIA

El presente trabajo de investigación le quiero dedicar a mis padres por su amor y sacrificios que me han dado todos estos años y que gracias a ellos he podido lograr llegar hasta este momento. Ha sido de mucho orgullo y privilegio ser su hijo.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito y en especial a aquellos que nos abrieron las puertas y compartieron su conocimiento.

Luis Chacon

AGRADECIMIENTO

Quiero agradecer primeramente a mis padres: Damián Larco y Patricia Borja por financiar mis estudios, por las charlas que me ayudaron a no rendirme, los valores y su amor incondicional.

A mi padrino John Larco por su apoyo incondicional en todo mi proceso universitario.

También quiero agradecer a mi tutor. Ing. Mg. José Cadena quien me guio en la realización de esta investigación con su paciencia y rectitud como docente.

Damián Larco

DEDICATORIA

El presente trabajo de investigación le quiero dedicar a mis padres por su amor y sacrificios que me han dado todos estos años y que gracias a ellos he podido lograr llegar hasta este momento. Ha sido de mucho orgullo y privilegio ser su hijo.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito y en especial a aquellos que nos abrieron las puertas y compartieron su conocimiento.

Damián Larco

ÍNDICE GENERAL

AGRADECIMIENTO	vi
DEDICATORIA	vii
AGRADECIMIENTO	viii
DEDICATORIA	ix
ÍNDICE GENERAL	x
ÍNDICE DE TABLAS	xiv
ÍNDICE DE FIGURAS	xv
RESUMEN	1
ABSTRACT.....	2
AVAL DE TRADUCCIÓN.....	3
1. INFORMACIÓN GENERAL.....	4
1. INTRODUCCIÓN	6
2.1. PROBLEMA	6
2.2.1. Situación Problemática	6
2.1.2. Formulación del problema.....	7
2.2. OBJETO Y CAMPO DE ACCIÓN	7
2.2.1. Objeto de estudio	7
2.2.2. Campo de acción.....	7
2.3. BENEFICIARIOS	7
2.4. JUSTIFICACIÓN.....	8
2.5. HIPÓTESIS.....	8
2.6. OBJETIVOS.....	9
2.6.1. General	9
2.6.2. Específicos	9
2.7. SISTEMAS DE TAREAS	10

3.	FUNDAMENTACIÓN TEÓRICA	11
3.1.	Antecedentes.....	11
3.2.	Machine Learning.....	12
3.3.	Visión por computador.....	13
3.4.	Reconocimiento de rostros	13
3.5.	Reconocimiento facial.....	14
3.5.1.	Aplicaciones.....	14
3.5.2.	Métodos basados en la imagen	15
3.5.3.	Sistemas de detección facial	15
3.5.4.	Estudio de las texturas de los canales de color.....	15
3.6.	Técnicas de reconocimiento facial	15
3.6.1.	EigenFaces.....	15
3.6.2.	LBPH	16
3.6.3.	Fisherfaces	17
3.7.	Python	20
3.8.	OpenCV	20
3.9.	Base de datos imágenes.....	21
3.9.1.	Base de datos REPLAY-ATTACK.....	21
3.9.2.	Orientado a objetos.	21
3.9.3.	Base de Datos de Imágenes de Rango del Signal Analysis and Machine Perception Laboratory	21
3.9.4.	Base de datos de USF (University of South Florida)	21
4.	MATERIALES Y MÉTODOS	22
4.1.	Enfoque de la investigación.....	22
4.2.	Tipos de investigación.....	22
4.2.1.	Investigación bibliográfica.....	22
4.2.2.	Investigación de campo	22

4.2.3.	Investigación descriptiva.....	22
4.3.	Métodos teóricos	22
4.3.2.	Método hipotético	22
4.4.	Métodos Empíricos	23
4.4.1.	Técnicas de investigación	23
4.4.2.	Instrumentos de la investigación	23
4.4.2.2.	Población.....	23
4.4.2.3.	Muestra	23
4.5.	Variables	24
4.5.1.	Variable independiente	24
4.5.2.	Variable dependiente.....	24
4.6.	Metodología de la investigación.....	24
4.6.2.	Unidad de análisis.....	25
4.6.3.	Técnicas de análisis de datos	25
4.7.	Detención de rostros	26
4.7.1.	Detección de un rostro en una imagen.....	26
4.7.2.	Haar Cascades Python – OpenCV	27
4.7.3.	DetectMultiScale	27
4.7.4.	ScaleFactor	27
4.7.5.	MinNeighbors	28
4.7.6.	Detección de rostros sobre una imagen	29
4.8.	Almacenando rostros	30
4.8.1.	Almacenar rostros presentes en una imagen	30
4.8.2.	Almacenar rostros presentes en un video.....	32
4.9.	Reconocimiento Facial.....	34
4.9.1.	Creación de la base de datos.....	34
4.9.2.	Preparando datos para entrenar.....	36

4.9.3. Entrenamiento	37
4.9.4. Guardar el modelo obtenido.....	38
5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	39
5.1. Funcionamiento del prototipo.....	39
5.1.1 Tiempo de respuesta del entrenamiento de las técnicas.....	39
5.1.2. Pruebas de las técnicas de reconocimiento facial.	39
5.2. Comparación de las técnicas de reconocimiento facial	41
5.2.1. Eficiencia de las técnicas	41
5.2.1.1. Eigenfaces	41
5.2.2. Tiempo del reconocimiento de las técnicas.....	43
5.3. Discusión	44
5.4. Validación de la hipótesis	44
5.5. Impacto económico	44
5.5.1. Gastos directos	44
5.6. Cuestionario del prototipo de reconocimiento facial.....	45
6. CONCLUSIONES Y RECOMENDACIONES	47
6.1. Conclusiones	47
6.2. Recomendaciones	47
7. BIBLIOGRAFÍA	48
8. ANEXOS.....	52

ÍNDICE DE TABLAS

Tabla 1: Beneficiario directos e indirectos.....	7
Tabla 2: Planificación de actividades.....	10
Tabla 3: Características Haar.....	13
Tabla 4: Tiempo de entrenamiento.....	39
Tabla 5: Tiempo y acierto de las técnicas.....	43
Tabla 6: Media, mediana, moda del tiempo del reconocimiento de las técnicas.....	43
Tabla 7: Cálculo del costo de desarrollo del prototipo.....	45
Tabla 8: Significado de los índices de juicio de expertos.....	45
Tabla 9: Cuestionario para la evaluación de juicio de expertos.....	45
Tabla 10: Cuestionario para la evaluación de juicio de expertos.....	46
Tabla 12: Ficha bibliográfica 1.....	54
Tabla 13: Ficha bibliográfica 2.....	54
Tabla 14: Ficha bibliográfica 3.....	54
Tabla 15: Ficha bibliográfica 4.....	55
Tabla 16: Total aciertos por técnica.....	55
Tabla 17: Firmas de los estudiantes.....	57

ÍNDICE DE FIGURAS

Figura 1: Imagen publicada por Pearson's Magazine [4].....	11
Figura 2: Uso del reconocimiento facial[17].....	15
Figura 3: Histograma por cada región de la imagen [22].....	18
Figura 4: Emociones y gestos.....	25
Figura 5: Proceso para crear un clasificador de rostros [34].....	26
Figura 6: Proceso para crear un clasificador de rostros empleando Haar-Cascade [34].	27
Figura 7: Ejemplo de pirámide de imágenes [35].....	28
Figura 8: Detección de rostro.....	28
Figura 9: Imagen de entrada para detención de rostros.....	29
Figura 10: Código para detectar el rostro.....	29
Figura 11: Imagen con detención de rostros.....	30
Figura 12: Imagen de entrada para almacenar rostros.....	31
Figura 13: Código para almacenar rostros de una imagen.....	31
Figura 14: Almacenando rostros de una imagen.....	32
Figura 15: Código para almacenar rostros de un video.....	33
Figura 16. Almacenando rostros.....	33
Figura 17. Rostros Almacenados.....	33
Figura 18: Código para almacenar rostros en la base de datos.....	34
Figura 19: Código para establecer el y número de imágenes.....	35
Figura 20. Base de datos creada a partir del código establecido.....	35
Figura 21: Código para preparar las imágenes para el entrenamiento.	36
Figura 22: Métodos para entrenar el conocedor.....	37
Figura 23: Entrenando el reconocedor de rostros.....	37
Figura 24: Guardar modelo obtenido.....	38
Figura 25: Entrenando el reconocedor de rostros.....	38

Figura 26: Guardando el modelo obtenido.....	38
Figura 27: Código reconocimiento EIGENFACES.....	39
Figura 28: Sujeto de prueba EIGENFACES.	40
Figura 29: Código reconocimiento FISHERFACES.....	40
Figura 30: Sujeto de prueba FISHERFACES.....	40
Figura 31: Código reconocimiento LBPH.	41
Figura 32: Sujeto de prueba LBPH.....	41
Figura 33: Porcentaje de acierto de la técnica EIGENFACES.....	41
Figura 34: Porcentaje de acierto de la técnica FISHERFACES.....	42
Figura 35: Porcentaje de acierto de la técnica LBPH.....	42
Figura 36: Recolección de fotografías.....	55

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.”

Autores:

Chacon Gavilanes Luis Fernando

Larco Borja Pablo Damián

RESUMEN

El presente trabajo de grado, es una investigación que trata sobre el uso de técnicas eficientes para el reconocimiento facial global utilizando Python y OpenCV realizada en la carrera de Sistemas de Información de la facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi. Recopilamos información a través de una visita a los estudiantes de la carrera de Sistemas de Información que nos sirven para el análisis y comparación de las tres técnicas utilizadas (Fisherfaces, EigenFaces, LBPH). El trabajo tiene como objetivo determinar una técnica eficiente que contribuya al área del reconocimiento facial global aportando significativamente al campo de la seguridad universitaria, teniendo a consideración el ahorro de recursos para una futura implementación. Terminando este trabajo con el respectivo análisis e interpretación de los resultados de la investigación, se ha determinado que, de las tres técnicas estudiadas, LBPH tienes los mejores resultados tanto en tiempo de entrenamiento como en la eficacia de reconocimiento de rostros alcanzando un 100% en nuestras pruebas.

Palabras Claves: Reconocimiento facial, OpenCV, Python, Fisherfaces, Fisherfaces, LBPH.

TECHNICAL UNIVERSITY OF COTOPAXI

FACULTY OF ENGINEERING SCIENCES AND APPLIED

THEME: “EFFICIENT TECHNIQUE FOR GLOBAL FACIAL RECOGNITION USING PYTHON AND OPENCV ON 2D IMAGES”

AUTHORS:

Chacon Gavilanes Luis Fernando

Larco Borja Pablo Damián

ABSTRACT

The present work of degree, is a research that deals with the use of efficient techniques for global face recognition using Python and OpenCV carried out in the career of Information Systems of the Faculty of Engineering and Applied Sciences of the Technical University of Cotopaxi. We collected information through a visit to the students of the Information Systems career that serve us for the analysis and comparison of the three techniques used (Fisherfaces, EigenFaces, LBPH). The work aims to determine an efficient technique that contributes to the area of global facial recognition, contributing significantly to the field of university security, taking into consideration the saving of resources for future implementation. Finishing this work with the respective analysis and interpretation of the research results, it has been determined that, of the three techniques studied, LBPH has the best results both in training time and face recognition efficiency reaching 100% in our tests.

Key words: Face recognition, OpenCV, Python, Fisherfaces, Fisherfaces, LBPH.



AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal CERTIFICO que:

La traducción del resumen al idioma Inglés del trabajo de titulación cuyo título versa: **“TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D”**, presentado por: **Chacon Gavilanes Luis Fernando** y **Larco Borja Pablo Damián**, estudiantes de la Carrera de **Ingeniería en Sistemas de Información** perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas** lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente aval para los fines académicos legales.

Latacunga, 23 marzo del 2022

Atentamente,



CENTRO
DE IDIOMAS

Mg. Marco Paúl Beltrán Semblantes

DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0502666514

1. INFORMACIÓN GENERAL

TÍTULO: Técnica eficiente para el reconocimiento facial global utilizando Python y OpenCV en imágenes 2d.

FECHA DE INICIO: Octubre del 2021

FECHA DE FINALIZACIÓN: Marzo del 2022

LUGAR DE EJECUCIÓN:

- **Provincia:** Cotopaxi
- **Cantón:** Latacunga
- **Parroquia:** Eloy Alfaro
- **Instituto:** Universidad Técnica de Cotopaxi

FACULTAD QUE AUSPICIA:

Ciencias de la Ingeniería y Aplicadas

CARRERA QUE AUSPICIA:

Ingeniería en sistemas de información

PROYECTO DE INVESTIGACIÓN VINCULADO:

Proyecto Generativo

EQUIPO DE TRABAJO:

DOCENTE TUTOR

Nombre: Ing. Mg. Cadena Moreano José Augusto

Nacionalidad: Ecuatoriano

Estado Civil: Casado

Residencia: Latacunga – Parroquia San Buenaventura

E-mail: jose.cadena@utc.edu.ec

Teléfono: 0984059929

ESTUDIANTES

Primer Estudiante

Nombre: Chacon Gavilanes Luis Fernando

Nacionalidad: Ecuatoriano

Estado Civil: Soltero

Residencia: Cotopaxi-Latacunga-San Martín

E-mail: luis.chacon1440@utc.edu.ec

Teléfono: 0998333687

Segundo Estudiante

Nombre: Larco Borja Pablo Damián

Nacionalidad: Ecuatoriano

Estado Civil: Soltero

Residencia: Pichincha-Sangolquí-Fajardo

E-mail: pablo.larco1838@utc.edu.ec

Teléfono: 0987423321

ÁREA DE CONOCIMIENTO:

06 Información y Comunicación (TIC) /061 Información y Comunicación (TIC) / 0611 El uso del ordenador

LÍNEA DE INVESTIGACIÓN:

Tecnologías de la información y comunicación (TICS).

SUBLÍNEAS DE INVESTIGACIÓN DE LA CARRERA:

Inteligencia artificial e inteligencia de negocios.

1. INTRODUCCIÓN

El presente proyecto de investigación describe las técnicas de reconocimiento facial utilizando Python y OpenCv, aplica algoritmos de inteligencia artificial en imágenes 2D.

Los sistemas de reconocimiento de rostros son un problema que aún es tema de investigación, debido a que gran parte de la problemática, se debe a los factores que pueden afectar el sistema al momento de efectuar el reconocimiento.

Según Franco, Ospina, Cuevas y Capacho [1], existen factores que dificultan el proceso de reconocimiento facial, como son las oclusiones que evita un buen reconocimiento, ya sean por gestos que realice la persona, elementos que cubra el rostro, iluminación, distancia, cicatrices, envejecimiento entre otros factores.

Se instaló el sistema operativo Windows, el lenguaje de programación Python, las librerías OpenCV, os y numpy, el IDE PyCharm en cual se desarrolló el algoritmo del reconocimiento de rostros empleando la técnica que mejor resultado nos proporcione para determinar una técnica eficiente.

Para Franco, Ospina, Cuevas y Capacho [1], para comprobar la capacidad en procesamiento de imágenes para el reconocimiento facial utilizó tres técnicas que son: Eigenfaces, LBPH (Local Binary Patterns Histograms) y Fisherfaces.

Por último, para comprobar si la técnica aplicada cumple con las expectativas previstas se realizará diferentes pruebas de campo con la detección de rostros y la comparación de resultados en las tres técnicas descritas.

2.1. PROBLEMA

2.2.1. Situación Problemática

Hoy en día, el uso del reconocimiento facial ha sido uno de los problemas más comunes mundialmente debido a la disponibilidad del software, el costo y hardware que se necesitan para su implementación en una organización pública o privada. Las potencias mundiales como Estados Unidos, Rusia, China, Japón, Alemania, Corea del Norte destinan grandes cantidades de recursos económicos para la implementación de herramientas en el reconocimiento facial en las empresas de estos países para que tengan una mayor seguridad en la autenticación de los trabajadores.

Según Astudillo y Mora [2] ,debido a la actual situación de violencia y delincuencia que atraviesa el Ecuador, es necesario proponer soluciones que permitan reducir los índices de inseguridad, se implementó una plataforma de reconocimiento facial, diseñado para ser expuesta al público mediante servicios web, la lógica desplegada se estructuró para satisfacer los requerimientos en el contexto del país, buscando la integración con los entes encargados de la seguridad ciudadana,al tener el control del desarrollo de la plataforma se combinaron diferentes técnicas y cambios a los algoritmos de detección de rostros y reconocimiento facial.

Existen varias técnicas que se utilizan para el reconocimiento facial, pero estas tienen falencias al momento de implementar en algunas empresas como el tiempo de respuesta y análisis de patrones de imágenes en 2D generando pérdidas tiempo al momento de una autenticación de un usuario y así perjudicando en la detención de los rostros.

2.1.2. Formulación del problema

¿Cuál es la técnica más eficiente para el reconocimiento facial utilizando Python y OpenCV para la obtención de características para el procesamiento de imágenes en 2D?

2.2. OBJETO Y CAMPO DE ACCIÓN

2.2.1. Objeto de estudio

Técnicas de reconocimiento facial.

2.2.2. Campo de acción

Prototipo para reconocimiento facial utilizando Python y OpenCV.

2.3. BENEFICIARIOS

En la tabla 1, se puede evidenciar que los beneficiarios de este proyecto de investigación es principalmente la Universidad Técnica de Cotopaxi y los beneficiarios indirectos son el grupo de investigación y personas externas.

Tabla 1: Beneficiario directos e indirectos

Beneficiarios Directos	Beneficiarios Indirectos
Universidad Técnica de Cotopaxi	Grupos de investigación Personas externas

2.4. JUSTIFICACIÓN

Este trabajo de investigación se realiza con el propósito de aportar al campo de estudio del reconocimiento facial global 2D, mediante una revisión bibliográfica sobre bases de datos, técnicas de extracción de características y algoritmos de clasificación de patrones en Python y OpenCV, en donde se resaltan los artículos más relevantes sobre el tema para determinar la metodología aplicada y poder ejecutar en una propuesta, para ser analizada en el área del reconocimiento facial, ya que se estaría comprobando que el uso de estos algoritmos mejoran favorablemente el reconocimiento de imágenes extraídas de una base de datos.

En la actualidad, el avance tecnológico aporta de manera significativa en el progreso de la sociedad, y en el área del reconocimiento facial que está impactando en la humanidad, sin embargo, es importante que la investigación avance hasta conseguir una técnica eficiente que logre un reconocimiento facial humano sin importar que esta se encuentre estática o en movimiento, color, rostro semicubierto ante el uso de mascarilla debido a la pandemia del Covid-19.

Ante esta realidad es vital trabajar en la búsqueda de una técnica más eficiente que asegure el proceso de reconocimiento de imágenes faciales por lo que resulta por demás justificado el tema propuesto. De allí la significación de continuar avanzando con la investigación propuesta, cabe mencionar que este tema está enmarcado dentro de la línea de investigación de inteligencia artificial e inteligencia de negocios pertenecientes a la carrera de sistemas de información.

2.5. HIPÓTESIS

La técnica encontrada para el reconocimiento facial utilizando Python y OpenCV, permitirá la detección de rostros en imágenes 2D de manera eficiente.

Variable dependiente: Reconocimiento facial.

Variable independiente: Eigenfaces, Fisherfaces y LBPH.

2.6. OBJETIVOS

2.6.1. General

Determinar una técnica eficiente de reconocimiento facial global con imágenes 2D utilizando Python y OpenCV para mejorar el uso del reconocimiento facial en diferentes áreas de estudio.

2.6.2. Específicos

- Revisar las fuentes bibliográficas referentes a bases de datos de imágenes 2D, extracción de características y algoritmos de reconocimientos de patrones para la identificación de rostros.
- Desarrollar un prototipo de modelo reconocimiento facial en 2D utilizando Python y OpenCV para identificación de rostros.
- Verificar el funcionamiento del prototipo mediante el criterio de juicio de expertos, el cual permita el desempeño para el reconocimiento facial.

2.7. SISTEMAS DE TAREAS

Tabla 2: Planificación de actividades.

Objetivos Específicos	Actividades	Resultados de las Actividades	Descripción (Técnicas e Instrumentos)
Revisar las fuentes bibliográficas referentes a bases de datos de imágenes 2D, extracción de características y algoritmos de reconocimientos de patrones para la identificación de rostros.	<p>Búsqueda de información sobre bases de datos con imágenes.</p> <p>Búsqueda de información de técnicas de extracción.</p> <p>Búsqueda de algoritmos de reconocimientos de patrones.</p>	<p>Base teórica de la investigación.</p> <p>Marco teórico.</p>	<p>Análisis documental.</p> <p>Ficha bibliográfica.</p>
Desarrollar un prototipo de modelo reconocimiento facial en 2D utilizando Python y OpenCV para identificación de rostros.	<p>Búsqueda de información sobre técnicas.</p> <p>Integración de la base de datos con Python.</p>	<p>Prototipo.</p> <p>Extracción de base de datos.</p> <p>Entrenamiento.</p>	<p>Uso de patrones, Python, técnicas de reconocimiento facial, Python, OpenCV.</p>
Verificar el funcionamiento del prototipo mediante el criterio de juicio de expertos, el cual permita el desempeño para el reconocimiento facial.	Opinión de expertos.	Diagnóstico del prototipo y probables errores.	Cuestionario de juicio de expertos.

3. FUNDAMENTACIÓN TEÓRICA

3.1. Antecedentes

Según Kuliah [3], en las civilizaciones primitivas las personas vivían en comunidades pequeñas, donde se reconocían sin dificultad por su número limitado de habitantes; sin embargo, debido al rápido aumento poblacional y la movilidad humana, la identificación se hizo un proceso complicado, de tal manera que las sociedades innovadoras han considerado necesario implementar sofisticadas técnicas de registro de identificación.

Manifestó Serratos [4], en 1882, el policía francés Alphonse Bertillon (1853-1914) presentó el primer sistema de identificación de las personas basado en las características físicas, es decir en los rasgos biométricos, y lo denominó antropometría y este se considera el primer sistema científico usado por la policía para identificar a criminales, está funciona al clasificar la forma de la nariz, la cara o del cuerpo de las personas: En la siguiente imagen se observa los diferentes tipos de narices que fueron publicadas en Pearson's Magazine.



Figura 1: Imagen publicada por Pearson's Magazine [4].

Afirmo Casanova [5], el concepto de detección facial se inició en los años 60's. Entre los años 1964 y 1965 Woodrow Wilson, Helen Chan y Charles Bisson desarrollaron el primer sistema semiautomático de reconocimiento facial mediante el uso del computador, en los años 70 Goldstein, Harmon, & Lesk, utilizaron 21 características físicas entre ellas están el tono del cabello, el espesor de labios, etc. Para mejorar la detección facial, pero para identificar estas características se debía seguir una secuencia manual.

A principios de los 90 Turk y Pentland [6], utilizando la "Técnica de eigenfaces", (método descubierto por Kirby y Sirovich) demostraron que: "el error fue utilizado en la identificación de rostros en imágenes, esto fue un hallazgo que permitió realizar sistemas de detección de rostros en tiempo real. Es por ello que la similitud era forzada por factores ambientales, sin embargo, esto ocasionó un interés significativo en posteriores desarrollos de estos sistemas".

Dijo Centeno [7], en el año 2001, se utilizaron cámaras de vigilancia en un partido de Super Bowl en donde el procedimiento fue la recolección de imágenes a través de las cámaras para después ser comparadas con imágenes digitalizadas de delincuentes que se encontraban en una base de datos que almacenaban dicha información”.

3.2. Machine Learning

Para Maisueche [8] ,El Machine Learning es una técnica asociada a la detección automática de patrones relevantes dentro de un conjunto de datos, en los últimos años, se ha convertido en una herramienta muy común en prácticamente todas las tareas que requieren extraer información a partir de grandes cantidades de datos.

Según Shwartz y David [9], en el día a día, estamos rodeados de tecnología basada en Machine Learning: el filtrado de correos electrónicos, los sistemas de recomendaciones, la detección facial y el reconocimiento del habla del smartphone, la previsión del tiempo atmosférico o la consulta del tráfico en la carretera., también se emplea en otros ámbitos como la medicina, el marketing, la logística o el mantenimiento de equipos industriales.

Manifiesta Shwartz y David [9], debido a la complejidad de todas estas aplicaciones, un ser humano no es capaz de programar una serie de especificaciones concretas para la realización de dichas tareas, sino que tiene que dotar a las propias computadoras con la habilidad de aprender de la experiencia y de adaptarse a las nuevas situaciones.

Dijo Moreno [10], el aprendizaje se refiere, como se ha visto, a un amplio espectro de situaciones en las cuales el aprendiz incrementa su conocimiento o sus habilidades para cumplir una tarea, el aprendizaje aplica inferencias a determinada información para construir una representación apropiada de algún aspecto relevante de la realidad o de algún proceso, una metáfora habitual en el área del aprendizaje automático dentro de la Inteligencia Artificial es considerar la resolución de problemas según afirmó Angluin [11], como un tipo de aprendizaje que consiste una vez resuelto un tipo de problema en ser capaz de reconocer la situación problemática y reaccionar usando la estrategia aprendida, actualmente la mayor distinción que se puede trazar entre un animal y un mecanismo de resolución de problemas es que ciertos animales son capaces de mejorar su actuación, en un amplio conjunto de tareas, como resultado de haber solucionado un cierto problema.

3.3. Visión por computador

Para Angluin [11], la visión por computador, también denominada visión artificial, visión de máquina, visión computacional, análisis de imágenes o interpretación de escenas, es el proceso de extracción de información del mundo real a partir de imágenes utilizando como herramienta un computador.

3.4. Reconocimiento de rostros

Manifestó Máster [12], la detección de rostros es una técnica que permite encontrar en una imagen, el rostro o cara de una o varias personas, mientras que ignora el fondo de la imagen u otros objetos que estén presentes dentro de ella.

Según OpenCV [13], está presente en muchas aplicaciones que usamos día a día, por ejemplo en facebook con las imágenes que subimos que ya detectan el rostro y lo que muchas veces nos pide es adicionar el nombre de la o las personas que allí aparecen, también tenemos los filtros de instagram en donde se necesita detectar el rostro para que los filtros que deseemos aplicar sean empleados. Incluso está presente en algunas aplicaciones bancarias o cuando usamos un smartphone para tomar una foto y muestra en un recuadro o círculo los rostros de las personas que aparecen allí.

Para Máster [12], la detección de objetos mediante clasificadores en cascada basados en funciones de Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones, es un enfoque basado en el aprendizaje automático en el que una función de cascada se entrena a partir de muchas imágenes positivas y negativas, luego se usa para detectar objetos en otras imágenes.

Afirma Máster [12], el método que utiliza Haar se puede describir como una pantalla de píxeles, de orientación y tamaño diferentes, separada por rectángulos, siendo así que cada rectángulo será bien positivos o negativos.

Tabla 3: Características Haar.

+	-
+	-

Para Máster [12], esta ventana compuesta por píxeles recorre toda la imagen, sumando los píxeles positivos y negativos respectivamente, luego se realiza una diferencia de que se denomina valor de la característica, este valor se puede representar a través de la siguiente ecuación:

$$h(x, y) = \sum_p I(x, y) - \sum_n I(x, y)$$

Donde:

$I(x, y)$ = Imagen a evaluar.

p y n = región positiva y negativa.

$H(x, y)$ = valor de la característica Haar en el punto x y y .

3.5. Reconocimiento facial

Según Ibarra y Paredes [14], el reconocimiento facial es un área que forma parte del reconocimiento de patrones, en los últimos años ha cobrado un gran interés especialmente por la amplia gama de aplicaciones que tiene en distintos campos tales como seguridad, vigilancia, tarjetas inteligentes, entre otros.

Por otro lado Mullainathan y Spiess [15], el reconocimiento facial es una parte de la visión por computador. El reconocimiento facial se viene usando hace muchas décadas principalmente por la fuerza armada.

Según Llapapasca y Zevallos [16], se entiende que el reconocimiento facial es una parte de la visión por computador, es necesario conocer su definición para así tener un mejor entendimiento acerca del tema, la misma es una manera de obtener, procesar las imágenes del mundo real para poder así obtener información numérica que pueden ser manejado por una computadora.

3.5.1. Aplicaciones

El uso del reconocimiento facial hoy en día se usa en diferentes aspectos y con nuevas tecnologías. Se utiliza en diferentes actividades como por ejemplo en las redes sociales para para poder ingresar a su cuenta social, también se utiliza para desbloquear diferentes dispositivos como celulares o puertas inteligentes, otro uso también se da en los aeropuertos o cámaras de seguridad para poder rastrear a personas desaparecidas o delincuentes siendo así que la aplicación de la tecnología ayuda de manera satisfactoria en el progreso de la humanidad

dando resultados óptimos y favorables. Según EcuRed [17], en la siguiente imagen se observa el uso del reconocimiento facial.



Figura 2: Uso del reconocimiento facial[17].

3.5.2. Métodos basados en la imagen

Para Centeno [7], “Se basa en patrones los cuales se pueden comprobar si una imagen contiene un rostro. Esto se consigue entrenando los métodos de redes neuronales con imágenes que contienen rostros y otras que no muestran imágenes”.

3.5.3. Sistemas de detección facial

Para Grado [18] los sistemas de detección facial tienen como objetivo detectar la presencia de una persona a través de sus rasgos faciales en una imagen digital, la principal ventaja de estos sistemas de detección, es que no es intrusivo, por lo que no requiere la colaboración del usuario más allá de estar frente a la cámara empleada en el sistema de seguridad, además de que solamente requiere de un solo dispositivo de captura.

3.5.4. Estudio de las texturas de los canales de color

Según Grado [18], este estudio en la actualidad es el más utilizado, se han realizado diferentes aproximaciones, pero la idea general consiste en analizar las texturas de la imagen teniendo en cuenta que la fisiología de la cara humana posee unas cualidades ópticas que no se pueden replicar en un spoofing, todo esto provoca en las imágenes veraces unas texturas que pueden tener alteraciones y permitirá identificarlo como no veraz.

3.6. Técnicas de reconocimiento facial

3.6.1. EigenFaces

Para Ibarra y Paredes [14], es una técnica que permite determinar, mediante la ortogonalidad dimensional, qué vectores ofrecen más información a un conjunto de datos de dimensión N , no obstante, la información N -dimensional obtenida con la técnica de Eigenfaces contiene datos redundantes que solo ocasionan que un sistema de clasificación tenga un alto costo computacional.

Para remediar se aplica el análisis de componentes (PCA), según Gottumukkal Asari [19] el cual toma una cantidad menor de los vectores entregados por las imágenes de la base de datos pero con información necesaria para la reconstrucción de los rostros de las imágenes ingresadas.

Para Ibarra y Paredes [14], el reconocimiento se realiza mediante la proyección de una imagen de un rostro en el subespacio formado por los eigenfaces y comparando su posición con la de los otros rostros conocidos, los eigenfaces son un conjunto de vectores representados gráficamente, convirtiéndose en una especie de mapa de las variaciones entre imágenes.

Según Ibarra y Paredes [14], estos vectores son el resultado de la aplicación de PCA a la matriz de covarianza de un conjunto de imágenes de rostros los cuales son denominados eigenvectores, siendo una imagen tratada como un vector en un espacio multidimensional, a su vez, cada cara individual puede ser representada exactamente en términos de una combinación lineal de las eigenfaces, la cual puede ser aproximada usando solamente los “mejores” eigenfaces que son las que tienen mayores eigenvalores, los M mejores eigenfaces conforman un subespacio M-dimensional de las caras de todas las posibles imágenes.

Este método realiza las siguientes operaciones y pasos:

- Adquisición de una serie de imágenes de caras iniciales [14].
- Cálculo de los eigenfaces del conjunto de entrenamiento, almacenando únicamente las M-dimensiones que correspondan con los eigenvalores mayores [14].
- Cálculo de un conjunto de pesos basados en la imagen de entrada y las M-eigenfaces mediante la proyección de la imagen de entrada sobre cada una de las eigenfaces [14].
- Determinar si la imagen de la cara pertenece o no al conjunto de entrenamiento, por medio de la distancia euclídea [14].

3.6.2. LBPH

Para Gottumukkal y Asari [19] ,el método de patrones binarios locales fue diseñado para la descripción de texturas. Según Alvarado y Fernández [20] ,el uso de descripciones locales en algunas regiones del rostro aportan más información que otras, por lo que los descriptores de texturas tienden a promediar la información que describen, lo cual no es conveniente al describir rostros puesto que mantener la información de las relaciones espaciales es importante. Para Álvarez [21],formar la descripción global, la imagen del rostro es dividida en diferentes regiones, a las que se les aplica un histograma con el que se obtiene el operador LBPH que

describe información independiente por región, estas descripciones locales son entonces concatenadas para construir una descripción global del rostro.

Según Casanova [5], el histograma se representa como un esquema donde se puede observar la distribución de las variables de intensidad de cada uno de los píxeles de la figura, cuya finalidad es conocer datos sobre la intensidad, el contraste, brillo, etc.

Para Casanova [5], la representación matemática de un histograma es: n como la cantidad de píxeles de una figura con un determinado grado de gris, donde r_k es el k -ésimo grado de gris dentro del rango $[0; L - 1]$, es definido que n_k es correspondiente a las variables del histograma, de esto se determinó la posibilidad de ocurrencia de todos los grados de gris:

$$p(r_k) = \frac{n_k}{n}$$

Manifestó Galindo y Gamboa [22], la técnica LBPH fue desarrollado para la describir la textura del área de las figuras segunda dimensión, estas se pudieron explicar a través de 2 medidas complementarias: De contraste de escala de grises y patrones espaciales locales, este algoritmo básicamente modela la estructura local de una imagen a través del cotejo de cada uno de los píxeles con su píxel próximo, al inicio el vecindario fue de 3×3 píxeles.

Para Galindo y Gamboa [22], se tomó un píxel para que sea umbral y núcleo de los píxeles próximos. Si la magnitud del píxel ubicado en el centro es igual o mayor a su píxel próximo, se da un valor en binario, en esta casuística 1 y 0 en opuesto caso, cuando termina el procedimiento, el píxel céntrico que está rodeado de 8 píxeles culmina con un valor numérico binario por píxel próximo, en otras palabras, un valor numérico de 8 bits con 28 combinaciones posibles que señalan la repartición de magnitud de los píxeles próximos.

3.6.3. Fisherfaces

Manifestó Martínez y Kak [23], Fisherface es una técnica de reconocimiento de rostros, que tiene en cuenta la luz y expresiones faciales. Este se encarga de clasificar y reducir las dimensiones de las caras usando el método FLD (Discriminant Lineal Fisher). Para Martínez y Kak [23], el análisis discriminante de Fisher intenta proyectar los datos de manera que su nueva dispersión sea óptima para la clasificación, mientras PCA busca los vectores que mejor describen los datos, LDA (Discriminant Lineal Analysis) busca los vectores que proporcionan mejor discriminación entre clases después de la proyección.

Según Belhumeur, Hespanha y Kriegman [24], Fisherfaces realiza un LDA, donde se busca aprovechar la información disponible, sobre la clasificación de las imágenes de entrenamiento, para buscar una proyección que maximice la separación entre imágenes de diferentes personas (o clases) y minimice la distancia entre imágenes de una misma clase, así logra concentrar las imágenes mejorando, en forma importante, la tasa de reconocimiento.

Según Galindo y Gamboa [22], este método cuyo objetivo es detallar las desemejanzas entre conjunto de elementos donde se pueden encontrar p variables, es decir, una vez realizado la detección del rostro de una persona, las figuras fueron asociadas por persona, dicho grupo se le conoce como clase, partiendo esto se rebusca dos matrices de covarianza, una inter-clase correspondiente a las distintas figuras del mismo sujeto, y una clase correspondiendo a las figuras de personas diferentes, de la misma forma que Eigenfaces es muy indispensable la etapa para entrenar, donde es necesario tomar múltiples figuras del sujeto que se analizará, pero con diferentes perfiles y variancias luminosas, el punto es usar figuras que muestren la gran parte de las disposiciones que puede adquirir el sujeto cuando se use la detección de rostros.

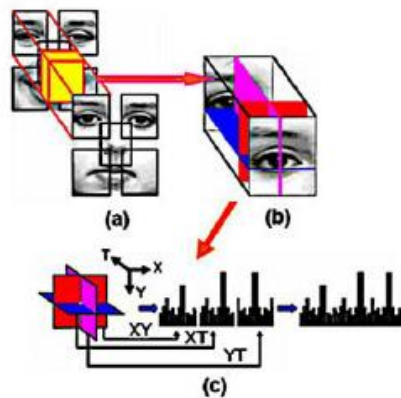


Figura 3: Histograma por cada región de la imagen [22].

Par el cálculo de los Fisherfaces se definió: La matriz de dispersión interclases:

S_w donde x_i^j es la instancia número i de la clase j , μ_j es la media de la clase j , c es el número de clases y $|T_j|$ el número de ejemplo de la clase j [22]:

$$S_w = \sum_{j=1}^c \sum_{i=1}^{|T_j|} (x_i^j - \mu_j)(x_i^j - \mu_j)^T \dots \dots$$

La matriz de dispersión extra clase, es la que simboliza la longitud las clases[22].

$$S_w = \sum_{j=1}^c (\mu_j - \mu)(\mu_j - \mu)^T$$

El incremento de la matriz extra-clase y la disminución de la matriz inter clase se alcanzan de la razón de los determinantes de las dos matrices[22].

$$\frac{\det(S_b)}{\det(S_x)}$$

La zona de exhibición resulta al solucionar la ecuación, donde W es la matriz de vectores asociados propiamente, que es la matriz diagonal con los propios valores, estas matrices están limitadas a la orden c-1 donde c es la cantidad de clases, cuya limitación se debe al cotejamiento de dos diferentes clases[22].

$$S_b W = \lambda S_w W$$

Para el reconocimiento de imágenes, esta se proyecta y se coteja con las figuras en la etapa de entrenar que de igual forma están proyectadas, el cotejamiento de estas imágenes se realiza a través de los clasificadores anteriormente precisados y la identidad se hace con la simbolización de la figura para entrenar la semejanza a la figura de prueba[22].

Para restringir el acceso a carpetas y archivos se desarrolló un prototipo para dispositivos móviles, fue basado en la librería OpenCV, donde se usará el algoritmo Fisherfaces, ya que reduce la problemática que ocasionan cambios de luz, deformaciones en la cara y cambios de postura, cuyo comportamiento supera a Eigenfaces siempre y cuando la cantidad de posiciones del perfil es una cantidad menor[22].

3.7. Python

Para Challenger, Díaz y García [25], es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible.

Manifiesta la Biblioteca Estándar de Python [26], que la sintaxis y semántica precisa del lenguaje Python, este manual de referencia de la biblioteca describe la biblioteca estándar que se distribuye con Python, también describe algunos componentes opcionales que son usualmente incluidos en las distribuciones de Python.

Según la Biblioteca Estándar de Python [26], la biblioteca estándar de Python es muy amplia, y ofrece una gran cantidad de producciones como puede verse en la larga lista de contenidos, la biblioteca contiene módulos incorporados (escritos en C) que brindan acceso a las funcionalidades del sistema como entrada y salida de archivos que serían de otra forma inaccesibles para los programadores en Python, así como módulos escritos en Python que proveen soluciones estandarizadas para los diversos problemas que pueden ocurrir en el día a día en la programación, algunos de estos módulos están diseñados explícitamente para alentar y reforzar la portabilidad de los programas en Python abstrayendo especificidades de las plataformas para lograr APIs neutrales a la plataforma.

3.8. OpenCV

Para OpenCV [27], es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto, OpenCV fue construido para proporcionar una infraestructura común para las aplicaciones de visión artificial y para acelerar el uso de la percepción de la máquina en los productos comerciales, al ser un producto con licencia BSD, facilita a las empresas la modificación del código.

Según OpenCV [27], la biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de aprendizaje por ordenador y de aprendizaje por ordenador tanto clásicos como de vanguardia, estos algoritmos se pueden usar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, encuentre imágenes similares de una base de datos de imágenes, elimine los ojos rojos de las imágenes tomadas con flash, siga los movimientos de los ojos, reconozca paisajes y establezca marcadores para superponerlos con realidad aumentada.

3.9. Base de datos imágenes

3.9.1. Base de datos REPLAY-ATTACK

Para Replay-Attack [28] , que consiste en un conjunto de vídeos de hasta 50 diferentes personas, con casos tanto veraces como spoofing en diferentes condiciones de entorno e instrumentos utilizados para el spoofing.

3.9.2. Orientado a objetos.

Según manifiesta Mora [29],aplica a los datos el paradigma de la orientación a objetos (OOP, object-oriented programming), irrumpió con fuerza en los años noventa debido a las nuevas necesidades de almacenamiento de las bases de datos relacionales (imágenes, documentos, ficheros de audio y vídeo).

3.9.3. Base de Datos de Imágenes de Rango del Signal Analysis and Machine Perception Laboratory

Afirmo Rodríguez, Ceballos, Hernández y Prieto [30],que esta base de datos contiene imágenes escaneadas con varios sensores de rango reales y en su mayoría con datos sintéticos, disponibles en los formatos GIF y formato neutral comprimido (formato “.txt”) con dimensiones de puntos en X, Y y Z.

3.9.4. Base de datos de USF (University of South Florida)

Para Rodríguez, Ceballos, Hernández y Prieto [30],esta base de datos contiene alrededor de 400 imágenes de rango, tomadas desde diferentes cámaras, como Odetics LADAR, Percetron LADAR, de Luz estructurada ABW y de luz estructurada K2T y, por consiguiente, dadas en diferentes formatos como PGM, ABW y K2t entre otros.

4. MATERIALES Y MÉTODOS

4.1. Enfoque de la investigación

El enfoque de esta investigación es cuantitativa, por consiguiente, se logró almacenar y analizar los datos obtenidos de una manera que facilite la investigación, con la finalidad de comprender de mejor manera el objeto de estudio sobre el reconocimiento facial global utilizando Python y Opencv, con esta información se planteó el problema, formulación del problema, fundamentación teórica, procesamiento y la discusión de los resultados.

4.2. Tipos de investigación

4.2.1. Investigación bibliográfica

La investigación bibliográfica permitirá obtener información relevante de artículos, libros, sitios web autorizados que tengan contenido verídico sobre el uso del reconocimiento facial y técnicas. La investigación se hará por medio de buscadores autorizados y repositorios bibliográficos que nos permita la obtención de información.

4.2.2. Investigación de campo

La recopilación de información se hará a los estudiantes de la Universidad Técnica de Cotopaxi pertenecientes a la carrera de sistemas de información recolectando imágenes de sus rostros con la cámara del computador, capturando distintos ángulos, variaciones de luz, gestos.

4.2.3. Investigación descriptiva

La investigación tiene como objetivo determinar las técnicas del reconocimiento facial global en imágenes 2D con la finalidad de conocer la eficiencia y dar un reporte con los datos obtenidos en esta investigación.

4.3. Métodos teóricos

4.3.1. Método Analítico

En esta investigación con la utilización de este método se separó las áreas de estudio para así obtener la información por diferentes fuentes bibliográficas de forma individual, los mismos que serán estudiados y analizados.

4.3.2. Método hipotético

Mediante este método se planteó la hipótesis y a partir de ella se pudo explicar el tema que se investigó, realizando deducciones que nos ayudó a comprobar la veracidad o falsedad del problema a solucionar.

4.4. Métodos Empíricos

4.4.1. Técnicas de investigación

4.4.1.1. Análisis Documental

Mediante esta técnica se pudo observar y registrar la información sobre las diferentes técnicas que existen para el reconocimiento facial global, así como también la información fundamental sobre el aprendizaje por visión del computador y así aplicarla a nuestra investigación.

4.4.2. Instrumentos de la investigación

4.4.2.1. Ficha bibliográfica

Mediante la ficha bibliográfica se va a identificar las fuentes de información en el funcionamiento del reconocimiento facial, técnicas, antecedentes de proyectos vinculados.

4.4.2.2. Población

En esta investigación la población estuvo conformado por los estudiantes de la Universidad Técnica de Cotopaxi pertenecientes a la carrera de sistemas de información para poder sacar características para el entrenamiento y realizar las pruebas respectivas.

4.4.2.3. Muestra

Para determinar la muestra se utilizó tres técnicas en el reconocimiento facial que son Eigenfaces, Fisherfaces y LBPH con un total de 31 estudiantes de diferentes características y etnias que serán utilizadas en esta investigación.

4.4.2.4. Cálculo de la muestra

La investigación es de tipo muestreo no probabilístico, debido a que se utiliza para extraer un muestreo de probabilidad aleatorio. Se consideró para no obtener un cálculo de muestra por el motivo que se tomó a estudiantes de la carrera de Sistemas de Información, para la realización de la investigación, obtención de datos y por lo tanto los resultados de la investigación.

4.4.2.5. Lenguaje de programación

Para el presente trabajo de investigación se utilizó el lenguaje de programación Python, debido a su adaptabilidad en diferentes entornos de desarrollo a su vez cuenta con gran cantidad de documentación referente a sus funciones y diversas aplicaciones.

4.4.2.6. Librería

Se utilizó la librería OpenCv (Open Computer Vision) debido que es desarrollada por Intel y está optimizada para aprovechar cada núcleo del procesador, evitando sobrecargarlo y aportando un mejor rendimiento.

Se utilizó NumPy para el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.

Empleamos os que nos permite acceder a funcionalidades dependientes del Sistema Operativo. Sobre todo, aquellas que nos refieren información sobre el entorno del mismo y nos permiten manipular la estructura de directorios.

4.4.2.7. Entorno de desarrollo

Para el presente trabajo usamos el entorno PyCharm debido a que está diseñado específicamente para el lenguaje de programación Python, adopta el enfoque centrado en el teclado para aprovechar al máximo las numerosas funciones de productividad de PyCharm.

4.5. Variables

4.5.1. Variable independiente

El lenguaje de programación Python y la librería OpenCV.

4.5.2. Variable dependiente

Reconocimiento facial.

4.6. Metodología de la investigación

4.6.1. Tipo y diseño de la investigación

Para la presente investigación se utilizó una metodología cuantitativa y una investigación experimental. Según manifestó Neill y Suárez [31], se basa en los aspectos numéricos para investigar, analizar y comprobar información y datos.

Para Dávila [32], consiste en contrastar hipótesis desde el punto de vista probabilístico y, en caso de ser aceptadas y demostradas en circunstancias distintas, a partir de ellas elaborar teorías generales

De acuerdo a la definición, la metodología cuantitativa nos permite realizar un análisis con los resultados obtenidos para que después en base a la estadística poder verificar la eficiencia de cada una de las técnicas utilizadas para el reconocimiento facial. La investigación experimental se refiere al cambio de la variable de entrenamiento en diferentes espacios controlados para realizar la prueba del reconocimiento facial. De acuerdo a Murillo [33], manipula una o más variables de estudio, para controlar el aumento o disminución de esas variables y su efecto en las conductas observada.

Ya dentro de la investigación se incluyen datos de entrenamiento y pruebas en las carpetas que contienen imágenes de las personas que fueron utilizadas para este proceso de reconocimiento facial, cabe recalcar que a partir del base de datos de las imágenes que se obtuvieron se obtienen todas las características relevantes para ser utilizadas por las tres técnicas (Eigenfaces, Fisherfaces y LBPH) en el reconocimiento facial en imágenes en 2d.

4.6.2. Unidad de análisis

Esta investigación va a manejar imágenes faciales en 2D. Las imágenes faciales poseen características únicas que las diferencian la una de la otra. Un rostro humano se establece en dos dimensiones tomando el plano y coordenadas (x, y) de tal forma podemos obtener imágenes, con gestos y emociones (alegre, triste, enojado, etc.).



Figura 4: Emociones y gestos.

4.6.3. Técnicas de análisis de datos

Como parte de la investigación experimental que aplica en este trabajo de investigación, para el análisis y proceso de datos obtenidos se utilizan algoritmos para extracción de carteristas de imágenes y algoritmos para el reconocimiento de las mismas dentro de los procesos de reconocimiento en 2D.

En forma generalizada se destaca dentro del proceso de reconocimiento en 2D, la utilización de algoritmos de vital importancia, tales como el filtrado y extracción de características y un clasificador de patrones.

4.7. Detección de rostros

La detección de rostros es una técnica que posibilita hallar en una imagen, la cara o caras de numerosas personas, mientras tanto que ignora el fondo de la imagen u otros objetos que se encuentren presentes en ella.

Está presente en muchas aplicaciones que usamos día a día, por ejemplo, en la red social Facebook con las imágenes que se suben detectan el rostro y lo que muchas veces nos pide es adicionar el nombre de la o las personas que allí aparecen. También en la red social Instagram tenemos los filtros en donde se necesita detectar el rostro para que sean aplicados.

4.7.1. Detección de un rostro en una imagen

No es una tarea fácil para el computador, por ello necesitamos que este ‘aprenda’ por lo cual empleamos Machine Learning o aprendizaje automático para la detección del rostro.

En un principio se necesita de una gran cantidad de imágenes para entrenar a un clasificador, para que pueda diferenciar entre la presencia de un objeto y la no presencia del mismo. Por ejemplo, para realizar un detector de rostros se requieren imágenes positivas (que contengan rostros) e imágenes negativas (imágenes que no contengan rostros).

Luego se procederá a la extracción de características de todas las imágenes, para después emplear un enfoque de Machine Learning y proceder con el entrenamiento. Finalmente se puede obtener el clasificador.

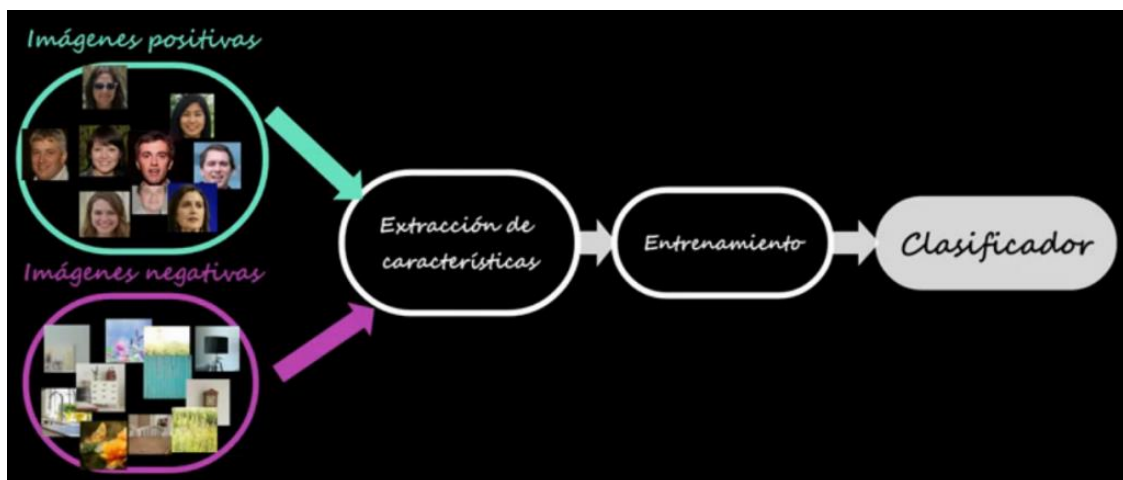


Figura 5: Proceso para crear un clasificador de rostros [34].

Ahora que tenemos en cuenta este proceso, veamos brevemente la detección facial usando Haar Cascade. Este detector de objetos usa el método propuesto en el paper “Rapid Object Detection using a Boosted Cascade of Simple Features”

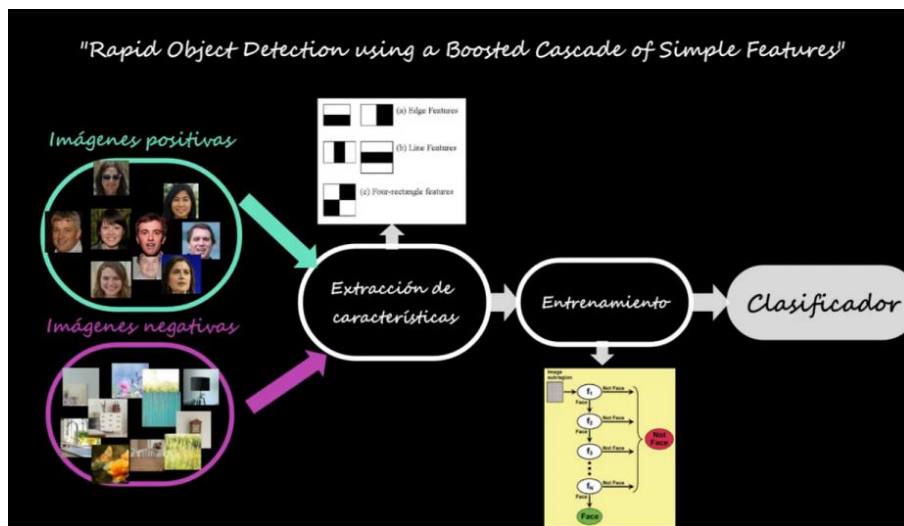


Figura 6: Proceso para crear un clasificador de rostros empleando Haar-Cascade [34].

4.7.2. Haar Cascades Python – OpenCV

OpenCV nos ofrece clasificadores pre entrenados no solos de rostros de personas el cual usaremos en el presente trabajo, sino de ojos, sonrisa, cuerpo entero entre otros. Se puede encontrar estos archivos XML en la carpeta: `opencv/data/haarcascades/` o en el repositorio de OpenCV en github.

4.7.3. DetectMultiScale

Para emplear la detección de rostros con haar cascade en OpenCV vamos a necesitar del módulo DetectMultiScale que ayudará a detectar los objetos de acuerdo al clasificador que se utilice. Este nos permitirá obtener un rectángulo delimitador en donde se encuentre el objeto que se desea encontrar dentro de una imagen, para ello se deben especificar algunos argumentos que se detallan a continuación.

4.7.4. ScaleFactor

Este parámetro especifica que tanto va a ser reducida la imagen. Por ejemplo, si se ingresa 1.1, quiere decir que se va a ir reduciendo la imagen en 10%, con 1.3 se reducirá 30%, creando de esta manera una pirámide de imágenes. Se debe tener en cuenta y es que, si damos un número muy alto, se pierden algunas detecciones. Mientras que para valores muy pequeños como por ejemplo 1,01 (es decir reducir en un 1% la imagen), llevará mayor tiempo de procesamiento,

ya que se tendrán más imágenes a analizar, además de que pueden incrementar los falsos positivos (que son detecciones presentadas como objetos u rostros, pero que en realidad no lo son).

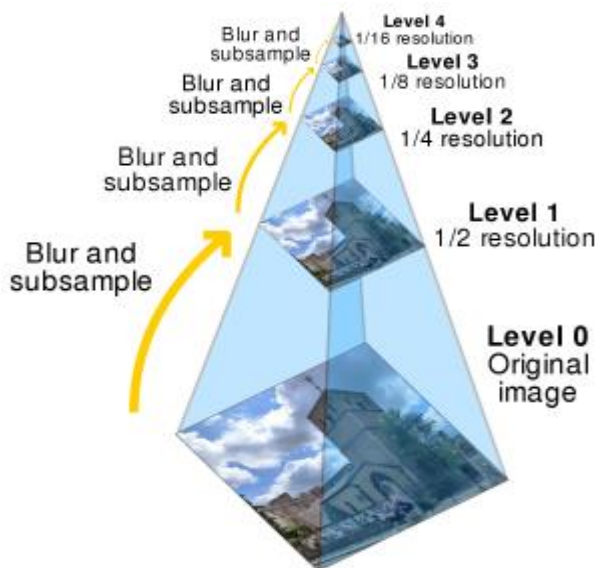


Figura 7: Ejemplo de pirámide de imágenes [35].

Se realiza una pirámide de imágenes debido al tamaño que presenten los rostros en la imagen, unos pueden ocupar mayor o menor área que otros, por lo que para que se trate de detectar todos en sus distintos tamaños se aplica la pirámide de imágenes.

4.7.5. MinNeighbors

Este parámetro especifica cuántos vecinos debe tener cada rectángulo candidato para retenerlo. Tenemos una pequeña ventana que va a ir pasando por una imagen buscando rostros, entonces puede que te encuentres que al final de todo el proceso ha identificado varios rostros (figura 7), pero puede que muchos de ellos correspondan a la misma persona. Entonces este parámetro, hace relación a todos esos rectángulos delimitadores de un mismo rostro. Por lo cual minNeighbors especifica el número mínimo de cuadros delimitadores o vecinos, que debe tener un rostro para que detectado como tal.

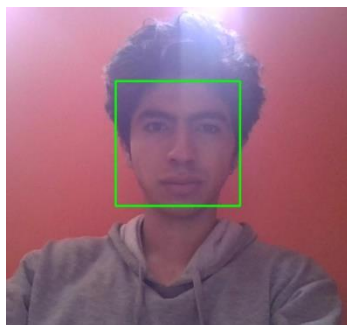


Figura 8: Detección de rostro.

Ahora, debemos tener en cuenta que entre más alto sea el valor que pongamos, menos caras se detectaran, mientras que si se da un valor muy bajo se pueden presentar falsos positivos.

4.7.6. Detección de rostros sobre una imagen

Para empezar, se guarda el archivo xml que contiene el clasificador de rostros frontales (haarcascade_frontalface_default.xml) en la misma carpeta donde está almacenado el script.

Vamos a detectar los rostros que aparecen en la siguiente imagen (figura 9):



Figura 9: Imagen de entrada para detención de rostros.

La programación dada para la detección de rostros será la siguiente:

```
1 import cv2
2
3 faceClassif = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
4
5 image = cv2.imread('imagenRostros.jpg')
6 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7
8 faces = faceClassif.detectMultiScale(gray,
9     scaleFactor=1.1,
10    minNeighbors=5,
11    minSize=(30, 30),
12    maxSize=(200, 200))
13
14 for (x, y, w, h) in faces:
15     cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
16
17 cv2.imshow('image', image)
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
```

Figura 10: Código para detectar el rostro.

Línea 1: Importar OpenCV.

Línea 3: Carga el clasificador con extensión xml con la ayuda de `cv2.CascadeClassifier`, dentro entre comillas especificar el nombre y extensión del archivo.

Línea 5 y 6: Leer la imagen en donde se van a detectar los rostros y luego proceder a transformar a escala de grises.

Línea 8: Cargar el clasificador en la línea 3 es necesario aplicar en la imagen, para ello se usa `detectMultiScale` que debe estar seguido de la variable con la cual se asignó la carga del clasificador y es aquí en donde los argumentos planteados recientemente: `image`, `scaleFactor`, `minNeighbors`, `minSize` y `maxSize`.



Figura 11: Imagen con detección de rostros.

4.8. Almacenando rostros

4.8.1. Almacenar rostros presentes en una imagen

Se procede a reutilizar el código de detección de rostros en una imagen con el uso de `haarcascades`, emplearemos el mismo procedimiento añadiendo nuevas líneas de código para guardar los rostros.



Figura 12: Imagen de entrada para almacenar rostros.

Código para detectar y almacenar los rostros de la imagen, será el siguiente:

```

1  import cv2
2
3  faceClassif = cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
4
5  image = cv2.imread('imagenRostros.jpg')
6  imageAux = image.copy()
7  gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
8
9  faces = faceClassif.detectMultiScale(gray, 1.1, 5)
10
11 count = 0
12
13 for (x,y,w,h) in faces:
14
15     cv2.rectangle(image, (x,y),(x+w,y+h),(128,0,255),2)
16     rostro = imageAux[y:y+h,x:x+w]
17     rostro = cv2.resize(rostro,(150,150), interpolation=cv2.INTER_CUBIC)
18     cv2.imwrite('dataRostros/rostro_{}.jpg'.format(count),rostro)
19     count = count + 1
20
21 cv2.imshow('rostro',rostro)
22 cv2.imshow('image',image)
23 cv2.waitKey(0)
24
25 cv2.destroyAllWindows()

```

Figura 13: Código para almacenar rostros de una imagen.

Línea 1: Importamos OpenCV.

Línea 3: Cargar el clasificador de rostros.

Línea 5: Se procede a leer la imagen de entrada que sería la correspondiente a la imagen de la figura 12.Línea 6: Crear una copia de la imagen de entrada, esta se mantendrá intacta, por lo que servirá para rectorar los rostros encontrados.

Línea 7: Transformar la imagen de entrada de BGR a escala de grises.

Línea 9: Se realiza la detección de rostros presentes en la imagen, según los parámetros de entrada de la función detectMultiScale.

Línea 11: Iniciar un contador count = 0, que nos ayuda al conteo de cada uno de los rostros detectados.

Línea 13: Extraer las coordenadas, ancho y alto de cada una de las caras detectadas.

Línea 15: Visualizar los rostros detectados en un rectángulo rosa.

Línea 16 y 17: En la línea 16 se procede a recortar los rostros de la imagen imageAux con ayuda de las coordenadas x, y ancho y alto que obtuvimos de la línea 13. En la línea 17 se redimensionan los rostros a 150 pixeles de ancho y alto.

Línea 18: Cada una de las imágenes tendrá un número diferente gracias al contador count.

Línea 21 a 25: Estas líneas son para la visualización del rostro y de las detecciones.

Al ejecutar el código se obtiene lo siguiente:



Figura 14: Almacenando rostros de una imagen.

Conforme se presione cualquier tecla se rodea un rostro y se va almacenando en la carpeta donde se encuentra el script. Se debe tomar en consideración que todos los rostros tienen el mismo ancho y alto ya que lo redimensionamos en el código.

4.8.2. Almacenar rostros presentes en un video.

Para la extracción y almacenamiento de rostros mediante el uso de videos se puede realizar en un video streaming o leer un video. Para el ejemplo lo realizaremos en un video streaming.

Código para detectar y almacenar los rostros del video streaming.

```
1 import cv2
2 import os
3 if not os.path.exists('Rostros encontrados'):
4     print('Carpeta creada: Rostros encontrados')
5     os.makedirs('Rostros encontrados')
6 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
```

Figura 15: Código para almacenar rostros de un video.

Línea 6: Especificar que se va a realizar un video streaming, sin embargo, se puede leer un video. Se usa `cv2.CAP_DSHOW` para que al momento de realizar la visualización ocupen totalmente la ventana.

Al ejecutar el código se obtiene lo siguiente:

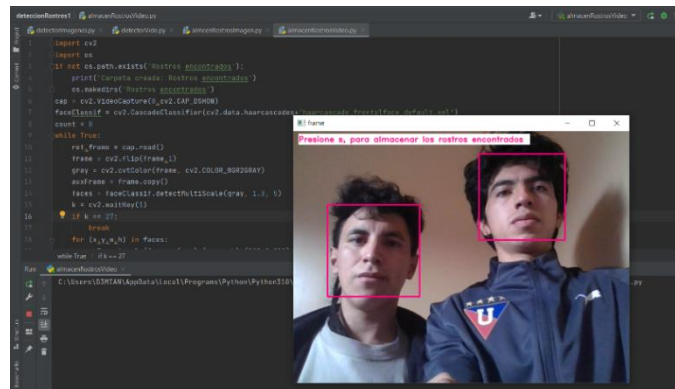


Figura 16. Almacenando rostros.

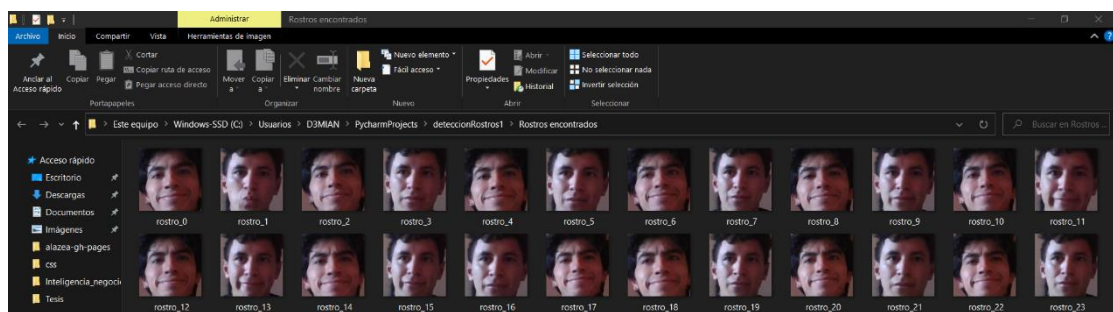


Figura 17. Rostros Almacenados.

Este procedimiento se usara más adelante para el reconocimiento facial, ya que necesitamos un gran cantidad de imágenes de personas que vayan a ser reconocidas por el algoritmo.

4.9. Reconocimiento Facial

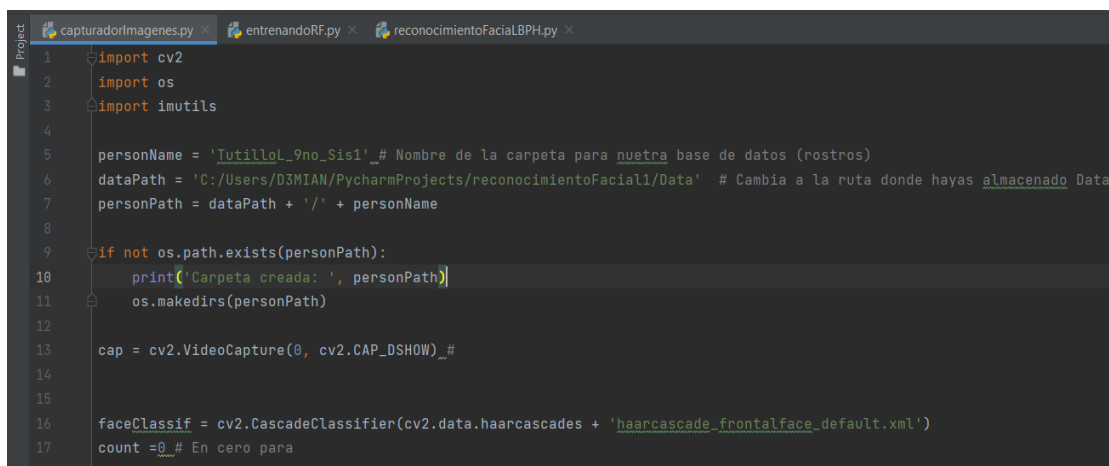
Para llevar a cabo el reconocimiento facial, en primer lugar es necesario recolectar datos, es decir los rostros de las personas que se desee reconocer, posteriormente se proceda entrenar el clasificador, para finalmente probarlo. Para todo este proceso será necesario usar la detección de rostros con haarcascades.

4.9.1. Creación de la base de datos

Para realizar el reconocimiento facial es necesario de los rostros de las personas que se quieran reconocer. Estos rostros deberán denotar variedad de expresiones como: felicidad, tristeza, aburrimiento, sorpresa, entre otros. Otro aspecto que deben tener estas imágenes es la variación de condiciones de luz, que las personas lleven lentes o que no los lleven, incluso que cierren los ojos o guiñen uno.

Se recomienda que se realice la recolección de estas imágenes en el escenario o ambiente en donde se vaya a aplicar el reconocimiento facial. Toda esta variedad de imágenes que se obtenga de los rostros contribuirá al desempeño de los algoritmos empleados en el presente trabajo.

En el código empleado para la demostración, se almacenaran automáticamente 130 rostros de un video streaming.



```
1 import cv2
2 import os
3 import imutils
4
5 personName = 'Tutillol_9no_Sis1' # Nombre de la carpeta para nuestra base de datos (rostros)
6 dataPath = 'C:/Users/D3MIAN/PycharmProjects/reconocimientoFacial1/Data' # Cambia a la ruta donde hayas almacenado Data
7 personPath = dataPath + '/' + personName
8
9 if not os.path.exists(personPath):
10     print('Carpeta creada: ', personPath)
11     os.makedirs(personPath)
12
13 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW) #
14
15
16 faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
17 count = 0 # En cero para
```

Figura 18: Código para almacenar rostros en la base de datos.

Línea 1 a 3: Importar OpenCV, os e imutils.

Línea 5 a 7: En estas líneas se crea una carpeta con el nombre de la persona que se desee reconocer, esta se creará dentro de la carpeta Data que ya había creado previamente de forma manual. Entonces en la línea 5 en personName se asigna el nombre de la persona, en la línea 6 se asigna a dataPath la ubicación del directorio donde se creará cada carpeta con el nombre de cada persona a reconocer. Finalmente personPath será la ruta completa.

Línea 9 a 11: Con la información de las líneas pasadas, se creará el directorio con el nombre de la persona a reconocer dentro de la carpeta Data. En la línea 5 por ejemplo puedes ver que se creará una carpeta llamada 'Tuttilo_9no_Sis1'.

```

13 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
14
15
16 faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
17 count = 0 # En cero para
18
19 while True:
20
21     ret, frame = cap.read()
22     if ret == False: break
23     frame = imutils.resize(frame, width=640)
24     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
25     auxFrame = frame.copy()
26
27     faces = faceClassif.detectMultiScale(gray, 1.3, 5)
28
29     for (x, y, w, h) in faces:
30         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
31         rostro = auxFrame[y:y + h, x:x + w]
32         rostro = cv2.resize(rostro, (150, 150), interpolation=cv2.INTER_CUBIC) # Tamaño de rostros
33         cv2.imwrite(personPath + '/' + rostro + '.jpg', rostro)
34         count = count + 1
35         cv2.imshow('frame', frame)
36
37     k = cv2.waitKey(1)
38     if k == 27 or count >= 130:
39         break
40
41 cap.release()
42 cv2.destroyAllWindows()

```

Figura 19: Código para establecer el y número de imágenes.

Línea 23: Se redimensiona con `imutils.resize`, esto se hace para redimensionar el tamaño de los fotogramas del video de entrada.

Línea 32: Se redimensiona las imágenes correspondientes a los rostros para que todos posean el mismo tamaño. Se ha establecido 150 píxeles.

Línea 38: En esta línea he añadido la condición de `count >= 130` para que el proceso de almacenamiento termine a los 130 rostros almacenados.

Al ejecutar el código se obtiene lo siguiente:

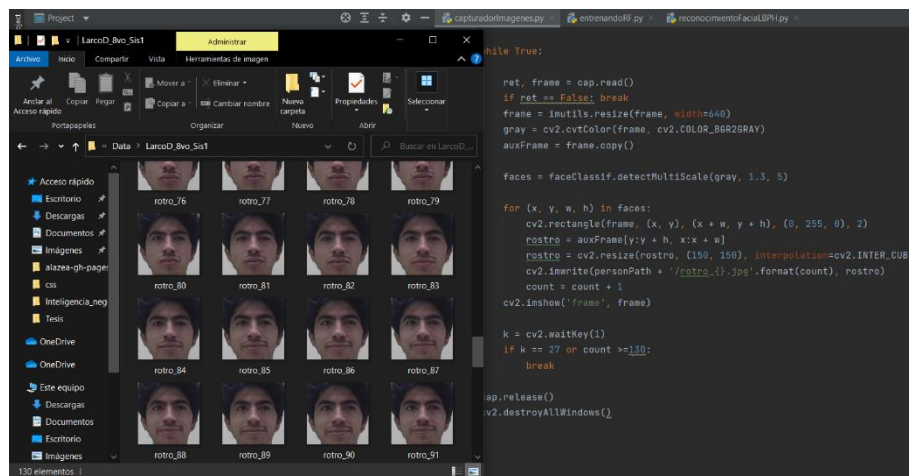


Figura 20. Base de datos creada a partir del código establecido.

4.9.2. Preparando datos para entrenar

Para proceder con el entrenamiento es necesario tener cada una de las imágenes con su respectiva etiqueta asociada a la persona a la que pertenecen los rostros. Por ejemplo cuando leamos la carpeta 'Abata todas esas imágenes se les asignará etiqueta 0, luego a todas imágenes de los rostros de 'Caizaluisa' se asignará 1, de 'Canizares' se asignará 2, y así sucesivamente. Con cada una de estas etiquetas el computador sabrá que las imágenes le corresponden a personas diferentes.

En otra clase a la que llamamos 'entrenadorRF.py' se prepara este proceso, es decir las imágenes y etiquetas para entrenarlos empleado: Eigenfaces, fisherfaces y Local Binary Patterns Histograms.

```
dataPath = 'C:/Users/D3MIAN/PycharmProjects/reconocimientoFacial1/Data'
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0

for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName_0))
        image = cv2.imread(personPath+'/'+fileName_0)
        cv2.imshow('image', image)
        cv2.waitKey(10)
        label = label + 1
```

Figura 21: Código para preparar las imágenes para el entrenamiento.

Línea 5 a 7: En primer lugar, se especifica la ruta de la carpeta 'Data' que se creó manualmente antes y dentro de ella estará cada carpeta con el nombre de las personas que se desee identificar. En la línea 6 se lista todos los nombres de las carpetas almacenadas en 'Data'. En la línea 7 se imprime la lista obtenida.

Línea 9: Se declara labels en este se almacenarán las etiquetas correspondientes a cada imagen según la persona.

Línea 10: Se declara facesData en donde se almacenará cada una de las imágenes de los rostros.

Línea 11: Se establece un contador label en 0, para conforme se termine de leer las imágenes de una persona, cambie a otro valor. Esto ayudará al clasificador a entender que se tarda de diferentes personas.

Línea 13 a 15: Lee cada una las carpetas dentro de 'Data'. En personPath se establece la ruta a la carpeta de cada persona. Mientras que en la línea 15 imprimimos el mensaje 'Leyendo las imágenes...' para que cuando se ejecute se pueda saber en qué parte del proceso se encuentra.

Línea 17 y 18: Se leen todas las imágenes correspondientes a cada rostro. En la línea 18 se imprime el nombre de la carpeta y la imagen.

Línea 19: En labels se añaden las etiquetas de cada imagen.

Línea 20: En el array facesData se añade cada imagen (rostro)..

Línea 24: Cada vez que se termine de almacenar los rostros y etiquetas de una carpeta, se incrementará label en 1.

4.9.3. Entrenamiento

Partiendo del reconocimiento facial usando OpenCV, pasamos al código.

```
31 # Métodos para entrenar el reconocedor
32 #face_recognizer = cv2.face.EigenFaceRecognizer_create()
33 #face_recognizer = cv2.face.FisherFaceRecognizer_create() # Dos modelos
34 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
```

Figura 22: Métodos para entrenar el conecedor.

```
37 # Entrenando el roconocedor de rostros
38 print("Entrenando...")
39 face_recognizer.train(facesData, np.array(labels))
```

Figura 23: Entrenando el reconocedor de rostros.

Línea 39: Para entrenar el reconocedor de rostros se necesita de face_recognizer.train, en donde face_recognizer será la variable donde se asignó el método (cualquiera de las líneas 31, 32 o 33). Dentro de los paréntesis se tiene que especificar el array en donde estén contenidos los rostros o imágenes de entrenamiento, mientras que el segundo parámetro corresponde a las etiquetas. Es necesario que este sea un numpy array por ello he puesto np.array(labels).

4.9.4. Guardar el modelo obtenido

Una vez entrenado el modelo que nos sirve para reconocer rostros se procede a guardarlo, con la finalidad de leerlo en otra clase. Para almacenar el modelo se usa write, dentro los paréntesis se especifica el nombre que se desea asignar, junto con la extensión XML o YAML.

```
41 # Almacenando el modelo obtenido
42 #face_recognizer.write('modeloEigenFace.xml')
43 #face_recognizer.write('modeloFisherFace.xml')
44 face_recognizer.write('modeloLBPHFace.xml')
45 print("Modelo almacenado...")
```

Figura 24: Guardar modelo obtenido

Al ejecutar el código se obtiene lo siguiente:

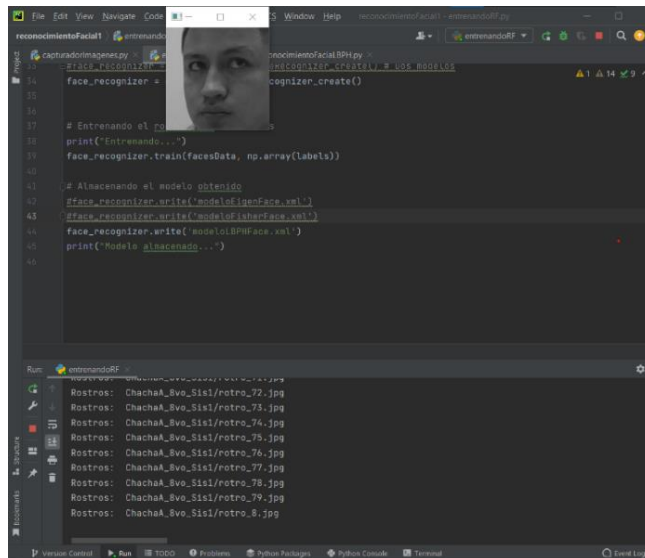


Figura 25: Entrenando el reconocedor de rostros.

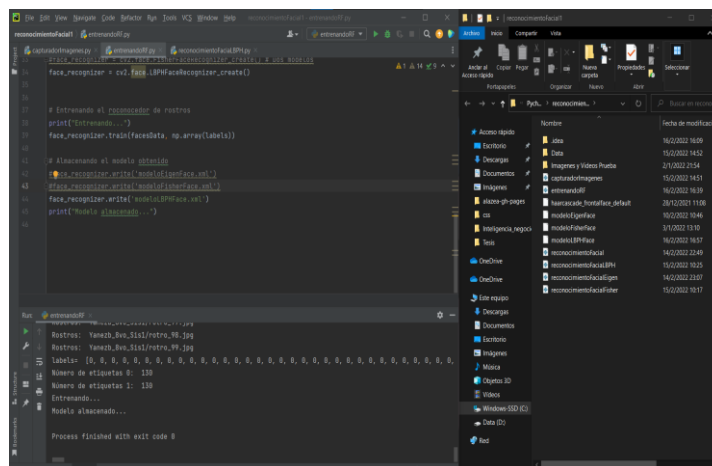


Figura 26: Guardando el modelo obtenido.

5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

5.1. Funcionamiento del prototipo

5.1.1 Tiempo de respuesta del entrenamiento de las técnicas

Se cuenta con 31 estudiantes de diferentes ciclos de la carrera de Sistemas de información cada uno con 130 fotografías de sus rostros, la cual se divide en 100 para entrenamiento y 30 para pruebas, cada imagen con un tamaño de 150 x 150 pixeles.

Tabla 4: Tiempo de entrenamiento

TIEMPO DE ENTRENAMIENTO	
31 estudiantes (3100 imágenes de 150 x 150 pixeles)	
PROCESADOR: INTEL CORE i7-9750H, RAM 16GB, GRÁFICA GTX 1060ti 6gb	
Técnicas	Tiempo
EIGENFACES	6365s (1 hora, 46 minutos y 5 segundos)
FISHERFACES	6284s (1 horas, 44 minutos y 44 segundos)
LBPH	7.86s

En la tabla 4 se observa que el que el menor tiempo de entrenamiento corresponde a la técnica LBPH con 7.86s, seguida de la técnica EIGENFACES con 6365s (1h, 46m, 5s) y en la última posición la técnica FISHERFACE con 6284s (1h, 44m, 44s) comprobando que la técnica más eficiente para el entrenamiento con la misma cantidad de imágenes corresponde a LBPH.

5.1.2. Pruebas de las técnicas de reconocimiento facial.

5.1.2.1. Primera Técnica (EIGENFACES)

```
# EigenFaces
if result[1] < 31000:
    cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y - 25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
else:
    cv2.putText(frame, 'Desconocido', (x, y - 20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)

# cv2.rectangle(frame, (x,y), (x+w, y+h),(0,255,0),2)
```

Figura 27: Código reconocimiento EIGENFACES.



Figura 28: Sujeto de prueba EIGENFACES.

La figura 27 representa en código para realizar el reconocimiento EIGENFACES el cual aplicaremos en una imagen para la respectiva prueba, se tomó un estudiante al azar (ver figura 28) obteniendo un resultado favorable en el tiempo de respuesta para el reconocimiento del estudiante Guañuna Alex perteneciente al sexto ciclo de la carrera de sistemas de información.

5.1.2.2. Segunda Técnica (FISHERFACES)

```
# FisherFaces
if result[1] < 500:
    cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y - 25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
else:
    cv2.putText(frame, 'Desconocido', (x, y - 20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

Figura 29: Código reconocimiento FISHERFACES.



Figura 30: Sujeto de prueba FISHERFACES.

La figura 29 representa en código para realizar el reconocimiento FISHERFACES el cual aplicaremos en una imagen para la respectiva prueba, se tomó un estudiante al azar (ver figura 30) obteniendo un resultado favorable en el tiempo de respuesta para el reconocimiento del estudiante Vargas Miguel perteneciente al séptimo ciclo de la carrera de sistemas de información.

5.1.2.3. Tercera Técnica (LBPH)

```
# LBPHFace
if result[1] < 70:
    cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x,y-25), 2, 1.1, (0,255,0), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
else:
    cv2.putText(frame, 'Desconocido', (x,y-20), 2, 0.8, (0,0,255), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)
```

Figura 31: Código reconocimiento LBPH.

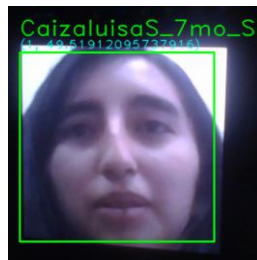


Figura 32: Sujeto de prueba LBPH.

La figura 31 representa en código para realizar el reconocimiento LBPH el cual se aplica en una imagen para la respectiva prueba, se tomó un estudiante al azar (ver figura 32) obteniendo un resultado favorable en el tiempo de respuesta para el reconocimiento de la estudiante Caizaluisa Sindy perteneciente al séptimo ciclo de la carrera de sistemas de información.

5.2. Comparación de las técnicas de reconocimiento facial

5.2.1. Eficiencia de las técnicas

5.2.1.1. Eigenfaces

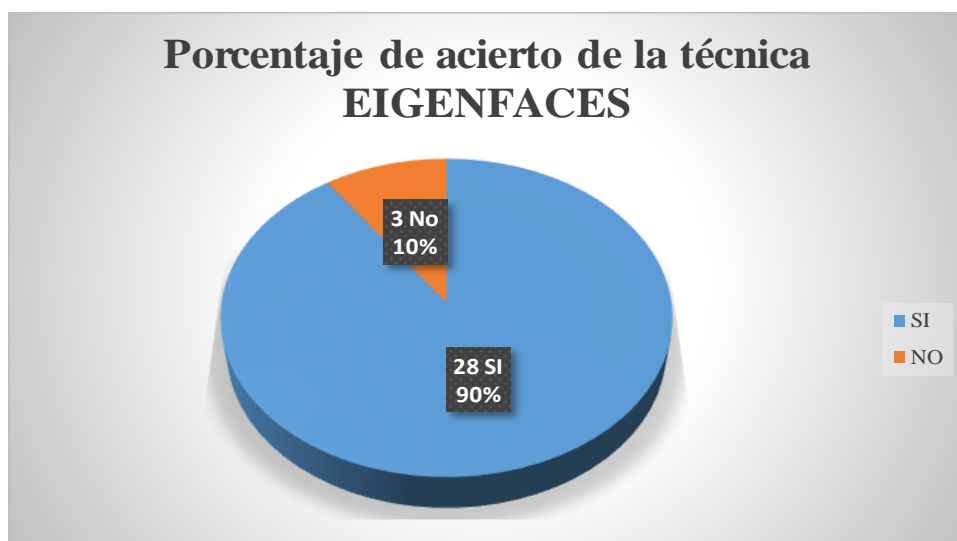


Figura 33: Porcentaje de acierto de la técnica EIGENFACES.

El acierto de la técnica EIGENFACES corresponde al 90% de efectividad y un 10% de ineficacia, identificando 28 aciertos y 3 negativas de un total de 31 estudiantes con 4030 fotografías de las cuales 3100 fueron utilizadas para el entrenamiento y 930 para las pruebas las mismas que tienen características como: gestos, variaciones de luz y accesorios.

5.2.1.2.Fisherfaces

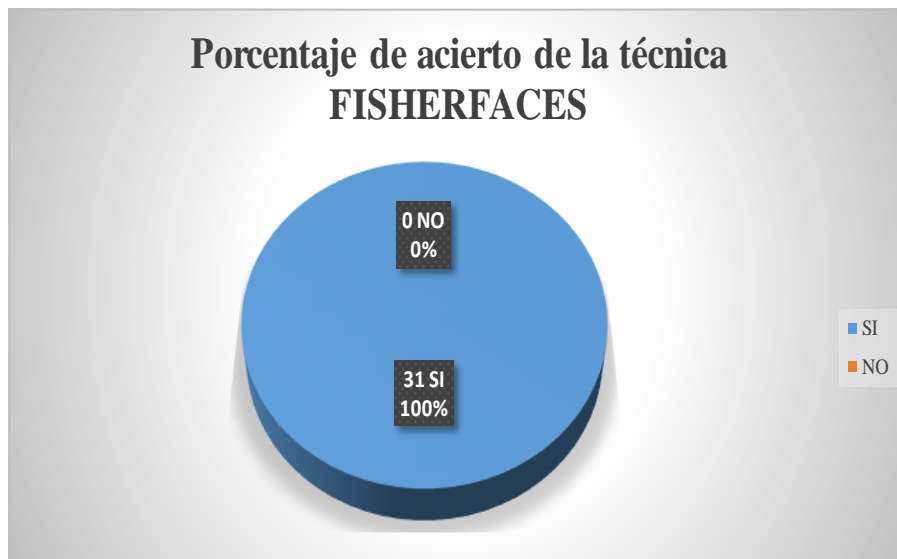


Figura 34: Porcentaje de acierto de la técnica FISHERFACES.

El acierto de la técnica FISHERFACES corresponde al 100% de efectividad, identificando 31 aciertos de un total de 31 estudiantes con 4030 fotografías de las cuales 3100 fueron utilizadas para el entrenamiento y 930 para las pruebas las mismas que tienen características como: gestos, variaciones de luz y accesorios.

5.2.1.3.LBPH



Figura 35: Porcentaje de acierto de la técnica LBPH.

El acierto de la técnica LBPH corresponde al 100% de efectividad, identificando 31 aciertos de un total de 31 estudiantes con 4030 fotografías de las cuales 3100 fueron utilizadas para el entrenamiento y 930 para las pruebas las mismas que tienen características como: gestos, variaciones de luz y accesorios.

5.2.2. Tiempo del reconocimiento de las técnicas

Tabla 5: Tiempo y acierto de las técnicas.

No	Nombre	Apellido	curso	Paralelo	Carrera	EIGENFACES		FISHERFACES		LBPH	
						Tiempo (Segundos)	Reconocio	Tiempo(Segundos)	Reconocio	Tiempo (Segundos)	Reconocio
1	Jessica	Abata	7mo	A	Sistemas de inform	4,16	Si	0,77	Si	2,27	Si
2	Sindy	Caizaluña	7mo	A	Sistemas de inform	0	No	1,17	Si	1,96	Si
3	Erick	Cañizares	8vo	A	Sistemas de inform	15,73	Si	1,37	Si	0,6	Si
4	Andres	Chacha	8vo	A	Sistemas de inform	10,75	Si	0,34	Si	1,52	Si
5	Luis	Chacon	8vo	A	Sistemas de inform	15,1	Si	1,94	Si	0,79	Si
6	Esteven	Chiluña	7mo	A	Sistemas de inform	0	No	3,34	Si	1,17	Si
7	Cristian	Diaz	7mo	A	Sistemas de inform	3,77	Si	0,8	Si	1,95	Si
8	Alex	Guañuna	6to	A	Sistemas de inform	7,12	Si	1,32	Si	2,07	Si
9	Damián	Larco	8vo	A	Sistemas de inform	6,57	Si	2,75	Si	1,22	Si
10	Leonel	Lema	7mo	A	Sistemas de inform	4,85	Si	2,72	Si	1,43	Si
11	Jonathan	Lituma	8vo	A	Sistemas de inform	8,16	Si	2,66	Si	0,27	Si
12	Erick	Lliguin	8vo	A	Sistemas de inform	5,78	Si	1,28	Si	1,11	Si
13	Denis	Martinez	7mo	A	Sistemas de inform	6,44	Si	1,94	Si	0,27	Si
14	Erick	Molina	7mo	A	Sistemas de inform	4,46	Si	1,32	Si	1,15	Si
15	Ronaldo	Molina	7mo	A	Sistemas de inform	0,87	Si	2,11	Si	1,56	Si
16	Jonathan	Montatixe	7mo	A	Sistemas de inform	5,92	Si	3,31	Si	1,23	Si
17	Danner	Morales	7mo	A	Sistemas de inform	7,42	Si	4,15	Si	0,44	Si
18	Jonathan	Oña	7mo	A	Sistemas de inform	0	No	1,65	Si	1,96	Si
19	Denis	Orellana	7mo	A	Sistemas de inform	7,3	Si	8,12	Si	1,9	Si
20	Denis	Pacheco	8vo	A	Sistemas de inform	12,78	Si	1,38	Si	3,68	Si
21	Jhon	Quinaucho	6to	A	Sistemas de inform	19,86	Si	1,71	Si	4,02	Si
22	Bryan	Quispe	7mo	A	Sistemas de inform	18,66	Si	3,26	Si	1,07	Si
23	David	Rea	6to	A	Sistemas de inform	23,38	Si	1,82	Si	3,28	Si
24	Diego	Reinoso	7mo	A	Sistemas de inform	5,53	Si	0,68	Si	1,05	Si
25	Wilson	Rojas	6to	A	Sistemas de inform	18,91	Si	1,64	Si	3,45	Si
26	Pamela	Sarabia	6to	A	Sistemas de inform	4,67	Si	3,4	Si	2,9	Si
27	Viviana	Telena	7mo	A	Sistemas de inform	16,49	Si	3,11	Si	0,27	Si
28	Luis	Tutillo	9no	A	Sistemas de inform	1,79	Si	0,95	Si	0,84	Si
29	Miguel	Vargas	7mo	A	Sistemas de inform	5,41	Si	0,58	Si	0,27	Si
30	Daniela	Viturco	7mo	A	Sistemas de inform	15,95	Si	1,62	Si	3,88	Si
31	Brayan	Yanez	8vo	A	Sistemas de inform	6,61	Si	0,87	Si	1,41	Si

Tabla 6: Media, mediana, moda del tiempo del reconocimiento de las técnicas.

TIEMPO			
TÉCNICAS	MEDIA	MEDIANA	MODA
EIGENFACES	8,5303226	6,57	0
FISHERFACES	2,0670968	1,65	1,94
LBPH	1,6448387	1,41	0,27

Una vez que se obtuvo el tiempo de acierto de cada estudiante por cada técnica (ver tabla 5) se realizó una comparación entre el tiempo de las técnicas, con la finalidad de identificar la eficiencia en el tiempo de respuesta. Siendo la técnica LBPH con una media de identificación de 1.64 segundos (ver tabla 6).

5.3. Discusión

En el presente proyecto de investigación se resumió de forma detallada los procesos para la ejecución de las técnicas, los resultados obtenidos mediante la ejecución se enfocaron en el tiempo de entrenamiento, reconocimiento y eficiencia de cada técnica con el respectivo registro de los datos obtenidos.

Seguido se realizó una comparativa de los datos obtenidos en la ejecución de las técnicas de reconocimiento facial desarrollada en este proyecto de investigación el mismo que nos muestra un resultado favorable para técnica LBPH utilizando Python y OpenCV.

5.4. Validación de la hipótesis

Después de haber realizado el respectivo proyecto de investigación en el cual se utilizó Python y OpenCV en el cual se emplearon tres distintas técnicas para el reconocimiento facial global que permitieron dar solución a la hipótesis planteada inicialmente, se pudo llegar a la conclusión que la técnica LBPH ha sido factible y eficiente para un reconocimiento facial. Por su tiempo de entrenamiento y reconocimiento menor a las dos técnicas restantes. Por lo tanto, se pudo demostrar que la investigación fue realizada y culminada satisfactoriamente.

5.5. Impacto económico

En el impacto económico se incluye la inversión generada, puesto que se requiere de componente específicos para la construcción del prototipo.

5.5.1. Gastos directos

Como parte de los gastos directos se tiene el conjunto de componentes utilizados que son necesarios para la elaboración del prototipo de reconocimiento facial en imágenes 2D, se considera oportuno utilizar componentes de calidad para el prototipo.

Para el costo del desarrollo del prototipo de reconocimiento facial, el tiempo de la realización fue de 60 días, con respecto al sueldo mínimos del programador es de \$426.20 por cada mes detallado por el Ministerio de trabajo, para realizar el respectivo cálculo se tomó como referencia 4 horas productivas de las 8 horas y 20 días al mes tal como se muestra en la tabla 6.

Tabla 7: Cálculo del costo de desarrollo del prototipo.

Costo de desarrollo del prototipo	
Cd (Costo Desarrollo)	Estimación en meses
S (Sueldo Programador)	$T_m = (T_d/N_p)/20$
N _p (Número de programadores)	$T_m = (60/2)/20$
T _d (Total de días)	$T_m = 20/20$
T _m (Tiempo en meses)	$T_m = 1,5$ meses
Og (Otros gastos)	
$Cd = (N_p * T_m * S) + Og$	
$Cd = (2 * 1 * 426,20) + 50$	
$Cd = 1328,60$ dólares Americanos	

5.6. Cuestionario del prototipo de reconocimiento facial

Marque con una "X" la respuesta que considere acertada con su punto de vista, según las siguientes alternativas:

Tabla 8: Significado de los índices de juicio de expertos.

TD	DA	MAD	ED	TDA
TOTALMENTE DE ACUERDO	DE ACUERDO	NI DE ACUERDO NI DESACUERDO	EN DESACUERDO	TOTALMENTE EN DESACUERDO

Tabla 9: Cuestionario para la evaluación de juicio de expertos.

N°	ÍTEMS	ÍNDICES				
		TD	DA	NAD	ED	TDA
1	Las imágenes obtenidas con la aplicación de las 3 técnicas son óptimas para el reconocimiento facial.	X				
2	Es necesario considerar las variaciones de luz en el reconocimiento facial.		X			
3	Al realizar el proceso de reconocimiento facial el prototipo funciona correctamente.	X				
4	Los rostros obtenidos se almacenan correctamente en una base de datos.	X				
5	Se actualiza en tiempo real las imágenes de rostros obtenidos por la cámara.	X				

6	Las técnicas utilizadas en el prototipo cumplen con los requisitos para validar esta investigación.	X				
7	El prototipo cumple con las expectativas del experto.	X				

Nombre del experto: Ing. Mg. Manuel Villa

Tabla 10: Cuestionario para la evaluación de juicio de expertos.

N°	ÍTEMS	ÍNDICES				
		TD	DA	NAD	ED	TDA
1	Las imágenes obtenidas con la aplicación de las 3 técnicas son óptimas para el reconocimiento facial.		X			
2	Es necesario considerar las variaciones de luz en el reconocimiento facial.	X				
3	Al realizar el proceso de reconocimiento facial el prototipo funciona correctamente.	X				
4	Los rostros obtenidos se almacenan correctamente en una base de datos.	X				
5	Se actualiza en tiempo real las imágenes de rostros obtenidos por la cámara.	X				
6	Las técnicas utilizadas en el prototipo cumplen con los requisitos para validar esta investigación.		X			
7	El prototipo cumple con las expectativas del experto.		X			

Nombre del experto: Ing. Mg. Mayra Albán

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- La revisión bibliográfica de varios libros, artículos científicos y páginas web relacionados a cerca de base de datos de imágenes 2D, extracción de características y algoritmos de reconocimientos de patrones fueron de vital importancia para el desarrollo de la fundamentación teórica y la determinación de las herramientas tecnológicas.
- Con el diseño del prototipo para el reconocimiento facial global utilizando Python y OpenCV, permitió la comparación de las técnicas utilizadas para determinar que técnica es la más eficiente.
- Mediante el aval de juicio de experto se determinó que el prototipo para el reconocimiento facial cumple con el funcionamiento correcto para la detección de rostros y en análisis de las técnicas empleadas.

6.2. Recomendaciones

- Es conveniente realizar la búsqueda bibliográfica en repositorios que garanticen la calidad de la información y también que la misma sea de actualidad para que la investigación sea exitosa.
- Es conveniente usar OpenCV en computadores con procesadores Intel debido a que estos aprovechan cada núcleo del procesador ayudando a la eficiencia del procesamiento de operaciones.
- Tener en cuenta la posibilidad de utilizar una cámara como mejor resolución y calidad debido a que en el periodo de pruebas se presentaron inconvenientes con respecto a la cámara.

7. BIBLIOGRAFÍA

- [1] C. E. Franco, C. T. Ospina, E. S. Cuevas, and D. V. Capacho, “RECONOCIMIENTO FACIAL BASADO EN EIGENFACES, LBHP Y FISHERFACES EN LA BEAGLEBOARD-xM,” *Rev. Colomb. Tecnol. Av.*, vol. 2, no. 26, 2017, doi: 10.24054/16927257.v26.n26.2015.2387.
- [2] J. González Astudillo and M. G. Zhindón Mora, “Plataforma de servicios de reconocimiento facial para detección de prófugos de la justicia en Ecuador,” *Rev. Cienc. e Investig.*, vol. 5, pp. 31–41, 2020.
- [3] M. Kulia and M. Kulia, “HERRAMIENTA DE RECONOCIMIENTO FACIAL CON TÉCNICA DE VISIÓN COMPUTACIONAL 2D,” no. April, pp. 33–35, 2019.
- [4] F. Serratos, “Biometria_ES_(Modulo_1),” p. 50, 2008.
- [5] J. J. I. CASANOVA, “RECONOCIMIENTO FACIAL EN AMBIENTES NO Autor : Asesor : Línea de Investigación : Infraestructura , Tecnología y Medio Ambiente,” 2020.
- [6] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” *Proceedings. 1991 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 586–591, doi: 10.1109/CVPR.1991.139758.
- [7] M. Centeno, “TECNICAS DE BIOMETRIA BASADAS EN PATRONES FACIALES DEL SER HUMANO,” *Экономика Региона*, p. 32, 2012.
- [8] A. Maisueche Cuadrado, “Utilización Del Machine Learning En La Industria 4.0,” pp. 1–118, 2019, [Online]. Available: <https://core.ac.uk/download/pdf/228074134.pdf#page=37&zoom=100,90,94>.
- [9] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, vol. 9781107057. 2013.
- [10] A. Moreno *et al.*, *Aprendizaje automático*. 1994.
- [11] D. Angluin, “Computational learning theory: Survey and selected bibliography,” *Proc. Annu. ACM Symp. Theory Comput.*, vol. Part F1297, pp. 351–369, 1992, doi: 10.1145/129712.129746.
- [12] P. F. De Máster, “Utilización de métodos de visión artificial para PC como apoyo en la automoción,” 2015.

- [13] “OpenCV: Clasificador en cascada.” https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html (accessed Jan. 15, 2022).
- [14] J. Ibarra-Estévez and K. Paredes, “Redes neuronales artificiales para el control de acceso basado en reconocimiento facial,” *Revistapuce*, 2018, doi: 10.26807/revpuce.v0i106.140.
- [15] S. Mullainathan and J. Spiess, “Machine learning: An applied econometric approach,” *J. Econ. Perspect.*, vol. 31, no. 2, pp. 87–106, 2017, doi: 10.1257/jep.31.2.87.
- [16] A. M. Llapapasca Montes and M. A. Ochoa Zevallos, “Creación de una librería de software de reconocimiento facial enfocado a la identificación de trabajadores de una empresa ’ Ingeniería de Software,” *Univ. Tecnológica del Perú*, 2019, [Online]. Available: <http://repositorio.utp.edu.pe/handle/UTP/2278>.
- [17] “Reconocimiento facial - EcuRed.” https://www.ecured.cu/Reconocimiento_facial (accessed Jan. 15, 2022).
- [18] T. D. F. De Grado, “VIDA EN EL RECONOCIMIENTO FACIAL Jaime Varela de la Escalera de Miguel,” 2019.
- [19] R. Gottumukkal and V. K. Asari, “System level design of real time face recognition architecture based on composite PCA,” *Proc. IEEE Gt. Lakes Symp. VLSI*, pp. 157–160, 2003, doi: 10.1145/764808.764849.
- [20] J. D. Alvarado and J. Fernández, “Análisis de textura en imágenes a escala de grises, utilizando patrones locales binarios (LBP),” 2012.
- [21] P. Álvarez, “Prototipo de sistema piloto para control de acceso basado en reconocimiento de rostros,” *Fac. Ing. Univ. Mil. Nueva Granada*, p. 72, 2013.
- [22] J. Galindo and A. Gamboa, “Control de Acceso a Archivos y Carpetas A Través del Reconocimiento Facial,” *UACISIS Edición 1*, vol. 0, no. 0, Oct. 2016, Accessed: Jan. 15, 2022. [Online]. Available: http://www.fuac.edu.co/recursos_web/documentos/ing.sistemas/ojs/index.php/UACISIS/article/view/27.
- [23] A. M. Martinez and A. C. Kak, “PCA versus LDA,” *IEEE Trans. Pattern Anal. Mach.*

- Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001, doi: 10.1109/34.908974.
- [24] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1064, no. 7, pp. 45–58, 1996, doi: 10.1007/bfb0015522.
- [25] I. Challenger, Y. Díaz, and R. B. García, “El lenguaje de programación Python/The programming language Python,” *Redalyc*, vol. XX, pp. 1–13, 2014, [Online]. Available: <https://www.redalyc.org/pdf/1815/181531232001.pdf>.
- [26] “La Biblioteca Estándar de Python — documentación de Python - 3.10.2.” <https://docs.python.org/es/3/library/index.html> (accessed Jan. 15, 2022).
- [27] “About - OpenCV.” <https://opencv.org/about/> (accessed Nov. 18, 2021).
- [28] “Replay-Attack — English.” <https://www.idiap.ch/en/dataset/replayattack> (accessed Nov. 18, 2021).
- [29] A. Mora, “Bases de datos. Diseño y gestión,” p. 199, 2014.
- [30] F. P. Isabel Rodríguez, Alexánder Ceballos, Jorge Hernández, “Disponible en: <http://www.redalyc.org/articulo.oa?id=75061112>,” *DISEÑO E IMPLEMENTACIÓN UNA BASE DATOS IMÁGENES RANGO CON DECIMACIÓN PARA LA Vis. EN WEB*, 2007.
- [31] D. Alan Neill and L. Cortez Suárez, *Procesos y fundamentos de la investigación científica*, vol. 53, no. 9. 2013.
- [32] G. Guerrero Dávila and M. Guerrero Dávila, “Metodología de la investigación,” pp. 9–9, 2014.
- [33] J. Murillo, “Asignatura: Métodos de investigación,” pp. 1–12, 2011, [Online]. Available: <https://d1wqtxts1xzle7.cloudfront.net/55568285/Experimental-with-cover-page-v2.pdf?Expires=1631141035&Signature=OkU~cybmdgDobOwDH1ARIdJY7CD57-5z3eU-Fg6exx579FcUm4MfOrSaiuUIofZObnU4gPeUGlwfyrctcHVk98UqrKR0Hog3AWno6x6yJibFFtGKSZLXF2adRExgPpdkBzdDcCDHieDUGx5D>.

- [34] “ALMACENANDO ROSTROS usando imágenes y video | Python - OpenCV » omes-va.com.” <https://omes-va.com/almacenando-rostros-usando-imagenes-y-video-python-opencv/> (accessed Feb. 16, 2022).
- [35] “DETECCIÓN DE ROSTROS con Haar Cascades Python – OpenCV » omes-va.com.” <https://omes-va.com/deteccion-de-rostros-con-haar-cascades-python-opencv/> (accessed Feb. 16, 2022).

8. ANEXOS

Anexo1: Datos personales de tutor y estudiantes

DATOS PERSONALES TUTOR ACADÉMICO

NOMBRES Y APELLIDOS: CADENA MOREANO JOSÉ AUGUSTO

DOCUMENTO DE IDENTIDAD: 0501552798

DIRECCIÓN: Latacunga, Cotopaxi.

TELÉFONO: 0984059929

E-MAIL: jose.cadena@utc.edu.ec

ESTUDIOS: INGENIERO EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES,
UNIVERSIDAD TÉCNICA DE COTOPAXI.

MAGISTER EN CIENCIAS DE LA EDUCACIÓN MENCIÓN PLANEAMIENTO Y
ADMINISTRACIÓN EDUCATIVA, UNIVERSIDAD TÉCNICA DE COTOPAXI.

DATOS PERSONALES ESTUDIANTE 1

NOMBRES Y APELLIDOS: CHACON GAVILANES LUIS FERNANDO

DOCUMENTO DE IDENTIDAD: 0504151440

DIRECCIÓN: Latacunga, Cotopaxi.

TELÉFONO: 0998333687

E-MAIL: luis.chacon1440@utc.edu.ec

ESTUDIOS: Bachillerato General, Unidad Educativa Vicente León

DATOS PERSONALES ESTUDIANTE 2

NOMBRES Y APELLIDOS: LARCO BORJA PABLO DAMIÁN

DOCUMENTO DE IDENTIDAD: 172185183-8

DIRECCIÓN: Sangolquí, Pichincha.

TELÉFONO: 0987423321

E-MAIL: pablo.larco1838@utc.edu.ec

ESTUDIOS: Bachillerato General, Unidad Educativa Mejía

Anexo2: Datos Personales de Expertos

DATOS PERSONALES EXPERTO 1

NOMBRES Y APELLIDOS: ALBAN TAIPE MAYRA SUSANA

DOCUMENTO DE IDENTIDAD: 050231198-8

DIRECCIÓN: Latacunga, Cotopaxi.

TELÉFONO: 0987773341

E-MAIL: mayra.alban@utc.edu.ec

ESTUDIOS: INGENIERA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES,
UNIVERSIDAD TÉCNICA DE COTOPAXI.

MAGISTER EN GESTIÓN DE LA PRODUCCIÓN, UNIVERSIDAD TÉCNICA DE
COTOPAXI.

DATOS PERSONALES EXPERTO 2

NOMBRES Y APELLIDOS: VILLA QUISHPE MANUEL WILLIAN

DOCUMENTO DE IDENTIDAD: 180338695-0

DIRECCIÓN: Latacunga, Cotopaxi.

TELÉFONO: 0983855980

E-MAIL: manuel.villa@utc.edu.ec

ESTUDIOS: INGENIERO EN SISTEMAS E INFORMÁTICA, UNIVERSIDAD
REGIONAL AUTÓNOMA DE LOS ANDES.

MAGISTER EN INTERCONECTIVIDAD DE REDES, ESCUELA SUPERIOR
POLITÉCNICA DE CHIMBORAZO.

Anexo3: Ficha Bibliográficas

Tabla 12: Ficha bibliográfica 1.

Ficha Bibliográfica 1	
Autor:	Luis Rodrigo Barba Guamán
Año de publicación:	2015
Título:	Utilización de métodos de visión artificial para PC como apoyo en la automoción.
Lugar y editorial:	Madrid/Universidad Técnica de Madrid

Tabla 13: Ficha bibliográfica 2.

Ficha Bibliográfica 2	
Autor:	Jefferson David Escobar Loayza
Año de publicación:	2019
Título:	Herramienta de reconocimiento facial con técnica de visión computacional 2d.
Lugar y editorial:	Quito/Udla

Tabla 14: Ficha bibliográfica 3.

Ficha Bibliográfica 3	
Autor:	Pablo Arturo Álvarez
Año de publicación:	2013
Título:	Prototipo de Sistema Piloto para Control de Acceso Basado en Reconocimiento de Rostros
Lugar y editorial:	Bogotá/Universidad Militar Nueva Granada

Tabla 15: Ficha bibliográfica 4.

Ficha Bibliográfica 4	
Autor:	Llapasca Montes Marco Antonio Ochoa Zevallos, Manuel Alexander
Año de publicación:	2019
Título:	Creación de una librería de software de reconocimiento facial enfocado a la identificación de trabajadores de una empresa
Lugar y editorial:	Lima/Universidad tecnología del Perú

Anexo4: Recolección de información

Tabla 16: Total aciertos por técnica.

RECONOCIMIENTO			
TÉCNICAS	SI	NO	TOTAL
EIGENFACES	28	3	31
FISHERFACES	31	0	31
LBPH	31	0	31




Figura 36: Recolección de fotografías

Anexo5. Permiso de derechos de autor de imágenes

Nosotros, Luis Fernando Chacon Gavilanes con C.I.: 0504151440 y Pablo Damián Larco Borja con C.I.: 1721851838, ser los autores del presente proyecto de Investigación:” TÉCNICA EFICIENTE PARA EL RECONOCIMIENTO FACIAL GLOBAL UTILIZANDO PYTHON Y OPENCV EN IMÁGENES 2D.”, siendo el Ing. Mg. José Augusto Cadena Moreano, tutor del presente trabajo, se nos autorice el uso de la imágenes tomadas para ser incluidas en esta investigación sin fines de lucro.

Atentamente,



.....

Chacon Gavilanes Luis Fernando

C.I.: 0504151440


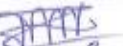




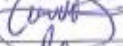

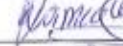

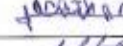





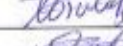







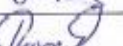


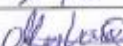
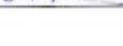



.....

Larco Borja Pablo Damián

C.I.: 1721851838

Tabla 17: Firmas de los estudiantes

No	Nombre	Apellido	curso	Paralelo	Carrera	Firma
1	Jessica	Abata	7mo	A	Sistemas de información	
2	Sindy	Caizaluisa	7mo	A	Sistemas de información	
3	Erick	Cañizares	8vo	A	Sistemas de información	
4	Andrés	Chacha	8vo	A	Sistemas de información	
5	Luis	Chacon	8vo	A	Sistemas de información	
6	Esteven	Chiluisa	7mo	A	Sistemas de información	
7	Cristian	Díaz	7mo	A	Sistemas de información	
8	Alex	Guañuna	6to	A	Sistemas de información	
9	Damián	Larco	8vo	A	Sistemas de información	
10	Leonel	Lema	7mo	A	Sistemas de información	
11	Jonathan	Lituma	8vo	A	Sistemas de información	
12	Erick	Lliguin	8vo	A	Sistemas de información	
13	Denis	Martínez	7mo	A	Sistemas de información	
14	Erick	Molina	7mo	A	Sistemas de información	
15	Ronaldo	Molina	7mo	A	Sistemas de información	
16	Jonathan	Montatixe	7mo	A	Sistemas de información	
17	Danner	Morales	7mo	A	Sistemas de información	
18	Jonathan	Oña	7mo	A	Sistemas de información	
19	Denis	Orellana	7mo	A	Sistemas de información	
20	Denis	Pacheco	8vo	A	Sistemas de información	
21	Jhon	Quinaucho	6to	A	Sistemas de información	
22	Bryan	Quispe	7mo	A	Sistemas de información	
23	David	Rea	6to	A	Sistemas de información	
24	Diego	Reinoso	7mo	A	Sistemas de información	
25	Wilson	Rojas	6to	A	Sistemas de información	
26	Pamela	Sarabia	6to	A	Sistemas de información	
27	Viviana	Telema	7mo	A	Sistemas de información	
28	Luis	Tutillo	9no	A	Sistemas de información	
29	Miguel	Vargas	7mo	A	Sistemas de información	

30	Daniela	Viturco	7mo	A	Sistemas de información	
31	Bryan	Yanez	8vo	A	Sistemas de información	