



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Apizaco

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

**ANÁLISIS Y CLASIFICACIÓN DE IMÁGENES DE ROSTROS
PARA LA PREDICCIÓN DE EMOCIONES.**

TITULACIÓN INTEGRAL TESIS

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

PRESENTA:

**LIZBETH MERLO HERNÁNDEZ
MARIA GUADALUPE ZECUA CAMACHO**

NOMBRE DEL ASESOR:

DR. JOSÉ CRISPÍN HERNÁNDEZ HERNÁNDEZ



APIZACO, TLAX., FEBRERO 2024

DEDICATORIA

Dedico este proyecto primordialmente a Dios y a la Santísima Virgen de Guadalupe, por darme la fuerza necesaria para superar todas las dificultades que se me presentaron durante este proceso de formación profesional, a mis padres por motivarme y brindarme su apoyo incondicional para lograr esta meta y acompañarme siempre en cada paso que doy. A los docentes por sus enseñanzas impartidas y su disposición para aclarar dudas, ayudándonos así a fortalecer nuestros conocimientos y culminar con éxito nuestra formación académica, el mismo que se ve reflejado mediante el desarrollo de este proyecto de titulación. A mi equipo de trabajo por los momentos compartidos y por el apoyo constante que me demostraron para no rendirme y estar firme en llegar a este objetivo.

Lizbeth Merlo Hernández

DEDICATORIA

A mis padres que han sabido guiarme por el camino del bien, forjando sentimientos, hábitos y valores que me han permitido llegar hasta este punto con su amor, paciencia y sobre todo fortaleza. A mis hermanos por su cariño, tiempo, paciencia y apoyo incondicional durante toda mi estancia en mi formación académica. A mi tío que desde el cielo me ilumina para seguir adelante con cada reto que se me presente. A mis tías que han estado en las buenas y malas, siempre sostenido mis manos y animarme a no rendirme, A nuestro asesor interno quien nos ha guiado durante toda esta travesía e impulsado, por todos los momentos compartidos y los conocimientos adquiridos. A nuestros docentes que siempre nos brindaron apoyo, motivado y todas aquellas veces que se tomaron el tiempo de extender nuestros conocimientos.

María Guadalupe Zecua Camacho

AGRADECIMIENTOS

A Dios

Agradezco principalmente a Dios por permitirme vivir momentos gratos y agradables a lo largo de mi niñez, de mi juventud y por cuidarme a lo largo de mi carrera universitaria dándome las fuerzas y voluntad para poder seguir preparándome.

A mis padres

No podría haber llegado a este momento de mi vida sin el apoyo constante de mis padres. A lo largo de mi vida, siempre estuvieron ahí para animarme, para impulsarme a seguir adelante y para creer en mí, incluso cuando yo mismo no lo hacía. Ahora, al final de mi camino de estudios, no puedo sino expresarles mi profundo agradecimiento. Sin su ayuda, no habría podido superar los obstáculos que se me presentaron a lo largo de mi camino. Es gracias a su amor y a su apoyo que he podido llegar hasta aquí. No tengo palabras suficientes para agradecerles todo lo que han hecho por mí, pero espero que estas pocas palabras sean un pequeño comienzo.

A mi abuelo

También se la dedico a mi abuelito Miguel Merlo Mistral, desde el cielo eres esa luz que me daba fuerzas para continuar. Hoy no estás para poder entregarte este logro que también es tuyo, gracias porque tus frutos de esfuerzo hoy también forman parte del mío.

A mi tutora

A mi profesora y guía, M.C. María Lorena Roldán Flores, le expreso mi gratitud por su apoyo constante, sabiduría y orientación en la realización de esta tesis. Su pasión por la enseñanza y su habilidad para transmitir conocimientos de manera clara y efectiva han sido fundamentales para mi crecimiento académico. Gracias por ser un faro de luz en mi camino hacia la excelencia en la investigación. Su compromiso con la excelencia académica y su habilidad para desafiar e inspirar a sus estudiantes han dejado una huella indeleble en mi vida. Gracias por marcar la diferencia en mi educación y carrera.

A nuestro asesor interno

Deseo expresar mi más sincero agradecimiento a mi profesor, Dr. José Crispín Hernández Hernández, por su apoyo, orientación y estímulo en la realización de esta tesis. Su entusiasmo por la enseñanza y su habilidad para inculcar la curiosidad en sus estudiantes me han motivado a perseverar en mi búsqueda del conocimiento. Gracias por ser una fuente constante de inspiración y sabiduría.

A Lic. María del Rocío Ojeda López

Le agradezco muy profundamente a la Lic. María del Rocío por su dedicación y paciencia, sin sus palabras y correcciones precisas no habiésemos podido lograr llegar a esta instancia tan anhelada.

A todos mis docentes

Son muchos los docentes que han sido parte de mi camino universitario, y a todos ellos les quiero agradecer por transmitirme los conocimientos necesarios para hoy poder estar aquí. Sin ustedes los conceptos serían solo palabras, y las palabras ya sabemos quién se las lleva, el viento.

Además, a mis compañeros

Agradecerles a todos mis compañeros los cuales muchos de ellos se han convertido en mis amigos, cómplices y hermanos. Gracias por las horas compartidas, los trabajos realizados en conjunto y las historias vividas.

Gracias a la casa de estudios

Por último, agradecer al Tecnológico Nacional de México / Instituto Tecnológico de Apizaco que me ha exigido tanto, pero al mismo tiempo me ha permitido obtener mi tan ansiado título. Agradezco a cada directivo por su trabajo y por su gestión, sin lo cual no estarían las bases ni las condiciones para aprender conocimientos.

Lizbeth Merlo Hernández

AGRADECIMIENTOS

A Dios

El principal agradecimiento es a Dios que me ha guiado y fortalecido en todo momento, así mismo al espíritu santo que ha derramado fortaleza y valentía, los cuales me permitieron llegar hasta este momento, con experiencias y enseñanzas que me guiarán durante mi vida profesional, así mismo agradezco porque gracias a Dios y al espíritu santo estoy terminado de una manera satisfactoria de mi persona.

A mi familia

Con su comprensión y estímulo para llegar a este punto de mi vida, porque a pesar de todo me fortalecieron día a día, forjándome en el camino correcto y de bien, sin todo su apoyo y ánimo no estaría hasta este punto de mi vida, estoy eternamente agradecida porque cuando yo no me creía capaz, ustedes mi familia sacaron a relucir lo mejor de mí y me convencieron de seguir el camino profesional con dedicación y entusiasmo, pero sobre todo por su amor incondicional que me abrigaba en mis momentos más difíciles, por su tiempo al escuchar mis problemas académicos y ayudarme a resolverlos, por siempre creer en mí, cuando más lo necesitaba y darme la confianza de que podría lograr mis metas.

A nuestro asesor interno

Quien me ha permitido desarrollarme he impulsarme, desde el día uno, le agradezco Dr. Crispín Hernández Hernández no solo por formar parte de mi vida estudiantil, sino también por animarme y creer en mí para realizar esta tesis, por ser mi mentor y guía, por sus pláticas constructivas que impactaron y se ven reflejadas en este trabajo, pero sobre todo su paciencia y dedicación hacia mi persona para que tuviera la oportunidad de titularme mediante esta tesis.

A la Lic. Roció Ojeda López

Le agradezco sinceramente por formar parte de este proceso, impulsándonos y acompañándonos en el camino administrativo y en la tesis, por otro lado, dándonos palabras de aliento y consejos de vida, que nos permitieron llegar hasta este punto, pero sobre todo por su valioso tiempo que nos dedicó.

A mis docentes

Quienes formaron parte de mis estudios académicos, apoyándome, escuchándome asesorándome, con sus conocimientos y me forjaron con las habilidades que hoy poseo, y es por ello que hoy en día estos hasta este punto de mi vida académica.

María Guadalupe Zecua Camacho

RESUMEN

La Tecnología es el pilar fundamental en la vida de muchas personas en la actualidad. Probablemente, de la mayoría de nosotros. Consciente o inconscientemente, vivimos rodeados de tecnología y ésta nos ayuda y nos facilita la vida de diversos modos. Las Tecnologías de la Información y la Comunicación (**TIC's**) son consideradas como unas herramientas que permiten el fácil acceso a la información y a su vez, garantizan un aprendizaje didáctico con plena adquisición de conocimientos para el mejor desenvolvimiento del ser humano en su entorno. La importancia de las TIC's es obvia cuando hablamos de que éstas son un conjunto de técnicas que permiten procesar la información utilizando la Informática, la Microelectrónica y las Telecomunicaciones. De esta forma, se crea un canal de comunicación novedoso y beneficioso en muchos aspectos.

La Inteligencia Artificial (IA) surge como una de las ramas de informática combinada con áreas de la lógica y las ciencias cognitivas. El desarrollo de esta tecnología se aplica desde hace tiempo en multitud de campos dentro de la empresa e incluso en nuestra vida más cotidiana. Cualquier ámbito de la sociedad es viable para que se utilice en él la IA. Al día de hoy, la Inteligencia Artificial se usa en cuestiones como el procesamiento de lenguaje natural, la automatización de procesos industriales o incluso la selección de contenidos.

Los diferentes tipos de IA ya son parte de procesos de toda clase. Muchas de las tareas diarias que realizamos han adoptado algún tipo de Inteligencia Artificial y aunque nos pueda resultar inquietante o imposible, la echaríamos de menos si desapareciese.

ABSTRACT

In the present thesis work we work on the recognition of emotions from the human speech signal. To tackle this problem we have adopted a psychological model of continuous emotions. It is envisaged that with this emotion model we study the analysis and classification of emotions in a broader sense of people as compared to what has been traditionally done as psychological studies of emotions. The facial acoustic space has been thoroughly explored in realistic scenarios. We applied human-computer interaction and probabilistic artificial intelligence techniques to estimate emotional states. This work contributes to the understanding of elements of human language that help determine emotions and to create a pattern recognition method based on a continuous emotional model suitable for spontaneous emotion detection. We experimented with various types of facial acoustic reflex features and easy response mechanisms using human language. We had used feature selection techniques to find the most important ones. In addition, we studied the importance of these attributes in facial expression data, proposed a method to represent emotional states based on fuzzy clustering, and studied the prediction of emotional states as a function of context. The results obtained in the estimation of emotion primitives, analysis and classification of basic emotions using our set of features and methods are comparable with the best state-of-the-art results already understood above.

PALABRAS CLAVE

- 1. Análisis**
- 2. Clasificación**
- 3. Datos**
- 4. Minería de datos**
- 5. Lenguaje**
- 6. Librerías**
- 7. Procesamiento**
- 8. Lematización**
- 9. Derivación**
- 10. Aprendizaje**
- 11. Información**
- 12. Algoritmo**
- 13. Extracción**
- 14. Implementación**
- 15. Modelos**
- 16. Parámetros**
- 17. Función**
- 18. Detección**
- 19. Técnica**
- 20. Mecanismo**

ÍNDICE

DEDICATORIA	I
DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN	VI
ABSTRACT.....	VII
PALABRAS CLAVE	VIII
ÍNDICE	IX
CAPITULO I. GENERALIDADES DEL PROYECTO.....	1
1.1 INTRODUCCIÓN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	2
1.3 PREGUNTA DE INVESTIGACIÓN.....	2
1.4 OBJETIVOS.	2
1.4.1 OBJETIVO GENERAL.	2
1.4.2 OBJETIVOS ESPECÍFICOS.....	2
1.5 JUSTIFICACIÓN.....	2
1.6 ALCANCES.	3
1.7 LIMITACIONES.	3
1.8 ORGANIZACIÓN DE TESIS.....	3
CAPITULO II. MARCO TEÓRICO.	5
2.1 ANÁLISIS DE EXPRESIONES FACIAL MEDIANTE TÉCNICAS DE IA: ESTADO DEL ARTE, TAXONOMÍAS Y RETOS.	5
2.2 NECESIDAD DE DETECCIÓN FACIAL, REDUCCIÓN DE DIMENSIONES Y NORMALIZACIÓN...7	
2.3 NECESIDAD DE EXTRACCIÓN DE CARACTERÍSTICAS.....	8
2.4 NECESIDAD DE CLASIFICACIÓN DE EMOCIONES.....	8
2.5 MÁQUINA DE VECTORES SOPORTE (SVM).....	9
2.6 RED NEURONAL ARTIFICIAL (RNA).....	11
2.7 DEFINICIÓN DE TÉRMINOS.	11
2.7.1 LIBRERÍA NUMPY.....	11
2.7.2 LIBRERÍA MATPLOTLIB.	12
2.7.3 BIBLIOTECA CV2.....	12
2.7.4 MÓDULO OS.....	13
2.7.5 FUNCIÓN IMUTILS.	14
2.7.6 TRADUCCIÓN DE IMÁGENES.	14
2.7.7 BIBLIOTECA TIME.	14
2.8 ALGORITMO DE CLASIFICACIÓN.....	14
2.8.4 HAAR CASCADA.....	15
	IX

2.8.5	FASTAI	18
2.8.6	RASGOS FACIALES CON DLIB, OPENCV Y PYTHON	21
CAPITULO III. INSTALACIÓN DE REQUERIMIENTOS		28
3.1	INSTALACIÓN DE PAQUETERÍAS	28
3.2	IMPLEMENTACIÓN DE SISTEMA.....	28
3.3	CREACIÓN DE LA BASE DE DATOS CON LOS ROSTROS QUE REFLEJAN LAS EMOCIONES. 29	
3.3.1	CAPTURACIÓN DE ROSTROS “Felicidad”	29
3.3.2	CAPTURACIÓN DE ROSTROS “Enojo”.....	30
3.3.3	CAPTURACIÓN DE ROSTROS “Sorpresa”	32
3.3.4	CAPTURACIÓN DE ROSTROS “Tristeza”	34
3.3.5	CAPTURACIÓN DE ROSTROS “Vergüenza”	35
3.4	ENTRENANDO.....	36
3.5	PROBANDO EIGENFACES, FISHERFACES Y LBPH PARA EL RECONOCIMIENTO DE EMOCIONES FACIALES	37
3.6	PRIMERAS PRUEBAS DE RECONOCIMIENTO	38
CAPITULO IV. RESULTADOS		40
4.1.	MODELO PROPUESTO.....	40
4.2.	IMPORTACIÓN DE PAQUETES	40
4.3.	CONFIGURACIÓN DE PARÁMETROS Y DEFINICIÓN DE ETIQUETAS	41
4.4.	CARGAS DE DATOS, CLASIFICADOR Y ALGORITMOS DE RECONOCIMIENTO DE EMOCIONES	41
4.5.	IMPRESIÓN DE PREDICCIONES DE RECOLECCIÓN	42
4.6.	RASGOS FACIALES CON DLIB, OPENCV Y PYTHON PARA LA CLASIFICACIÓN DE EMOCIONES	47
4.7.	CLASIFICADOR CON RED NEURONAL DE APRENDIZAJE PROFUNDO	48
4.8.	RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE HAAR CASCADAS	51
4.9.	RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE RASGOS FACIALES	52
4.10.	RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE FASTAI-RED NEURONAL DE APRENDIZAJE PROFUNDO	54
CAPITULO V. CONCLUSIONES.....		55
COMPETENCIAS DESARROLLADAS		57
1.	INSTRUMENTALES	57
2.	INTERPERSONALES.....	57
3.	COMPETENCIAS SISTÉMICAS	57
FUENTES DE INFORMACIÓN		58
ANEXOS		61
ANEXO A.....		61
	SISTEMA INTERACTIVO	61
ANEXO B.....		63
	TUTORIAL 1	63

TUTORIAL 2	63
ANEXO C	64
EXPERIMENTO 1: THE ORL DATABASE OF FACES.....	64

ÍNDICE DE FIGURAS

FIGURA 2.1 VERSIÓN NUMPY EN EL EQUIPO.....	12
FIGURA 2.2 VERSIÓN MATPLOTLIB EN EL EQUIPO	12
FIGURA 2.3 VERSIÓN DE OPENCV EN LA TERMINAL DEL SISTEMA OPERATIVO.....	13
FIGURA 2.4 VERSIÓN OS DE WINDOWS.....	13
FIGURA 2.5 VERSIÓN IMUTILS.....	14
FIGURA 2.6 COMPORTAMIENTO DE LA BIBLIOTECA TIME	14
FIGURA 3.1 INSTALACIÓN DE PAQUETERÍAS.....	28
FIGURA 3.2 ESQUEMA DE GENERACIÓN DE CARPETAS DEL CONTENIDO DE RECONOCIMIENTO DE EMOCIONES	28
FIGURA 3.3 CREACIÓN Y ALMACENAMIENTO DE BASE DE DATOS POR EMOCIONES	29
FIGURA 3.4 RESULTADOS DE LA CAPTACIÓN DE ROSTROS GUARDADOS AUTOMÁTICAMENTE DE LAS TOMAS PARA LAS CAPTURAS DE "FELICIDAD"	29
FIGURA 3.5 CARPETA "FELICIDAD" CREADA PARA EL ALMACENAMIENTO DE DATOS	29
FIGURA 3.6 EJECUCIÓN DE GIF "FELICIDAD"	30
FIGURA 3.7 ALMACENAMIENTO DE DATOS EN LA CARPETA "FELICIDAD"	30
FIGURA 3.8 APERTURA DE CARPETA DE RECONOCIMIENTO DE EMOCIONES MEDIANTE EL CMD..	31
FIGURA 3.9 RESULTADOS DE LA CAPTACIÓN DE ROSTROS GUARDADOS AUTOMÁTICAMENTE DE LAS TOMAS.....	31
FIGURA 3.10 EJECUCIÓN DE GIF "ENOJO"	31
FIGURA 3.11 ALMACENAMIENTO DE DATOS EN LA CARPETA "ENOJO"	32
FIGURA 3.12 APERTURA DE LA CARPETA DE RECONOCIMIENTO DE EMOCIÓN "SORPRESA"	32
FIGURA 3.13 RESULTADO DE CREACIÓN DE LA CARPETA	32
FIGURA 3.14 EJECUCIÓN DE GIF "SORPRESA"	33
FIGURA 3.15. RECOLECCIÓN DE DATOS A LA CARPETA.....	33
FIGURA 3.16. APERTURA DE LA CARPETA DE RECONOCIMIENTO DE EMOCIÓN "TRISTEZA"	34
FIGURA 3.17. RESULTADO DE CREACIÓN DE LA CARPETA	34
FIGURA 3.18. EJECUCIÓN DE GIF "TRISTEZA"	34
FIGURA 3.19. RECOLECCIÓN DE DATOS A LA CARPETA	35
FIGURA 3.20. APERTURA DE LA CARPETA DE RECONOCIMIENTO DE EMOCIÓN "SORPRESA"	35
FIGURA 3.21. CREACIÓN Y RECOLECCIÓN DE DATOS DE LA EMOCIÓN "VERGÜENZA"	35
FIGURA 3.22. RESULTADO DE CREACIÓN DE LA CARPETA	36
FIGURA 3.23. RECOLECCIÓN DE DATOS A LA CARPETA "VERGÜENZA"	36
FIGURA 3.24. MODELO DE ENTRENAMIENTO "ENTRENANDO.PY"	36
FIGURA 3.25. MODELO DE ENTRENAMIENTO "EIGENFACES, FISHERFACES, LBPH"	37
FIGURA 3.26. VISUALIZACIÓN DE DOCUMENTOS XML DE LOS MODELOS CREADOS	37
FIGURA 3.27. PARÁMETROS DE FUNCIONAMIENTO PARA LOS FOTOGRAMAS EN FUNCIONAMIENTO	38

FIGURA 3.28. LECTURA DEL MODELO Y LLAMADO DE EMOCIONES.....	38
FIGURA 3.29. PRIMERA PRUEBA DE EJECUCIÓN DE RECONOCIMIENTO DE EMOCIONES	38
FIGURA 3.30. SEGUNDA PRUEBA DE EJECUCIÓN DE RECONOCIMIENTO DE EMOCIONES	39
FIGURA 3.31. CAPTACIÓN DE ROSTROS NO IDENTIFICADOS, AL NO TENER REGISTRO EN EL ENTRENAMIENTO	39
FIGURA 4.1 IMPORTACIÓN DE PAQUETES	40
FIGURA 4.2. CONFIGURACIÓN DE PARÁMETROS Y DEFINICIÓN DE ETIQUETAS	41
FIGURA 4.3. CARGA DE DATOS, DE MODELO Y DE PREDICCIONES	41
FIGURA 4.4. RESULTADO DE CARGAS	42
FIGURA 4.5. IMPRESIONES CLASES REALES, DE PREDICCIÓN E IMPRESIÓN DE PREDICCIONES REPORTE FINAL	42
FIGURA 4.6. SEGUNDA PARTE DE LA IMPRESIÓN DE PREDICCIONES	43
FIGURA 4.7. TERCERA PARTE DE LA IMPRESIÓN DE PREDICCIONES	43
FIGURA 4.8. CUARTA PARTE DE LA IMPRESIÓN DE PREDICCIONES	43
FIGURA 4.9. QUINTA PARTE DE LA IMPRESIÓN DE PREDICCIONES	44
FIGURA 4.10. SEXTA PARTE DE LA IMPRESIÓN DE PREDICCIONES	44
FIGURA 4.11. SÉPTIMA PARTE DE LA IMPRESIÓN DE PREDICCIONES	44
FIGURA 4.12. OCTAVA PARTE DE LA IMPRESIÓN DE PREDICCIONES	45
FIGURA 4.13. NOVENA PARTE DE LA IMPRESIÓN DE PREDICCIONES	45
FIGURA 4.14. DÉCIMA PARTE DE LA IMPRESIÓN DE PREDICCIONES	45
FIGURA 4.15. ÚLTIMO REPORTE DE IMPRESIÓN DE PREDICCIONES	46
FIGURA 4.16. REPORTE DE MÉTRICAS Y CLASIFICACIÓN DE CLASES REALES Y CLASES DE PREDICCIÓN	46
FIGURA 4.17. PUNTOS DE REFERENCIA FACIALES CON DLIB, OPENCV Y PYTHON	47
FIGURA 4.18. VISUALIZACIÓN DE LAS 68 COORDENADAS DE PUNTOS DE REFERENCIA	48
FIGURA 4.19. DISTRIBUCIÓN DE LAS EMOCIONES EN LAS IMÁGENES DE APRENDIZAJE Y PRUEBA PÚBLICOS Y PRIVADOS	49
FIGURA 4.20. RESULTADOS DE CLASIFICACIÓN	49
FIGURA 4.21 RESULTADOS OBTENIDOS POR MATRIZ DE CONFUSIÓN	50
FIGURA 4.22 RESULTADOS OBTENIDOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE HAAR CASCADAS	51
FIGURA 4.23 RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE RASGOS FACIALES (LIZBETH)	52
FIGURA 4.24 . RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE RASGOS FACIALES (GUADALUPE)	53
FIGURA 4.25 RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO FASTAI-RED NEURONAL DE APRENDIZAJE PROFUNDO(LIZBETH)	54

<i>FIGURA 4.26 . RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO FASTAI-RED NEURONAL DE APRENDIZAJE PROFUNDO(GUADALUPE).....</i>	<i>54</i>
<i>FIGURA 1 ANEXO A DIAGRAMA DE FLUJO DEL FUNCIONAMIENTO DEL SISTEMA INTERACTIVO</i>	<i>61</i>
<i>FIGURA 2 ANEXO A. MATRIZ DE CONFUSIÓN DEL SISTEMA INTERACTIVO.....</i>	<i>62</i>
<i>FIGURA 3 ANEXO C.....</i>	<i>64</i>

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 2.1 MARCO DE COMPORTAMIENTO	15
ILUSTRACIÓN 2.2 CARACTERÍSTICAS DEL MODELO HAAR CASCADA	16
ILUSTRACIÓN 2.3 MULTIPLICACIÓN DE CONVOLUCIÓN	17
ILUSTRACIÓN 2.4 . CUADRO DE CARACTERÍSTICAS DE HAAR	17
ILUSTRACIÓN 2.5 JERARQUÍA DE API	19
ILUSTRACIÓN 2.6 JERARQUÍA API	20
ILUSTRACIÓN 2.7 PUNTOS DE REFERENCIA FACIALES	26
ILUSTRACIÓN 4.1 FUNCIONAMIENTO DEL MODELO DE IDENTIFICACIÓN FACIAL	40

CAPITULO I. GENERALIDADES DEL PROYECTO.

1.1 INTRODUCCIÓN.

Es necesario conocer las necesidades, demandas y usos de información que tienen la variedad de los usuarios en las TI, sobre todo enfocado en dicho proyecto de tesis integral. En este caso en las emociones de personas aplicando la inteligencia artificial, se llevará a cabo el proyecto de análisis y clasificación de imágenes de rostros para la predicción de emociones.

Si bien se sabe la ciencia de datos es un área de trabajo interdisciplinar que incluye procesos para recopilar, preparar, analizar, visualizar y modelizar datos que permitan generar conocimiento útil para comprender problemas complejos y ayudar en la toma de decisiones. Estos datos con frecuencia son no estructurados y heterogéneos. En muchas ocasiones, se trata de grandes volúmenes de datos que por su complejidad y diversidad requieren de arquitecturas y técnicas innovadoras para extraer conocimiento relevante como lo es el conocido Big Data. El estudio de la ciencia de datos es un campo emergente con una alta aplicabilidad en ciencias de la salud, marketing, negocios, mercados financieros, transporte, comunicaciones, redes sociales, etc. Cabe mencionar que la inteligencia artificial forma parte de corazón de la ciencia de datos. El uso de técnicas inteligentes de análisis de datos ha propiciado el desarrollo de tecnologías que están siendo aplicadas de forma exitosa para la extracción de conocimiento y que junto con la disponibilidad de datos masivos han hecho que la ciencia de datos y las tecnologías de Big Data alcancen un peso muy importante entre las tecnologías fundamentales de la transformación digital que está acelerando la transformación de la sociedad y economía basada en el conocimiento extractor. Por otro lado, la minería de datos es el proceso de detectar la información procesable de los conjuntos grandes de datos, utilizando el análisis matemático para deducir los patrones y tendencias que existen en los datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque hay demasiados datos.

1.2 PLANTEAMIENTO DEL PROBLEMA.

En los últimos años, el reconocimiento de emociones faciales (FER, Face Expression Recognition) incremento su interés de estudio en diversas disciplinas, por citar: la industria, la medicina, la psicología cognitiva, y en especial, en la computación. A pesar de existir nuevas técnicas de aprendizaje automático y de aprendizaje profundo que abordan dicho FER, el medir las emociones es una tarea difícil en ambientes de la vida real, en expresiones débilmente posadas, y los atributos faciales.

1.3 PREGUNTA DE INVESTIGACIÓN.

¿Cuál de los algoritmos: Haar Cascada, Fast.ai o dlib es el idóneo para la clasificación de emociones?

1.4 OBJETIVOS.

1.4.1 OBJETIVO GENERAL.

Analizar y clasificar las emociones en rostros de personas presentes ante diversas situaciones.

1.4.2 OBJETIVOS ESPECÍFICOS.

- Identificar el área del rostro
- Identificar rasgos en el rostro
- Analizar expresiones faciales
- Clasificar automáticamente expresiones faciales
- Predecir las emociones fáciles básicas: felicidad, tristeza, enojo, asco, miedo, sorpresa, y neutro

1.5 JUSTIFICACIÓN.

Las expresiones faciales son usadas por los humanos para comunicarse, expresarse e interactuar con los de su mismo género. El estudio de las expresiones faciales cobra mayor relevancia cuando se identifican: en la interacción con otros individuos, el desenvolvimiento en ambientes controlados o como se presentan en determinadas situaciones. Mediante

expresiones faciales es posible mostrar estados emociones específicos, aportando de esta manera información para el análisis y clasificación de las emociones básicas.

Este trabajo es relevante ya se busca analizar y clasificar emociones en imágenes de rostros que se presentan cuando se reacciona a una situación en particular. Se podrá utilizar para identificar reacciones inadvertidas al momento de realizar compras, seguir cursos en línea, entrevistas u otros usos.

1.6 ALCANCES.

El alcance de este proyecto está directamente centrado al análisis, la clasificación emociones en rostros, y sobre todo sistemáticamente a la predicción de emociones para nuevas imágenes de rostros.

1.7 LIMITACIONES.

Debido al surgimiento de nuevos algoritmos que requieren de un equipo especializado (con GPU integrados), teniendo en cuenta esto se buscó alternativas de solución, se llegó a la determinación del uso de Google COLAB que trabaja directamente en la nube. Desafortunadamente, los tiempos de procesamiento son de uso compartido, y por lo mismo se está sujeto a disponibilidad de recursos.

1.8 ORGANIZACIÓN DE TESIS.

En el presente proyecto de Tesis se pretende alcanzar el análisis y clasificación de expresiones faciales de rostros para la clasificación de emociones de personas, en el cual se utilizará ciencia de los datos y técnicas de visión por computadora. mediante el estudio del arte del reconocimiento automático de emociones, de igual manera se usará minería de datos y aprendizaje automático, se procederá a la preparación de imágenes y el análisis de imágenes en donde se llevará el reconocimiento rostros, así mismo se hará la extracción de rasgos faciales que se realizará en fases: supervisados, y semi supervisados, por otra parte, se aplicarán algoritmos de reconocimiento de emociones y se identificarán emociones ocultas en personas, en base a ello la tesis se construye en cuatro capítulos.

Capítulo I se constituye de las generalidades del proyecto, es decir, introducción, planteamiento del problema, pregunta de investigación, objetivos generales y específicos, justificación, alcances y sus limitaciones.

Capítulo II, es la parte mas extensa del desarrollo ya que en este se describen las diferentes investigaciones el cual respaldara la validez y confiabilidad del estudio, tales como: Análisis del sentimiento facial mediante técnicas de IA: Estado del arte y taxionomas, Necesidad de detección facial, reducción de dimensiones y normalización, Necesidad de extracción de características, Necesidad de clasificación de emociones, Máquina de vectores soporte (SVM), Red neuronal artificial (RNA) y definición del término o librerías.

Capítulo III se implementan las instalaciones de requerimiento, de los cuales son: Instalación de paqueterías, implementación de sistema, creación de la base de datos con los rostros que reflejan las emociones, entrenando, probando Eigenfaces, Fisherfaces y LBPH para el reconocimiento de emociones faciales y primeras pruebas de reconocimiento.

Capítulo IV En esta parte se plantean los resultados obtenidos a lo largo del proyecto de la tesis, es decir, los Modelos propuestos, Importación de paquetes HAARCASCADA, Configuración de parámetros y definición de etiquetas, Cargas de extracción de datos, Impresión de predicciones de recolección, Rasgos faciales con DLIB, OPENCV Y PYTHON para la clasificación de emociones, Clasificador de red neuronal de aprendizaje profundo y resultados de las predicciones usando el Algoritmo de haar cascadas, Algoritmo de rasgos faciales y Algoritmo de fastai-red neuronal de aprendizaje profundo.

Capitulo V es en donde se constituye la tesis, se dará una opinión cercana a la investigación, la importancia del estudio y la demostración de como los hallazgos constituyen a este proyecto.

CAPITULO II. MARCO TEÓRICO.

2.1 ANÁLISIS DE EXPRESIONES FACIAL MEDIANTE TÉCNICAS DE IA: ESTADO DEL ARTE, TAXONOMÍAS Y RETOS.

Las emociones son eficaces y auto explicativas en las interacciones humanas cotidianas. La emoción humana más notable se manifiesta a través de las expresiones faciales. El Reconocimiento de Expresiones Faciales (FER) es bastante complejo y tedioso, pero ayuda en diversas áreas de aplicación, como la asistencia sanitaria, los robots emocionales y la interacción persona-ordenador. Aunque los avances en FER aumentan su eficacia, lograr una alta precisión sigue siendo una tarea difícil. Las seis emociones más genéricas de un ser humano son la ira, la felicidad, la tristeza, el asco, el miedo y la sorpresa. Además, la emoción llamada desprecio se añadió como una de las emociones básicas.

La EER es una tarea desconcertante y su precisión depende completamente de los parámetros seleccionados, como los factores de iluminación, la oclusión, es decir, la obstrucción en la cara como la mano, la edad y las gafas de sol. Los investigadores de este campo tienen en cuenta estos parámetros a la hora de crear sus modelos de EERR para conseguir una precisión considerable. A continuación, se describen algunos factores importantes para la EERR.

- **Factor de iluminación:** La intensidad de la luz que incide sobre el objeto afecta a la clasificación del modelo. Los valores de textura aumentan la tasa de falsa aceptación, ya sea minimizando la distancia entre clases o aumentando el contraste.
- **Intensidad de la expresión:** El reconocimiento de expresiones depende en gran medida de la intensidad de la expresión. La expresión se reconoce con mayor precisión cuando la expresión es menos sutil. Esto afecta en gran medida a la precisión del modelo.
- **Oclusión:** Si hay oclusión en una imagen, resulta difícil para el modelo extraer características de la parte ocluida debido a una alineación imprecisa de la cara y a una localización imprecisa de las características. También introduce ruido en los valores atípicos y en las características extraídas.

Los sistemas FER pueden ser estáticos o dinámicos en función de la imagen. La EER estática sólo tiene en cuenta la información de localización de los puntos faciales a partir de

la representación de características de una única imagen, mientras que la EER de imagen dinámica tiene en cuenta la información temporal con fotogramas continuos.

- **Conjunto de datos:** Para evitar el sobreajuste, se discuten los siguientes algoritmos de EERR, que necesitan amplios datos de entrenamiento. Para probar, entrenar y validar los algoritmos para el desarrollo de la EERR es esencial disponer de un conjunto de datos con etiquetas de emoción de expresión facial bien definidas. Estos conjuntos de datos contienen una secuencia de imágenes con emociones distintas, como se ha mencionado anteriormente. Hemos revisado muchos conjuntos de datos para entrenar diferentes modelos para obtener beneficios en el mundo real.
- **Preprocesamiento:** En este paso se preprocesa el conjunto de datos eliminando el ruido y comprimiendo los datos. Los distintos pasos del preprocesamiento de datos son los siguientes:
 - I. **La detección facial:** Es el poder de detectar la ubicación de la cara en cualquier imagen o fotograma. A menudo se considera un caso especial de la detección de clase de objeto, que determina si la cara está presente en una imagen o no,
 - II. **La reducción de dimensión:** Se utiliza para reducir las variables mediante un conjunto de variables principales. Si el número de características es mayor, resulta más difícil visualizar el conjunto de entrenamiento y trabajar con él. En este caso, se pueden utilizar PCA y LDA para manejar la situación mencionada.
 - III. **Normalización:** También se conoce como escalado de características. Tras el paso de reducción de dimensión, las características reducidas se normalizan sin distorsionar las diferencias en el rango de valores de las características. Existen varios métodos de normalización, como la Normalización Z, la Normalización Min-Max o la Normalización Vectorial Unitaria, que mejoran la estabilidad numérica y aceleran el entrenamiento del modelo.
- **Extracción de características:** Es el proceso de extracción de características que son importantes para la EERR. Se obtienen conjuntos de atributos más pequeños y ricos que contienen características como los bordes de la cara, las esquinas, la diagonal y otra información importante como la distancia entre los labios y los ojos,

la distancia entre dos ojos, que ayuda a acelerar el aprendizaje de los datos entrenados.

- **Clasificación de emociones:** Se trata de algoritmos que clasifican las emociones en función de las características extraídas. La clasificación tiene varios métodos, que clasifican las imágenes en varias clases. La clasificación de una imagen FER se lleva a cabo después de pasar por los pasos de preprocesamiento de detección facial y extracción de características. En el estudio propuesto se analizan varias técnicas de clasificación.

El sistema de EERR tiene diversas aplicaciones, como las interacciones ordenador-humano, el sistema sanitario y el marketing social. En el estudio propuesto, analizamos los estudios existentes sobre diferentes enfoques de EERR propuestos por autores de todo el mundo. Comparamos los estudios y desarrollamos una taxonomía sobre varios pasos de preprocesamiento, extracción de características y clasificación de emociones. También analizamos las diversas cuestiones abiertas y los futuros retos de investigación relacionados con la EERR. (Alberca, 2020)

2.2 NECESIDAD DE DETECCIÓN FACIAL, REDUCCIÓN DE DIMENSIONES Y NORMALIZACIÓN.

En el proceso FER, el primer paso previo es la detección facial, que implica la detección de una cara en la imagen o fotograma y elimina los píxeles insignificantes. El algoritmo de detección de rostros da como resultado las coordenadas del cuadro delimitador que se coloca sobre el rostro. Detectar un rostro es bastante complejo, ya que los rostros humanos pueden tener diferentes tamaños y formas. Por lo tanto, el algoritmo de detección de caras desempeña un papel fundamental en esta situación. Existen varios algoritmos para la detección de caras, como Viola-Jones, PCA, LDA y algoritmos genéticos. El algoritmo de Viola-Jones es uno de los más utilizados para la detección de caras. Diferencia las caras de las que no lo son. El PCA es el otro método de detección de caras más utilizado. Se utiliza para reducir las dimensiones de la imagen y consta de cuatro partes principales: covarianza de características, descomposición eigen, transformación de componentes principales y elección de componentes. Reducir las dimensiones de m-dimensiones a n-dimensiones: $\forall m > n$ no significa que estemos perdiendo las propiedades de la imagen, es más, las

preserva. Después de la reducción dimensional, se puede utilizar la normalización para escalar la imagen. (Alberca, 2020)

2.3 NECESIDAD DE EXTRACCIÓN DE CARACTERÍSTICAS.

El análisis de la expresión facial comprende varios métodos, como la identificación de puntos de referencia faciales, la extracción de características y diferentes bases de datos de extracción de características. Los puntos de referencia faciales se obtienen a partir de los puntos clave faciales que se derivan de la geometría de la cara. La extracción de características se realiza tras la fase de preprocesamiento. Hay dos métodos disponibles para la extracción de características: la extracción basada en la apariencia y la extracción basada en la geometría. El método basado en la geometría extrae características como los bordes y las esquinas. Neha analizo el rendimiento de la técnica de extracción de características del filtro de Gabor. También probaron el filtro de Gabor medio y compararon ambas técnicas de filtrado para mejorar la tasa de reconocimiento.

- **Esquinas:** Las esquinas de una imagen son una propiedad importante, que puede inferirse de los objetos complejos de la imagen. Cho y al desarrollaron la técnica de detección de esquinas, que mide la distancia y el ángulo entre dos líneas rectas.
- **Bordes:** Son características unidimensionales que representan el límite de una región de la imagen.

El segundo método, basado en la apariencia, se ocupa de los estados de distintos puntos de la cara, como la posición del ojo, la forma de puntos importantes como la boca y las cejas, utilizando las características de los puntos salientes. La mayoría de los métodos tradicionales han utilizado el patrón binario local (LBP) como técnica de extracción de características, que es un marco genérico para la extracción de características de la imagen estática. Convierte las características más importantes de la imagen de entrada, como se ha mencionado anteriormente, en un histograma.

2.4 NECESIDAD DE CLASIFICACIÓN DE EMOCIONES.

El tercer paso de la EERR es la clasificación de emociones. Hay varios métodos que se utilizan para la clasificación de emociones después de aplicar algoritmos de detección facial y extracción de características. Los algoritmos de clasificación son redes neuronales convolucionales (CNN), SVM y máquinas de Boltzmann restringidas (RBM). El método de

clasificación más utilizado es la CNN. Es el algoritmo más eficiente, ya que puede aplicarse directamente a la imagen de entrada sin aplicar ningún algoritmo de extracción de características ni de detección de rostros, y aun así obtiene una mayor precisión sobre los datos de entrada. El número de imágenes en el conjunto de datos de entrenamiento también tiene un gran impacto en los resultados de la clasificación. La CNN se enfrenta a un gran reto en el entrenamiento de conjuntos de datos de imágenes limitadas. Por ello, los modelos que se construyen sobre un conjunto de datos limitado pueden utilizar el algoritmo SVM para la extracción de características y la detección de caras. Las emociones de un ser humano no son estáticas, sino que varían en el tiempo. Por lo tanto, la clasificación de las emociones basadas en la situación es un reto. (Carvalho & Plastino, 2021)

2.5 MÁQUINA DE VECTORES SOPORTE (SVM).

SVM es un clasificador diseñado para clasificar entre dos clases. Si el SVM tiene más de dos clases, entonces hay que implementar más de un SVM. Hay tres métodos por los que podemos implementar SVM para más de dos clases.

- **Uno contra todos:** Fue propuesto en Construye k modelos SVM para datos de entrenamiento que tienen k número de clases. Si hay tres clases, entonces SVM se puede realizar tres veces para cada clase.
- **Uno contra uno:** Fue introducido en este método construye $k(k-1)/2$ clasificadores, donde se toman dos clases a la vez para entrenar el modelo. En este método, SVM se realiza entre cada clase que se va a clasificar.

SVM de grafo acíclico dirigido (DAGSVM): Fue propuesto en su fase de entrenamiento es similar al método uno-vs-uno. La fase de prueba hace uso de un DAG binario enraizado que tiene como máximo $k(k-1)/2$ nodos internos y el máximo k hojas. Una ventaja de utilizar un DAGSVM es que permite generalizar el análisis.

El objetivo de la SVM es identificar el plano de margen máximo entre las clases. El plano de margen máximo puede obtenerse a partir de la distancia máxima entre el plano de margen positivo y el negativo, respectivamente, de las dos clases. La distancia entre el plano de separación y el plano de margen positivo debe ser igual en ambos lados.

Para resolver el problema del reconocimiento de emociones a partir de expresiones faciales de una manera simple y rápida, Datta y Al presentaron un sistema de clasificación que utilizaba la concatenación de características geométricas, así como basadas en la textura para clasificar las emociones utilizando SVMs. Han utilizado la arquitectura SVM jerárquica para aprovechar los beneficios de la clasificación binaria multiclase. Para la clasificación se utilizó el conjunto de datos CK+. Han conseguido mejoras significativas en la precisión utilizando características SVM híbridas en comparación con las características LBP.

Nuno Lopes y Al en 2018 dieron un modelo de clasificación para EERR en ancianos y también presentan las diferencias de EERR en ancianos y personas de otras edades. Utilizaron la máquina de vectores de soporte con una clasificación multiclase para clasificar las emociones. Propusieron dos arquitecturas, la primera elimina las arrugas, el pliegue nasolabial y otros rasgos faciales, utilizando técnicas de suavizado que preservan los bordes. Mientras que, en la segunda arquitectura, introdujeron un algoritmo de la API Microsoft, que detecta la edad de la persona. Se utilizó el conjunto de datos lifespan para entrenar y probar la SVM multiclase. Utilizaron un 80% de imágenes para probar la precisión de la SVM y un 20% para probar la precisión de la aplicación. Obtuvieron una precisión del 95,24% en el grupo de edad joven y una precisión del 90,32% en el grupo de edad avanzada.

SVM es un clasificador lineal que puede aplicarse a datos linealmente separables. Pero SVM también puede tomar datos de alta dimensión como entrada, que la mayoría de las veces son datos no lineales. Por tanto, se aplica una función de mapeo al entrenamiento de la SVM, que es no lineal y convierte los datos en linealmente separables, pero en una dimensión superior. Esta función se denomina función kernel. Hay varias funciones, pero, en su trabajo, utilizó la función kernel de base radial (RBF). En este trabajo utilizaron un enfoque de uno contra uno.

Ibrahim Adeyanju y Al propusieron un método en el que utilizaron cuatro kernels SVM para clasificar diferentes emociones de caras. Utilizaron funciones de Base Radial, Polinómica, Lineal y Cuadrática como kernels SVM. Probaron su modelo en 467 conjuntos de entrenamiento y 238 conjuntos de prueba para clasificar 7 emociones. Obtuvieron una

precisión media máxima del 86,4% con el núcleo RBF, del 99,33% con la función cuadrática, del 97,65% con la polinómica y del 97,86% con la lineal. (Alberca, 2022)

2.6 RED NEURONAL ARTIFICIAL (RNA).

La RNA se inspira en las redes neuronales biológicas que constituyen el cerebro [160]. Nuestro cerebro consta de millones de neuronas que forman una red neuronal. Estas neuronas están interconectadas entre sí y procesan las señales hacia/desde el cerebro a las demás partes de nuestro cuerpo. Este tipo de enlace se denomina sinapsis. Hay aproximadamente 100.000 millones de neuronas y están interconectadas por miles o más de sinapsis. En las RNA, la señal es un número real o binario y la salida de estas neuronas se obtiene mediante alguna función de activación de la suma de todas las entradas a esa neurona. La conexión entre dos neuronas/nodos se denomina borde. Este borde tiene un peso, que describe la intensidad de la señal. Normalmente, las neuronas se agrupan en capas. La primera y la última capa son la de entrada y la de salida, respectivamente. Las capas intermedias se denominan capas ocultas. (Alberca, 2022)

2.7 DEFINICIÓN DE TÉRMINOS.

Las librerías de programación son conjuntos de archivos de código que se utilizan para desarrollar software. Su objetivo es facilitar la programación, al proporcionar funcionalidades comunes, que ya han sido resueltas previamente por otros programadores. Las librerías (o bibliotecas) son un componente esencial para que los desarrolladores puedan programar de forma sencilla, evitando la duplicidad de código y minimizando errores. También permiten una mayor agilidad, al reducir el tiempo de desarrollo, así como los costes. (Patel et al., 2020)

2.7.1 LIBRERÍA NUMPY.

NumPy es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos. Incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación.

Por otro lado, la ventaja de Numpy frente a las listas predefinidas en Python es que el procesamiento de los arrays se realiza mucho más rápido (hasta 50 veces más) que las listas, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.

El principal beneficio de NumPy es que permite una generación y manejo de datos extremadamente rápido. NumPy tiene su propia estructura de datos incorporada llamado arreglo que es similar a la lista normal de Python, pero puede almacenar y operar con datos de manera mucho más eficiente. (Cardellino, 2021) como se muestra en la Figura 2.1

```
In [3]: import numpy
        print(numpy.__version__)

1.23.2
```

Figura 2.1 Versión numpy en el equipo

2.7.2 LIBRERÍA MATPLOTLIB.

Matplotlib es una librería open source que pertenece a Python en la que se pueden crear visualizaciones animadas, estáticas e interactivas en Python. Matplotlib se utiliza en servidores de aplicaciones web, shells y scripts de Python y es representativamente más útil para aquellas personas que trabajan con NumPy. . (KeepCoding, 2022) como se muestra en la Figura 2.2

```
In [1]: import matplotlib
        print(matplotlib.__version__)
```

Figura 2.2 Versión matplotlib en el equipo

2.7.3 BIBLIOTECA CV2.

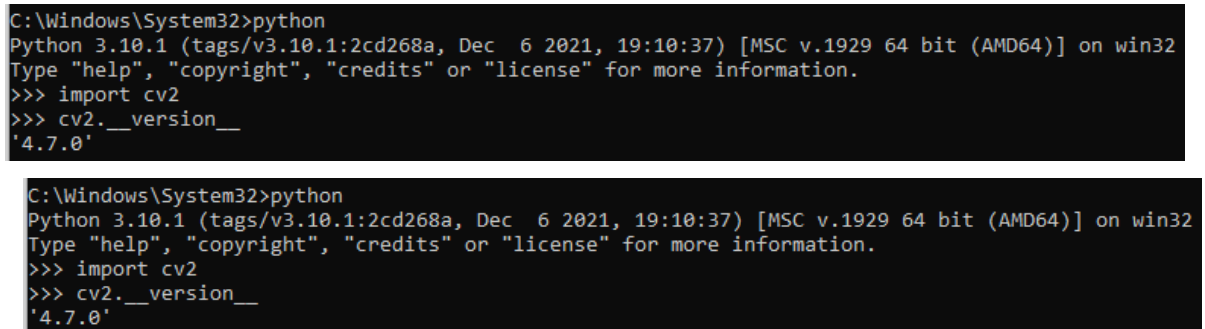
La visión artificial o por computadora se está utilizando cada vez más para el análisis y tratamiento de imágenes mediante algoritmos de inteligencia artificial. Uno de los usos más importantes de OpenCV en la visión por computadora es la detección de rostros y objetos, sobre todo en ámbitos como la fotografía, el marketing o la seguridad.

Una biblioteca de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos.

Además, está especializada en el sistema de visión artificial y machine learning. Con sus más de 2500 algoritmos auestas, OpenCV es capaz de acciones como las siguientes:

- Identificar objetos o caras (reconocimiento facial).
- Encontrar imágenes similares.
- Eliminar los ojos rojos de las fotografías.
- Reconocer escenarios.
- Seguir los movimientos de los ojos.
- Clasificar acciones humanas que estén en videos.
- Extraer modelos 3D.
- Útil en campos como la robótica y la realidad aumentada.

Con esta herramienta tenemos a la mano una biblioteca para aplicaciones de visión artificial. (Fecha y hora. Python. Bartolomé Sintés Marco. www.mclibre.org, s/f) ver



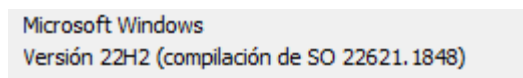
```
C:\Windows\System32>python
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.7.0'
```

```
C:\Windows\System32>python
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.7.0'
```

Figura 2.3 Versión de OpenCV en la terminal del sistema operativo

2.7.4 MÓDULO OS.

El módulo os 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. (Fecha y hora. Python. Bartolomé Sintés Marco. www.mclibre.org, s/f) ver Figura 2.4



Microsoft Windows
Versión 22H2 (compilación de SO 22621.1848)

Figura 2.4 Versión OS de Windows

2.7.5 FUNCIÓN IMUTILS.

Imutils es un paquete basado en OpenCV, para lograr el propósito de llamar a la interfaz OpenCV de manera más concisa, pudiendo realizar fácilmente una serie de operaciones como traducción, rotación, zoom y esqueletización de imágenes. . (*imutils - programador clic, s/f*) como lo muestra la Figura 2.5

```
Requirement already satisfied: imutils in c:\users\lizbt\AppData\Local\Programs\Python\Python310\lib\site-packages (0.5.4)
```

Figura 2.5 Versión imutils

2.7.6 TRADUCCIÓN DE IMÁGENES.

OpenCV también proporciona la implementación de la traducción de imágenes. Primero se debe calcular la matriz de traducción y luego se utiliza la transformación afín para lograr la traducción. La traducción de imágenes se puede realizar directamente en imutils. (*imutils - programador clic, s/f*)

2.7.7 BIBLIOTECA TIME.

La biblioteca time contiene una serie de funciones relacionadas con la medición del tiempo. Como ocurre con todas las bibliotecas, antes de utilizar una función se debe importar la función o la biblioteca entera. La función time() devuelve el número de segundos transcurridos desde el origen de los tiempos del sistema, esta función nos permite cronometrar la duración de un intervalo de tiempo, restando dos llamadas a la función time(). El resultado es un número decimal, pero la precisión en los decimales depende del ordenador. . (*os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.13, s/f*) ver Figura 2.6

```
time.gmtime()
time.struct_time(tm_year=2023, tm_mon=6, tm_mday=15, tm_hour=21, tm_min=46, tm_sec=0, tm_wday=3, tm_yday=166, tm_isdst=0)
```

Figura 2.6 Comportamiento de la biblioteca TIME

2.8 ALGORITMO DE CLASIFICACIÓN.

Un sistema de identificación facial es una tecnología capaz de identificar el rostro de una persona a partir de una imagen digital o un fotograma de vídeo a partir de una fuente de vídeo. (Hariramani, 2020)

2.8.4 HAAR CASCADA.

El clasificador en cascada, o, es decir, la cascada de clasificadores potenciados que trabajan con características similares a las de Haar, es un caso especial de aprendizaje por conjuntos, llamado boosting. Por lo general, se basa en los clasificadores Adaboost (y otros modelos como Real Adaboost, Gentle Adaboost o Logitboost).

Los clasificadores en cascada se entrenan con unos pocos cientos de imágenes de muestra de imágenes que contienen el objeto que se desea detectar y otras imágenes que no contienen esas imágenes hablando en términos expansivos de lectura de datos.

Existe un algoritmo, llamado marco de detección de objetos Viola-Jones, que incluye todos los pasos necesarios para la detección de rostros en vivo como se mencionan a continuación:

- Selección de características de Haar, características derivadas de ondículas de Haar
- Crear una imagen integral
- Entrenamiento de Adaboost
- Clasificadores en cascada

En el siguiente diagrama se puede ver el lineamiento de marco de comportamiento del algoritmo HAAR CASCADA, en primera instancia de comportamiento realiza la selección de funciones, después crea una imagen integral, se dirige hacer el entrenamiento con el modelo real Adaboost, para posteriormente finalizar el proceso con los clasificadores en cascada. Ver Ilustración 2.1.

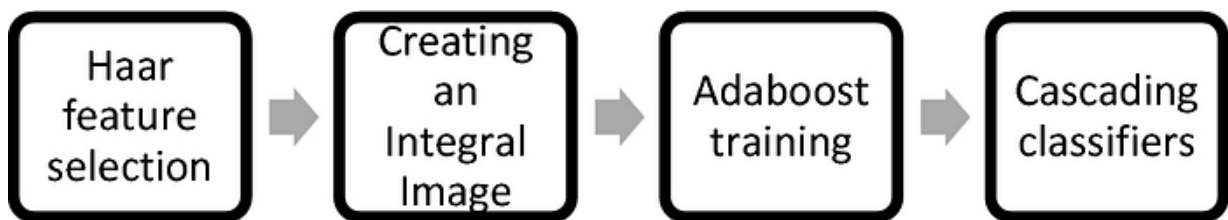


Ilustración 2.1 Marco de comportamiento

Cabe mencionar que el clasificador Haar Cascade se basa en la técnica Haar Wavelet para analizar los píxeles de la imagen en cuadrados por función. Esto utiliza conceptos de "imagen integral" para calcular las "características" detectadas. El

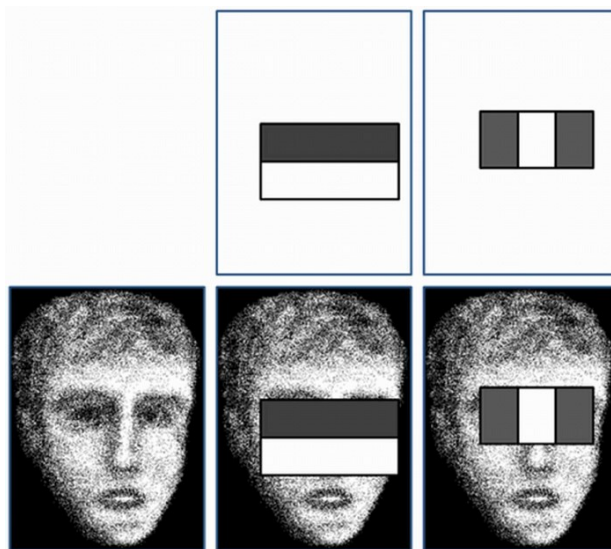
clasificador en cascada de Haar se basa en el algoritmo de detección de Viola-Jones, que se entrena en algunas caras de entrada y no caras y se entrena un clasificador que identifica una cara. (Hariramani, 2020)

CARACTERÍSTICAS HAAR CASCADA

Hay algunas características comunes que encontramos en los rostros humanos estudiados en el paso de selección entre los que se encontraron los siguientes puntos importantes para tomar en cuenta:

- Una región oscura de los ojos en comparación con la parte superior de las mejillas
- Una región brillante del puente de la nariz en comparación con los ojos
- Alguna localización concreta de los ojos, la boca, la nariz.

Las características se denominan características de Haar. El proceso de extracción de características se verá de la siguiente manera :Ilustración 2.2.



*Ilustración 2.2 Características del modelo
Haar Cascada*

Las características de Haar son similares a estos núcleos de convolución que se utilizan para detectar la presencia de esa característica en la imagen dada.

DETECCIÓN DE BORDES MEDIANTE KERNELS DE CONVOLUCIÓN

Dada una imagen de entrada y un kernel de convolución, colocamos el kernel en una esquina y hacemos la multiplicación de convolución cambiando los kernels como se muestra a continuación en la ilustración Ilustración 2.3 Multiplicación de convolución del presente documento. (Hariramani, 2020b)

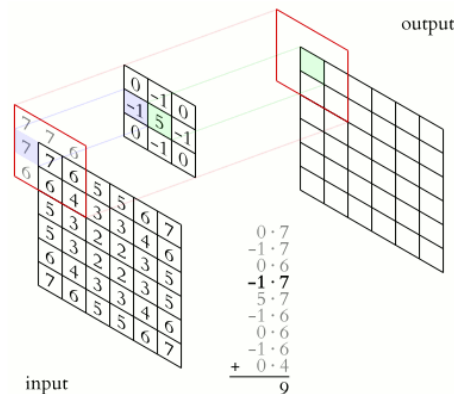


Ilustración 2.3 Multiplicación de convolución

Estas son algunas de las características de Haar que se muestran en la Ilustración 2.4 . Cuadro de características de Haar en donde contemplamos que las dos primeras son "entidades de borde", que se utilizan para detectar bordes. La tercera es una "característica de línea", mientras que la cuarta es una "característica de cuatro rectángulos", que probablemente se usa para detectar

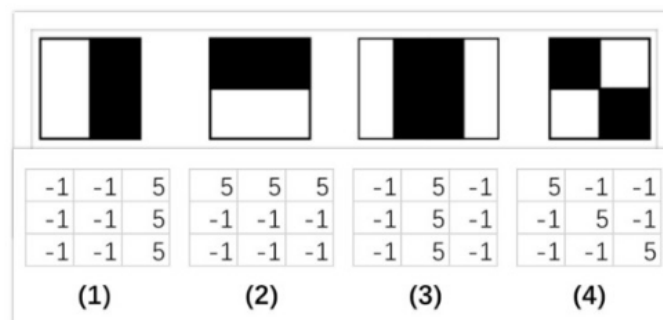


Ilustración 2.4 . Cuadro de características de Haar

una línea inclinada. (Hariramani, 2020b)

Viola-Jones usa 24*24 como tamaño de ventana base y calcula las características anteriores en todo el desplazamiento de la imagen en 1 PX.

Si consideramos todos los parámetros posibles de las características de haar como la posición, la escala y el tipo, terminamos calculando alrededor de 160,000+ características. Por lo tanto, necesitamos evaluar un gran conjunto de características para cada PX de $24 * 24$. Entonces, para evitar esto, tenemos la idea de evitar las funciones redundantes y elegir solo aquellas características que son muy útiles para nosotros. Esto se puede hacer usando AdaBoost. (Hariramani, 2020)

ENTRENAMIENTO DE CLASIFICADORES

Los clasificadores se entrenan utilizando Adaboost y ajustando el umbral para minimizar la tasa de falsos. Al entrenar dicho modelo, las variables son las siguientes:

- El número de etapas del clasificador
- El número de entidades de cada fase
- El umbral de cada etapa

El trabajo de cada fase se utiliza para determinar si una subventana determinada definitivamente no es una cara o puede ser una cara. Una subventana determinada se descarta inmediatamente como no una cara si falla en alguna de las etapas. Afortunadamente, en OpenCV, todo este modelo ya está preentrenado para la detección de rostros. (Hariramani, 2020b)

2.8.5 FASTAI.

En primera instancia se entiende que Fastai es un marco de alto nivel sobre Pytorch para entrenar modelos de aprendizaje automático y lograr un rendimiento de vanguardia en muy pocas líneas de código. Fastai es a Pytorch, lo que Keras es a Tensorflow.

FastAI es una biblioteca de aprendizaje profundo que proporciona a los profesionales componentes de alto nivel que pueden proporcionar rápida y fácilmente resultados de vanguardia en dominios estándar de aprendizaje profundo, y proporciona a los investigadores componentes de bajo nivel que se pueden mezclar y combinar para construir nuevos enfoques. Su objetivo es hacer ambas cosas sin comprometer sustancialmente la facilidad de uso, la flexibilidad o el rendimiento. Esto es posible gracias a una arquitectura cuidadosamente estratificada, que expresa patrones subyacentes comunes de muchas técnicas de aprendizaje profundo y procesamiento

de datos en términos de abstracciones desacopladas. Estas abstracciones se pueden expresar de forma concisa y clara aprovechando el dinamismo del lenguaje Python subyacente y la flexibilidad de la biblioteca PyTorch. Fastai incluye:

- Un nuevo sistema de distribución de tipos para Python junto con una jerarquía de tipos semánticos para tensores
- Una biblioteca de visión artificial optimizada para GPU que se puede ampliar en Python puro
- Un optimizador que refactoriza la funcionalidad común de los optimizadores modernos en dos piezas básicas, lo que permite que los algoritmos de optimización se implementen en 4-5 líneas de código
- Un novedoso sistema de devolución de llamada de 2 vías que puede acceder a cualquier parte de los datos, el modelo o el optimizador y cambiarlo en cualquier momento durante el entrenamiento
- Una nueva API de bloque de datos
- Y mucho más...

FastAI se organiza en torno a dos objetivos de diseño principales: ser accesible y rápidamente productivo, a la vez que profundamente hackeable y configurable. Se basa en una jerarquía de API de nivel inferior que proporcionan bloques de creación que admiten composición. De esta manera, un usuario que desee reescribir parte de la API de alto nivel o agregar un comportamiento particular para satisfacer sus necesidades no tiene que aprender a usar el nivel más bajo. (Tanwar, 2021)

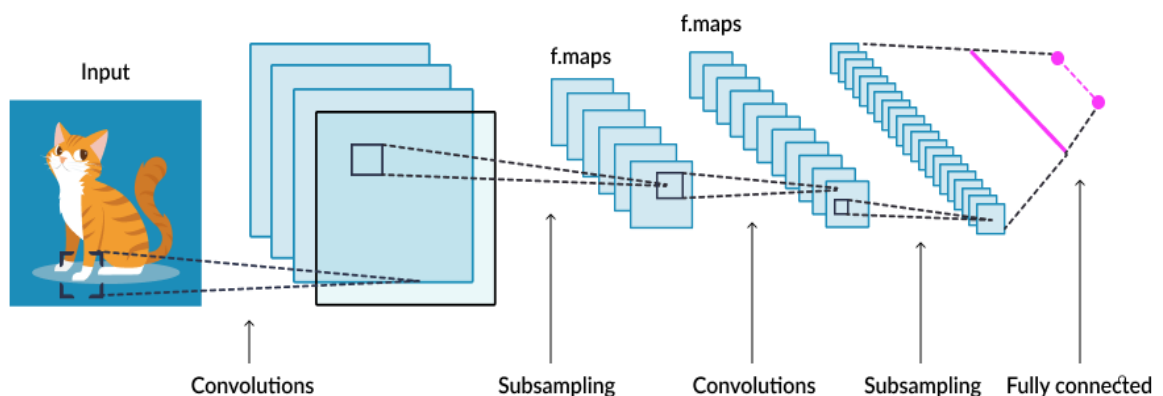


Ilustración 2.5 Jerarquía de API

A continuación, en la Ilustración 2.6 podemos ver el diagrama de los niveles de API y las aplicaciones posteriores.

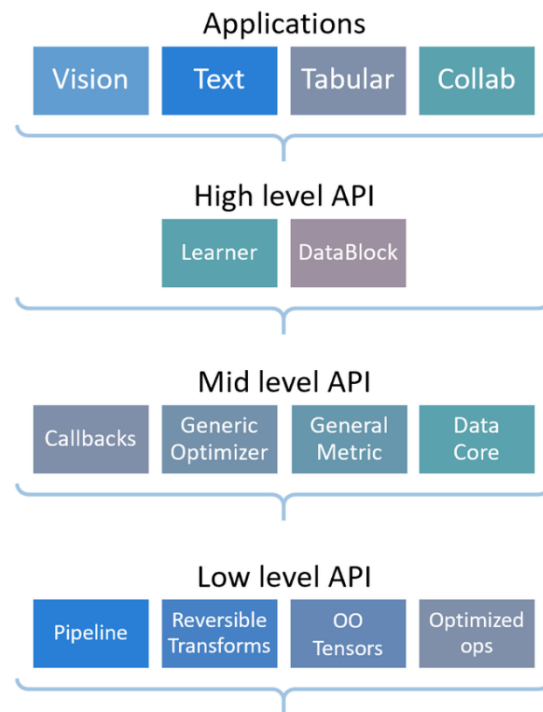


Ilustración 2.6 Jerarquía API

Se ha dividido el marco en 3 secciones: tabular, visión y texto.

- **Tabular:** El aprendizaje automático cubre los problemas de aprendizaje automático en los que tenemos una lista de características para cada usuario o dispositivo o, como las llamamos técnicamente, muestras.
- **Visión:** Cubre los problemas que tienen imágenes como conjunto de datos y es posible que tengamos que hacer clasificación, segmentación semántica o detección de objetos.
- **Texto:** Cubre las secciones en las que queremos hacer el procesamiento del lenguaje natural.

MIGRACIÓN DESDE OTRAS BIBLIOTECAS

Es muy fácil migrar desde PyTorch, Ignite o cualquier otra biblioteca basada en PyTorch, o incluso usar fastai junto con otras bibliotecas. Por lo general, podrá usar todo el código de procesamiento de datos existente, pero podrá reducir la cantidad de código que necesita para el entrenamiento y aprovechar más fácilmente las

prácticas recomendadas modernas. Estas son las guías de migración de algunas bibliotecas populares para ayudarte en tu camino:

- **PyTorch Simple:** La migración desde PyTorch puro le permite eliminar una gran cantidad de código y no requiere que cambie ninguna de sus canalizaciones de datos, optimizadores, funciones de pérdida, modelos, etc. existentes. Es de suma relevancia mencionar que fastai es mucho más que un simple bucle de entrenamiento (aunque solo estamos usando el bucle de entrenamiento en este ejemplo): es un marco completo que incluye transformaciones aceleradas por GPU, inferencia de extremo a extremo, aplicaciones integradas para visión, texto, tabular y filtrado colaborativo, etc. Puede usar cualquier parte del marco por sí sola o combinarlas
- **Ignite:** Fastai es muy diferente de Ignite, ya que es mucho más que un simple bucle de entrenamiento, es un marco completo que incluye transformaciones aceleradas por GPU, inferencia de extremo a extremo, aplicaciones integradas para visión, texto, tabular y filtrado colaborativo, etc.
- **Lightning:** La migración desde Lightning nos permite reducir la cantidad de código y no requiere que cambie ninguna de sus canalizaciones de datos, optimizadores, funciones de pérdida, modelos, etc. existentes. Una vez que haya realizado este cambio, puede beneficiarse del amplio conjunto de devoluciones de llamada, transformaciones, visualizaciones, etc. de fastai.
- **Catalyst:** En el código Catalyst, un bucle de entrenamiento se define manualmente, que es donde se aplanan los tensores de entrada. En fastai, no es necesario definir su propio bucle de entrenamiento, puede insertar su propio código en cualquier parte del proceso de entrenamiento mediante una devolución de llamada, que incluso puede modificar datos, gradientes, la función de pérdida o cualquier otra cosa en el bucle de entrenamiento.
(Tanwar, 2021)

2.8.6 RASGOS FACIALES CON DLIB, OPENCV Y PYTHON.

Dlib es un moderno kit de herramientas de C++ que contiene algoritmos de aprendizaje automático y herramientas para crear software complejo en C++ para resolver problemas del mundo real. Se utiliza tanto en la industria como en el mundo académico en una amplia gama de dominios incluyendo robótica, dispositivos

integrados, teléfonos móviles y grandes entornos de computación de rendimiento. La licencia de código abierto de Dlib le permite usarlo en cualquier aplicación, de forma gratuita. (Rosebrock, 2017)

CARACTERÍSTICAS PRINCIPALES

- **Documentación**

- A diferencia de muchos proyectos de código abierto, este proporciona información completa y precisa documentación para cada clase y función. También hay modos de depuración que comprueban las condiciones previas documentadas para las funciones. Cuando esto está habilitado, capturará la gran mayoría de los Errores causados por llamar a funciones de forma incorrecta o por utilizar objetos de forma incorrecta.
- Se proporcionan muchos programas de ejemplo
- Considero que la documentación es la parte más importante de la biblioteca. Así que, si encuentras algo que no está documentado, no está claro o tiene documentación desactualizada, dímelo y lo arreglaré.

- **Código portátil de alta calidad**

- Buena cobertura de pruebas unitarias. La relación entre las líneas de código de prueba unitaria y las líneas de biblioteca de El código es de 1 a 4.
- La biblioteca se prueba regularmente en sistemas MS Windows, Linux y Mac OS X. Sin embargo, debería funcionar en cualquier sistema POSIX y se ha utilizado en Solaris, HPUX y BSD.
- No se requiere ningún otro paquete para usar la biblioteca. Solo las API que son proporcionados por un sistema operativo listo para usar.
- No es necesario ningún paso de instalación o configuración antes de poder utilizar la biblioteca.
- Todo el código específico del sistema operativo está aislado dentro de las capas de abstracción del sistema operativo, que son lo más pequeño posible. El resto de la biblioteca se superpone al sistema operativo capas de abstracción o es puro estándar ISO C++. (*dlib C++ Library*, s/f)

- **Algoritmos de aprendizaje automático**

- Aprendizaje profundo
- Máquinas de vectores de soporte convencionales basadas en SMO para clasificación y regresión
- Métodos de rango reducido para la clasificación y regresión a gran escala
- Máquinas de vectores de relevancia para clasificación y regresión
- Herramientas de clasificación multiclase de uso general
- Una SVM multiclase
- Una herramienta para resolver el problema de optimización asociado a las máquinas de vectores de soporte estructural.
- Herramientas SVM estructurales para el etiquetado de secuencias
- Herramientas SVM estructurales para resolver problemas de asignación
- Herramientas SVM estructurales para la detección de objetos en imágenes, así como herramientas de aprendizaje profundo más potentes (pero más lentas) para la detección de objetos.
- Herramientas SVM estructurales para etiquetar nodos en grafos
- Una implementación de SVM-Rank a gran escala
- Un algoritmo de regresión RLS del kernel en línea
- Un algoritmo de clasificación de SVM en línea
- Aprendizaje métrico semidefinido
- Un estimador/detector de novedades de centroide kernelizado en línea y Clasificación de una clase del vector de soporte fuera de línea
- Algoritmos de agrupamiento: k-means lineales o kernel, susurros chinos y agrupamiento de Newman.
- Redes de funciones de base radial
- Perceptrones multicapa (*dlib C++ Library*, *s/f*)

- **Algoritmos numéricos**

- Un objeto de matriz rápida implementado mediante la expresión técnicas de plantillas y capaz de utilizar las librerías BLAS y LAPACK cuando estén disponibles.

- Se definen numerosas operaciones matemáticas y de álgebra lineal para el objeto matricial, como la descomposición de valores singulares, la transposición, las funciones trigonométricas, etc.
- Algoritmos de optimización no lineal sin restricciones de propósito general que utilizan las técnicas de gradiente conjugado, BFGS y L-BFGS
- Levenberg-Marquardt para resolver problemas no lineales Problemas de mínimos cuadrados
- Optimización libre de derivados restringida por caja a través del algoritmo BOBYQA
- Una implementación del algoritmo de plano de corte optimizado
- Varios solucionadores de programas cuadráticos
- Herramientas de optimización combinatoria para resolver problemas de asignación óptima y flujo mínimo/máximo, así como el algoritmo CKY para encontrar el árbol de análisis sintáctico más probable
- Un objeto entero grande
- Un objeto de número aleatorio (*dlib C++ Library*, s/f)
- **Algoritmos de inferencia de modelos gráficos**
 - Algoritmo de árbol de unión para una inferencia exacta en una red bayesiana.
 - Gibbs sampler markov chain monte Algoritmo de Carlo para inferencia aproximada en una red bayesiana.
 - Rutinas para realizar inferencias MAP en gráficos de factores generales, de Potts o estructurados en cadena. (*dlib C++ Library*, s/f)
- **Tratamiento de imágenes**
 - Rutinas para leer y escribir formatos de imagen comunes.
 - Conversión automática del espacio de color entre varios tipos de píxeles
 - Operaciones comunes de imagen, como la búsqueda de bordes y las operaciones morfológicas

- Las implementaciones de los algoritmos de extracción de características SURF, HOG y FHOG.
- Herramientas para detectar objetos en imágenes, incluida la detección de la cara frontal y la estimación de la posición del objeto.
- Reconocimiento facial de alta calidad (*dlib C++ Library*, s/f)
- **Interfaces gráficas de usuario**
 - La biblioteca proporciona una API de interfaz gráfica de usuario básica portátil y simple
 - Implementados en la parte superior de la API de la interfaz gráfica de usuario central se encuentran numerosos widgets
 - A diferencia de muchos otros kits de herramientas de interfaz gráfica de usuario, todo el kit de herramientas de interfaz gráfica de usuario dlib es seguro para subprocessos (*dlib C++ Library*, s/f)
- **Algoritmos de compresión e integridad de datos**
 - Un objeto CRC 32
 - Funciones MD5
 - Varios objetos abstractos que representan partes de algoritmos de compresión de datos. Se incluyen muchas formas del algoritmo PPM. (*dlib C++ Library*, s/f)

PUNTOS DE REFERENCIA FACIALES

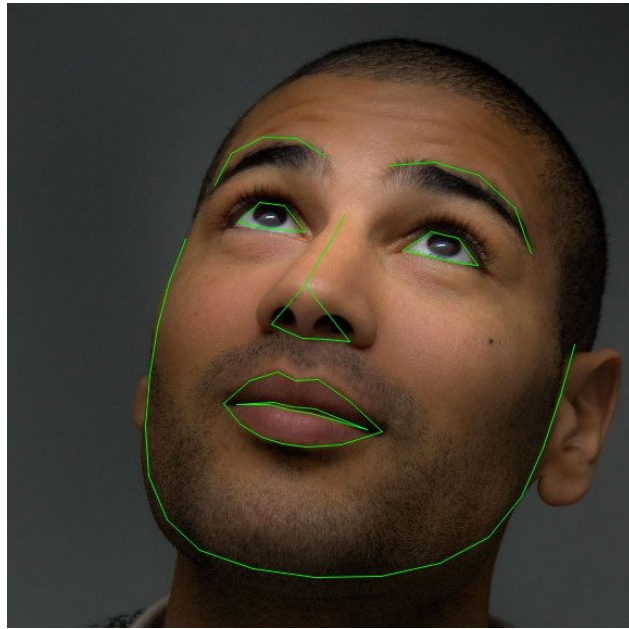


Ilustración 2.7 Puntos de referencia faciales

La detección de puntos de referencia faciales se utiliza para etiquetar e identificar los atributos faciales clave en una imagen (fuente), independientemente en otros términos más explícitos es un subconjunto del problema de predicción de formas. Dada una imagen de entrada (y normalmente un ROI que especifica el objeto de interés), un predictor de forma intenta localizar puntos clave de interés a lo largo de la forma. Ver Ilustración 2.7

La detección de puntos de referencia faciales es un proceso de dos pasos:

Paso 1: Localiza la cara en la imagen.

Paso 2: Detecta el ROI de las estructuras faciales clave en el rostro.

Es importante tener en cuenta que la detección de rostros (En este caso el paso 1 ya mencionado) se puede lograr de varias maneras. Por ejemplo:

- Podríamos usar las cascadas Haar incorporadas en OpenCV.
- Podríamos aplicar un detector de objetos HOG + Linear SVM preentrenado específicamente para la tarea de detección de rostros.
- O incluso podríamos usar algoritmos basados en el aprendizaje profundo para la localización de rostros. (Rosebrock, 2017)

En cualquier caso, el algoritmo real utilizado para detectar la cara en la imagen no importa. En cambio, lo importante es que a través de algún método obtengamos el cuadro delimitador de la cara (es decir, las coordenadas (x, y) de la cara en la imagen).

Dada la región de la cara, podemos aplicar el Paso 2 que en este caso se encarga de detectar estructuras faciales clave en la región de la cara. Hay una variedad de detectores de puntos de referencia faciales, pero todos los métodos intentan esencialmente localizar y etiquetar las siguientes regiones faciales:

- Boca
- Ceja derecha
- Ceja izquierda
- Ojo derecho
- Ojo izquierdo
- Nariz
- Mandíbula

El detector de puntos de referencia faciales incluido en la biblioteca dlib es una implementación del artículo One Millisecond Face Alignment with an Ensemble of Regression Trees de Kazemi y Sullivan.

Es importante tener en cuenta que existen otros tipos de detectores de puntos de referencia faciales, incluido el modelo de 194 puntos que se puede entrenar en el conjunto de datos HAREN.

Ahora bien, independientemente del conjunto de datos que se utilice, se puede aprovechar el mismo marco dlib para entrenar un predictor de forma en los datos de entrenamiento de entrada, lo que es útil si desea entrenar detectores de puntos de referencia faciales o predictores de forma personalizados propios. (Rosebrock, 2017)

CAPITULO III. INSTALACIÓN DE REQUERIMIENTOS

3.1 INSTALACIÓN DE PAQUETERÍAS

En la Figura 3.1 podemos observar la instalación de las paqueterías utilizadas para la implementación del proyecto dado.

```
C:\Windows\system32\cmd.exe
C:\Users\Lupita>py -m pip install opencv-python
Defaulting to user installation because normal site-packages is not writeable
Collecting opencv-python
  Downloading opencv_python-4.7.0.72-cp37-abi3-win_and64.whl (38.2 MB)
    Requirement already satisfied: numpy>=1.17.3 in c:\users\lupita\appdata\roaming\python\python310\site-packages (from opencv-python) (1.22.3)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.7.0.72

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Lupita>pip install -U imutils
Defaulting to user installation because normal site-packages is not writeable
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
    Preparing metadata (setup.py) ... done
Installing collected packages: imutils
  WARNING: imutils is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8557
  Running setup.py install for imutils ... done
Successfully installed imutils-0.5.4

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Lupita>pip install opencv-contrib-python
Defaulting to user installation because normal site-packages is not writeable
Collecting opencv-contrib-python
  Downloading opencv_contrib_python-4.7.0.72-cp37-abi3-win_and64.whl (44.9 MB)
    Requirement already satisfied: numpy>=1.21.2 in c:\users\lupita\appdata\roaming\python\python310\site-packages (from opencv-contrib-python) (1.22.3)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.7.0.72

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Figura 3.1 Instalación de paqueterías

3.2 IMPLEMENTACIÓN DE SISTEMA

Con el fin de desarrollar un sistema que permita identificar las emociones se generó una serie de pasos antes de la realización del código con ello en la Figura 3.2 se puede observar un pequeño esquema en el cual la primera parte es la generación de una carpeta que se llame: “Reconocimiento de personas”, en donde se guardaran 3 carperas, 6 gifs, 3 tipos de código y 3 modelos, los cuales durante el avance se irán ocupando.



Figura 3.2 Esquema de generación de carpetas del contenido de reconocimiento de emociones

3.3 CREACIÓN DE LA BASE DE DATOS CON LOS ROSTROS QUE REFLEJAN LAS EMOCIONES.

Una vez realizado lo anterior se creará la base de datos de rostros mediante Gifs y estas se almacenarán en las correspondientes carpetas.



Figura 3.3 Creación y almacenamiento de base de datos por emociones

3.3.1 CAPTURACIÓN DE ROSTROS “Felicidad”

Con ayuda del cogido y el cmd se hace la toma de fotos, lo primero se llama la carpeta de reconocimientos de emociones, posteriormente en código fuente en Python, la carpeta Data y finalmente se crea una carpeta llamada felicidad. Ver Figura 3.4 y Figura 3.5

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>capturadoRostros.py
Carpeta creada: C:/Users/Lupita/Desktop/Reconocimiento de emociones/Data/Felicidad
C:\Users\Lupita\Desktop\Reconocimiento de emociones>

2 import os
3 import imutils
4
5 #emotionName = 'Enojo'
6 emotionName = 'Felicidad'
7 #emotionName = 'Sorpresa'
8 #emotionName = 'Tristeza'
9
```

Figura 3.4 Resultados de la captación de rostros guardados automáticamente de las tomas para las capturas de “FELICIDAD”

Creacion de la carpeta “Felicidad”

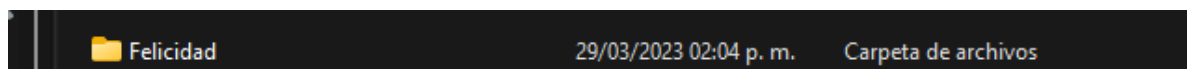


Figura 3.5 Carpeta "Felicidad" creada para el almacenamiento de datos

Reconocimiento de la emoción trabajando con el Gif “felicidad” previamente almacenado en la carpeta principal. Ver Figura 3.6

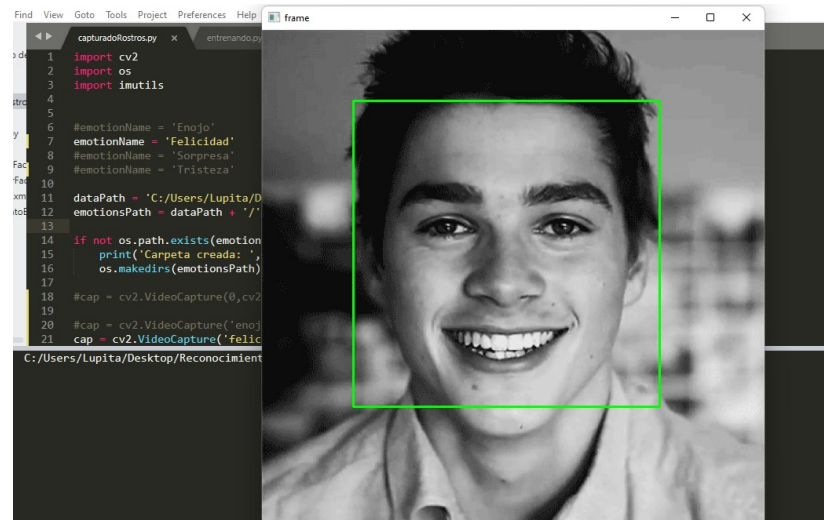


Figura 3.6 Ejecución de GIF "Felicidad"

Automáticamente este empieza a capturar la emoción y se almacena en la carpeta “Felicidad”. Ver Figura 3.7

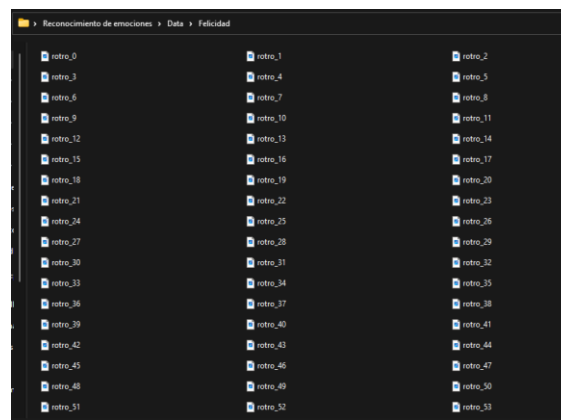


Figura 3.7 Almacenamiento de datos en la carpeta "Felicidad"

3.3.2 CAPTURACIÓN DE ROSTROS “Enojo”

En primera instancia se dio apertura a la carpeta de reconocimiento de emociones, pero por medio del CMD, en donde se manda a llamar la carpeta de reconocimiento de emociones. Ver Figura 3.8 y Figura 3.9

```
C:\WINDOWS\system32\cmd.exe - capturadoRostros.py
Microsoft Windows [Versión 10.0.22000.1455]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Lupita>cd Desktop

C:\Users\Lupita\Desktop>cd Reconocimiento de emociones

C:\Users\Lupita\Desktop\Reconocimiento de emociones>capturadoRostros.py
Carpeta creada: C:/Users/Lupita/Desktop/Reconocimiento de emociones/Data/Enojo
```

Figura 3.8 Apertura de carpeta de reconocimiento de emociones mediante el CMD

Creación de la carpeta “ENOJO”

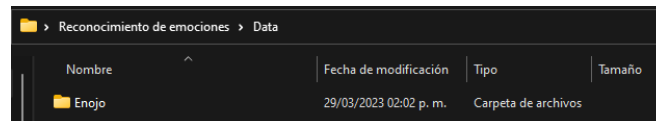


Figura 3.9 Resultados de la captación de rostros guardados automáticamente de las tomas

Reconocimiento de la emoción trabajando con el Gif “enojo” previamente almacenado en la carpeta principal ver Figura 3.10 y Figura 3.11

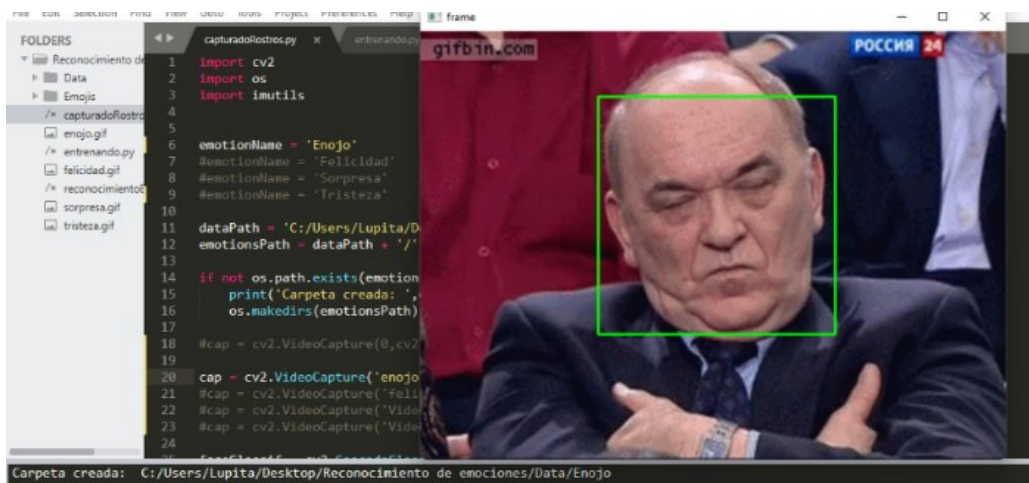


Figura 3.10 Ejecución de GIF "Enojo"

Capturar de imagen y almacenamiento en carpeta “Enojo”

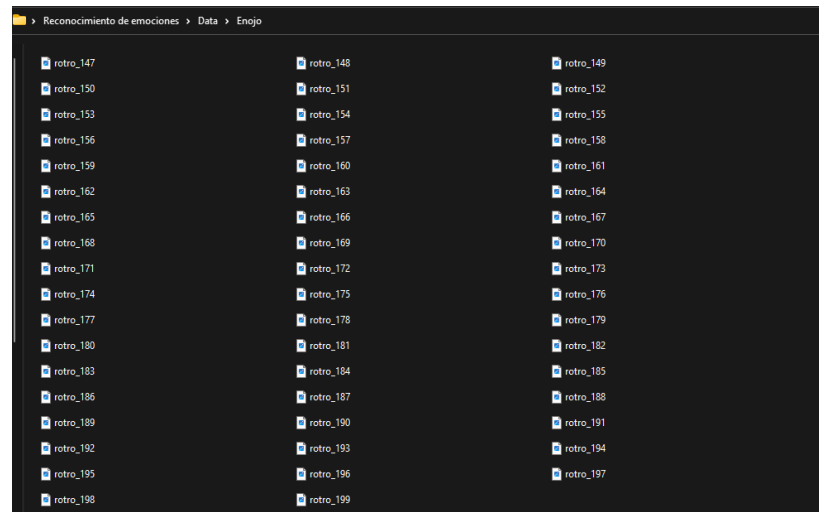


Figura 3.11 Almacenamiento de datos en la carpeta "Enojo"

3.3.3 CAPTURACIÓN DE ROSTROS “Sorpresa”

Apertura de carpeta de reconocimiento de emociones mediante el CMD. Ver Figura 3.12

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>capturadoRostros.py
Carpeta creada: C:/Users/Lupita/Desktop/Reconocimiento de emociones/Data/Sorpresa
C:\Users\Lupita\Desktop\Reconocimiento de emociones>

7 emotionName = 'Sorpresa'
8 #emotionName = 'Tristeza'
9
10 dataPath = 'C:/Users/Lupita/Desktop/Reconocimiento de emocion
11 emotionsPath = dataPath + '/' + emotionName
12
```

Figura 3.12 Apertura de la carpeta de reconocimiento de emoción "Sorpresa"

Creación de la carpeta “**SORPRESA**”

Resultados de la captación de rostros guardados automáticamente de las tomas para las capturas de “**SORPRESA**”.

Previamente encontramos la entrada de reconocimiento de la emoción trabajando

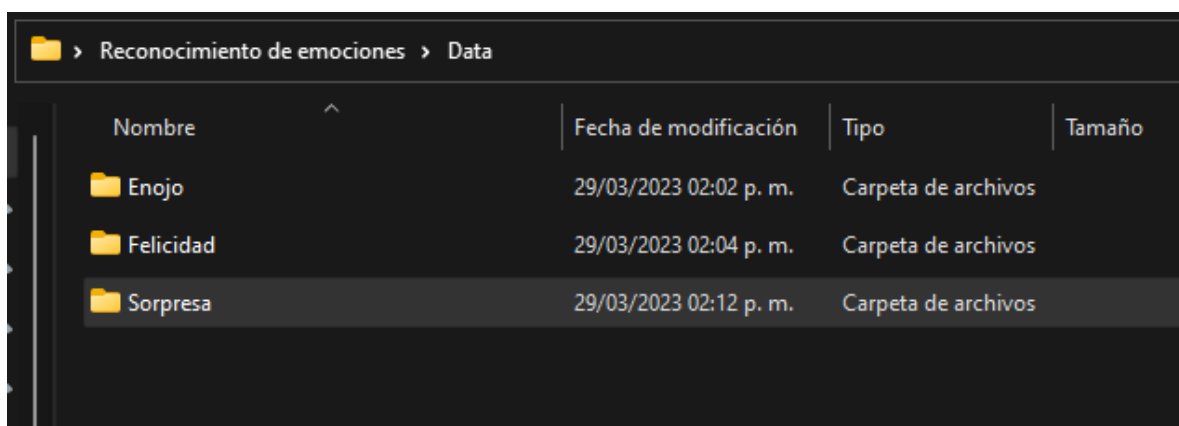


Figura 3.13 Resultado de creación de la carpeta

con el Gif “Sorpresa” previamente almacenado en la carpeta principal.

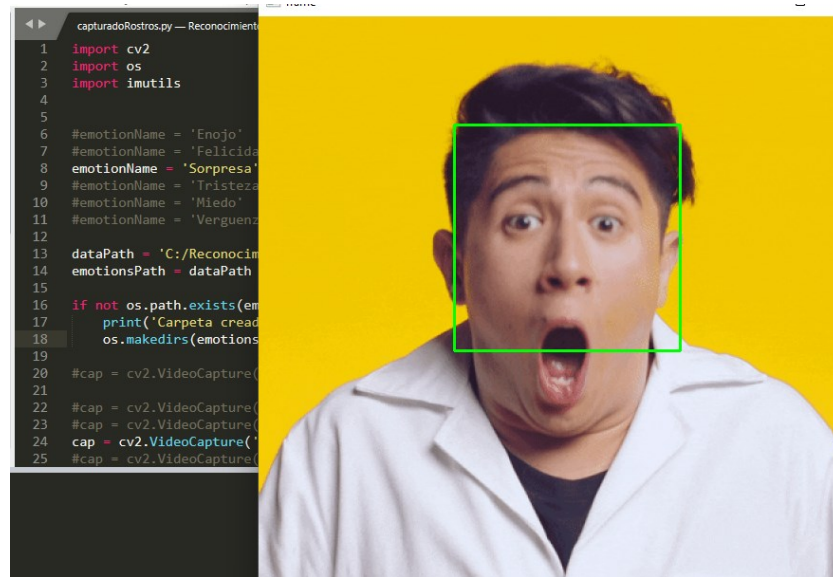


Figura 3.14 Ejecución de GIF "Sorpresa"

Capturar de imagen y almacenamiento en carpeta “Sorpresa” ver Figura 3.13, Figura 3.14 y Figura 3.15

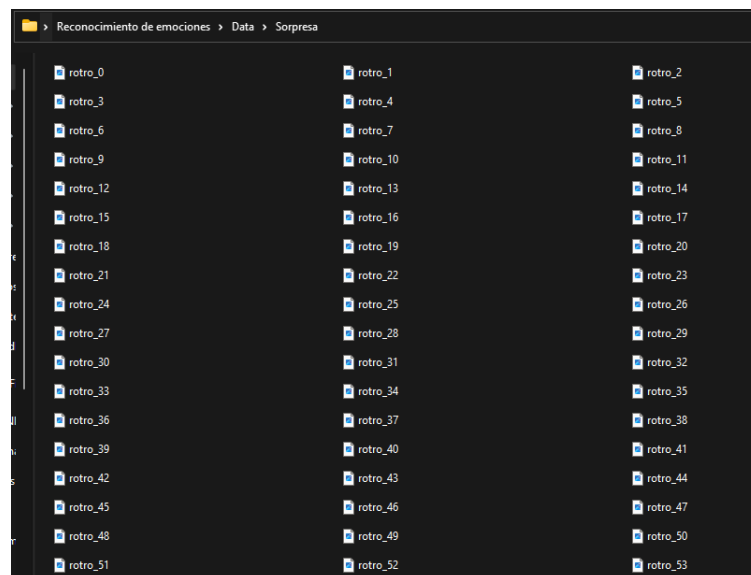


Figura 3.15. Recolección de datos a la carpeta

La Figura 3.19 muestra la captura de imagen y almacenamiento en carpeta “Tristeza”

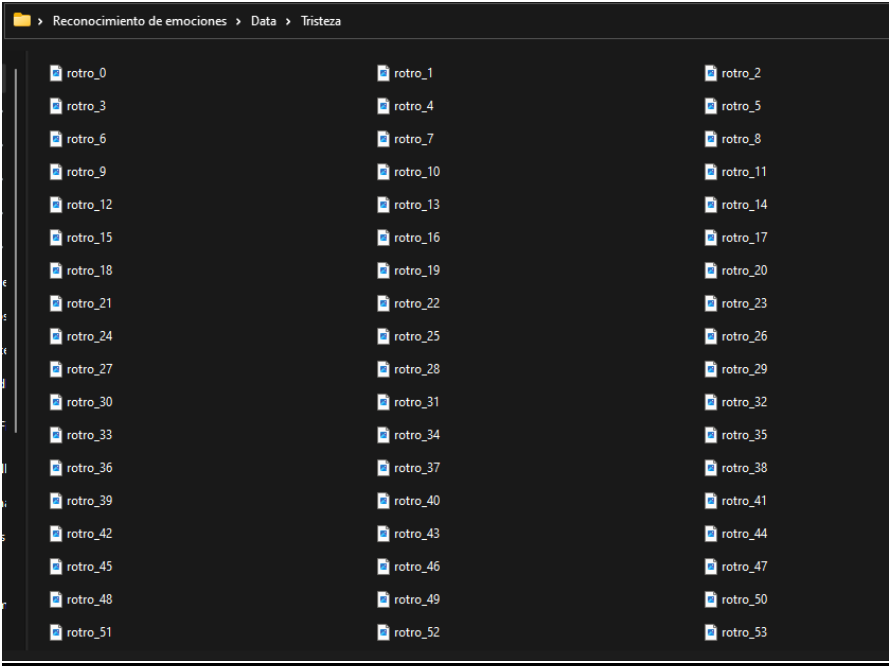


Figura 3.19. Recolección de datos a la carpeta

3.3.5 CAPTURACIÓN DE ROSTROS “Vergüenza”

Apertura de carpeta de reconocimiento de emociones mediante el CMD.

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>capturadoRostros.py
Carpeta creada: C:/Users/Lupita/Desktop/Reconocimiento de emociones/Data/Sorpresa
C:\Users\Lupita\Desktop\Reconocimiento de emociones>
7 emotionName = 'Sorpresa'
8 #emotionName = 'Tristeza'
9
10 dataPath = 'C:/Users/Lupita/Desktop/Reconocimiento de emocion
11 emotionsPath = dataPath + '/' + emotionName
12
```

Figura 3.20. Apertura de la carpeta de reconocimiento de emoción “Sorpresa”

Creación de la carpeta “**Vergüenza**” y resultados de la captación de rostros guardados automáticamente de las tomas para las capturas de “**Vergüenza**”

Enojo	12/06/2023 06:36 p. m.	Carpeta de archivos
Felicidad	12/06/2023 06:34 p. m.	Carpeta de archivos
Miedo	12/06/2023 06:34 p. m.	Carpeta de archivos
Sorpresa	12/06/2023 06:33 p. m.	Carpeta de archivos
Tristeza	12/06/2023 06:35 p. m.	Carpeta de archivos
Vergüenza	12/06/2023 06:36 p. m.	Carpeta de archivos

Figura 3.21. Creación y recolección de datos de la emoción “Vergüenza”

Reconocimiento de la emoción trabajando con el Gif “Vergüenza” previamente almacenado en la carpeta principal. Ver Figura 4.21 y Figura 4.22

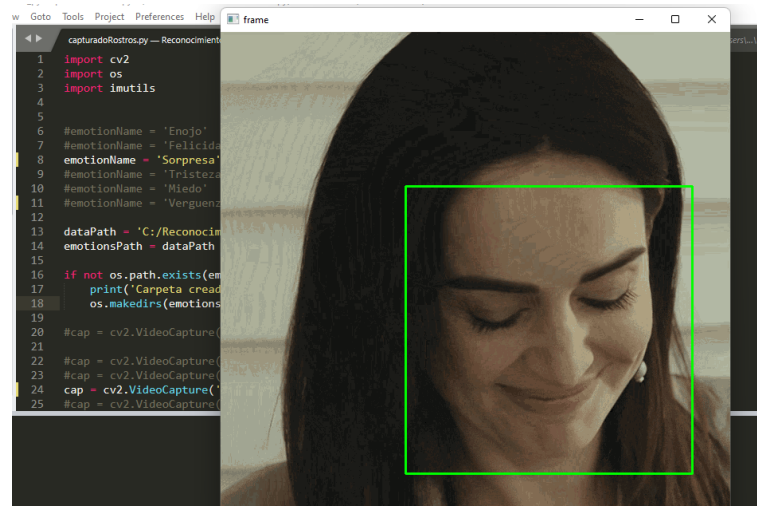


Figura 3.22. Resultado de creación de la carpeta

La Figura 3.23 muestra la capturar de imagen y almacenamiento en carpeta “Vergüenza”

The image shows a file explorer window displaying the contents of the 'Vergüenza' folder. The folder contains 13 image files named 'rotro_0' through 'rotro_12'. All files are of type 'Archivo JPG' and have a size of 10 KB, except for 'rotro_10', 'rotro_11', and 'rotro_12' which are 11 KB. The files were all captured on 02/06/2023 at 07:42 a. m.

Nombre	Fecha	Tipo	Tamaño	Etiqueta
rotro_0	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_1	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_2	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_3	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_4	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_5	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_6	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_7	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_8	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_9	02/06/2023 07:42 a. m.	Archivo JPG	10 KB	
rotro_10	02/06/2023 07:42 a. m.	Archivo JPG	11 KB	
rotro_11	02/06/2023 07:42 a. m.	Archivo JPG	11 KB	
rotro_12	02/06/2023 07:42 a. m.	Archivo JPG	11 KB	

Figura 3.23. Recolección de datos a la carpeta "Vergüenza"

3.4 ENTRENANDO.

El primer paso que se realizó fue abrir el modelo entrenando.py desde el sistema para ejecutar así desde el reconocimiento de emociones. Ver Figura 3.24

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>entrenando.py
Lista de personas: ['Enojo', 'Felicidad', 'Sorpresa', 'Tristeza']
Entrenando ( EigenFaces )...
```

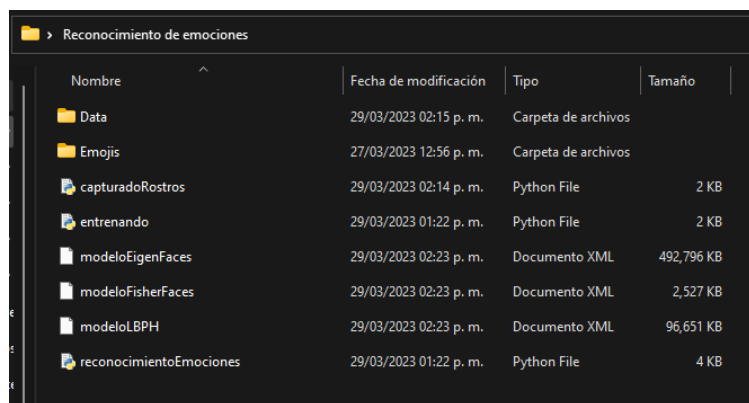
Figura 3.24. Modelo de entrenamiento "entrenando.py"

Posteriormente a la ejecución del entrenamiento como podemos visualizar en la Figura 3.25 nos direcciona a proceder al tener éxito en los entrenamientos de EigenFaces, FisherFaces y finalmente LBPH.

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>entrenando.py
Lista de personas: ['Enojo', 'Felicidad', 'Sorpresa', 'Tristeza']
Entrenando ( EigenFaces )...
Tiempo de entrenamiento ( EigenFaces ): 86.45906829833984
Entrenando ( FisherFaces )...
Tiempo de entrenamiento ( FisherFaces ): 37.59746837615967
Entrenando ( LBPH )...
Tiempo de entrenamiento ( LBPH ): 1.8892223834991455
C:\Users\Lupita\Desktop\Reconocimiento de emociones>
```

Figura 3.25. Modelo de entrenamiento "EigenFaces, FisherFaces, LBPH"

Después nos direccionamos a la carpeta correspondiente de reconocimiento de emociones para poder encontrar los 3 documentos XML que entrenamos desde el sistema en este caso el modeloEigenFaces, modeloFisherFaces y el modeloLBPH. Ver Figura 3.26



Nombre	Fecha de modificación	Tipo	Tamaño
Data	29/03/2023 02:15 p. m.	Carpeta de archivos	
Emojis	27/03/2023 12:56 p. m.	Carpeta de archivos	
capturadoRostros	29/03/2023 02:14 p. m.	Python File	2 KB
entrenando	29/03/2023 01:22 p. m.	Python File	2 KB
modeloEigenFaces	29/03/2023 02:23 p. m.	Documento XML	492,796 KB
modeloFisherFaces	29/03/2023 02:23 p. m.	Documento XML	2,527 KB
modeloLBPH	29/03/2023 02:23 p. m.	Documento XML	96,651 KB
reconocimientoEmociones	29/03/2023 01:22 p. m.	Python File	4 KB

Figura 3.26. Visualización de documentos XML de los modelos creados

3.5 PROBANDO EIGENFACES, FISHERFACES Y LBPH PARA EL RECONOCIMIENTO DE EMOCIONES FACIALES

Una vez que hemos entrenado el o los modelos será necesario probar su funcionamiento, para ello crearemos otro script llamado reconocimientoEmociones.py, en el cual hay una función llamada "emotionImage", como parámetros tendremos a "emotion" que puede ser felicidad, enojo, sorpresa, tristeza, vergüenza y enojo. Dependiendo de estos se visualizará un emoji diferente a la derecha de los fotogramas. Ver Figura 3.27

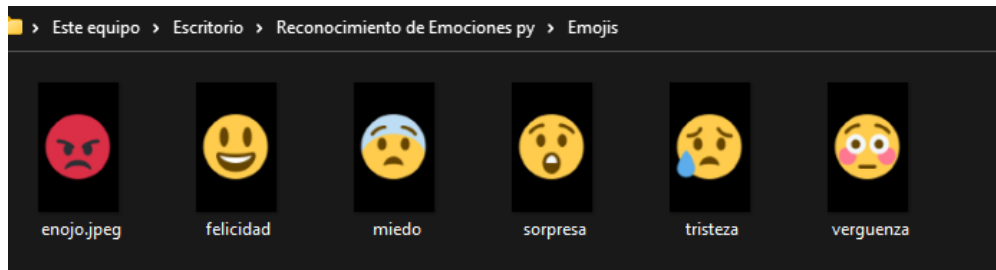


Figura 3.27. Parámetros de funcionamiento para los fotogramas en funcionamiento

Dependiendo del método se asignará a la emoción reconocida y se lea el modelo correspondiente a EigenFaces, FisherFaces o LBPH previamente entrenado, finalmente llama a todas las emociones. Ver Figura 3.28

```
C:\Users\Lupita\Desktop\Reconocimiento de emociones>reconocimientoEmociones.py
imagePaths= ['Enojo', 'Felicidad', 'Sorpresa', 'Tristeza']
```

Figura 3.28. Lectura del modelo y llamado de emociones

3.6 PRIMERAS PRUEBAS DE RECONOCIMIENTO.

En este caso como ya se hizo la creación de carpetas y el entrenamiento correcto de la persona en la Figura 3.29 y Figura 3.30 podemos visualizar en las primeras pruebas de almacenamiento de información del usuario que nos arroja un resultado positivo a las pruebas de reconocimiento mostrándonos que al interactuar de manera directa con nuestro rostro puede identificar concretamente la emoción al momento de la captura.

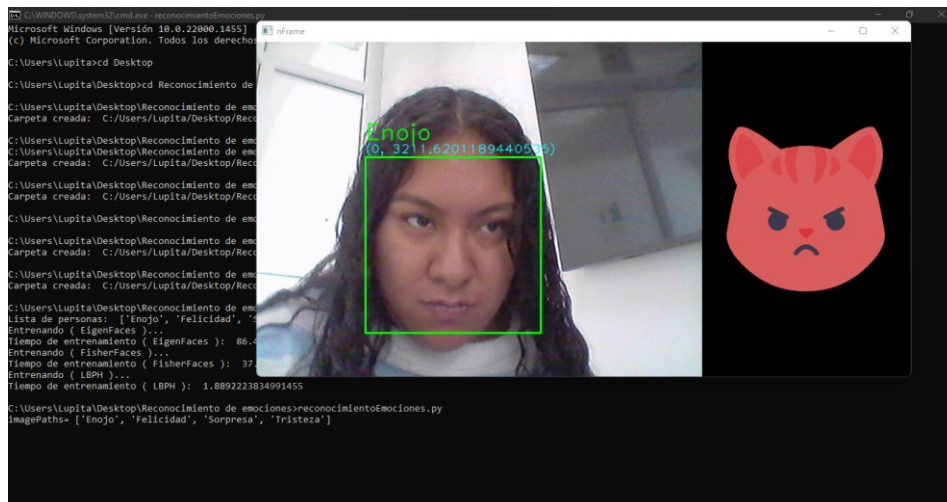


Figura 3.29. Primera prueba de ejecución de reconocimiento de emociones

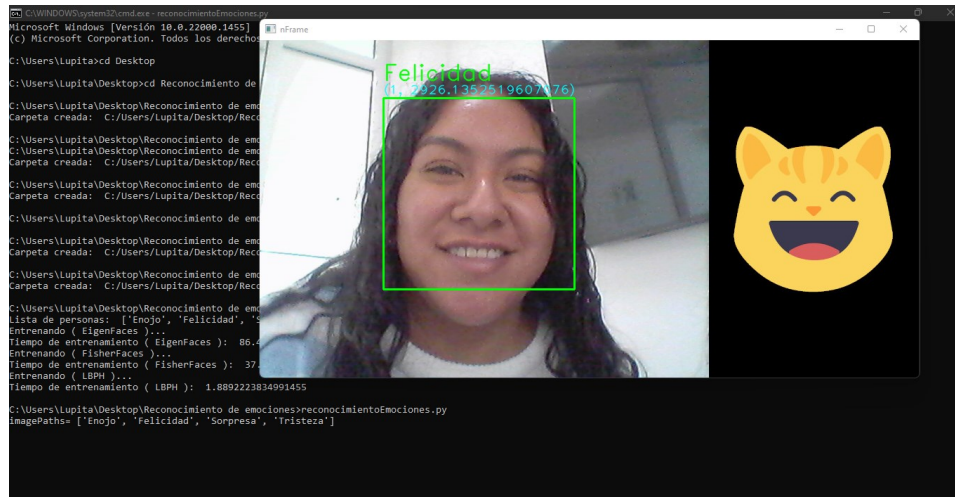


Figura 3.30. Segunda prueba de ejecución de reconocimiento de emociones

En la Figura 3.31 muestra el caso de rostro no identificado, significa que el rostro que se está mostrando no fue el registrado para los resultados en el entrenamiento, es por ello que nos arroja que no puede identificar el rostro de la persona y por lo tanto no puede identificar qué tipo de emoción es la que está concretando en ese momento.

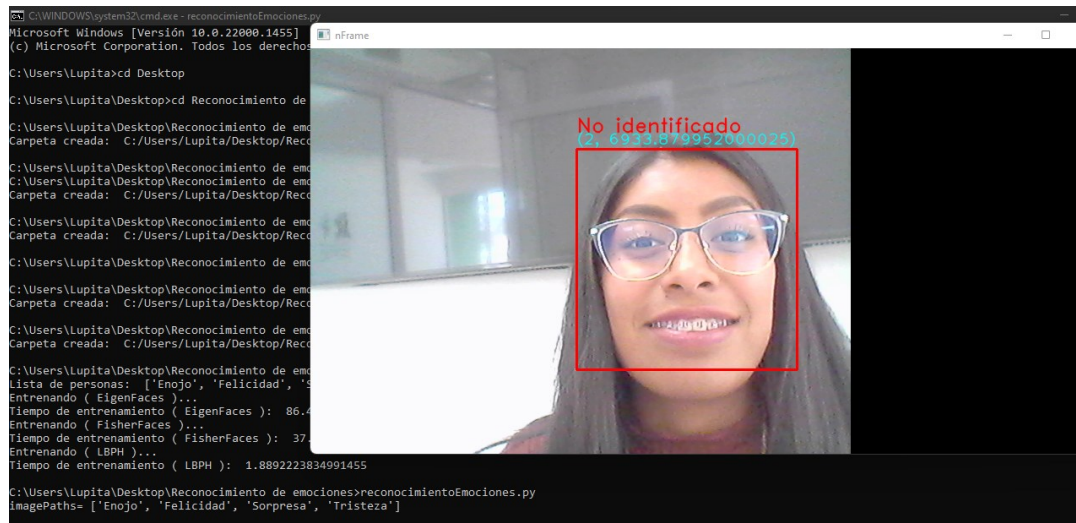


Figura 3.31. Captación de rostros no identificados, al no tener registro en el entrenamiento

4. CAPITULO IV. RESULTADOS

4.1. MODELO PROPUESTO

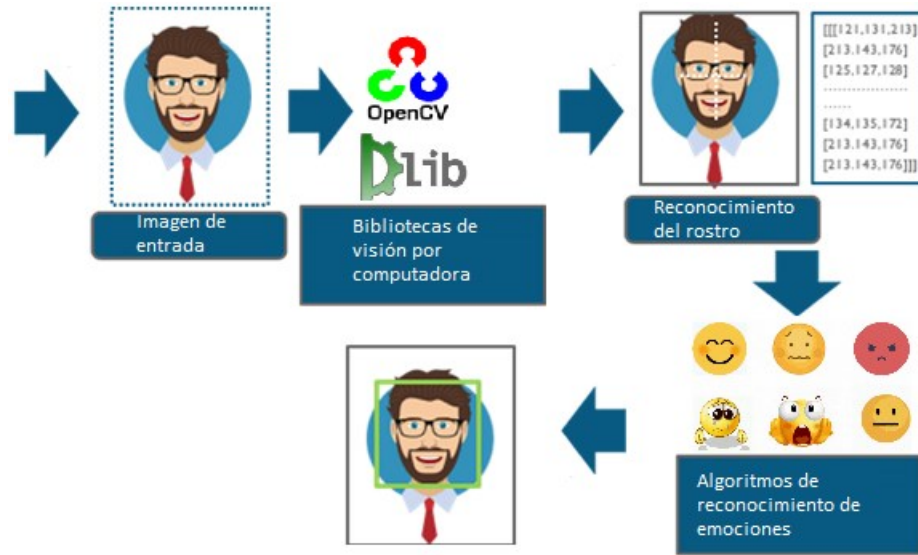


Ilustración 4.1 Funcionamiento del modelo de identificación facial

El modelo propuesto adoptado en esta tesis, se muestra en la Ilustración 4.1. Primeramente, se utiliza una imagen del rostro de una persona. Posteriormente, utilizando bibliotecas compatibles con Python (OpenCV y Dlib), se realizó el preprocesamiento de la imagen de entrada y se obtuvo, únicamente el área del rostro. Por último se usaron algoritmos de reconocimiento de emociones faciales (Haar cascada, rasgos faciales, y una red neuronal de aprendizaje profundo), para llevar a cabo la clasificación y predicción de emociones en personas.

4.2. IMPORTACIÓN DE PAQUETES

En primera instancia se realizó la importación de paquetes como se muestra en la Figura 4.1 Importación de paquetes ver Figura 4.1. En este caso es necesario contar con los paquetes indispensables para que funcione el algoritmo de reconocimiento de emociones.

```
In [58]: #Importacion de paquetes
from sklearn.metrics import confusion_matrix, f1_score, roc_curve, precision_score, recall_score, accuracy_score, roc_auc_score
from sklearn import metrics
from mlxtend.plotting import plot_confusion_matrix
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Figura 4.1 Importación de paquetes

4.3. CONFIGURACIÓN DE PARÁMETROS Y DEFINICIÓN DE ETIQUETAS

En la figura Figura 4.2 se visualiza el desarrollo de la configuración de parámetros que se plantearon para el desarrollo y con ello la definición de las etiquetas. En este caso, cada una de las emociones que se trabajaron en dicho proyecto.

```
In [59]: #configuración de parametros
width_shape = 224
height_shape = 224
batch_size = 32

#Definición de etiquetas
names = ['Felicidad', 'Enojo', 'Tristeza', 'Sorpresa', 'Vergüenza', 'Miedo']
```

Figura 4.2. Configuración de parámetros y definición de etiquetas

4.4. CARGAS DE DATOS, CLASIFICADOR Y ALGORITMOS DE RECONOCIMIENTO DE EMOCIONES

La carga de datos o imágenes de entrada, dependió del algoritmo que se utilizó. Para haar cascada y rasgos faciales, las imágenes se ubicaban de manera local (laptop), mientras que para la red neuronal de aprendizaje profundo, se subían al servidor remoto de Google Colab. Por lo que las imágenes locales eran de mayor calidad, mientras que las remotas, solo eran imágenes de 48 x 48 pixeles. Posteriormente se cargaron los paquetes de visión por computadora y los algoritmos de reconocimiento de emociones faciales. Ver Figura 4.3

Los resultados mostrados en este trabajo, para el algoritmo de la red neuronal, únicamente son las pantallas de salida, por los tiempos de procesamiento, y por usar el servidor remoto de Google Colab.

```
In [62]: #Carga de datos
test_data_dir = 'C:/Users/Lupita/Desktop/Reconocimiento de Emociones py/EmocionesT'

test_datagen = ImageDataGenerator()

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(width_shape, height_shape),
    batch_size = batch_size,
    class_mode='categorical',
    shuffle=False)

#Carga de modelo
custom_Model = load_model("model_VGG16.h5")

#Carga de predicción
predictions = custom_Model.predict(test_generator)

y_pred = np.argmax(predictions, axis=1)
y_real = test_generator.classes
```

Figura 4.3. Carga de datos, de modelo y de predicciones

RESULTADO DE IMÁGENES ENCONTRADAS POR CLASES EN HAAR CASCADA

En la Figura 4.4 se puede visualizar los resultados de las cargas que se hicieron para el reconocimiento de emociones.

```

Found 57 images belonging to 6 classes.
WARNING:tensorflow:AutoGraph could not transform <function Model.make_predict_function.<locals>.predict_function at 0x000001AF376C940> and will run it as-is.
Cause: Unable to locate the source code of <function Model.make_predict_function.<locals>.predict_function at 0x000001AF376C940>. Note that functions defined in certain environments, like the interactive Python shell, do not expose their source code. If that is the case, you should define them in a .py source file. If you are certain the code is graph-compatible, wrap the call using @tf.autograph.experimental.do_not_convert. Original error: lineno is out of bounds
To silence this warning, decorate the function with @tf.autograph.experimental.do_not_convert
WARNING: AutoGraph could not transform <function Model.make_predict_function.<locals>.predict_function at 0x000001AF376C940> and will run it as-is.
Cause: Unable to locate the source code of <function Model.make_predict_function.<locals>.predict_function at 0x000001AF376C940>. Note that functions defined in certain environments, like the interactive Python shell, do not expose their source code. If that is the case, you should define them in a .py source file. If you are certain the code is graph-compatible, wrap the call using @tf.autograph.experimental.do_not_convert. Original error: lineno is out of bounds
To silence this warning, decorate the function with @tf.autograph.experimental.do_not_convert
2/2 [=====] - 26s 6s/step

```

Figura 4.4. Resultado de cargas

4.5. IMPRESIÓN DE PREDICCIONES DE RECOLECCIÓN

A continuación, en la Figura 4.5, Figura 4.6, Figura 4.7, Figura 4.8, Figura 4.9, Figura 4.10, Figura 4.11, Figura 4.12, Figura 4.13 y Figura 4.14 se muestra el resultado de impresión de las predicciones que se hicieron de la recolección de carga de predicciones de (y_{real}) y (y_{pred}).

```
In [63]: #Impresion de predicciones V_real clases reales
print(y_real)

[0 0 0 0 0 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]
 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5]
```

```
In [64]: #Impresion de predicciones y_pred clases de prediccion
print(y_pred)

[7 3 2 2 3 7 2 7 3 7 7 7 9 9 9 4 4 2 4 4 4 2 2 2 2 7 4 4 4 2 2 4 4 4
 3 3 2 2 4 4 2 2 2 2 7 4 4 3 2 2 4]
```

```
In [65]: #Impresion de predicciones
print(predictions)
```

```
[[3.94889280e-03 2.85110623e-03 2.46985346e-01 6.48393482e-02
 2.12219841e-02 2.8389941e-03 1.49858079e-03 4.15258616e-01
 1.43928736e-04 2.40413249e-01]
 [3.10270026e-03 1.41540830e-04 6.3542223e-04 4.90250404e-05
 1.58102828e-02 2.22780990e-04 7.3393577e-05 1.83759538e-03
 1.64057812e-04 2.33012866e-02]
 [4.21520555e-03 6.77283015e-03 6.42892599e-01 2.38365809e-01
 4.49426598e-05 5.68380857e-04 5.44904678e-04 1.57389250e-02
 4.74047614e-04 4.54852246e-02]
 [1.43163755e-04 4.34152689e-03 6.36017501e-01 3.09210509e-01
 1.23255644e-02 3.79656063e-04 1.75454697e-04 6.06445735e-03
 6.50494801e-02 2.93951724e-02]
 [3.46183804e-02 2.95832897e-02 2.00102061e-01 6.03905121e-01
 8.85091648e-07 7.07830640e-04 2.72426289e-04 2.32780892e-02
 2.45224364e-04 1.88935306e-02]
 [9.18287085e-04 2.09436878e-03 7.28449374e-02 1.37886658e-01
 2.89591193e-01 1.32770790e-02 3.93966632e-03 3.42351676e-01
 1.43164891e-01 1.34800871e-01]
 [2.90313794e-04 6.26240810e-03 3.93314660e-01 1.26359135e-01
```

Figura 4.5. Impresiones clases reales, de predicción e impresión de predicciones reporte final

```
In [65]: #Impresion de predicciones
print(predictions)

[2.90313794e-04 6.26240810e-03 3.93314660e-01 1.26359135e-01
 2.18741253e-01 3.34762642e-03 1.40961565e-04 1.23826176e-01
 5.79667394e-04 1.27137795e-01]
[7.60869589e-04 1.26778865e-02 4.85121697e-01 3.89426500e-02
 2.32504129e-01 2.75706337e-03 2.70495162e-04 1.36053205e-01
 6.30756607e-04 9.02812183e-02]
[5.37763000e-04 1.10890381e-01 7.69261047e-02 2.20020860e-01
 1.68767691e-01 2.08140239e-02 2.22618455e-05 2.46668249e-01
 6.64770021e-04 1.54687867e-01]
[2.19985167e-03 5.50468452e-04 2.71315962e-01 4.89860475e-01
 1.95857674e-01 1.13463150e-02 1.77778740e-04 1.46802794e-02
 3.19976709e-04 1.36912400e-02]
[6.91470644e-03 1.47199370e-02 1.37148827e-01 8.80521834e-02
 1.09325409e-01 1.30777573e-03 2.17196633e-04 4.46181238e-01
 3.81659222e-04 1.95751131e-01]
[3.08076735e-03 5.21532912e-03 2.66460925e-01 5.39047569e-02
 9.99218747e-02 6.94274146e-04 3.43892782e-04 5.12398541e-01
 2.95200618e-04 5.76843210e-02]
```

Figura 4.6. Segunda parte de la impresión de predicciones

```
In [65]: #Impresion de predicciones
print(predictions)

[3.08076735e-03 5.21532912e-03 2.66460925e-01 5.39047569e-02
 9.99218747e-02 6.94274146e-04 3.43892782e-04 5.12398541e-01
 2.95200618e-04 5.76843210e-02]
[4.90607170e-04 9.90391360e-04 8.90209228e-02 1.30708292e-02
 1.53613314e-02 9.46993066e-04 5.38154854e-04 8.42176378e-01
 4.16867733e-05 3.73626910e-02]
[1.71800348e-03 7.93264422e-04 5.41527681e-02 5.19225653e-03
 7.98005238e-03 5.50468452e-04 1.63650594e-03 9.10704255e-01
 2.32460989e-05 1.72562767e-02]
[4.92696883e-03 4.09412719e-02 2.94960767e-01 1.99675448e-02
 7.45998323e-02 5.88899711e-03 3.77440942e-04 6.58214018e-02
 9.09005298e-03 4.83417749e-01]
[3.18180793e-03 3.33683938e-02 2.77967364e-01 1.32806217e-02
 4.18048091e-02 2.19595199e-03 2.26777745e-04 1.26708075e-01
 1.54034025e-03 4.99725848e-01]
[3.18180793e-03 3.33683938e-02 2.77967364e-01 1.32806217e-02
 4.18048091e-02 2.19595199e-03 2.26777745e-04 1.26708075e-01
 1.54034025e-03 4.99725848e-01]
[1.07172299e-02 2.83058045e-02 1.38271168e-01 2.16531590e-01
 2.44761750e-01 5.54652791e-03 1.51604498e-02 1.97929397e-01
 1.29931991e-03 1.41476601e-01]
```

Figura 4.7. Tercera parte de la impresión de predicciones

```
In [65]: #Impresion de predicciones
print(predictions)

[3.18180793e-03 3.33683938e-02 2.77967364e-01 1.32806217e-02
 4.18048091e-02 2.19595199e-03 2.26777745e-04 1.26708075e-01
 1.54034025e-03 4.99725848e-01]
[1.07172299e-02 2.83058045e-02 1.38271168e-01 2.16531590e-01
 2.44761750e-01 5.54652791e-03 1.51604498e-02 1.97929397e-01
 1.29931991e-03 1.41476601e-01]
[2.79810484e-02 2.28243251e-03 4.20758612e-02 4.03856598e-02
 7.75034130e-01 1.12185790e-03 1.67904294e-03 4.28133681e-02
 2.39483761e-05 6.66026250e-02]
[6.62371190e-03 1.45943917e-03 5.68440080e-01 1.56195819e-01
 1.29670069e-01 1.02091227e-02 8.82049650e-03 1.04280524e-01
 1.03129225e-03 1.32695064e-02]
[2.48174928e-03 1.13175635e-03 8.49638954e-02 1.78351164e-01
 6.75180674e-01 1.67702953e-03 2.68065458e-04 3.55030186e-02
 6.11131254e-05 2.03815978e-02]
[8.70556291e-03 1.92406250e-03 1.47573933e-01 2.70246387e-01
 3.65316391e-01 4.51386720e-03 1.95643093e-04 1.74718872e-01
 6.48606365e-05 2.67404877e-02]
[2.60227211e-02 6.48256317e-02 8.54962841e-02 1.04727723e-01
 3.99161398e-01 8.41100235e-04 1.01815662e-04 2.49871835e-01
 2.99161398e-01 8.41100235e-04]
```

Figura 4.8. Cuarta parte de la impresión de predicciones

```
In [65]: #Impresion de predicciones
print(predictions)

[8.70556291e-03 1.92406250e-03 1.47573933e-01 2.70246387e-01
 3.65316391e-01 4.51386720e-03 1.95643093e-04 1.74718872e-01
 6.48606365e-05 2.67404877e-02]
[2.60227211e-02 6.48256317e-02 8.54962841e-02 1.04727723e-01
 3.99161398e-01 8.41100235e-04 1.01815662e-04 2.49671835e-01
 1.42835823e-04 6.88087121e-02]
[2.27315370e-02 7.45549472e-03 2.01794952e-01 1.10234685e-01
 4.51006532e-01 1.02655459e-02 7.99921528e-03 8.07117224e-02
 5.73090324e-03 1.02069326e-01]
[5.01600653e-03 2.18024431e-03 7.28460014e-01 1.18929930e-01
 1.01098664e-01 1.45137974e-03 6.95566880e-03 2.30534449e-02
 1.82510016e-03 1.10296439e-02]
[2.94188713e-03 3.53289302e-03 6.98015869e-01 1.18466280e-01
 4.56566624e-02 3.76541074e-03 8.10407847e-03 2.65512168e-02
 4.90911072e-04 9.24748331e-02]
[3.20372847e-03 5.04363468e-03 7.85704851e-01 6.57532960e-02
 3.07415854e-02 5.33510884e-03 3.70518444e-03 3.28200646e-02
 4.19039163e-04 6.72735050e-02]
[3.42572504e-03 1.27814617e-03 5.41248441e-01 2.66784206e-02
 3.28133315e-01 4.06947210e-02 8.06902442e-03 3.80745381e-02
 4.27708495e-03 8.12056940e-03]
[3.11916857e-03 1.35983527e-01 2.70697534e-01 3.73433296e-03
 3.00386064e-02 4.04341767e-04 1.28270837e-03 3.80237043e-01
 4.17238043e-05 1.74461022e-01]
[2.28112470e-02 1.09202066e-03 7.28866532e-02 2.68351380e-02
 6.63148344e-01 2.07569404e-03 4.33897227e-03 1.15628652e-01
 1.36800948e-04 9.10472497e-02]
[1.72579773e-02 2.32757605e-03 2.58277506e-01 6.71362951e-02
 4.86126453e-01 1.03312870e-02 5.68310672e-04 4.74235564e-02
 4.73328284e-04 1.10077634e-01]
[1.16340453e-02 2.73516588e-03 2.69223303e-01 6.08895235e-02
 5.89154363e-01 3.88298417e-03 1.54857174e-03 3.33397686e-02
 3.97031181e-05 2.75526270e-02]
[3.69727169e-03 1.39280371e-02 4.81097817e-01 1.14645876e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
[2.15768651e-03 6.16069734e-02 5.31152666e-01 2.24683546e-02
 2.89404541e-01 4.12150379e-03 1.07644555e-04 2.69221161e-02
 7.45184952e-03 5.46067059e-02]
[8.71247239e-03 8.84280261e-03 2.36197710e-01 5.00720665e-02
 6.31933451e-01 2.78106122e-03 8.82199686e-03 2.87087038e-02
 9.40328930e-04 2.29894104e-02]
[4.77366671e-02 2.96909246e-03 3.29283118e-01 7.38595501e-02
 3.79914135e-01 1.59694590e-02 6.00642944e-03 1.07915299e-02
 1.62751821e-04 1.33307219e-01]
[4.10270579e-02 3.71982664e-04 2.78987169e-01 1.99822292e-01
 3.72524261e-01 6.80198940e-03 7.12675508e-03 2.86234654e-02
 9.70491456e-05 6.46180138e-02]
[3.5776505e-03 4.10402172e-03 1.71670055e-01 5.41554313e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
```

Figura 4.9. Quinta parte de la impresión de predicciones

```
In [65]: #Impresion de predicciones
print(predictions)

[3.20372847e-03 5.04363468e-03 7.85704851e-01 6.57532960e-02
 3.07415854e-02 5.33510884e-03 3.70518444e-03 3.28200646e-02
 4.19039163e-04 6.72735050e-02]
[3.42572504e-03 1.27814617e-03 5.41248441e-01 2.66784206e-02
 3.28133315e-01 4.06947210e-02 8.06902442e-03 3.80745381e-02
 4.27708495e-03 8.12056940e-03]
[3.11916857e-03 1.35983527e-01 2.70697534e-01 3.73433296e-03
 3.00386064e-02 4.04341767e-04 1.28270837e-03 3.80237043e-01
 4.17238043e-05 1.74461022e-01]
[2.28112470e-02 1.09202066e-03 7.28866532e-02 2.68351380e-02
 6.63148344e-01 2.07569404e-03 4.33897227e-03 1.15628652e-01
 1.36800948e-04 9.10472497e-02]
[1.72579773e-02 2.32757605e-03 2.58277506e-01 6.71362951e-02
 4.86126453e-01 1.03312870e-02 5.68310672e-04 4.74235564e-02
 4.73328284e-04 1.10077634e-01]
[1.16340453e-02 2.73516588e-03 2.69223303e-01 6.08895235e-02
 5.89154363e-01 3.88298417e-03 1.54857174e-03 3.33397686e-02
 3.97031181e-05 2.75526270e-02]
[3.69727169e-03 1.39280371e-02 4.81097817e-01 1.14645876e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
[2.15768651e-03 6.16069734e-02 5.31152666e-01 2.24683546e-02
 2.89404541e-01 4.12150379e-03 1.07644555e-04 2.69221161e-02
 7.45184952e-03 5.46067059e-02]
[8.71247239e-03 8.84280261e-03 2.36197710e-01 5.00720665e-02
 6.31933451e-01 2.78106122e-03 8.82199686e-03 2.87087038e-02
 9.40328930e-04 2.29894104e-02]
[4.77366671e-02 2.96909246e-03 3.29283118e-01 7.38595501e-02
 3.79914135e-01 1.59694590e-02 6.00642944e-03 1.07915299e-02
 1.62751821e-04 1.33307219e-01]
[4.10270579e-02 3.71982664e-04 2.78987169e-01 1.99822292e-01
 3.72524261e-01 6.80198940e-03 7.12675508e-03 2.86234654e-02
 9.70491456e-05 6.46180138e-02]
[3.5776505e-03 4.10402172e-03 1.71670055e-01 5.41554313e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
```

Figura 4.10. Sexta parte de la impresión de predicciones

```
In [65]: #Impresion de predicciones
print(predictions)

4.73328284e-04 1.10077634e-01]
[1.16340453e-02 2.73516588e-03 2.69223303e-01 6.08895235e-02
 5.89154363e-01 3.88298417e-03 1.54857174e-03 3.33397686e-02
 3.97031181e-05 2.75526270e-02]
[3.69727169e-03 1.39280371e-02 4.81097817e-01 1.14645876e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
[2.15768651e-03 6.16069734e-02 5.31152666e-01 2.24683546e-02
 2.89404541e-01 4.12150379e-03 1.07644555e-04 2.69221161e-02
 7.45184952e-03 5.46067059e-02]
[8.71247239e-03 8.84280261e-03 2.36197710e-01 5.00720665e-02
 6.31933451e-01 2.78106122e-03 8.82199686e-03 2.87087038e-02
 9.40328930e-04 2.29894104e-02]
[4.77366671e-02 2.96909246e-03 3.29283118e-01 7.38595501e-02
 3.79914135e-01 1.59694590e-02 6.00642944e-03 1.07915299e-02
 1.62751821e-04 1.33307219e-01]
[4.10270579e-02 3.71982664e-04 2.78987169e-01 1.99822292e-01
 3.72524261e-01 6.80198940e-03 7.12675508e-03 2.86234654e-02
 9.70491456e-05 6.46180138e-02]
[3.5776505e-03 4.10402172e-03 1.71670055e-01 5.41554313e-01
 2.45146677e-01 1.93024252e-03 4.84138698e-04 1.21421665e-01
 4.12500987e-04 1.72357671e-02]
```

Figura 4.11. Séptima parte de la impresión de predicciones

Figura 4.12. Octava parte de la impresión de predicciones

Figura 4.13. Novena parte de la impresión de predicciones

Figura 4.14. Décima parte de la impresión de predicciones

Hasta esta última salida, Figura 4.15 llegó el reporte arrojado de predicciones. Como podemos ver en las figuras, fueron resultados continuos.

```
In [65]: #Impresion de predicciones
print(predictions)

1.63987625e-05 6.09643795e-02]
[2.37368024e-03 1.95722771e-03 2.66994357e-01 3.61375585e-02
5.86578429e-01 2.08931975e-03 5.05744611e-05 5.11106960e-02
3.77685428e-05 5.26703857e-02]
[3.32951255e-04 1.90187246e-02 8.85130465e-02 5.06910443e-01
1.89351767e-01 4.47034510e-03 1.04129365e-04 1.54264331e-01
4.19108001e-05 3.69923636e-02]
[1.82535301e-03 3.83427329e-02 7.23890126e-01 1.00933509e-02
4.53024326e-02 1.20989401e-02 2.21359683e-03 1.31990612e-01
4.55985864e-04 3.37068215e-02]
[2.57967948e-03 6.20529056e-02 5.83579719e-01 2.78514922e-02
1.71503007e-01 5.41655347e-03 1.77049034e-04 6.48772269e-02
2.46009236e-04 8.17163810e-02]
[2.29786732e-03 1.85259078e-02 6.27183795e-01 1.27169758e-01
1.75272778e-01 1.03612174e-03 3.11468262e-04 4.20047827e-02
7.59404938e-05 6.12161867e-03]
[9.18600708e-04 4.00883518e-03 5.16478680e-02 4.72490452e-02
7.35266507e-01 1.44215615e-03 6.04625419e-03 1.48904249e-01
1.77037145e-03 2.74620135e-03]]
```

Figura 4.15. Último reporte de impresión de predicciones

En la última Figura 4.16 se puede visualizar la impresión del reporte de métricas y clasificación para clases reales y clases de predicción: La **precisión**, (o "*precision*") se refiere a lo cerca que está el resultado de una predicción del valor verdadero. La **sensibilidad** (o *recall*) representa la **tasa de verdaderos positivos** (*True Positive Rate*) ó **TP**, es decir, la proporción entre los casos positivos bien clasificados por el modelo, respecto al total de positivos. observar las 4 columnas de datos: precisión, sensibilidad, puntuación f1 y soporte.

```
In [89]: print(metrics.classification_report(y_real,y_pred, digits = 4))
```

	precision	recall	f1-score	support
0	0.0000	0.0000	0.0000	6
1	0.0000	0.0000	0.0000	4
2	0.0000	0.0000	0.0000	8
3	0.3333	0.0870	0.1379	23
4	0.1176	0.2857	0.1667	7
5	0.0000	0.0000	0.0000	9
7	0.0000	0.0000	0.0000	0
9	0.0000	0.0000	0.0000	0
accuracy			0.0702	57
macro avg	0.0564	0.0466	0.0381	57
weighted avg	0.1490	0.0702	0.0761	57

Figura 4.16. Reporte de métricas y clasificación de clases reales y clases de predicción

Se entrena el algoritmo de detección de rostros de viola-Jones y los pesos se almacenan de manera local. Todo lo que hacemos es tomar las características del archivo y aplicarlas a nuestra imagen, si la cara está presente en la imagen, obtenemos la ubicación de la cara.

4.6. RASGOS FACIALES CON DLIB, OPENCV Y PYTHON PARA LA CLASIFICACIÓN DE EMOCIONES

En este clasificador se utilizó el paquete de DLIB en el que la detección de rostros se puede obtener de varias maneras, en cualquier caso, el algoritmo real utilizado para detectar la cara en la imagen no importa. En cambio, lo importante es que a través de algún método se obtenga el cuadro delimitador de la cara (es decir, las coordenadas (x, y) de la cara en la imagen). Este método se comenzó usando un conjunto de entrenamiento de puntos de referencia faciales etiquetados en una imagen. Estas imágenes se etiquetan manualmente, especificando coordenadas específicas (x, y) de las regiones que rodean cada estructura facial. En la Figura 4.17 podemos observar el resultado de la salida a la inicialización de detector de caras.

```
17. # inicializar el detector de caras de dlib (basado en HOG) y luego crear
18. # El predictor de puntos de referencia faciales
19. detector = dlib.get_frontal_face_detector()
20. predictor = dlib.shape_predictor(args["shape_predictor"])
```

Figura 4.17. Puntos de referencia faciales con dlib, OpenCV y Python.

Posteriormente, a priori, más específicamente, la probabilidad de distancia entre pares de píxeles de entrada, teniendo en cuenta estos datos de entrenamiento, se entrena un conjunto de árboles de regresión para estimar las posiciones de los puntos de referencia faciales directamente a partir de las propias intensidades de píxeles (es decir, no se está llevando a cabo ninguna "extracción de características").

Entendiendo el detector de puntos de referencia faciales previamente entrenado dentro de la biblioteca dlib se utiliza para estimar la ubicación de las 68 coordenadas(x, y) que se asignan a las estructuras faciales de la cara. Los índices de las 68 coordenadas se pueden visualizar en la Figura 4.18.

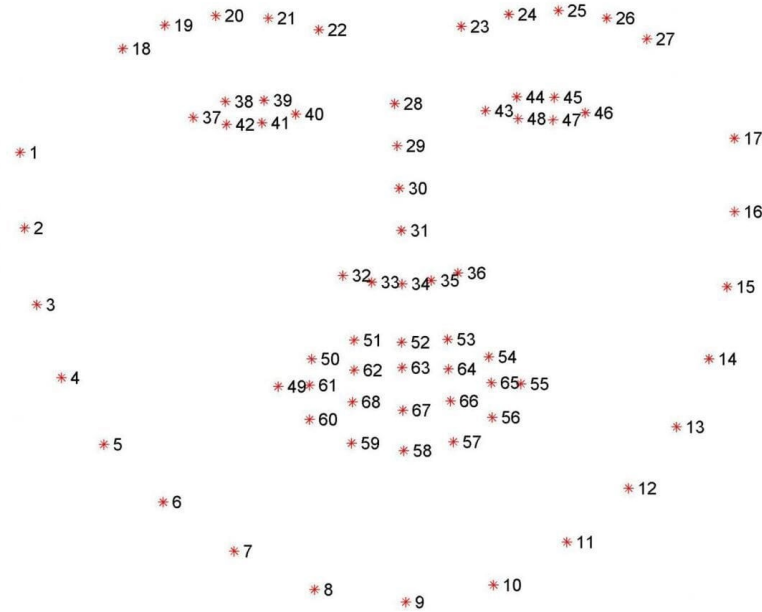


Figura 4.18. Visualización de las 68 coordenadas de puntos de referencia

El resultado es un detector de puntos de referencia faciales que se puede utilizar para detectar puntos de referencia faciales en tiempo real con predicciones de alta calidad (igualmente de manera local). Estas anotaciones forman parte del conjunto de datos iBUG 300-W de 68 puntos con el que se entrenó el predictor de puntos de referencia faciales dlib.

4.7. CLASIFICADOR CON RED NEURONAL DE APRENDIZAJE PROFUNDO

Para este clasificador se utilizó el paquete de Fastai, que incluye un modelo de red neuronal de aprendizaje profundo y como datos se utilizó los conjuntos de datos Kaggle's Facial Expression Recognition Challenge y CK+ que se compone de 28709 imágenes para entrenamiento (Training) + 3589 de imágenes de prueba publicos (PublicTest) + 3589 de imágenes de prueba privados (PrivateTest) dando un total de 35887 imágenes usadas en este apartado ver Figura 4.19

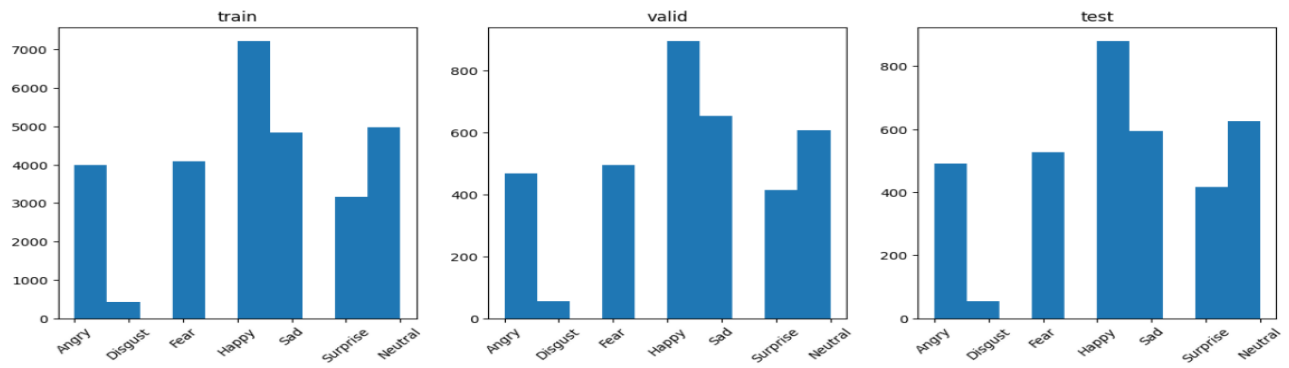


Figura 4.19. Distribución de las emociones en las imágenes de aprendizaje y prueba públicos y privados

Como podemos observar en la Figura 4.20 se muestra el resultado total de la clasificación con el modelo de red neuronal de aprendizaje profundo que fue utilizado en el paquete ya mencionado anteriormente.



Figura 4.20. Resultados de clasificación

En la Figura 4.21 obtenemos así mismo también la matriz de confusión en donde nos refleja las predicciones y las muestras actuales que se tienen al momento.

Confusion matrix

Actual	Angry	302	1	58	112	114	139	41
	Disgust	17	3	7	15	15	23	8
	Fear	124	0	165	88	124	211	140
	Happy	84	0	34	1050	124	98	34
	Neutral	75	0	56	111	513	182	47
	Sad	124	0	72	117	230	396	29
	Surprise	36	0	60	59	62	20	421
		Predicted						
		Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise

Figura 4.21 Resultados obtenidos por matriz de confusión

4.8. RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE HAAR CASCADAS

El algoritmo Haar cascadas arrojando los resultados de Lizbeth Merlo Hernández, Maria Guadalupe zecua Camacho y actores famosos, donde se muestran que el algoritmo de reconocimiento de emociones humanas Haar cascada (ver Figura 4.22), acepta imágenes de gran resolución. Para los actores, sus predicciones fueron muy buenas. Sin en cambio, cuando se probó con imágenes propias, los resultados fueron pésimos. Es un algoritmo que depende mucho en está configuración `smile_detector.detectMultiScale3(roi, 2.5, 20, outputRejectLevels=True)`, y de la posición del rostro.

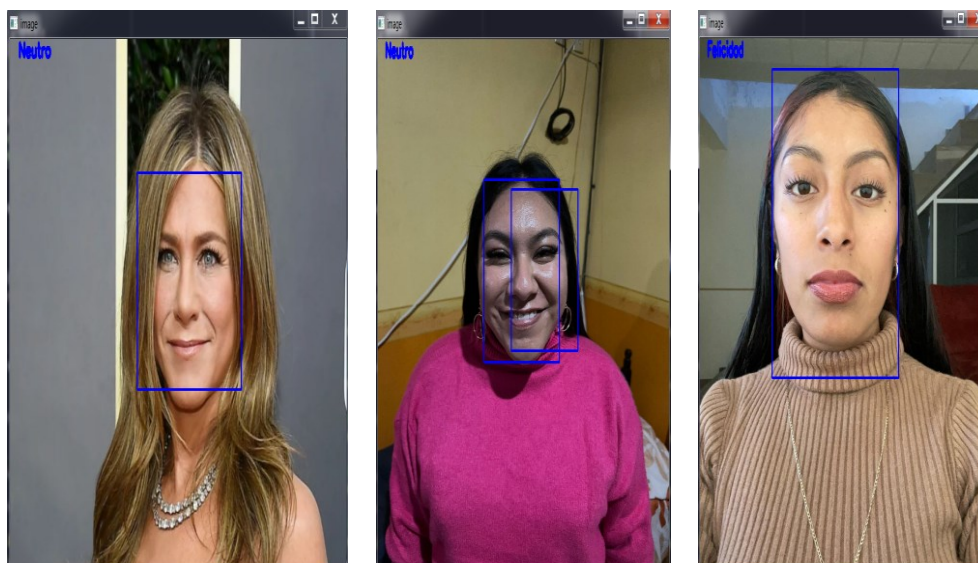
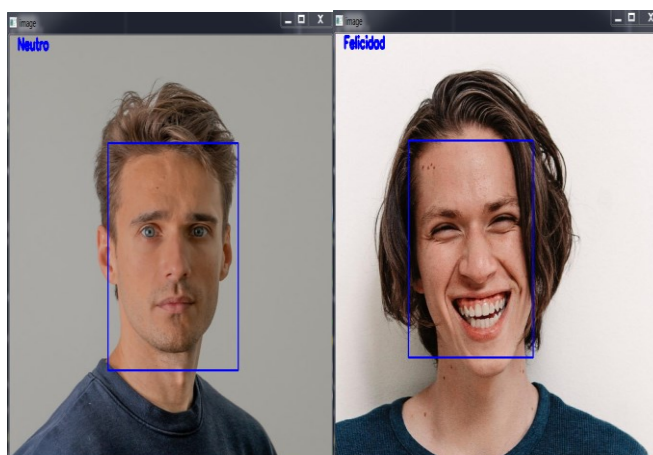


Figura 4.22 Resultados obtenidos de la predicción usando el algoritmo de Haar cascadas

4.9. RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE RASGOS FACIALES

Se observa claramente que en el algoritmo de rasgos faciales (ver Figura 4.23 y Figura 4.24), se utilizaron imágenes de mayor precisión, donde los actores son Lizbeth Merlo y Maria Guadalupe Zecua.



Figura 4.23 Resultados de la predicción usando el algoritmo de rasgos faciales (Lizbeth)



Figura 4.24 . Resultados de la predicción usando el algoritmo de rasgos faciales (Guadalupe)

4.10. RESULTADOS DE LA PREDICCIÓN USANDO EL ALGORITMO DE FASTAI-RED NEURONAL DE APRENDIZAJE PROFUNDO

En la red neuronal de aprendizaje profundo (ver Figura 4.25 y Figura 4.26), eran imágenes de baja resolución (48x48 píxeles).

Los tiempos de procesamiento, aprendizaje, clasificación y de predicción, fueron mayores en la red de aprendizaje profundo y con mejores resultados en la predicción.



Figura 4.25 Resultados de la predicción usando el algoritmo fastai-red neuronal de aprendizaje profundo(Lizbeth)

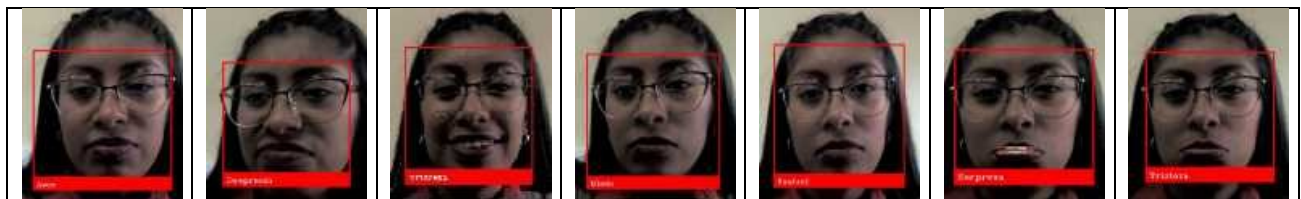


Figura 4.26 . Resultados de la predicción usando el algoritmo fastai-red neuronal de aprendizaje profundo(Guadalupe)

CAPITULO V. CONCLUSIONES

Maria Guadalupe Zecua Camacho: En conclusión, puedo decir que durante la realización de la tesis amplié mis conocimientos profesionales, en cuanto a análisis y comprensión del campo, además de cumplir con los objetivos para que este proyecto sea exitoso, por otro lado, comprender los diferentes algoritmos pueden hacer predicciones precisas o requerir recursos para funcionar y obtener resultados óptimos. Actualmente la aplicación de la Inteligencia Artificial con las emociones es una de las ramas denominada Inteligencia Emocional Artificial y nos permite conocer el estado emocional de una persona, por ello se decidió realizar el análisis y clasificación de imágenes emocionales con tres algoritmos que serán analizados, permitiéndonos observar cuál de los tres tiene mejores predicciones, y así clasificar las emociones automáticamente. El algoritmo de Haar cascada, implica recopilar una gran cantidad de imágenes positivas que contienen el objeto deseado y una cantidad de imágenes negativas que no contener el objeto deseado, Utilizando estas imágenes para el entrenamiento, el algoritmo aprende las características distintivas de un objeto y crea un modelo de clasificación, mientras que el algoritmo de rasgos faciales con DLIB es un detector de puntos de referencia facial con un modelo previamente entrenado. Dlib se utiliza para estimar ubicaciones en 68 coordenadas (x, y) que Mapea puntos faciales en la cara de una persona, los puntos se identifican a partir de un modelo previamente entrenado que detecta rasgos faciales con mayor énfasis. Ambos algoritmos devuelven datos que no son muy favorables por diferentes razones, su resolución, ambos dependen de otros factores o configuraciones, la posición de la cara, pero para el algoritmo de red neuronal de aprendizaje profundo fastai, crea una de las redes neuronales convolucionales (CNN), que es una red multicapa diseñada específicamente para la clasificación de imágenes y el reconocimiento facial. En cuanto a los resultados de predicción de emociones, es más satisfactorio porque siendo una imagen de baja resolución, detecta y analiza emociones, obteniendo mejores resultados. Esta tesis demuestra que, aunque los tres tienen diferentes complicaciones y dificultades, van más allá de los resultados de las bases de datos y amplían el conocimiento de aquellos algoritmos predichos en función de su elaboración.

Lizbeth Merlo Hernández: En lo personal puedo decir que este proyecto me dejó un aporte a mis conocimientos más comprendidos en cuanto a dirección de objetivos en este caso al análisis y modelado mediante la captación de sentimientos ocultos en pequeñas oraciones del lenguaje humano, puedo concluir que los objetivos han sido cumplidos satisfactoriamente, el trabajo realizado ha tratado de integrar un sistema capaz de reconocer e identificar rostros de acuerdo a una base de datos. Para ello, se propuso el estudio y evaluación de diferentes técnicas de reconocimiento y posteriormente se realizaron las adaptaciones necesarias para que el algoritmo elegido sea capaz de incorporar nuevos usuarios o identificar a los ya existentes, cabe mencionar que se ha desarrollado un sistema de reconocimiento facial robusto, rápido y con un coste de implantación dentro de un sistema domótico sustancialmente económico, haciendo énfasis en el apartado de los resultados es satisfactorio apreciar como los tiempos de cómputo para un número de usuarios entorno a varios supera los 2 segundos en medio tiempo más que suficiente para un sistema de reconocimiento, es importante mencionar que el estudio del reconocimiento facial por medio de una computadora siempre será un área de investigación que continua en pleno desarrollo por los avances que se van obteniendo en la AI, por otro lado la emulación del proceso de reconocimiento facial mediante sistemas artificiales ha sido un desafío para algunas disciplinas de nuestra área como la informática, la visión artificial, entre otras. En condiciones controladas, estos sistemas tienen la ventaja de ser rápidos, precisos aclarando la precisión es posible medirla en el rendimiento, es decir se debe de llevar a cabo un estudio y mucho trabajo para poder lograr llegar a la precisión decidida y correcta para los buenos resultados de nuestros estudios, sin embargo en condiciones no controladas me quedo con el aprendizaje de que se enfrentan con diferentes problemas relacionados con los datos a analizar como las variaciones de escalas, de orientación, expresión facial, condiciones de iluminación, oclusiones, entre otros problemas vistos en la implementación del mismo. En cuanto al último método utilizado, Redes Neuronales de Convolución, una técnica reciente en las tareas de reconocimiento ha quedado probada la importancia de reunir conjuntos de datos a gran escala. Este trabajo demuestra que esta técnica es capaz de superar las limitaciones de los métodos anteriores superando las tasas de acierto para bases de datos realistas, en realidad me llevo un buen aprendizaje en cuanto a la implementación y manejo de las técnicas utilizadas.

COMPETENCIAS DESARROLLADAS

1. INSTRUMENTALES

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos generales básicos.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas).
- Solución de problemas.
- Toma de decisiones.

2. INTERPERSONALES

- Capacidad crítica y autocrítica.
- Trabajo en equipo.
- Habilidades interpersonales.
- Capacidad de trabajar en equipo interdisciplinario.
- Capacidad de comunicarse con profesionales de otras áreas.
- Habilidad para trabajar en un ambiente laboral.
- Compromiso ético.

3. COMPETENCIAS SISTÉMICAS

- Capacidad de aplicar los conocimientos en la práctica.
- Habilidades de investigación.
- Capacidad de aprender.
- Capacidad de adaptarse a nuevas situaciones.
- Capacidad de generar nuevas ideas (creatividad).
- Liderazgo.
- Habilidad para trabajar en forma autónoma.

FUENTES DE INFORMACIÓN

- Alberca, A. S. (2022, 12 mayo). *La librería Numpy*. Aprende con Alf. Recuperado el 12 de Enero de 2024, de <https://aprendeconalf.es/docencia/python/manual/numpy/>
- Cardellino, F. (2021, marzo 20). *La guía definitiva del paquete NumPy para computación científica en Python*. freecodecamp.org.
<https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy-para-computacion-cientifica-en-python/>
- Carvalho, J., & Plastino, A. (2021). On the evaluation and combination of state-of-the-art features in Twitter sentiment analysis. *Artificial Intelligence Review*, 54(3), 1887–1936. <https://doi.org/10.1007/s10462-020-09895-6>
- Dertat, A. (2017, noviembre 13). *Applied Deep Learning - Part 4: Convolutional Neural Networks*. Medium. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>
- Dlib C++ Library*. (s/f). Recuperado el 12 de enero de 2024, de <http://dlib.net/>
- Fecha y hora. Python*. Bartolomé Sintés Marco. *Www.mclibre.org*. (s/f). Recuperado el 12 de febrero de 2024, de <https://www.mclibre.org/consultar/python/lecciones/python-biblioteca-time.html>
- Hariramani, V. (2020, abril 22). *Geeky baba*.
<https://vaibhavhariramani.blogspot.com/2020/04/a-full-guide-to-face-detection.html>
- Hariramani, V. (2020b, octubre 7). Face Identification using Haar cascade classifier.
GEEKY BAWA. <https://medium.com/geeky-bawa/face-identification-using-haar-cascade-classifier-af3468a44814>

imutils—Programador clic. (s/f). Recuperado el 12 de enero de 2024, de

<https://programmerclick.com/article/60071608406/>

Jaiswal, A. (2022, abril 1). Guide to Haar Cascade Algorithm with Object Detection

Example. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/>

KeepCoding, R. (2022, noviembre 25). *¿Qué es Matplotlib y cómo funciona? | KeepCoding*

Bootcamps. <https://keepcoding.io/blog/que-es-matplotlib-y-como-funciona/>

La guía definitiva del paquete NumPy para computación científica en Python. (2021, marzo

20). freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy-para-computacion-cientifica-en-python/>

Mittal, A. (2024, enero 18). Haar Cascades, Explained. *Analytics Vidhya*.

<https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

Os—Interfaces misceláneas del sistema operativo—Documentación de Python—3.10.13.

(s/f). Recuperado el 12 de enero de 2024, de

<https://docs.python.org/es/3.10/library/os.html>

Parimala, M., Swarna Priya, R. M., Praveen Kumar Reddy, M., Lal Chowdhary, C., Kumar

Poluru, R., & Khan, S. (2021). Spatiotemporal-based sentiment analysis on tweets for risk assessment of event using deep learning approach. *Software: Practice & Experience*, 51(3), 550–570. <https://doi.org/10.1002/spe.2851>

Patel, K., Mehta, D., Mistry, C., Gupta, R., Tanwar, S., Kumar, N., & Alazab, M. (2020).

Facial sentiment analysis using AI techniques: State-of-the-art, taxonomies, and challenges. *IEEE access: practical innovations, open solutions*, 8, 90495–90519.

<https://doi.org/10.1109/access.2020.2993803>

ResearchGate. (s/f). Recuperado el 12 de febrero de 2024, de

https://www.researchgate.net/publication/341230468_Facial_Sentiment_Analysis_Using_AI_Techniques_State-of-the-Art_Taxonomies_and_Challenges/link/5eb9f9db299bf1287f7fbd6f/download?tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19

Rosebrock, A. (2017, abril 3). Facial landmarks with dlib, OpenCV, and Python.

PyImageSearch. <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

Tanwar, K. (2021, abril 21). Introduction to fastai v2. *Analytics Vidhya*.

<https://medium.com/analytics-vidhya/introduction-to-fastai-v2-902520c9466>

Zhang, Y., Tiwari, P., Song, D., Mao, X., Wang, P., Li, X., & Pandey, H. M. (2021). Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis. *Neural Networks: The Official Journal of the International Neural Network Society*, 133, 40–56. <https://doi.org/10.1016/j.neunet.2020.10.001>

Zhou, J., Tian, J., Wang, R., Wu, Y., Xiao, W., & He, L. (2020). SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis. *Proceedings of the 28th International Conference on Computational Linguistics*.

ANEXOS

ANEXO A

SISTEMA INTERACTIVO

El sistema interactivo desarrollado para este proyecto se ha realizado con una base de datos propia, para su consulta acudir a Anexo. La dificultad que presentan estas imágenes es mayor que las anteriores dadas las diferentes condiciones del ambiente y propias de los sujetos: La posición de la cara, la iluminación, la expresión facial (muecas, sonrisas...) y apariencia (gafas, barbas...). El método elegido para la implementación del sistema es el LBP, el sistema pretende la evaluación del update por ello, el experimento consta de dos partes. La primera de ellas es la de notificar que el sujeto es desconocido al modelo de partida. La segunda parte, tras actualizar el modelo, consiste en evaluar el reconocimiento de emoción del usuario. La Figura 1 Anexo A Diagrama de flujo del funcionamiento del sistema interactivo muestra el estado de actividad del funcionamiento que tiene dicho sistema de reconocimiento de emociones de los usuarios.

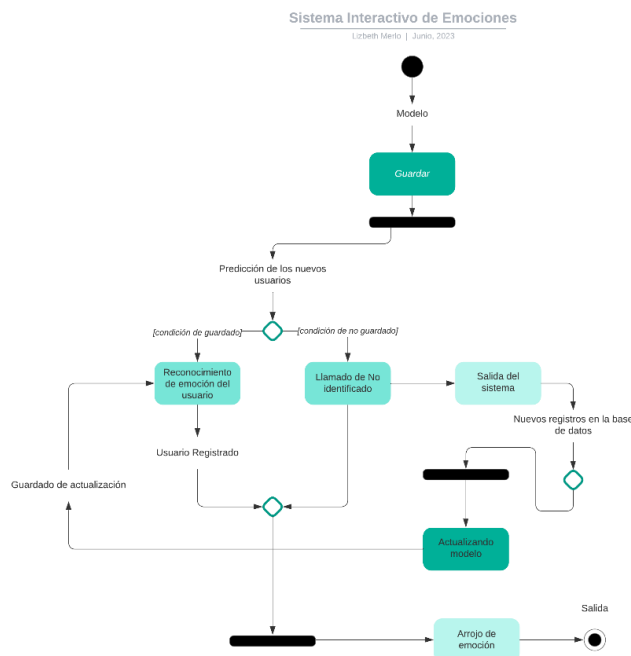


Figura 1 Anexo A Diagrama de flujo del funcionamiento del sistema interactivo

Tras la experimentación con datos reales se puede concluir que tiene algunos problemas lógicos dentro de las capacidades que ofrece el método y que cuando es un sujeto nuevo lo notifica bastante bien, sin embargo, cuando las imágenes no son controladas (gestos variados, iluminación cambiante...) empieza a fallar como se visualiza en la Figura 2 Anexo A. Matriz de confusión del sistema interactivo, en la entrada de un usuario nuevo dando de resultado una matriz de confusión del sistema en el reconocimiento de la emoción del usuario.

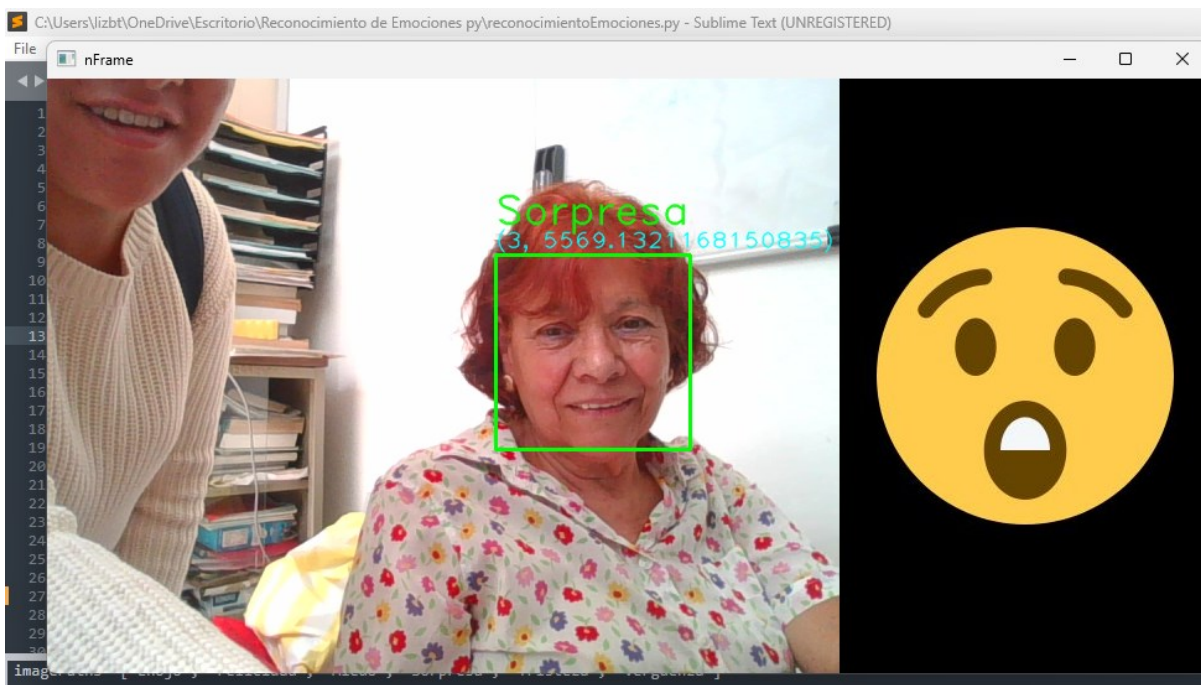


Figura 2 Anexo A. Matriz de confusión del sistema interactivo

ANEXO B

OpenCV proporciona enlaces en los que se describen algunos tutoriales para principiantes. Todos los códigos proporcionados son una parte regular y básica en los lanzamientos de OpenCV. Existe un listado de tutoriales proporcionado por el mismo. Se realizaron y ampliaron aquellos más interesantes para el presente trabajo.

TUTORIAL 1

En este tutorial se aprendió a cargar y mostrar una imagen por pantalla. Sus objetivos eran:

- Cargar la imagen usando funciones propias del entorno.
- Crear un entorno de programación utilizando OpenCV como la principal biblioteca de trabajo para el reconocimiento.
- Conseguir mostrar cualquier imagen por pantalla.

Además, por motivación propia se mostraron imágenes y vídeo capturadas desde una cámara web.

TUTORIAL 2

Con los conocimientos adquiridos en el anterior tutorial, se trató de modificar una imagen tomando en cuenta la comparación con un emoji siendo la principal función del sistema. Una vez realizada la transformación, se aprendió cómo guardarla en una carpeta. También se desarrolló un pequeño algoritmo que permitía mostrar simultáneamente el video capturado por la webcam con su respectiva captación de emoción definida en el emoji y su propia definición de la emoción al instante.

ANEXO C

EXPERIMENTO 1: THE ORL DATABASE OF FACES

MODELO ENTRENANDO "EigenFaces"

```
Entrenando ( EigenFaces )...  
Tiempo de entrenamiento ( EigenFaces ): 86.45906829833984
```

MODELO ENTRENANDO "FisherFaces"

```
Entrenando ( FisherFaces )...  
Tiempo de entrenamiento ( FisherFaces ): 37.59746837615967
```

MODELO ENTRENANDO "LBPH"

```
Entrenando ( LBPH )...  
Tiempo de entrenamiento ( LBPH ): 1.8892223834991455
```

Figura 3 Anexo C