

Informe de Proyecto: Detección de Emociones Faciales

Manuel Auqui, Leoni Guambo, Mayerli Méndez, Jorge Ortiz, Luis Valencia Escuela Politécnica Nacional (EPN), Quito - Ecuador

Resumen – En este documento se muestra los pasos para realizar el reconocimiento de emociones (enojo, felicidad, sorpresa, tristeza, disgusto y neutral) usando Python y OpenCV, para ello se usa métodos como EigenFace, FisherFaces y LBPH que se utilizan de igual forma para el reconocimiento facial.

I. INTRODUCCIÓN

Lo primero que se va a realizar es capturar los rostros en los que se reflejen las distintas emociones: enojo, felicidad, sorpresa, tristeza, disgusto y neutral cada uno de ellos serán almacenados en carpetas diferentes. Luego procederemos a entrenar los reconocedores con 3 métodos: EigenFace, FisherFace y LBPH. Finalmente se prueba el reconocimiento de cada una de las emociones entrenadas, además se implementa emojis que indican la emoción reconocida.

II. CONOCIMIENTO TEÓRICO

a) Eigen Faces

Es una colección de vectores cuando se implementa en un problema de visión artificial de reconocimiento de rostros humanos. Los vectores propios se derivan de la matriz de covarianza de la distribución de probabilidades en el espacio vectorial de alta dimensión de las imágenes faciales. Las mismas caras forman un conjunto base de todas las imágenes que son usadas para construir la matriz de covarianza. La clasificación se puede lograr comparando como se representan las caras por el conjunto base.

b) Fisher Faces

Es un método que se encarga del reconocimiento de caras haciendo referencia como se refleja la luz y las expresiones faciales. Este tipo de algoritmo maximiza la relación entre la división de las clases y la distribución intra-clases. Este algoritmo clasifica y reduce la dimensión de las caras utilizando el método discriminante lineal de Fisher y PCA. Este método crea una proyección lineal que maximiza las diferentes imágenes de caras proyectadas.

Este trabajo fue apoyado por Auqui Manuel, Guambo Leoni, Méndez Mayerli, Ortiz Jorge y Valencia Luis.

c) Local Binary Pattern

Es un operador de textura simple pero muy eficiente que etiqueta los píxeles de una imagen mediante el umbral de la vecindad de cada píxel y considera el resultado como un número binario. Debido a su poder discriminativo y simplicidad computacional, el operador de textura LBP se ha convertido en un enfoque popular en varias aplicaciones. Puede verse como un enfoque unificador de los modelos estadísticos y estructurales tradicionalmente divergentes del análisis de textura.

III. DESARROLLO

A. Creación de la base de datos con las emociones

Para realizar el dataset las emociones que queremos ejecutamos el código denominado “capturandoRostro.py” en donde encontramos las importaciones de las librerías “cv2, os e imutils”. Utilizamos variables emotionName que contendrán cada una de las emociones a utilizar alojadas a una carpeta que se creará denominada “Data” ahí es donde cada una de las 200 imágenes serán contenidas en su correspondiente directorio. La lógica del código consiste en crear los directorios y que cada imagen a cargar logre identificar el rostro del usuario dependiendo de la emoción a realizar Fig 1. Los directorios de las emociones que se lograron alojar en los directorios lo podemos ver en la imagen Fig. 2. Las fotos que logramos capturar se alojan en el directorio neutral por ejemplo como lo podemos ver en la imagen Fig. 3.

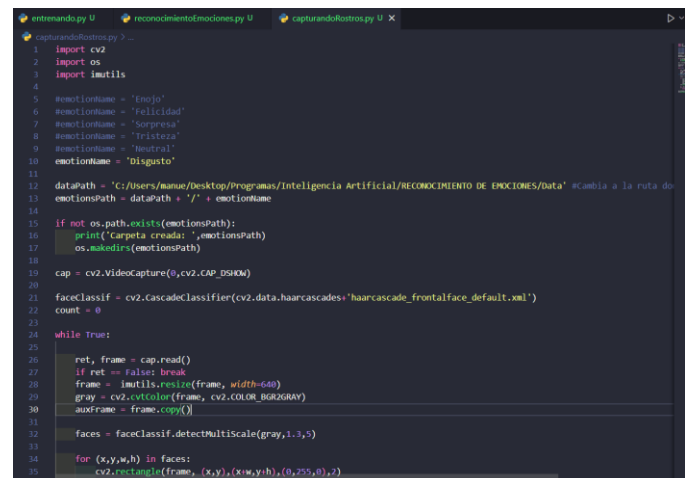


Fig. 1. Script capturandoRostros.py.



Fig. 2. Directorios alojados en "Dataset".

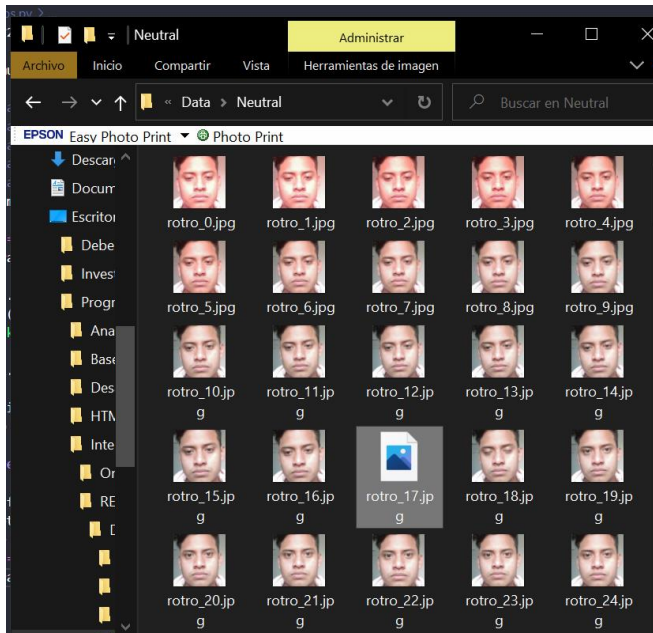


Fig. 3. Fotos alojadas en el directorio Neutral

B. Entrenamiento

Se va a realizar el entrenamiento con los tres métodos que OpenCV ofrece, para ello se requiere optimizar el código anterior para entrenar todos los métodos en una sola ejecución, esto nos servirá para realizar pruebas sobre cada uno de ellos. Se crea un script llamado entrenando.py como muestra la Fig.4.

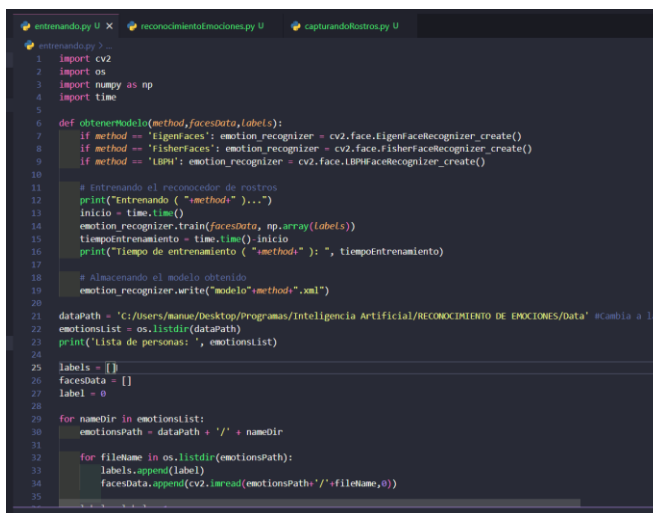


Fig. 4. Script entrenando.py

Una vez entrenado se llama a la función obtenerModelo se especifica el método que se utilizara. En este caso se va a realizar el entrenamiento con los 3 métodos como muestra la siguiente Fig. 5.

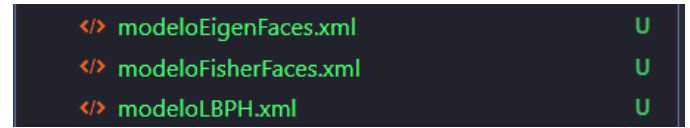


Fig. 5. Métodos de entrenamiento.

C. Probando Funcionamiento de EigenFaces, FisherFaces y LBPH.

Una vez que se ha realizado los entrenamientos es necesario probar el funcionamiento por ello se crea un script: reconocimientoEmociones.py como muestra la siguiente Fig. 6.

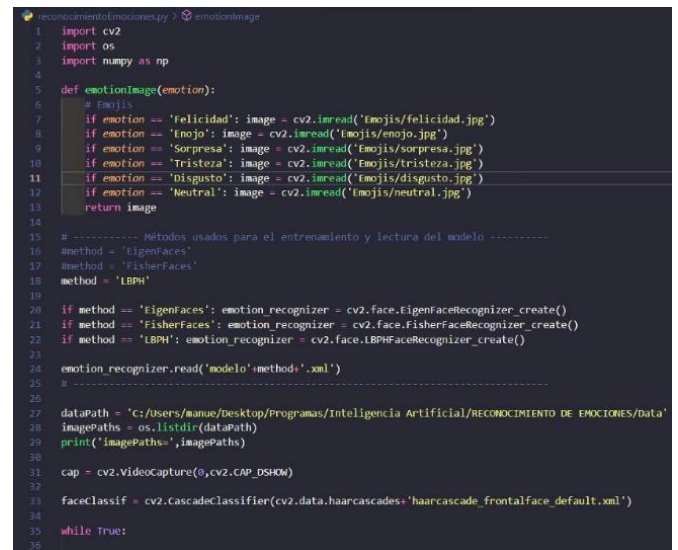


Fig. 6. Script reconocimientoEmociones.

Ahora se crea una carpeta Emoji en donde se coloca los emojis que se desee para que se identifique y muestre la emoción que corresponda como muestra la siguiente Fig.7.

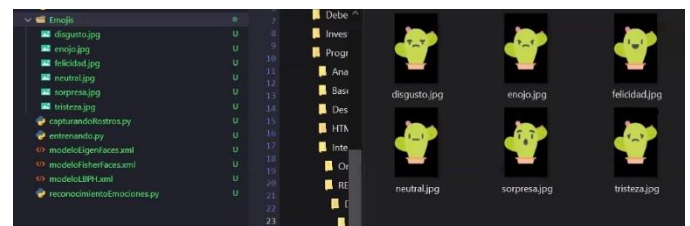


Fig. 7. Emojis

Una vez ejecutado los datos de entrenamiento en la siguiente sección es donde comprobaremos el funcionamiento del programa por medio del modelo más factible según las ejecuciones realizadas.

IV. RESULTADOS OBTENIDOS

Como resultados obtenidos ejecutamos el archivo “reconocimientoEmocione.py” donde hacemos referencia a las imágenes que se utilizaron para crear las emociones en sus debidos directorios, ahí podemos escoger que tipo de modelo queremos aplicar (Fisher faces y LBPH Eigen Faces), todo en función de las emociones y la cámara que capta las reacciones de la persona y relaciona el tipo de emoción como se ven las imágenes.

Verificación con el método LBPH se realiza las pruebas previas con todas las emociones como muestra la Fig. 8, Fig. 9, Fig. 10, Fig.11, Fig.12.,Fig 13.

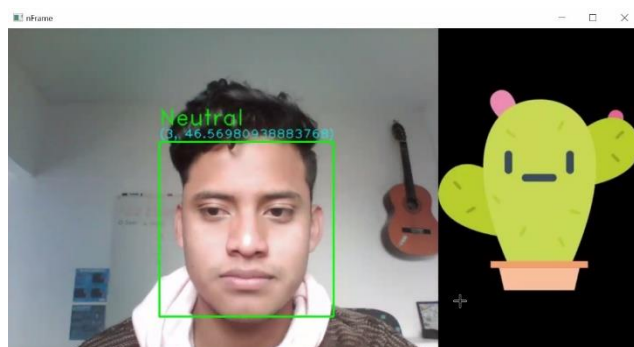


Fig. 8. Emoción neutral

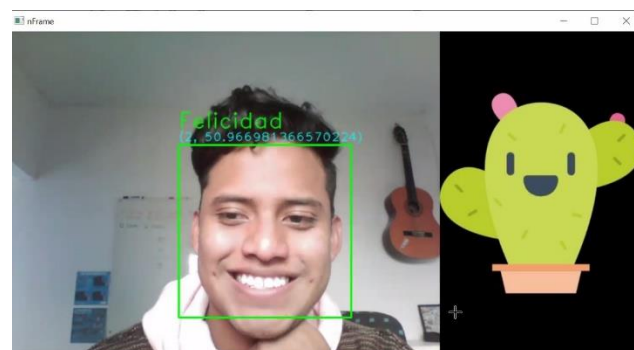


Fig. 9. Emoción Felicidad

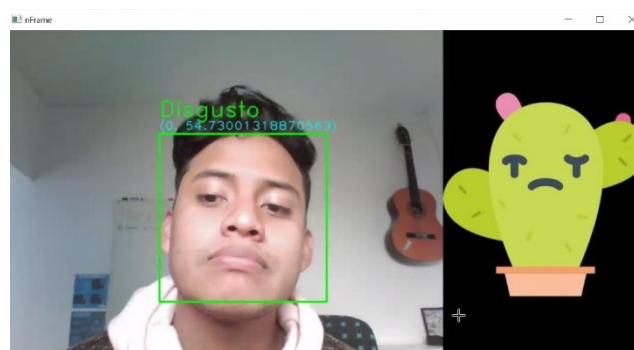


Fig. 10. Emoción de Disgusto

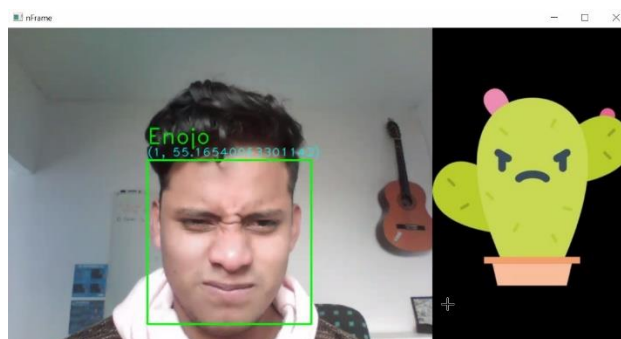


Fig. 11. Emoción Enojado

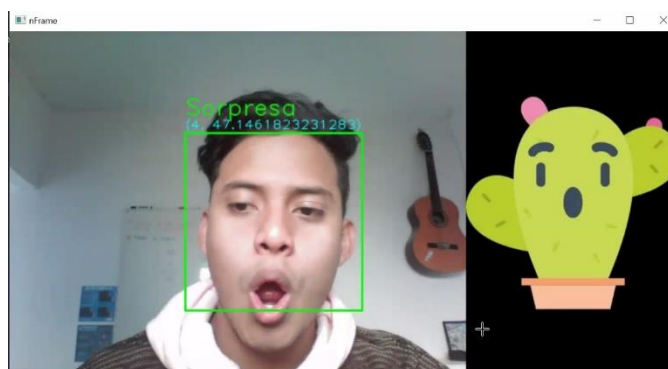


Fig. 12. Emoción Sorpresa

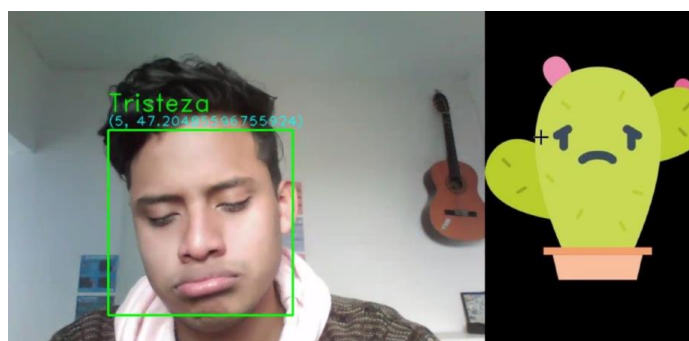


Fig. 13. Emoción Tristeza

La verificación con los demás métodos el resultado es similar siendo el método LBPH y Eigen Faces los más efectivos.

V. CONCLUSION

- La implementación de este programa se realizó con el uso de ciertas librerías como OpenCV e Imultis . Donde OpenCV ayudara con el estudio y despliegue de imágenes mediante las características de pixeles, dimensiones, y colores por medio del análisis de imágenes o el estudio en video de captura de rasgos faciales por medio de una cámara web. Imultis es una librería para el procesamiento de imágenes como rotación, dimensiones de imágenes y la examinación de cada imagen que esta emparejado con OpenCV y

Matplotlib.

- Para el desarrollo de este programa se basó en 3 fases donde primero se capturará datos fotográficos y almacenarlos en carpetas con su respectiva clasificación, donde el segundo paso será el entrenamiento de varios métodos con la utilización de los datos de imágenes y último paso en el despliegue del resultado mediante el testeo y prueba de cámara.

VI. ENLACE DE GITHUB

<https://github.com/ManuEly19/Reconocimiento-de-emociones>

VII. ENLACE DE VIDEOS

Video de Explicación y Funcionalidad:

<https://youtu.be/n-Qi77-4QCK>

Video Manual de Usuario:

https://youtu.be/lpGNHv_qRFs

VIII. REFERENCIAS

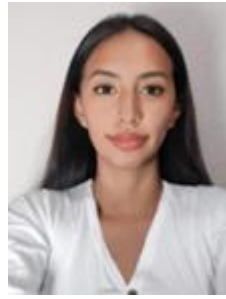
- [1] G. Solano, «OMES,» 31 Agosto 2022. [En línea]. Available: <https://omes-va.com/reconocimiento-de-emociones-opencv-python/>.

IX. BIOGRAFÍAS



Manuel Auqui, nació en Quito-Ecuador el 17 de febrero de 2002. Realizó sus estudios secundarios en la Unidad Educativa Técnica Fiscal “Arturo Borja”. Actualmente cursa sus estudios en Tecnología Superior en Desarrollo de Software en la Escuela Politécnica Nacional. Además, es miembro de la irrelevante organización del Club de los F.

Áreas de interés: Desarrollo Fronted y Backend, Organización y Liderazgo, autosaboteo y música.
(manuel.auqui@epn.edu.ec)



(leoni.guambo@epn.edu.ec)

Leoni Guambo, nació en Quito-Ecuador el 1 de septiembre del 2001. Realizó sus estudios secundarios en el Colegio Don Bosco de la Kennedy. Actualmente estudia en la Escuela Politécnica Nacional en la carrera Tecnología Superior en Desarrollo de Software y está cursando el cuarto semestre. Áreas de interés: diseño web y de interfaces, soporte técnico y mantenimiento de equipos.



(mayerli.mendez@epn.edu.ec)

Mayerli Méndez, nació en Quito-Ecuador el 14 de febrero de 2002. Realizó sus estudios secundarios en la Unidad Educativa “Consejo Provincial de Pichincha”. Actualmente estudia en la Escuela Politécnica Nacional en la carrera Tecnología Superior en Desarrollo de Software y está cursando el cuarto semestre.

Áreas de interés: diseño web y de interfaces soporte técnico y mantenimiento de equipos.



(Jorge.ortiz@epn.edu.ec)

Jorge Ortiz, nació en Ambato-Ecuador el 23 de mayo del 2000. Realizó sus estudios secundarios en el Colegio Nuestra Señora del Rosario. Está ejerciendo su carrera en Tecnología Superior en Desarrollo de Software en la Escuela Politécnica nacional desde el 2020 y está en el 4to nivel de dicha carrera.

Áreas de interés: programación, bases de datos, deportes, arquitectura y mantenimiento de computadores y ofimática.



(Luis.valencia@epn.edu.ec)

Luis Valencia, nació en Quito-Ecuador el 5 de marzo de 1998. Realizó sus estudios secundarios en el Colegio Nacional Vicente Rocafuerte. En la actualidad estudia en La Escuela Politécnica Nacional la carrera tecnológica en Desarrollo de Software en 4to semestre.

Áreas de interés: Desarrollo de páginas web, seguridad informática, testeo de programas, computación y las tecnologías de la información.