

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
DEPARTAMENTO DE CIENCIAS DE LA COMUNICACIÓN
SOFTWARE



MATERIA:

Vinculación con la sociedad.

Tema:

Clasificador de emociones.

Nombres:

Alexander Maila

Erick Criollo

Stalin Rivera

Henry Tiamba

Docente:

Omar Quimbita

Índice

Índice	2
Objetivos de aprendizaje	3
Objetivo general	3
Objetivos específicos	3
Introducción	3
Procedimiento	3
Resultado	5

Objetivos de aprendizaje

Objetivo general

Desarrollar una aplicación que mediante la recolección de imágenes de los usuarios y mediante el uso de inteligencia artificial sea capaz de clasificar emociones de dichos usuarios, mediante el uso del lenguaje de programación python.

Objetivos específicos

- Capturar imágenes de los rostros de los usuarios que utilizaran la aplicación con el fin de generar un conjunto de datos para el clasificador de emociones.
- Entrenar el modelo de inteligencia artificial en base al conjunto de imágenes faciales de los usuarios recolectadas .
- Ejecutar el programa de reconocimiento de emociones que funcione mediante los archivos de entrenamientos generados de acuerdo a los datos recolectados.

Introducción

Esta es una guía básica para ejecutar y generar los diferentes archivos necesarios para lograr ejecutar de manera satisfactoria la aplicación de reconocimiento de aplicaciones desarrollado en Python y OpenCV.

Procedimiento

Paso 1: Para poder llevar a cabo la ejecución del modelo es necesario instalar las siguientes librerías en nuestro entorno de trabajo. Para esto es necesario tener abierta la carpeta del proyecto en Visual Studio Code.

Una vez abierto el proyecto abrir una terminal e instalar las librerías mediante las siguientes sentencias.

```
pip install numpy  
pip install opencv-contrib-python  
pip install imutils
```

Paso 2: Dentro de la carpeta del proyecto se encuentran 3 archivos con los que debemos interactuar los cuales son:

```
emociones.py  
entrenamiento.py  
rostro.py
```

En la carpeta del proyecto es necesario crear una carpeta llamada “Data” y dentro de los archivos mencionados anteriormente se debe localizar la declaración “dataPath” y dentro de esta colocar la dirección de la carpeta “Data” que se acaba de crear como se ve puede apreciar a continuación:

```
dataPath = 'C:/Users/ckan1/Downloads/Vinculacion-main/Data'
```

Paso 3: El primer archivo Python que se debe ejecutar es el denominado “rostro.py” es necesario tener en cuenta que en este archivo es donde tomaremos captura de imágenes correspondientes a las emociones que se analizarán en este proyecto en este caso: enojo, felicidad, sorpresa, tristeza, neutral, disgusto. De este modo se puede tener control de las emociones que estamos cargando para el entrenamiento del modelo.

Para cada ejecución la cámara del dispositivo se encenderá y empezará a tomar imágenes del rostro del usuario y a la vez, dentro de la carpeta “Data” generará subcarpetas correspondientes a cada emoción definida en el código.



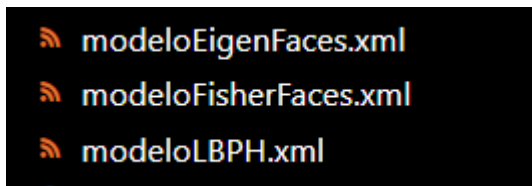
Paso 4: A continuación se debe ejecutar el archivo “entrenamiento.py” el cual mediante las imágenes alojadas en las subcarpetas de las emociones generará archivos xml que pertenecen a los tres tipos de entrenamiento que han sido desarrollados para este trabajo los cuales son:

EigenFaceRecognizer

FisherFaceRecognizer

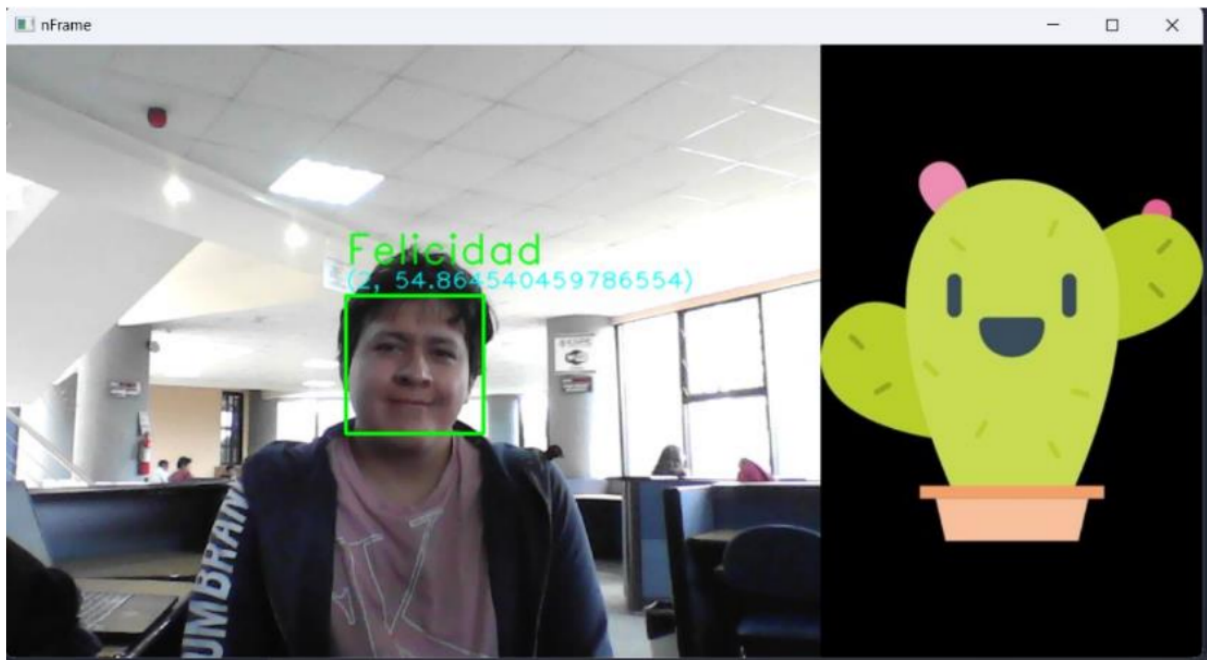
LBPHFaceRecognizer

Los cuales serán usados como base para el analizador de emociones.



Paso 4: Finalmente si los pasos anteriores se han realizado de manera correcta podremos ejecutar el archivo “emociones.py”, este es como tal el encargado de analizar las emociones del usuario a partir del modelo de entrenamiento xml generado en el punto anterior, cabe tener en cuenta que, este al hacer uso del resultado de los pasos anteriores depende exclusivamente del resultado de los mismos, es decir la captura de imágenes y el el entrenamiento del modelo debe haber sido realizado.

El clasificador de emociones ejecutará un frame que hará uso de la cámara del dispositivo para captar el rostro del usuario y generar un resultado a partir de los datos y el modelo de entrenamiento.



Resultado

El resultado final de este proyecto se trata de una aplicación que a partir de la captura de los datos faciales de un usuario y mediante el entrenamiento de modelos de reconocimiento facial, esta aplicación es capaz de arrojar un resultado de acuerdo a las expresiones del usuario, cabe mencionar que para tener un resultado el usuario debió haber guardado sus datos a manera de imágenes de modo que el modelo de entrenamiento los considere y ejecute.