

DISEÑO DEL DETECTOR DE EMOCIONES FACIALES:

MEMORIA

TODOR KRASIMIROV IVANOV

ALBA MEJÍAS COBOS

PINZHANG CHEN

ÍNDICE

PÁGINAS

| | |
|---|---------------|
| 1. OBJETIVO..... | 3 |
| 2. PLANTEAMIENTO DEL PROBLEMA..... | 4 |
| 3. RESUMEN..... | 5 |
| 4. REQUISITOS Y CONTENIDO..... | 6 - 7 |
| 5. GUÍA DE USO..... | 8 |
| 6. DESARROLLO..... | 9 - 11 |
| 7. PRUEBAS Y CONCLUSIONES..... | 12 |
| 8. BIBLIOGRAFÍA..... | 13 |

1. OBJETIVO

El objetivo principal de este proyecto es crear una inteligencia artificial que sea capaz de obtener la emoción predominante que muestra una persona a partir de una imagen de su rostro. Las emociones objetivo serán felicidad, tristeza, sorpresa y enfado.

2. PLANTEAMIENTO DEL PROBLEMA

Para imitar los resultados que puede dar un ser humano real desde su inteligencia natural, la inteligencia artificial ha de estar programada así tal que sea capaz de realizar los procesos que seguiría (subconscientemente) un humano para entender una emoción. La boca es una de las partes más representativas para esto pues, viendo si la persona sonríe, está “de morros”, etcétera, se puede saber su estado de ánimo en términos generales. Las cejas también juegan un importante papel en cierto tipo de emociones como por ejemplo la sorpresa o la tristeza.

Este detector se fijará en la boca y las cejas, es decir, solo sacará y utilizará características de la boca y de ambas cejas para determinar la emoción.

Ahora bien, ¿qué características son las más llamativas para detectar un estado de ánimo? Como seres humanos, además de hacerlo por comunicación, podemos detectar una emoción si nos fijamos en los gestos faciales, como pueden ser: ojos más o menos cerrados, boca más o menos cerrada o inclinada, cejas más o menos inclinadas... Al final, por la expresión facial somos capaces de entender, en general, cómo se siente una persona. Para crear la inteligencia artificial se medirán y procesarán dichas inclinaciones y aperturas.

Una vez hecho esto, no quedará más que “enseñar” a la inteligencia artificial que, con ciertas pendientes y niveles de apertura de los rasgos faciales, el ser humano tiende a expresar una emoción u otra.

3. RESUMEN

Para cumplir el objetivo, se han utilizado herramientas para el manejo y manipulación de imágenes que se encuentran en las librerías *OpenCV*, *dlib*, *imutils* de Python (entre otras). Con estas herramientas se detectan las caras de la imagen y, a su vez, los rasgos faciales de estas (ojos, cejas, mandíbula, boca, nariz) mediante puntos en la imagen que, posteriormente, se podrán tratar según los intereses buscados.

Se ha utilizado un *dataset* del cual se obtendrán ciertas características para entrenar un clasificador mediante la técnica de regresión lineal (en el entorno de desarrollo WEKA) para que resulte una función que determine el nivel de una emoción concreta en función de las características utilizadas.

Finalmente, se han obtenido un total de cuatro funciones, una por emoción, las cuales se utilizan para obtener valores numéricos de cada emoción, en función de las características extraídas del rostro y, mediante un árbol de decisión, determinar la emoción predominante.

4. REQUISITOS Y CONTENIDO

REQUISITOS:

Para utilizar este detector de emociones se requiere, además de tener una versión actualizada de Python, las librerías *dlib*, *imutils*, *matplotlib*, *numpy* y *opencv*, un compilador de C++ y *CMake* actualizado. Excepto el compilador de C++, la instalación de todos los demás recursos mencionados anteriormente se facilita en un archivo llamado *requirements.txt*. Basta con abrir el terminal de comandos y escribir el siguiente comando: `pip install -r *ruta absoluta de requirements.txt*`, por ejemplo, `pip install -r C:/Users/Desktop/requirements.txt`

La imagen a detectar deberá contener solamente un rostro, puesto que el programa no está implementado para funcionar correctamente con varios.

En general, se requiere que la calidad y resolución de la imagen sean lo más óptimas posibles, al igual que la distancia del foco al rostro no sea muy grande, para que el programa funcione con más eficacia. En cualquier caso, las imágenes en las que la boca se encuentra moderadamente abierta suelen clasificarse como sorpresa, puesto que en el *dataset* utilizado, las únicas caras que tienen la boca bastante abierta son las que expresan sorpresa.

CONTENIDO:

ConjEntrenamiento, *ConjPrueba* y *ConjValidacion* son directorios en los cuales solo almacenaremos imágenes. En el de entrenamiento se encuentran las 99 imágenes utilizadas para entrenar la inteligencia artificial, en el de prueba están las imágenes de las que queremos saber el estado de ánimo y en el de validación hay imágenes prueba cuyo nombre de archivo contiene la emoción *a priori* escrito como AN (angry), SA (sad), SU (surprised) y HA (happy).

Existen cuatro programas *.py* con distintas funcionalidades:

- *Entrenamiento.py* se utiliza para sacar los ficheros *.arff* que posteriormente se utilizarán para obtener las funciones predictivas del resto de programas.
- *SingleTest.py* se utiliza para predecir la emoción de una imagen, escribiendo la ruta en la que se encuentra.
- *MultipleTest.py* saca las emociones de todas las imágenes que se encuentran en una carpeta (*ConjPrueba*), una a una, cambiando de imagen pulsando cualquier tecla.
- *Validacion.py* es igual que el anterior, pero en lugar de mostrar las emociones lo que hace es comparar la emoción predicha con la emoción real (escrita *a priori* en el nombre de la imagen) e imprime los fallos, los aciertos y el porcentaje de acierto de predicción.

valuesSur, *valuesHap*, *valuesSad* y *valuesAng* contienen los valores de cada emoción del *dataset* en orden para cada imagen del conjunto, fragmentadas emoción por emoción a partir del archivo *DatasetValues.txt*.

requirements.txt es el archivo que utilizamos para instalar las librerías necesarias.

trainSur, *trainHap*, *trainSad* y *trainAng* son los ficheros .arff que se obtienen tras ejecutar por primera vez el programa de *Entrenamiento.py* y los que se utilizan para entrenar en WEKA y obtener la función de cada emoción, así como el árbol de decisión.

shape_predictor_68_face_landmarks.dat es el fichero que utiliza la función de *dlib* para crear el predictor de marcas faciales.

5. GUÍA DE USO

Para que muestre las emociones de varias imágenes, en la carpeta *ConjPrueba* se pueden meter tantas imágenes como se quiera. Se ha de ejecutar el programa *MultipleTest.py*, presionando cualquier tecla para ir avanzando imagen por imagen dentro de dicha carpeta.

Si se quiere detectar el estado de una sola imagen, se puede abrir el programa *SingleTest.py* y sustituir, al principio del programa, la ruta escrita en *#IMAGEN A LEER* por la ruta de la imagen deseada.

Para validar el porcentaje de acierto de emoción, se tendrán que almacenar tantas imágenes como se quiera en la carpeta *ConjValidacion* cuyo nombre tendrá que contener la emoción real de esta forma: AN (angry), SA (sad), SU (surprised) y HA (happy). Después, bastará con ejecutar el programa *Validacion.py*.

Si bien se desea comprobar el porcentaje de acierto con las imágenes utilizadas para el entrenamiento, solamente se ha de abrir el programa *Validacion.py* y cambiar en las primeras líneas el nombre de *ConjValidacion* por el de *ConjEntrenamiento*, puesto que las imágenes de esta carpeta ya contienen en su nombre la emoción real escrita como AN, SA, SU o HA.

6. DESARROLLO

Declaramos todas las librerías a utilizar y, posteriormente, abrimos un detector de caras y un predictor de marcas faciales (funciones implementadas en la librería *dlib*).

Utilizaremos un conjunto de 99 imágenes para el entrenamiento, almacenadas en *ConjEntrenamiento*.

Abrimos, una por una, las imágenes cuyas emociones se quieren detectar, se redimensionan, se convierten a escala de grises y se detectan los rostros existentes (solo 1 por imagen).

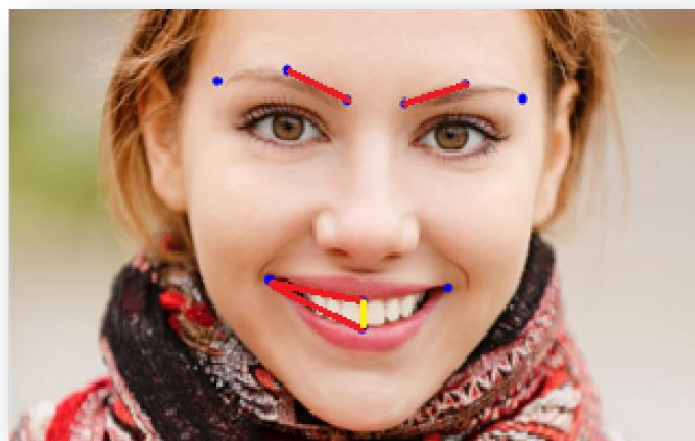
Se guardan, en diferentes listas, los puntos más significativos de la boca y de las cejas, obtenidos de la función *facial_landmarks*. Con ellos se puede saber con bastante exactitud la forma de la boca y cejas observando y comparando, por ejemplo, sus posiciones respecto al eje Y para determinar la curvatura.



Se declaran tres listas, una para los puntos de la boca, otra para los de la ceja derecha y otra para los de la ceja izquierda. Posteriormente, de todos los puntos obtenidos, solo se mantienen guardados los que se consideran de mayor interés tal y como se puede ver en la imagen. Con dichos puntos, se ha calculado la pendiente de la recta que pasa por dos de ellos y, al ser la cara prácticamente simétrica, solo se utilizan las pendientes de la mitad del rostro. Las cejas, aunque sean casi simétricas, se ha decidido mantener ambas pendientes para tener en cuenta las imágenes que están algo ladeadas y parezca que las cejas están en una posición diferente.

*Rectas rojas: pendientes.

*Recta amarilla: distancia.



Con esto, se hace un clasificador utilizando la técnica de regresión lineal con cada emoción que se ha escogido (felicidad, tristeza, enfado y sorpresa) para tener una función de cada emoción, la cual dependa de las características (pendiente, apertura) de cada imagen. Para ello, se han creado cuatro archivos .arff mediante el entrenamiento, uno por cada emoción, con las pendientes obtenidas de las imágenes de nuestro *dataset* y, como objetivo, el grado que tiene de dicha emoción (valor numérico). El grado de emoción lo extraemos del propio *dataset* según la foto que se esté analizando. Los cuatro ficheros .arff se crean en el programa de *Entrenamiento* y, en cada uno, se escriben las pendientes y el grado de emoción correspondiente a la imagen, cogido del *dataset*.

Cabe mencionar que se han tenido que apartar algunas imágenes del *dataset* para realizar el entrenamiento por exceso de sombras o porque no eran lo suficiente expresivas y provocaban que la ecuación resultante no fuera satisfactoria.

Al realizar la regresión lineal en los diferentes archivos, se ha obtenido una función por cada emoción:

$$\text{Angry} = 3.2488 * \text{pendMup1} + -0.0497 * \text{apertura} + 5.1385 * \text{pendEBL} + 1.6319$$

$$\text{Happy} = -3.6945 * \text{pendMup1} + -2.1962 * \text{pendMdown1} + 2.157$$

$$\text{Sad} = 3.4623 * \text{pendMup1} + -1.9049 * \text{pendMdown1} + -0.0864 * \text{apertura} + -2.5612 * \text{pendEBL} + 3.6316$$

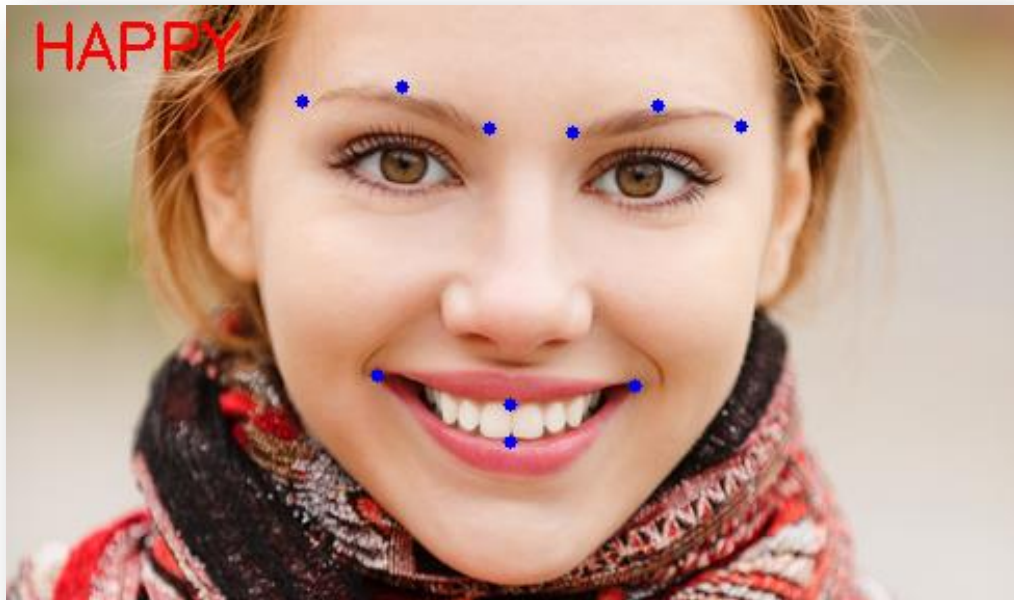
$$\text{Surprised} = 6.7817 * \text{pendMup1} + -4.0118 * \text{pendMdown1} + -0.0583 * \text{apertura} + -2.1956 * \text{pendEBR} + 1.2609$$

“PenMup1” se corresponde a la pendiente de la recta que une el punto de la comisura y el punto medio del labio superior y, por consiguiente, “pendMdown1” es la pendiente de la recta que une la comisura con el punto medio del labio inferior.

“Apertura” es la distancia entre los puntos medios de los labios. “PendEBR” y “pendEBL” corresponden a las pendientes de las rectas que unen el extremo interno y el punto medio de la ceja (EB = eyebrow, R = “right” y L = “left”).

Con estas ecuaciones ya se puede obtener el grado de las emociones a partir de las pendientes obtenidas de las imágenes. Los resultados se interpretan mediante un árbol de decisión, creado con el algoritmo J48 utilizando los resultados obtenidos y los valores numéricos dados en el *dataset* (en WEKA), el cual determina el estado de ánimo de la persona de la imagen. Sin embargo, se ha querido mostrar el grado de todas las emociones analizadas porque una persona no está en un solo estado de ánimo si no que, aunque haya uno que predomine, está en varios simultáneamente.

Al finalizar, el programa imprimirá los valores numéricos de cada emoción y mostrará la imagen con los puntos determinantes que se han observado, con la emoción predicha escrita arriba a la izquierda.



7. PRUEBAS Y CONCLUSIONES

Para comprobar la fiabilidad del programa se han introducido diferentes imágenes, algunas del propio *dataset* y otras obtenidas por internet, todas ellas con buena resolución, sin muchas sombras y el rostro a una distancia apropiada (tamaño carné).

Con las imágenes que hay en el directorio *ConjValidacion* se obtiene un porcentaje de acierto del 85.19%.

Utilizando el conjunto de entrenamiento *ConjEntrenamiento*, el porcentaje de acierto es del 87.88%.

Mencionado anteriormente, el problema esencial que baja la tasa de acierto es el hecho de que el *dataset* utilizado no es muy variado, por ello, se confunde bastante a una persona enfadada o incluso muy feliz con una persona sorprendida, puesto que en el *dataset* una apertura moderada de la boca siempre representa sorpresa.

Realizando un *dataset* con más variedad y con personas de todo el mundo, el porcentaje de éxito sería mayor.

Mediante lógica borrosa se podrían implementar más emociones que, a partir de los niveles de los valores numéricos de las cuatro principales que se han tenido en cuenta, predigan otra emoción como “preocupado” o “disgustado”.

Algunos ejemplos de la ejecución de pruebas:



8. BIBLIOGRAFÍA

Detector facial y predictor de marcas faciales en Python:

<https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/>

Dataset utilizado para el entrenamiento:

<http://www.kasrl.org/jaffe.html>