

SOLIDAttendance - Computer Vision

Para identificar rostros humanos se necesitan hacer 4 pasos principales:

1. Detección de rostro
2. Preprocesamiento de rostro
3. Reunir y aprender rostros
4. Identificación de rostros

Puede existir el caso de que no encuentre ninguna similitud con los rostros guardados anteriormente, de ser así se puede almacenar para después identificar a esa persona.

Detección de rostro

La detección del rostro la hice con una técnica llamada Haar-Cascade. Esta técnica fue creada principalmente para detectar patrones en general, pero con el paso del tiempo tomó mucha popularidad por su exitosa implementación al representar rostros.

La librería OpenCV tiene varios documentos XML en donde se especifican las medidas que debe de tomar para detectar las regiones del rostro.

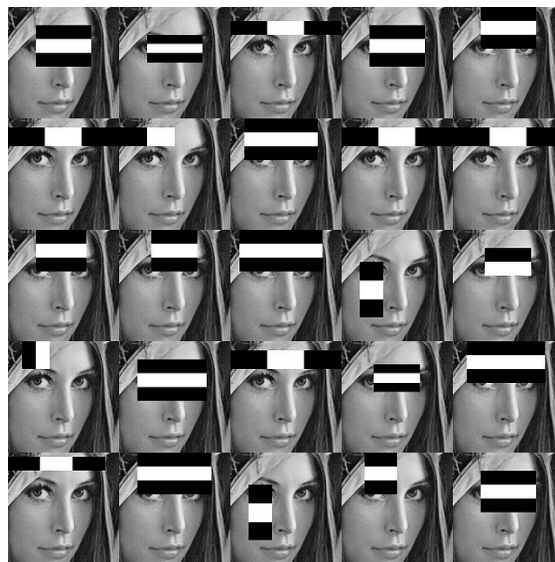


Fig 1. Regiones dibujadas que Harr-Cascade toma para representar rostros

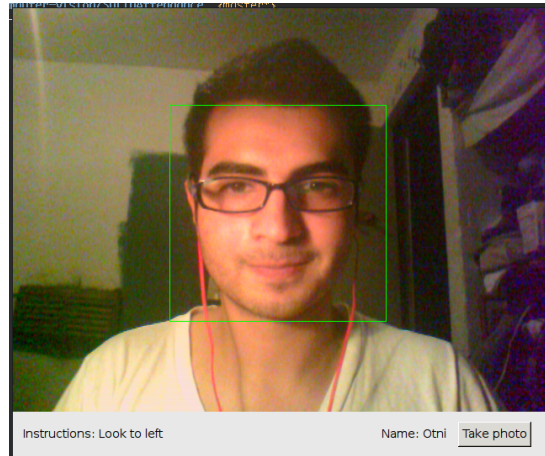


Fig 2. Reconocimiento de rostro desde el sistema.

Preprocesamiento de rostro

Una vez detectado el rostro, se tiene que preparar para poder pasarlo al algoritmo de reconocimiento. Lo que se hizo fué simplemente un recorte de la imagen y una normalización en escala de grises, para que así fuera lo más preciso posible.



Fig 3. Imágenes preprocesadas

Reunir y aprender rostros

Para tener un mejor reconocimiento, se escogió hacer el entrenamiento con 12 diferentes muecas (y de cada mueca se sacan 5 capturas), de tal manera que se puede reconocer que es cierta persona bajo diferentes circunstancias.

Al momento de capturar cada mueca, se van almacenando en un folder temporal. Hasta que se captura la última mueca se crea un folder por persona y se mueven las imágenes, esto por si el usuario quiere modificar su nombre o su cargo en la empresa.

Lo que se hace para entrenar el algoritmo es lo siguiente:

- Crear dos listas, una para las imágenes y otra para las etiquetas que servirán para poder identificar una vez que se haya reconocido el rostro.
- Listar y ordenar de menor a mayor los folders de las imágenes de entrenamiento
 - Al momento de crear el folder se nombra con un número partiendo desde el 0
- Recorrer cada folder y tomar todas las imágenes dentro del mismo
- Tomar cada una de las imágenes y convertirlo a un array bidimensional
- Agregar las imágenes del folder actual y a cada una de ellas asignarles un ID que corresponde al nombre del folder.
- Hacer este proceso para todos los folders.

Después, metemos las 2 listas a opencv de la siguiente forma:

```
self.recognizer.train(pictures,labels)
```

Con esto, ya tenemos listo nuestro reconocedor.

Identificación de rostros

Para identificar rostros, se utilizó un algoritmo llamado FisherFaces, el cual ya viene implementado en OpenCV.

Este método regresa dos valores, uno es la etiqueta la cual asocia el rostro. Esta nos servirá para traernos el nombre y el cargo desde la base de datos.

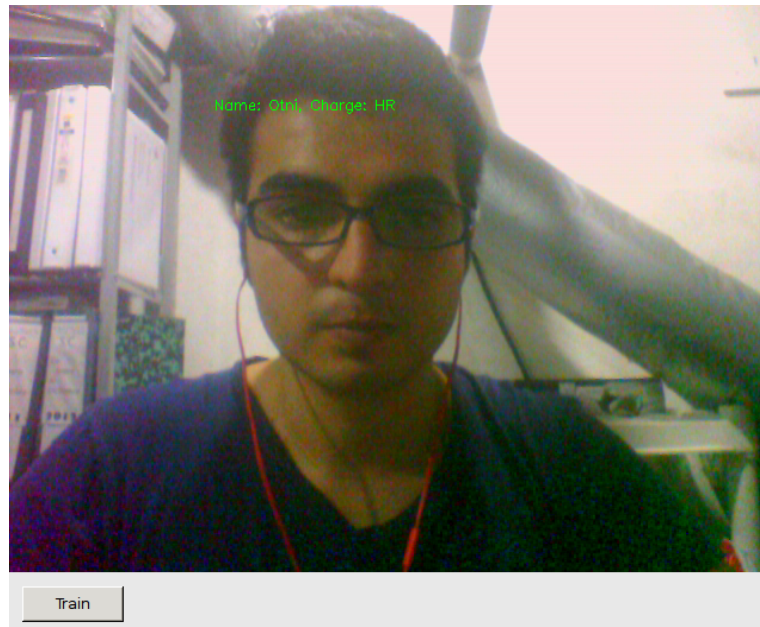


Fig. 4 Reconocimiento de rostro

En el caso de que no se haya entrenado el sistema con el rostro, la distancia será mucho más grande. Entonces se implementó un umbral para saber si ya se entrenó o no.

Se borró el usuario asignado a un servidor para probar esta característica.



Fig 5. Rostro identificado pero no reconocido de acuerdo a su distancia.

Conclusiones y trabajo futuro.

Se concluyó una exitosa implementación de identificación y reconocimiento de rostros con algoritmos existentes. Este proyecto solo es una de las muchas formas de implementar reconocimiento facial, una de ellas es etiquetar a personas automáticamente en las fotos dentro de una red social, sistemas de seguridad, etc.

Se plantean los siguientes aspectos para trabajar en un futuro:

1. Implementar otros factores para reconocer con mayor exactitud los rostros, por ejemplo identificación de ojos y las medidas de los mismos.
2. Exportar el proyecto a un sistema web.
3. Extender el soporte para múltiples cámaras.
4. Hacer reportes de asistencia y por cargo.

Referencias

Baggio, D. (2012), Mastering OpenCV with practical computer vision projects. Birmingham, Editorial: Packt Pub

OpenCV Documentation: docs.opencv.org