

Informe: Detector de caras en imágenes



Guillermo Berasategui

Asier Gurpegui

Urtzi Sotes

8/1/2019

Preprocesamiento de imágenes

Para el preprocesamiento de imágenes hemos realizado la normalización de las imágenes y hemos aumentado el conjunto de prueba realizando la operación de flip sobre las imágenes. Al aumentar el dataset hay más ejemplos para obtener las características por lo que tiene más datos para poder saber cuáles son las más importantes.

Extracción de características

Hemos extraído todas las características de las imágenes con lo que luego se hará el entrenamiento.

Entrenamiento del clasificador

Para encontrar las características importantes hemos entrenado con un AdaBoost las características. Para realizar este entrenamiento hemos probado con diferente número de clasificadores débiles y hemos llegado a la conclusión de que a partir de unas 30 clasificadores los resultados son bastante parecidos y se obtiene una precisión similar dando unos buenos resultados. Con 30 clasificadores obtenemos una precisión de entrenamiento de 0.98% mientras que el resultado óptimo se obtiene con 60 clasificadores con una precisión del 0.99%. En cambio con menos de treinta clasificadores los resultados detectando caras no eran buenos.

Por lo tanto hemos llegado a la conclusión de que 60 se obtiene el mejor resultado aunque el tiempo para el entrenamiento es mas elevado

(aproximadamente 11 minutos).

Hemos entrenado 5 clasificadores cada uno con un número de características entre 5 y las 60 características para así realizar la detección de caras en cascada.

Pruebas en tiempo real

Para detectar las caras en una imagen recorreremos con una ventana toda la imagen. Para cada ventana inicialmente extraemos 5 características y se predice con el primer clasificador. Si predice como cara se sigue con la detección en cascada aumentando el número de características con las que hemos entrenado anteriormente. Solo aquellas ventanas que se predigan como caras en todos los clasificadores se tomaran como posibles caras.

Una vez hemos obtenido las posibles caras se da solapamiento entre ventanas positivas. Para solucionar esto obtenemos la cara con mayor probabilidad y aplicamos la función de `intersection_over_union` con todas posibles caras que tienen solapamiento, y si la intersección entre ambas es mayor que 0.5 la cara con menor probabilidad se descarta porque posiblemente son la misma cara.

A continuación, dibujamos el rectángulo de las caras en la imagen, con el tamaño de la ventana actual para su futura visualización.

Por último reducimos y ampliamos la imagen para detectar las caras con otro tamaño, ya que la ventana es de tamaño fijo (24x24). Si la cara detectada es de un tamaño menor que nuestra ventana original a la hora de dibujar el rectángulo en la imagen original también hay que ajustarla a la imagen original.

Resultado de la pruebas

Las pruebas han sido realizadas con los clasificadores entrenados con el dataset1 por lo que no muy precisas.

Imagen original



Imagen con la detección



Bibliografia

<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

https://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_face_detection.html

https://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html