Schneiderman and Kanade: implementación

Ignacio Rodríguez Ferrero

Mayo 2023

1. Introducción

Schneiderman and Kanade (SnK) es un método de detección facial estadístico. En este los cuadrados de imagen pretendientes a ser clasificados son divididos en cuadrículas. Estos parches son clasificados mediante un método de *clustering* y finalmente se calculan sus probabilidades para comprobar si es una cara o no. Este proceso se repite en todos los cuadrados de la imagen disponibles, a través de una pirámide de la imagen para encontrar todas las caras a diferentes escalas.

2. Entrenamiento: clasificación y conteo

Para el entrenamiento se ha utilizado el conjunto de datos *Labeled Faces in the Wild*. En concreto se ha usado una versión que tiene las caras centradas, para que recortando solamente el cuadrado central, que siempre es del mismo tamaño se obtiene la cara. Luego esta es reescalada para que tenga un tamaño de 32×32 píxeles de escala de grises. Luego estas caras son separadas por una cuadrícula de 4 de lado y alto, obteniendo así 16 parches cuadrados de 8 píxeles de lado, todos correspondientes a partes clave de la cara 1.

Una vez obtenido el conjunto de datos de caras se concatena a este el de no caras, ya que por ahora el objetivo es entrenar un modelo de *clustering k-means*

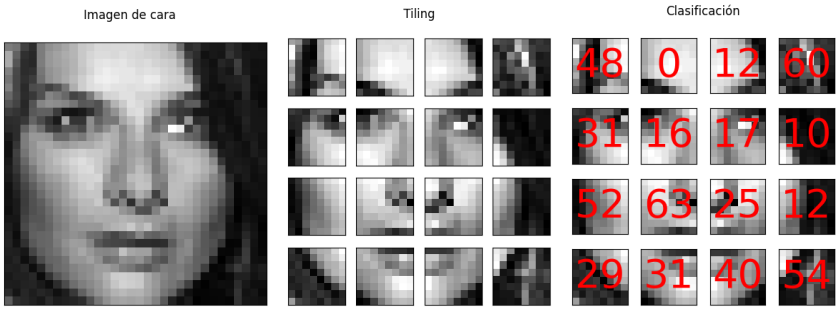

Figura 1: Proceso de recorte de imágenes, trozeado en parches y clasificación para el entrenamiento y la inferencia.

de 64 *clusters*. Para obtener el conjunto de datos de no-cara se usa el *dataset Caltech 256*, eliminando algunas categorías que potencialmente podrían tener caras como la de «gorros». Estas imágenes son troceadas también en parches cuadrados de 8 píxeles de lado. Un modelo de clustering es entrenado sobre estos datos.

El conteo se lleva a cabo sobre el conjunto de datos de caras extendido con *data augmentation*. Se obtiene un vector que cuenta el número de veces que cada parche entra en cada *cluster* y una matriz que además tiene en cuenta la posición que toma el parche dentro de las 16 posibles. Estas dos matrices son $P(q|cara)$ y $P(pos|q, cara)$.

Sobre el conjunto de datos no-cara se sigue un proceso similar, salvo porque se asume que $P(pos|q, nocara) \approx \frac{1}{nr}$, donde n es el número de parches en la cuadrícula y r es el número de *clusters*. Por lo tanto solo se tiene que clasificar y contar $P(q|nocara)$. Además, este conteo es extendido con otro conjunto de datos, *Natural Images Dataset*.

El resultado de este entrenamiento es un clasificador *k-means* y tres matrices de conteo: $P(q|nocara)$,$P(q|cara)$ y $P(pos|q, cara)$.

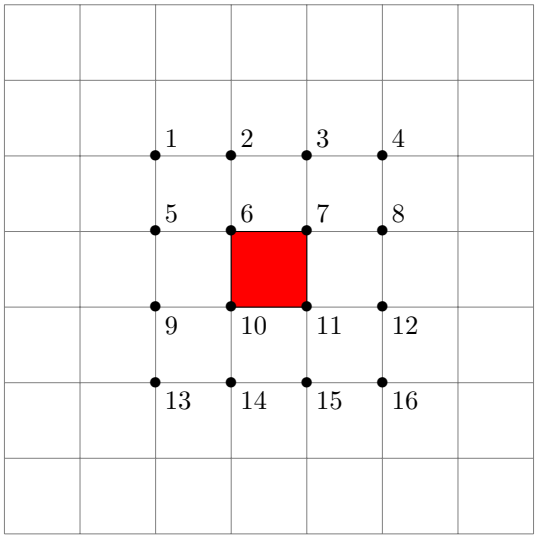

Figura 2: Un parche de píxeles (rojo) es utilizado cada vez que la ventana es centrada en uno de los 16 puntos marcados en la figura.

3. Inferencia

Para la inferencia se convierte la imagen a escala de grises y se atraviesa toda la pirámide, reduciendo su tamaño en un factor de 0,8 en cada nivel. Para cada uno de los niveles de la pirámide se trocea toda la imagen en una cuadrícula de 8 píxeles de lado. Estos parches son clasificados y una ventana deslizante se

Figura 3: Resultado de la búsqueda de caras en una multitud.

mueve en ambos ejes y, usando los conteos del entrenamiento normalizados, se calcula el *score* de esa ventana con la siguiente formula:

$$\Pi_{i=1}^{nr} \frac{P(pos|q, cara)P(q|cara)}{\frac{P(q|nocara)}{nr}}$$

El *score* obtenido es entonces comparado con un *score* mínimo que determinará si esa región es una cara o no. En caso de que lo sea se guardan las coordenadas de la ventana, teniendo en cuenta el nivel en la pirámide y el desplazamiento de la cuadrícula.

Una ventaja de cuadricular al imagen entera primero, para después clasificar y calcular desde ventanas de clases en vez de ventanas de píxeles es que se reduce el número de veces que se debe clasificar cada parche. En el método original cada vez que la ventana se mueve sobre los píxeles estos se deben clasificar, independientemente de si esos mismo píxeles ya han sido clasificados antes. En total, con el método de ventana deslizante sobre los píxeles un mismo parche de píxeles es clasificado 16 veces como se explica en la figura 2 (exceptuando casos cercanos al borde). Esto se reduce a solo 1 clasificación por parche con el método propuesto ya que este ya habrá sido clasificado antes.

Por otro lado, el método propuesto depende de que la cuadrícula se alinee con los mejores parches para encontrar una cara. Esta es la razón por la que se desplaza la cuadrícula y se repite la clasificación. El número de veces que se desplaza la cuadrícula puede ser desde 1 por píxel (8 veces) a ninguna vez. En

este trabajo se ha usado un desplazamiento de dos píxeles, desplazando entonces 4 veces la cuadrícula tanto vertical como horizontalmente.

4. Resultados

El sistema resultante es capaz de encontrar caras que miran casi directamente a cámara. No es especialmente robusto a rotaciones de la cara en ningún eje. Casi cualquier obstrucción a la cara resulta en un falso negativo. Asimismo, algunos parches con tonos más oscuros en la línea horizontal de los tres cuartos superior lanzan falsos positivos, además de las zonas con mucho ruido.

Un ejemplo de un resultado obtenido puede verse en la figura 3. Aquí se puede ver cómo, en el centro de la imagen, solo dos de las caras son detectadas. Sin embargo, aquellas que miran hacia el lado o están obstruidas son ignoradas completamente. A su vez, las zonas del fondo, donde la imagen es más ruidosa, se encuentran mucho falsos positivos.

Sin embargo, la ventaja de este sistema es una rápida inferencia, un fácil reentreno para expandir los datos y una gran portabilidad debido a la pequeña cantidad de parámetros.