Visión por Computador (2018-2019)

Grado en Ingeniería Informática y Matemáticas

Universidad de Granada

Descriptores HOG en la detección de peatones

Ignacio Aguilera Martos y Diego Asterio de Zaballa

17 de enero de 2019

Índice

1. Descripción del problema y enfoque de la resolución **3**

1.1. Problema a resolver 3

1.2. Descriptores HOG 4

1.2.1. Normalización Gamma 5

1.2.2. Computo del gradiente 5

1.2.3. Generación de los histogramas 5

1.2.4. Normalización de los bloques 6

1.3. Fases de la resolución 6

1.3.1. Procedimiento para entrenar el modelo SVM 6

1.3.2. Procedimiento para la detección de un peatón en una imagen de test 7

2. Valoración de los resultados **8**

3. Trabajo futuro: propuestas de mejora **8**

1. Descripción del problema y enfoque de la resolución

En la resolución de este trabajo nos hemos planteado el siguiente problema a resolver, dada una imagen de una persona andando por la calle (un peatón), ¿cómo podemos reconocer que es un peatón?

Esta misma cuestión fue planteada por Navneet Dalal y Bill Triggs en su paper "Histogram of Oriented Gradients for Human Detection" en el que explican el desarrollo de unos descriptores que aplicados a su dataset de personas obtienen unos resultados muy buenos en la detección de las mismas.

Estos descriptores son los descriptores HOG o descriptores de histogramas basados en gradientes. Como veremos a lo largo del trabajo se han empleado distintas variaciones a la hora de hallar los descriptores por Dalal y Triggs, quedándonos nosotros con las elecciones que han resultado más fructíferas para ellos en su análisis.

Para comenzar hay que saber que la detección de personas es un problema difícil de abordar y que actualmente resulta muy interesante en aplicaciones por ejemplo en coches, de forma que si detecta un peatón andando por delante del vehículo este entienda que tiene que detenerse.

Las herramientas usadas en este proyecto han sido:

- Python para la implementación.
- OpenCV y NumPy para las operaciones de los algoritmos.
- El módulo de SVM incluido en OpenCV para poder predecir la existencia o no de un humano en una imagen en base a los descriptores calculados.
- El dataset dado por los investigadores empleado en la elaboración de su artículo.

A continuación hacemos una descripción un poco más elaborada del problema propuesto.

1.1. Problema a resolver

El problema consiste en, dada una imagen que contiene o no un peatón, debemos determinar si es que lo contiene y además una región aproximada en la que se encuentra el mismo.

Para la elaboración de la solución del problema hemos utilizado varios ingredientes entre los que tenemos la fase de creación de descriptores y entrenamiento de la SVM y la fase de test.

En la fase de entrenamiento de los descriptores obtenemos un dataset ya formado por parte del grupo de investigación con parches de personas y con parches que no son personas todos con el mismo tamaño. De esta forma podemos obtener descriptores comparables entre sí para las imágenes positivas y negativas con lo que entrenaremos una SVM.

La segunda fase es la de test que comparte dificultad e importancia con la primera. Como explicaremos más en detalle posteriormente resolvemos el problema de dada una imagen, ¿cómo la tenemos que analizar para poder identificar si hay o no personas y dónde están? En secciones siguientes veremos que la solución adoptada ha sido una ventana deslizante con algunas modificaciones importantes cuyos parches extraídos se pasan a la SVM y se predice sobre ellos para poder detectar el área de la imagen en la que obtenemos respuestas positivas a la pregunta de si hay o no una persona.

1.2. Descriptores HOG

El problema mencionado anteriormente se ha enfrentado desde distintos puntos de vista lo que ha dado lugar a diversos descriptores. Nuestra implementación se basa en el paper de Navneet Dalal y Bill Triggs en el que desarrollan un descriptor basado en el gradiente de una imagen llamado descriptor HOG. El metodo se basa en calcular los gradientes de una imagen y a continuación construir histogramas locales basado en la orientación de dichos gradientes.

La idea básica sobre la que se desarrolla la técnica es que las características de forma y apariencia de un objeto de forma local se pueden resumir gracias a la distribución de la orientación de los gradientes de la imagen.

A continuación se describe la implementación de la técnica a grandes rasgos. En primer lugar cada imagen se procesa con una normalización Gamma tras un suavizado se calculan los gradientes de la imagen. A esto le sigue que la ventana de la imagen sobre la que se va a construir el vector de características se divide en celdas. Estas celdas son pequeñas regiones de la imagen. En cada una de estas celdas se calcula el histograma de la dirección de los gradientes que en ella se encuentran. Por ultimo estas celdas se agrupan en bloques para que sea posible normalizar los histogramas. Finalmente, el vector de características lo conforman los histogramas normalizados concatenados uno detrás de otro.

Conviene detallar cada uno de los pasos que se han seguido en el proceso de construcción del descriptor.

1.2.1. Normalización Gamma

La fase de normalización consiste en corregir los valores de la imagen gracias a una transformación del tipo

$$v_{\text{out}} = \alpha \cdot v_{\text{in}}^{\beta}$$

esta transformación se aplica a cada píxel de la imagen.

1.2.2. Computo del gradiente

Previo al computo del gradiente se puede realizar un suavizado de la imagen convolucionando una mascara Gaussiana. Sin embargo, los mejores resultados se alcanzan sin ese suavizado. A continuación se calculan las derivadas direccionales $\frac{df}{dx}$ y $\frac{df}{dy}$ convolucionando el kernel uno dimensional

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

por filas y columnas respectivamente.

1.2.3. Generación de los histogramas

Para generar los histogramas hace falta establecer el tamaño de las celdas en nuestro caso son celdas 6×6 píxeles. A continuación en cada píxel de la celda se considera el gradiente de la imagen en ese píxel. Se calcula el ángulo que forma con el eje OX y la magnitud del gradiente. Si el ángulo α es mayor que 180 entonces se toma el ángulo $\alpha - 180$.

El gradiente de la imagen en ese píxel va a realizar un voto que se acumulara en el histograma de la celda en la que se encuentre dicho gradiente. El voto que realiza cada gradiente coincide con la magnitud del propio gradiente. Para determinar sobre cuales de las columnas del histograma se acumula el voto se utiliza el ángulo θ del gradiente. Cada columna del histograma se identifica con un ángulo entre 0 y 180 hay tantos ángulos como columnas tenga el histograma (en nuestro caso 9) y están a la misma distancia unos de otros. A continuacion se eligen los dos angulos θ_i, θ_{i+1} de entre los nueve mencionados anteriormente de forma que $\theta_i \leq \theta < \theta_{i+1}$. A la columna i-esima se le vota con

$$(1 - \frac{\theta - \theta_i}{\theta_{i+1} - \theta_i}) \|\nabla f(x, y)\|_2$$

y a la $i + 1$-esima con

$$(\frac{\theta - \theta_i}{\theta_{i+1} - \theta_i}) \|\nabla f(x, y)\|_2$$

1.2.4. Normalización de los bloques

A continuación las celdas se agrupan en bloques de tamaño 3×3 celdas. Los histogramas de las celdas se concatenan juntos y el vector resultante v se normaliza utilizando la norma euclídea :

$$\frac{v}{\|v\|_2}$$

En esta seccion es importante que los bloques se solapen, es decir habra bloques que compartan celdas. Para conseguir esto empezamos con el bloque formado por las 3×3 primeras celdas una vez esta construido y normalizado se almacena el vector resultado y se siguen construyendo bloques. Cada bloque en construccion se rellena con la matriz formada por las 3×3 celdas que estan un paso a la derecha. Cuando se agota la fila cambiamos de columna. De esta forma, recorremos toda la ventana.

En ultimo lugar los vectores que se han formado siguiendo el procedimiento comentado arriba se concatenan. Este es el descriptor final.

1.3. Fases de la resolución

1.3.1. Procedimiento para entrenar el modelo SVM

Las palabras del Test se han generado a partir de las imágenes de la base de datos INRIA. Esta es la base de datos con la que se obtuvieron por primera vez los resultados del descriptor HOG.

En primer lugar se han procesado las imágenes del test etiquetadas como positivas. Con ese fin hemos utilizado el archivo de anotaciones que tiene cada imagen positiva del test. Cada imagen etiquetada como positiva tiene asociado un archivo con anotaciones en las que se almacenan las coordenadas de cada peatón dentro de la imagen. A partir de esa información cada peatón de la imagen se ha recortado y redimensionado al tamaño de la ventana, es decir 64×128. A este conjunto de imágenes positivas se le han añadido sus reflejadas con respecto al borde derecho de la imagen, de forma que continuamos teniendo peatones con la cabeza en la parte superior y los pies en la inferior, obteniendo el doble de ejemplos positivos a partir de los recortes.

En segundo lugar se han procesado las imágenes negativas. Para obtener recortes de las imágenes negativas se ha recorrido el directorio de las imágenes negativas y por cada imagen se han obtenido 10 recortes del tamaño considerado que se han etiquetado como negativas. Los 10 recortes se han elegido de forma aleatoria dentro de las imágenes. A partir de todos esos recortes se han generado los descriptores HOG asociados con los que se ha entrenado la SVM.

Hemos utilizado la implementación de openCV de SVM. La SVM es lineal y tiene el mismo coeficiente C que en el paper de referencia es decir $C = 0{,}01$.

Después de obtener un predictor preliminar se ha buscado dentro de las imágenes negativas recortes de la ventana deslizante que hagan falsos positivos. Estos se han almacenado en un directorio llamado falsos_positivos una vez determinados esos falsos positivos se ha reentrenado la SVM.

En total el numero de imágenes de ventanas positivas que se han conseguido es 1237 más sus imágenes reflejadas con respecto al borde derecho.. Para las imágenes negativas hay 10 ventanas por cada imagen de modo que en total son 12180. A todo eso además se le añade un numero considerable de falsos positivos para mejorar los resultados de la predicción.

1.3.2. Procedimiento para la detección de un peatón en una imagen de test

El el paper no se discute en ningún momento el método de test utilizado por los investigadores por lo que, para que sea comparable con otras técnicas actuales, hemos empleado un método de test utilizado actualmente para medir el acierto de la detección.

El dataset INRIA tambien incluye un fichero de anotaciones en el caso en el que la imagen es positiva. Como en la fase de entrenamiento las anotaciones indican el rectángulo de la imagen en el que aparece un peatón. De esta forma, una vez estimada la región en la que nuestro predictor encuentra un peatón se puede comprobar si coincide con las regiones provistas por el fichero de anotaciones.

En primer lugar se toman los tres primeros niveles de la pirámide Gaussiana para la imagen original i.e. una lista con la imagen original y los dos niveles siguientes de la pirámide (se incluye a la imagen original como el primer nivel de la pirámide Gaussiana).

Cada uno de estos niveles se recorre con una ventana deslizante de 128×64 píxeles. Tras la obtención de todos los parches de un nivel de la piramide Gaussiana se calculan los descriptores asociados a estos parches. Estos descriptores son la entrada de la SVM que predice si hay o no un peatón en cada parche. Es entonces cuando se obtienen ventanas en las que se predice que haya un peatón y ventanas en las que se predice lo contrario.

Se toma en este momento, una matriz de ceros del mismo tamaño que el nivel de la pirámide Gaussiana que se esta evaluando y en cada posición se suma 1 cada vez que el predictor afirme que dicho píxel está en una región en la que aparece un peatón. Se termina con una matriz que tiene 0 en las posiciones en las que nunca se ha predicho que hay un peatón y números mayores que 0 indicando cuántas veces dicho píxel ha estado en una ventana con un peatón. Esta estructura es conocida como mapa de calor (Heat map). Una vez obtenida esta matriz se truncan los valores menores que un umbral a 0 para conservar únicamente las regiones mas significativas, es decir, tomando como umbral 2 entonces todos los píxeles con valor menor estricto que 2 tendrían un 0 en el mapa de calor.

Con esta estructura ya realizada se pueden estudiar las regiones conexas de la misma, es decir, las regiones en las que existen números mayores que 0 que son adyacentes entre sí. De esta forma se obtienen regiones disjuntas del mapa de calor. Sólo queda en cada region encontrar el punto central y generar una ventana de tamano 128×64.

Por último se comprueba que las regiones encontradas solapen al menos en un 50 % con las regiones aportadas por los ficheros de anotaciones y devolvemos para cada imagen $\frac{número\ de\ peatones\ acertados}{número\ de\ peatones\ totales}$. De esta forma tenemos una medida entre 0 y 1 que nos dice cuánto hemos acertado en cada imagen. Si se suman todos estos números se alclanza una medida global del acierto, de forma que la puntuación máxima que podríamos obtener sería el número de imágenes.

De igual modo para hacer una predicción y medida equivalente en las imágenes negativas, para cada imagen de test negativa tomamos 10 ventanas de tamaño 128×64 de forma aleatoria y devolvemos el número de ventanas acertadas partido por el número de ventanas totales, en este caso 10. Las medidas de las imágenes negativas y positivas acertadas son comparables entre sí.

Al realizar el análisis de los peatones en cada uno de los niveles de la pirámide Gaussiana obtenemos una ventaja, y es que los posibles peatones que haya en la misma serán de menor tamaño y por tanto será más probable que el peatón ocupe por completo una ventana al ir avanzando en los niveles de la pirámide con lo que el parche que obtengamos será más parecido a los pasados para entrenamiento de la SVM.

Así mismo se ha realizado un test ortodoxo del modelo tomando imágenes ya recortadas del test de tamaño 128×64 con personas y sin personas de forma que es capaz de decirnos si hay o no personas en esa imagen (sólo hay una persona por recorte). Este test proporciona también una medida del método que subyace a lo realizado en el proceso anterior. Se tiene que este proceso de test de los recortes realizados por nosotros a mano empleando los ficheros de anotaciones es la parte del reconocimiento de los parches empleada en el método anterior al obtenerlos con la ventana deslizante. En la siguiente sección detallaremos más en profundidad los resultados obtenidos y los comentaremos.

2. Valoración de los resultados

3. Trabajo futuro: propuestas de mejora