UNIVERSIDAD DE GRANADA

Visión por Computador

grado en ingeniería informática

Cuestionario 2

Clasificación de escenas y objetos

Autor
José María Sánchez Guerrero

Rama
Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2019-2020

Índice

Ejercicio 1

Identifique las semejanzas y diferencias entre los problemas de: a) clasificación de imágenes; b) detección de objetos; c) segmentación de imágenes; d) segmentación de instancias.

La **clasificación de imágenes** y consiste en, dada una imágen, catalogarla dependiendo del contenido de ésta. Por ejemplo, en la siguiente imágen podemos clasificarla perfectamente como "gato":

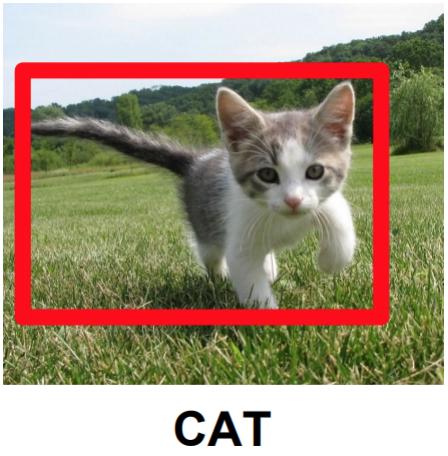

Figura 1: Clasificación de imágenes + localización

Por otra parte, la **detección de objetos** se parece a la clasificación en que el objetivo principal es el mismo, dependiendo de lo que aparezca en la imagen, decir qué está reflejado en ella.

Lo que diferencia a la detección de objetos es que encuentra todos los objetos de una imagen, dibujando un cuadro delimitador alrededor de ellos.

En la imágen anterior, hemos visto que se dibuja un rectángulo, ya que ha detectado sólo un gato. Sin embargo, esto es lo que hace la detección de objetos cuando hay más cosas en la escena:

Figura 2: Detección de objetos

Algo parecido ocurre con la segmentación de imágenes y la segmentación de instancias. Ambas consisten en, dada una imagen, clasificar cada píxel como perteneciente a una clase en concreto. Si tomamos de ejemplo la misma imagen del gato, sería algo así:

Figura 3: Segmentación de imágenes

No obstante, lo que las diferencia la una de la otra es que, la **segmentación de imágenes** no crea máscaras para cada objeto individual en la escena. Por ejemplo, si tuviesemos más de una gato en la foto anterior, nos lo fusionaría y detectaría todo como çat".

Esto con la **segmentación de instancias** no ocurre, ya que separa todos los objetos de una imagen y los clasifica como clases distintas (al igual que ocurría con la detección de objetos). Veamos un ejemplo con la imagen anterior:

Figura 4: Segmentación de instancias

Ejercicio 2

¿Cuál es la técnica de búsqueda estándar para la detección de objetos en una imagen? Identifique pros y contras de la misma e indique posibles soluciones para estos últimos.

Esta técnica se la conoce como ***Sliding Window Detection***, la cual es capaz de detectar objetos en varios pasos bastante simples: primero inspecciona la imagen desplazando una ventana de detección con un tamaño prefijado (a varias escalas ya que los objetos pueden aparecer con distintos tamaños en la imagen), después extrae las características para cada ventana; y por último, clasifica los resultados obtenidos y filtra los que se superpongan (dos ventanas detecten el mismo objeto) eligiendo la que mayor precisión tenga.

Las **ventajas** que tiene son que es un algoritmo bastante simple y que realiza una búsqueda exhaustiva sobre la posición y la escala (puede usar una ventana del mismo tamaño sobre una pirámide espacial de imágenes).

Las **desventajas** que tiene son las siguientes:

- *Objetos de distintos tamaños.* Como dijimos anteriormente, los objetos pueden aparecer con tamaños distintos en la imagen, pero para solucionar esto

se ha propuesto analizar la imagen a múltiples escalas con un tamaño de ventana prefijado. Altenativamente, también podemos analizar la imagen con distintos tamaños y proporciones de ventana.

- *Variada relación de aspecto.* La ventan que utilizemos no será la misma para analizar, por ejemplo, un gato de frente que un gato de perfil. Si utilizamos varias ventanas con diferente relación de aspecto, el número de imagenes escaneadas será el número de escalas por los diferentes tamaños de ventana que queramos asignar, y esto puede ser muy poco eficiente computacionalmente. Es difícil encontrar una solución concreta para este problema, pero lo que se podría es ajustar la ventana a lo que estamos intentando hallar (para un semáforo, no elegir una ventana cuadrada, sería más bien alargada).
- *Superposición de objetos y múltiples respuestas.* Dos o más ventanas pueden reconocer el mismo objeto, no obstante, cada una de estas múltiples respuestas tienen distintos pesos. Tendremos que realizar una supresión de no máximos (también para las distintas escalas) y quedarnos con el mejor.

Ejercicio 3

¿Cuál es la diferencia "esencial" entre el filtro de convolución y el de mediana? Justificar la respuesta.

La principal diferencia entre estos filtros es que el filtro de convolución, como hemos visto antes, es lineal, mientras que el filtro de mediana no lo es. Para justificar la respuesta veamos un ejemplo: si tenemos los filtros $A = (0, 1, 2, 3, 4)$, $B = (5, 0, 1, 4, 5)$ y $(A + B) = (5, 1, 3, 7, 9)$, el cálculo de sus medianas es:

$$Med(A) = 2 \qquad Med(B) = 4 \tag{1}$$

Y por tanto, como:

$$Med(A) + Med(B) = 6 \quad \neq \quad Med(A + B) = 5 \tag{2}$$

podemos justificar que los filtros de mediana no son lineales.

Ejercicio 4

Identifique el "mecanismo concreto" que usa un filtro de máscara para transformar una imagen.

El mecanismo que utilizan todos los filtros de máscara para transformar una imagen es que lo hace utilizando **información local**, es decir, que el valor de cada píxel se calcula teniendo en cuenta los píxeles cercanos a él (los que están dentro de la máscara).

Podremos encontrar varios tipos de filtros, ya sean lineales o no lineales como hemos visto antes, pero ninguno realizará cálculos en la imagen sin tener en cuenta sus píxeles asociados en la máscara.

Ejercicio 5

¿De qué depende que una máscara de convolución pueda ser implementada por convoluciones 1D? Justificar la respuesta.

Depende de que la máscara de convolución solo se pueda descomponer como producto de dos matrices de dimensión 1. Por ejemplo, la siguiente máscara Gaussiana se puede descomponer en:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{3}$$

Para comprobar si esto es posible, podemos realizar la **descomposición en valores singulares** de la matriz. Si esta descomposición nos da un valor distinto de 0, la máscara será separable en dos matrices 1D, y por tanto, la podremos represetar como:

$$\sum_{r}^{i=0} \sigma_i u_i v_i^T \tag{4}$$

siendo σ_i los valores de la matriz diagonal, y u_i y v_i los valores de las respectivas matrices 1D resultantes.

Otra forma que tenemos de comprobarlo es que, si nos fijamos en este tipo de máscaras, el **rango** de la matriz siempre va a ser igual a 1.

Ejercicio 6

Identificar las diferencias y consecuencias desde el punto de vista teórico y de la implementación entre:

(a) **Primero alisar la imagen y después calcular las derivadas sobre la imagen alisada**

(b) **Primero calcular las imágenes derivadas y después alisar dichas imágenes.**

Justificar los argumentos.

Desde un punto de vista teórico dependerá del tipo de filtro que usemos, pero lo lógico sería elegir un filtro de convolución. Como hemos visto en el ejercicio 2, este filtro tiene cumple la propiedad **asociativa**, por lo que tanto la opción *a* como la *b* darían el mismo resultado.

$$(f * g) * h = f * (g * h) \tag{5}$$

En cuanto a la implementación, es más conveniente **alisar primero la imagen y posteriormente calcular las derivadas** sobre la imagen alisada. Si lo hacemos de esta forma, tendremos que hacer solo una convolución y dos derivadas después. De la otra forma, haremos dos derivadas al principio y posteriormente dos alisamientos, uno para la X y otro para la Y.

Ejercicio 7

Identifique las funciones de las que podemos extraer pesos correctos para implementar de forma eficiente la primera derivada de una imagen. Suponer alisamiento Gaussiano.

Ejercicio 8

Identifique las funciones de las que podemos extraer pesos correctos para implementar de forma eficiente la Laplaciana de una imagen. Suponer alisamiento Gaussiano.

Ejercicio 9

Suponga que le piden implementar de forma eficiente un algoritmo para el cálculo de la derivada de primer orden sobre una imagen usando alisamiento Gaussiano. Enumere y explique los pasos necesarios para llevarlo a cabo.

Los pasos necesarios serían:

1. Seleccionar un tamaño para la máscara o calcularlo en función del *sigma*. Podemos poner por ejemplo $6\sigma + 1$, ya que con la densidad de la función Gaussiana que nos ofrece abarcamos casi toda la máscara.
2. Calcular los filtros de la primera derivada de la función Gaussiana. Para ello tendremos que muestrear en tantos puntos como tamaño hayamos seleccionado anteriormente. También tendremos en cuenta que obtendremos un array para las filas y otro para las columnas.
3. Normalizar el filtro multiplicando por sigma y hacer la convolución, tanto por filas como por columnas, sobre la imagen.

Ejercicio 10

Identifique semejanzas y diferencias entre la pirámide gaussiana y el espacio de escalas de una imagen, ¿cuándo usar una u otra? Justificar los argumentos.

Ambas son una representación a distintas escalas de una imagen en las que se utiliza como filtro de suavizado principal el Gaussiano. También coinciden en la forma de hacer el submuestreo, ya que ambas obtienen la octava de la imagen original, y a partir de ahí siguen bajando niveles con la imagen resultante.

Se diferencian en que la pirámide Gaussiana únicamente reduce la imagen y obtiene las frecuencias bajas de ella; mientras que en un espacio de escalas, también se hace uso de estas frecuencias bajas, pero lo hace junto a una serie de técnicas (como puede ser el filtro Laplaciano sobre el alisamiento o supresión de no máximos) ya que el objetivo de este es detectar las distintas regiones más relevantes de la imagen. Otra diferencia es que, pese a que se trabaja con una octava de la imagen en cada nivel, esa octava es diferente en una técnica y otra, porque en el espacio de escalas se realiza cada una de las técnicas comentadas para cada nivel.

Ejercicio 11

¿Bajo qué condiciones podemos garantizar una perfecta reconstrucción de una imagen a partir de su pirámide Laplaciana? Dar argumentos y discutir las opciones que considere necesario.

Únicamente con las frecuencias altas de la pirámide Laplaciana no podríamos reconstruir la imagen original, también necesitaríamos la última submuestra de la imagen, es decir, las frecuencias bajas residuales.

Si disponemos de ella, se podría reconstruir con el siguiente algoritmo, el cual consiste en sumar al último nivel de la Laplaciana esta última submuestra de frecuencias bajas, y ampliada con una función F:

$$G_k = L_k + F(G_{k+1}) \tag{6}$$

Cuando terminemos (llegemos a G_1) será cuando tengamos la imagen original completamente reconstruida.

Ejercicio 12

¿Cuáles son las contribuciones más relevantes del algoritmo de Canny al cálculo de los contornos sobre una imagen? ¿Existe alguna conexión entre las máscaras de Sobel y el algoritmo de Canny? Justificar la respuesta

Las contribuciones que tiene este algoritmo frente a otros es que este utiliza las siguientes técnicas:

1. **Non-maximum supression.** Esta técnica recorre todos los píxeles de cada imagen y, para cada uno de ellos, comprueba que sus píxeles fronterizos no tienen un valor más alto que el central. En caso de que exista, lo pondrá a cero. Para tener más claro cómo funciona, vamos a ejemplificarlo con una matriz ejemplo que represente los píxeles de una imagen:

$$\begin{bmatrix} 3 & 2 & 3 \\ 6 & 4 & 1 \\ 1 & 3 & 3 \end{bmatrix} \underset{(x,y)>=(0,0)}{\rightarrow} \begin{bmatrix} 3 & 2 & 3 \\ 6 & 0 & 1 \\ 1 & 3 & 3 \end{bmatrix} \qquad \begin{bmatrix} 3 & 2 & 3 \\ 1 & 4 & 1 \\ 1 & 3 & 3 \end{bmatrix} \underset{(x,y)<(0,0)}{\rightarrow} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

   A la izquierda vemos que hay valores mayores que el central, el $(1,0) = 6$, asi que el píxel central termina siendo cero. Por otra parte, en la matriz de la izquierda, ningun valor es más grande que el central, por lo que podemos decir que es un máximo.

2. **Linking and thresholding (hysteresis).** Tras hacer la supresión de no máximos, define dos umbrales (thresholding), uno alto y otro bajo. Volveremos a recorrer cada píxel y dependiendo su valor con los umbrales se hará lo siguiente: en caso de ser inferior que el bajo, no se le considerará como borde; si está entre bajo y el alto, será parte del borde si está conectado a otro que ya está considerado como borde; y si el valor es superior al umbral alto, se le considerará como borde.

   Antes de aplicar estas dos técnicas, Canny tiene que encontrar la magnitud y orientación del gradiente. Para ello se servirá de las **máscaras de Sobel**, las cuales se basan en el cálculo de la primera derivada respecto de X e Y.

Ejercicio 13

Identificar pros y contras de k-medias como mecanismo para crear un vocabulario visual a partir del cual poder caracterizar patrones. ¿Qué ganamos y que perdemos? Justificar los argumentos

K-means es un algoritmo de clasificación no supervisado el cual selecciona un número de clústeres y sus centros. Después, clasifica cada punto de los descriptores calculando la distancia entre ellos y el centro más cercano, minimizando el error de la suma de cuadrados; y por último recalcula los centros de cada cluster.

Es de los algoritmos más utilizados y más populares, ya que entre sus principales **ventajas** tenemos:

- Simplicidad y velocidad que le permite ejecutarse en grandes conjuntos de datos. Su complejidad computacional por cada iteración es de: asignar cada punto al centro del cluster más cercano $O(n * k)$, y recalcular el centro del cluster a la media de sus puntos asignados $O(n)$.
- Utiliza "hard assignment", es decir, cada punto se asigna a únicamente un solo cluster.
- Su simplicidad facilita la demostración de los resultados obtenidos, a diferencia de las redes neuronales o las SVM.
- Que sea eficiente implica que el algoritmo es bueno por si necesitamos segmentar el conjunto de datos.

Entre sus **desventajas** tenemos que:

- Los centros de los k clusters iniciales se inicializan aleatoriamente, asi que no se produce el mismo resultado tras cada ejecución. Una mala inicialización nos puede llevar a: una mala velocidad de convergencia y a un mal agrupamiento de los clusteres.
- Necesita el número de clústeres que se especificarán. Demasiados clústeres pueden causar escasez de datos y muy pocos pueden provocar que llegen a converger, por eso hay que elegir con cuidado.
- No garantiza que el resultado sea un mínimo global de la varianza, ya que converge a un óptimo local.
- Sensible a los valores atípicos. Cuando hay valores atípicos, los centros de los clusteres resultantes pueden no ser tan representativos y, por lo tanto, el error de la suma de cuadrados (varianza) también será más alto, y nuestro objetivo en minimizarlo.
- No trabaja bien con grupos no lineales.

Ejercicio 14

Identifique pros y contras del modelo de "Bolsa de Palabras" como mecanismo para caracterizar el contenido de una imagen. ¿Qué ganamos y que perdemos? Justificar los argumentos.

El modelo de 'Bolsa de Palabras' extrae las características locales de una imagen, muestrea un subconjunto de ellas y construye un diccionario visual mediante *k-means* visto en el ejercicio anterior, cuantifica las característica utilizando este diccionario, y por último, representa imágenes mediante la frecuencia de cada 'palabra visual'. Veamos las **ventajas** de usar este método:

- Es bastante flexible a la geometría o las deformaciones producidas por el punto de vista (donde hayamos tomado la imagen).
- Resume bastante bien el contenido de la imagen.
- Proporciona una representación vectorial para imágenes y poder crear así un histograma.
- En la práctica, nos ofrece buenos resultados.

Como **desventajas** tenemos que:

- Generar el vocabulario a partir de grandes cantidades de datos suele ser computacionalmente costoso.
- El modelo básico no tiene en cuenta la geometría, asi que tendrá que verificar después o codificar mediante funciones.
- Para los humanos, intuitivamente vemos la división de los objetos en partes, pero realmente esa información no existe. Por tanto, no tenemos grantías de capturar todos los puntos de interés.
- Si la bolsa de palabras cubre toda la imagen, se puede mezclar el fondo y el primer plano. Para solucionarlo se suelen utilizar las pirámides espaciales.
- Sigue siendo complicado la formación del vocabulario óptimo, al igual que todavía no ha sido ampliamente testeado para invarianza en la escala y en el punto de vista.

Ejercicio 15

Suponga que dispone de un conjunto de imágenes de dos tipos de clases bien diferenciadas. Suponga que conoce como implementar de forma eficiente el cálculo de las derivadas hasta el orden N de la imagen. Describa como crear un algoritmo que permita diferenciar, con garantías, imágenes de ambas clases. Justificar cada uno de los pasos que proponga.

Como sabemos calcular de manera eficiente las derivadas hasta el orden N, yo utilizaría un espacio de escalas y así obtener las características invariantes a la escala. Después haríamos la supresión de no máximos, pero eliminando los contornos porque si no, nos encontraríamos ahí a la mayoría de máximos. A continuación, realizamos el cálculo del gradiente para cada punto obtenido y se crea el histograma con las mayores frecuencias de direcciones. Por último, ya podemos pasar a extraer las características con un algoritmo.

Este algoritmo consistirá en una 'Bolsa de palabras', en el cual aprenderemos un 'vocabulario visual' muestreando un subconjunto de características y después utilizando un algoritmo de clustering como el *k-means*. Cuantificamos las características utilizando este diccionario y para terminar ya sólo nos faltaría un clasificador, y como las características deben de estar bien diferenciadas si hemos hecho correctamente los pasos anteriores, yo utilizaría un SVM o kNN para separarlas.