



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

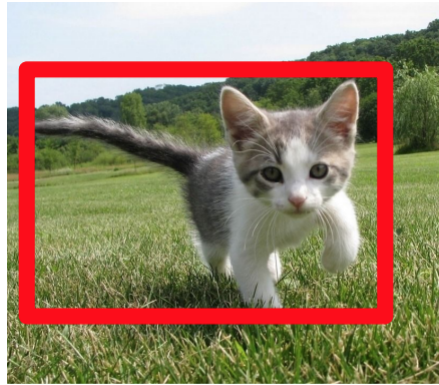
Índice

Ejercicio 1	2
Ejercicio 2	4
Ejercicio 3	5
Ejercicio 4	5
Ejercicio 5	6
Ejercicio 6	6
Ejercicio 7	7
Ejercicio 8	7
Ejercicio 9	7
Ejercicio 10	7
Ejercicio 11	8
Ejercicio 12	8
Ejercicio 13	8
Ejercicio 14	8
Ejercicio 15	8

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 imagen, catalogarla dependiendo del contenido de ésta. Por ejemplo, en la siguiente imagen podemos clasificarla perfectamente como 'cat':



CAT

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 imagen 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 'cat'.

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, después extrae las características para cada ventana; y por último, clasifica los resultados obtenidos y filtra los que se superpongan.

Las **ventajas** que tiene es 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.* Los objetos pueden aparecer con tamaños distintos en la imagen. Para solucionar esto se ha propuesto analizar la imagen a múltiples escalas con un tamaño de ventana prefijado. Alternativamente,

también podemos analizar la imagen con distintos tamaños y proporciones de ventana.

- *Variada relación de aspecto.* La ventana necesaria para analizar, por ejemplo, un gato de frente que un gato de perfil, no será la misma. Es difícil encontrar una solución concreta para este problema, pero se podría 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

Considere la aproximación que extrae una serie de características en cada píxel de la imagen para decidir si hay contorno o no. Diga si existe algún paralelismo entre la forma de actuar de esta técnica y el algoritmo de Canny. En caso positivo identifique cuales son los elementos comunes y en que se diferencian los distintos.

Esta aproximación es la *Selective Search*, la cual no tiene aspectos muy similares al algoritmo de Canny. La búsqueda selectiva extrae una serie de características para cada píxel y realiza una agrupación jerárquica basándose en similitudes entre texturas, tamaños, forma y color; mientras que Canny simplemente realiza una detección de contornos mediante histéresis. En caso de utilizar la búsqueda selectiva para detección de contornos, se podrían obtener diferenciando entre unos segmentos detectados y otros.

Como similitud podemos comentar que ambos tienen en cuenta las distintas escalas de la imagen, con la diferencia de que la búsqueda selectiva utiliza un algoritmo jerárquico, y Canny detecta los contornos a gran escala aumentando un valor de σ y viceversa. También hacen uso de la derivada de la Gaussiana para hacer un filtrado de la imagen.

Ejercicio 4

Tanto el descriptor de SIFT como HOG usan el mismo tipo de información de la imagen pero en contextos distintos. Diga en que se parecen

y en que son distintos estos descriptores. Explique para que es útil cada uno de ellos.

Ambos están basados en los gradientes de primer orden de la imagen, pero como su propio nombre indica, HoG simplemente realiza un histograma de los gradientes agrupándolos según su orientación; mientras que SIFT, después de hacer este histograma, realiza una interpolación entre ángulos vecinos y pondera los valores de todo el descriptor mediante la Gaussiana.

Debido a esta ponderación Gaussiana que tiene SIFT, veo más útil utilizarlo para describir la importancia en un único punto; y HoG es más adecuado usarlo en la clasificación de imágenes en general, ya que no tiene ese sesgo.

Ejercicio 5

Observando el funcionamiento global de una CNN, identifique que dos procesos fundamentales definen lo que se realiza en un pase hacia delante de una imagen por la red. Asocie las capas que conozca a cada uno de ellos.

1. Utilizar capas convolucionales con su activación 'ReLU', 'linear', 'tanh'..., capas de *pooling* o algunas funciones de regularización como *Dropout* o *Batch-Normalization*.
2. Utilizar un clasificador con capas totalmente conectadas para realizar la predicción. También contarán con su función de activación (siendo la última 'softmax' para obtener las probabilidades de cada clase como salida).

Ejercicio 6

Se ha visto que el aumento de la profundidad de una CNN es un factor muy relevante para la extracción de características en problemas complejos, sin embargo este enfoque añade nuevos problemas. Identifique cuales son y qué soluciones conoce para superarlos.

Aumentar la profundidad en una red neuronal de forma excesiva nos puede llevar al **sobreaprendizaje**, ya que función de salida obtenida se ajusta casi por completo al conjunto de entrenamiento. Una solución que podemos aplicar es el aumento del conjunto de datos (técnicas como *horizontal flip*, o hacer zoom en partes de la

imagen son un ejemplo de esto). Otra solución para evitar el sobreaprendizaje es regularizar los datos introduciendo capas como *Dropout* o *BatchNormalization*.

Por otra parte, al hacer ***backpropagation*** se pierde precisión y los gradientes también van disminuyendo. Este problema es más difícil de solucionar, ya que tendríamos que buscar nuevos métodos, como puede ser la ejecución capa por capa o utilizar clasificadores auxiliares en distintos niveles de la red neuronal (como hace GoogleNet).

Ejercicio 7

Existe actualmente alternativas de interés al aumento de la profundidad para el diseño de CNN. En caso afirmativo diga cuál/es y como son.

Ejercicio 8

Considere una aproximación clásica al reconocimiento de escenas en donde extraemos de la imagen un vector de características y lo usamos para decidir la clase de cada imagen. Compare este procedimiento con el uso de una CNN para el mismo problema. ¿Hay conexión entre ambas aproximaciones? En caso afirmativo indique en que parecen y en que son distintas.

Ejercicio 9

¿Cómo evoluciona el campo receptivo de las neuronas de una CNN con la profundidad de la capas? ¿Se solapan los campos receptivos de las distintas neuronas de una misma profundidad? ¿Es este hecho algo positivo o negativo de cara a un mejor funcionamiento?

Ejercicio 10

¿Qué operación es central en el proceso de aprendizaje y optimización de una CNN?

Ejercicio 11

Compare los modelos de detección de objetos basados en aproximaciones clásicas y los basados en CNN y diga que dos procesos comunes a ambas aproximaciones han sido muy mejorados en los modelos CNN. Indique cómo.

Ejercicio 12

Es posible construir arquitecturas CNN que sean independientes de las dimensiones de la imagen de entrada. En caso afirmativo diga cómo hacerlo y cómo interpretar la salida.

Ejercicio 13

Suponga que entrenamos una arquitectura Lenet-5 para clasificar imágenes 128x128 de 5 clases distintas. Diga que cambios deberían de hacerse en la arquitectura del modelo para que se capaz de detectar las zonas de la imagen donde aparecen alguno de los objetos con los que fue entrenada.

Ejercicio 14

Argumente por qué la transformación de un tensor de dimensiones 128x32x32 en otro de dimensiones 256x16x16, usando una convolución 3x3 con stride=2, tiene sentido que pueda ser aproximada por una secuencia de tres convoluciones: convolución 1x1 + convolución 3x3 + convolución 1x1. Diga también qué papel juegan cada una de las tres convoluciones.

Ejercicio 15

Identifique una propiedad técnica de los modelos CNN que permite pensar que podrían llegar a aproximar con precisión las características

del modelo de visión humano, y que sin ella eso no sería posible. Explique bien su argumento.