

EL7007-1 - Introducción al Procesamiento Digital de Imágenes

TAREA 1

Profesor: Claudio Pérez F.
Auxiliar: Jorge Zambrano I.
Semestre: Otoño 2022

DETECCIÓN Y TRACKING DE OBJETOS BASADO EN MODELO DE HUBEL Y WIESEL

1. Objetivo:

El objetivo de esta tarea es desarrollar un sistema simple de detección de objetos, y adaptarlo a un sistema básico de tracking, basado en la teoría de campos receptivos de Hubel y Wiesel.

2. Descripción:

El reconocimiento y tracking de objetos son problemas altamente desafiantes y que continúan bajo investigación, debido a la alta variabilidad de patrones y las diferentes estrategias para abordar este problema. En la actualidad, las Redes Neuronales Convolucionales (CNN) han presentado un mejor desempeño ante otros tipos de sistemas, basándose en la extracción de características por medio de filtros convolucionales. Estos tipos de filtros se basan en la investigación de Hubel y Wiesel, quienes descubrieron que ciertas neuronas se excitan cuando un patrón simple se sitúa en una zona particular de la retina, y que dichos estímulos decrecen cuando el patrón se aleja. Esta tarea tiene como finalidad implementar detectores simples de patrones y extrapolarlo a un sistema de tracking, mediante el diseño de filtros capaces de encontrar dichos objetos por medio de estímulos dentro de una imagen.

3. Implementación de un detector de Patrones:

En esta sección se explicará el procedimiento para desarrollar los detectores de patrones en imágenes.

- i. Lea la imagen *img1.png* y conviértala a imagen binaria. De la imagen leída, recorte el patrón mostrado “p”:



Figura 1: Patrón “p”.

- ii. Utilizando un kernel de 3x3 balanceado (-1 en los bordes y 8 en el centro) “k”, genere un filtro detector “f” para el patrón siguiendo el esquema de Hubel y Wiesel. Para ello, haga la convolución entre el patrón recortado “p” y el kernel “k”.
- iii. Aplique el filtro generado “f” a la imagen *img1.png*, haciendo la convolución de esta imagen con el filtro encontrado. De esta forma se está haciendo la detección “d” del objeto.
- iv. Presente los valores de la matriz del filtro detector “f” y del resultado de la convolución “d” en forma de mapas de calor (colorbar) y explique los resultados.
- v. Convierta esta matriz a un vector fila, y muestre los resultados en una gráfica. Esto le servirá para decidir sobre un umbral de detección, por ejemplo, para la Figura 2 un umbral adecuado sería 500 (o un 80% del estímulo máximo).

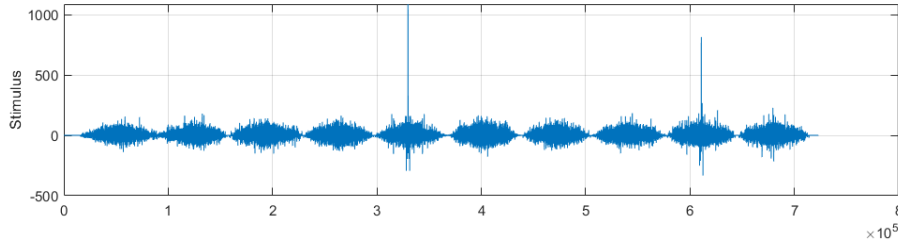


Figura 2: Matriz resultante en forma de vector

- vi. Recupere las posiciones (x, y) de los valores que superan el umbral, y grafique sobre la imagen original “bounding boxes” sobre dichas posiciones. Puede utilizar la siguiente línea como referencia (para el caso de Matlab). HINT: El tamaño del “bounding box” debe ser del mismo tamaño que el patrón “p”.

```
[x, y] = find(d >= 0.8 * max(d(:)));
```

- vii. Detecte sobre las imágenes Img1 e Img2 los patrones indicados. Para la imagen Img2, solo debe mostrar los resultados de las detecciones, es decir los bounding boxes sobre los objetos.
- viii. Implemente un algoritmo para evitar múltiples detecciones (Non Maximum Suppression NMS). Se debe indicar en pantalla la cantidad de objetos encontrados después de la supresión.
- ix. Comente las ventajas y desventajas de este método y explique por qué se dan falsos positivos y negativos al momento de la detección.

A continuación, se presentan un ejemplo de las detecciones sin y con NMS:

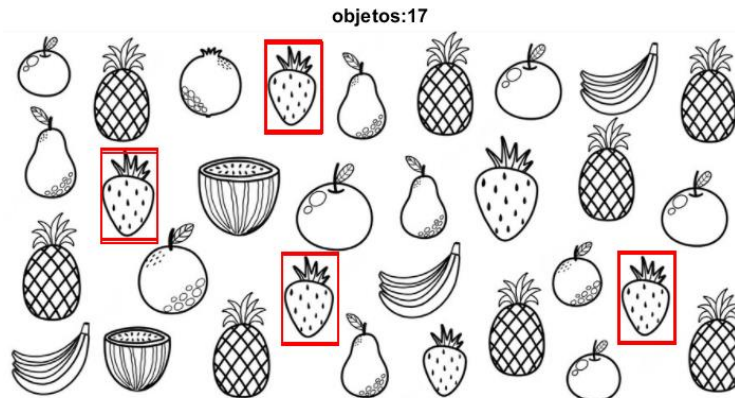


Figura 3: Ejemplo de detección sin NMS.

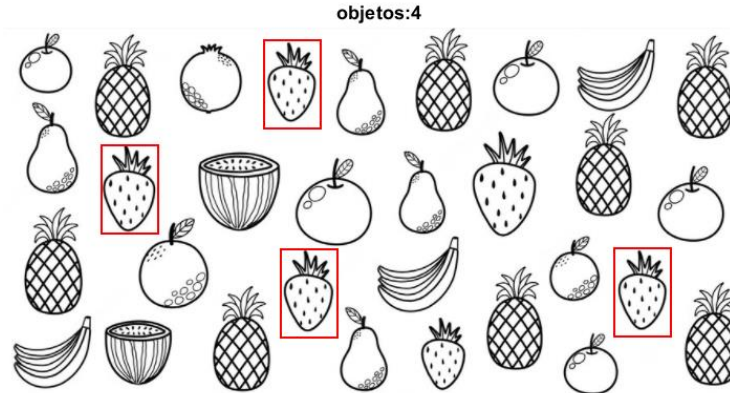


Figura 4: Ejemplo de detección con NMS

x. Listas de detecciones:

- Img1: Head-phones
- Img2: Frutilla y Piña

4. Implementación de un algoritmo de Tracking en Super Mario Bros:

Una vez desarrollado un algoritmo de detección correcto, se aplicará en una secuencia de imágenes correspondiente a un video del videojuego “Super Mario Bros”. Se detallan los pasos a continuación:

- i. Lea el video y transfórmelo a una secuencia de imágenes. (330 frames).
- ii. Para realizar el tracking, debe aplicar el algoritmo anterior a la secuencia de imágenes. Es decir, primero detecta el filtro detector “f” entre el patrón solicitado y el frame indicado, luego se aplica la detección sobre el resto de las imágenes.
- iii. Se realizará el tracking de 3 patrones por separado. A continuación, se detallan los patrones y el frame del cual deben ser extraídos.
 - Posición del salto hacia delante de Mario (frame 145)
 - Goomba (frame 297)
 - Cara de Mario hacia adelante (frame 10)
- iv. Finalmente genere un video con los frames resultantes.

HINTS:

- Recuerde ajustar finamente los umbrales para evitar los falsos positivos.
- Trabaje con imágenes binarias (en blanco y negro), y presente los resultados sobre los frames originales a color.
- Encuentre el mejor padding para la convolución, las posibles opciones generalmente son: *same*, *full*, *valid*, tanto en Matlab como en Python.
- Para su facilidad, puede mostrar el valor de la detección junto con el bounding box. Esto le podría ayudar a seleccionar correctamente un umbral.

A continuación, se muestran unos ejemplos de los resultados esperados:



Figura 5: Cara de Mario hacia adelante



Figura 6 Posición del salto hacia delante de Mario



Figura 7: Goomba

5. Entregables:

- Presente un reporte individual del trabajo. Además, incluya en el informe las detecciones solicitadas en las imágenes *Img1* e *Img2*, con las indicaciones mostradas anteriormente.
- Entregue los tres videos generados del tracking solicitado, y en el informe incluya ejemplos de buenas y malas detecciones para cada caso. Recuerde explicar el motivo de las falsas detecciones.
- El código también debe ser entregado, asegúrese de que pueda ser ejecutado por los revisores para comprobar el algoritmo. Se puede implementar en Python o Matlab.

La fecha de entrega de la tarea será el día 07/04/2022 a las 18:00 hrs por medio de U-cursos.