
	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Marzo – Agosto 2024

			<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>		
<b>CARRERA:</b> COMPUTACIÓN			<b>ASIGNATURA:</b> VISIÓN POR COMPUTADOR		
<b>NRO. PRÁCTICA:</b>	4-1	<b>TÍTULO PRÁCTICA:</b> Clasificación de Objetos Usando Técnicas de Deep Learning e identificación de formas empleando el descriptor Shape Signature (Transformada de Fourier)			
<b>OBJETIVO:</b> Reforzar los conocimientos adquiridos en clase sobre clasificación de objetos usando deep learning y la extracción del descriptor Shape Signature para caracterización de la forma de figuras.					
<b>INSTRUCCIONES:</b>		1. Revisar el contenido teórico del tema			
		2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje y la documentación disponible en fuentes académicas en línea			
		3. Deberá desarrollar dos códigos, uno que permita clasificar objetos usando técnicas de deep learning y otro para reconocer formas empleando el descriptor Shape Signature a partir de un corpus dado.			
		4. La práctica tiene una sola sección y el objetivo es aplicar la búsqueda de templates y determinar cómo se realiza almacenamiento de vídeos usando OpenCV C++			
<b>ACTIVIDADES POR DESARROLLAR</b>					

### PARTE 1A

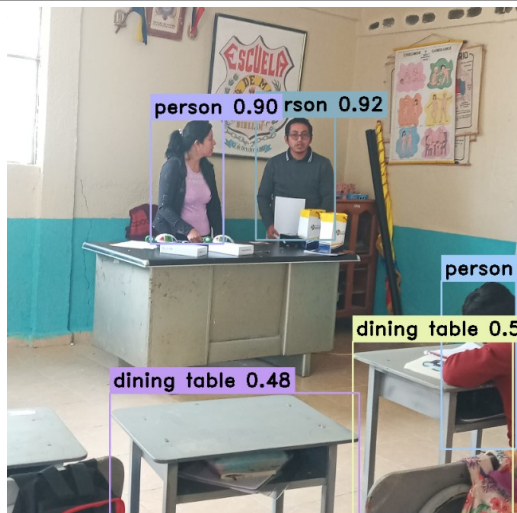
- Emplear una red YOLOv10 (como se ha visto en clase) o similar a la que se haya realizado un proceso de **transfer learning** para reconocer 5 tipos de objetos del Ecuador con imágenes captadas con la cámara del computador. Deberá realizar pruebas de rendimiento usando GPUs vs CPU (para ello puede emplear las computadoras del laboratorio de Cómputo 8). **El entrenamiento y validación de resultados de la red puede hacerlo en Google Colab o en cualquier entorno de su preferencia.** Por ejemplo, podría realizar el reconocimiento de especies endémicas (iguana marina, tortugas gigantes, rana enana de colorado, colibrí de Esmeraldas y Viscacha Ecuatoriana). Se sugiere emplear imágenes que se puedan encontrar en repositorios para realizar la tarea. **No puede haber ningún grupo que tenga las mismas imágenes o código que otros grupos.**
- Por otra parte, como podemos apreciar en la Ilustración 1, se ha realizado un proceso de detección de objetos usando la red neuronal YOLOv8l (versión large) en diferentes imágenes:



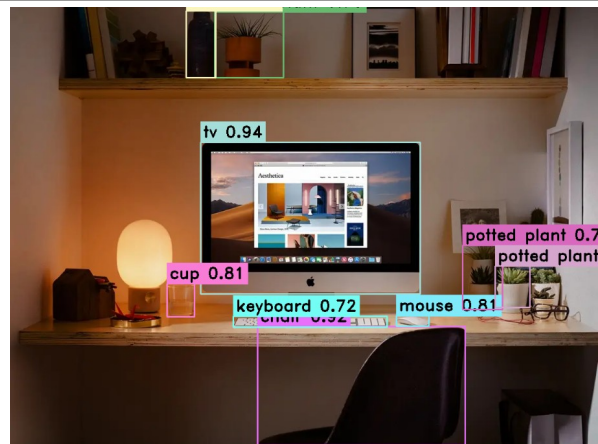
(a)



(b)



(c)

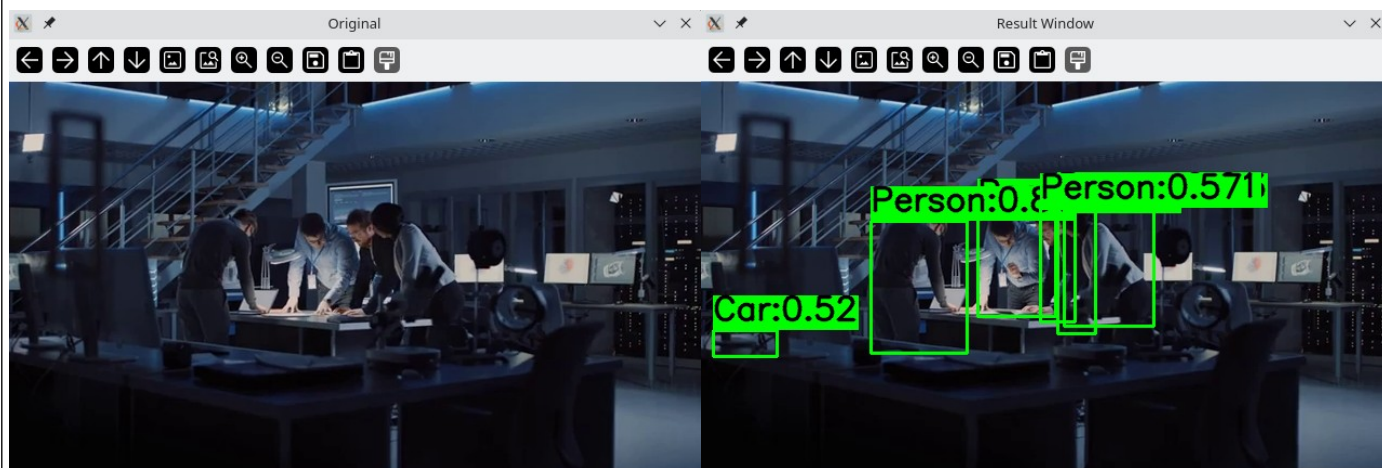


(d)

**Ilustración 2.** Ejemplos de reconocimiento de objetos usando YOLO8l en una imagen de una niña (a), en imágenes de escritorios (b) y (d) y en un aula en Biblián (c).

## PARTE 1B

- Realizar pruebas de rendimiento de la red neuronal MobileNetV3 (como se ha visto en clase), comparando GPUs vs CPU (para ello debe emplear las computadoras del laboratorio de Cómputo 8). Para ello, deberá visualizar la cantidad de Frames por Segundo que se tiene en CPU frente a GPU (ver Ilustración 2).



(a)

(b)

**Ilustración 2.** Ejemplos de detección de objetos (b) usando la red neuronal MobileNetV3, dada un vídeo de una escena de oficina (a).

- Cada grupo deberá usar un computador distinto**
- Deberá grabar un vídeo del resultado que se obtiene tanto en CPU como en GPU y mostrar la siguiente información:
  - Uso de la memoria con el comando **nvidia-smi**
  - Número de FPS (*frames per second*) en GPU vs CPU
  - Uso de memoria RAM en GPU vs CPU

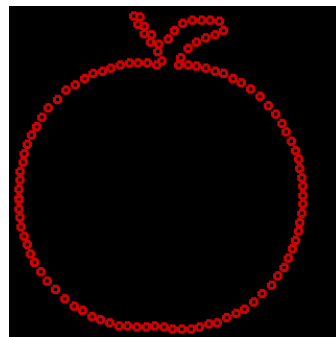
## Documentación de Soporte:

- El enlace donde se encuentra el código de ejemplo para crear aplicaciones que usan YOLOv5 y YOLOv8 es el que se detalla a continuación:

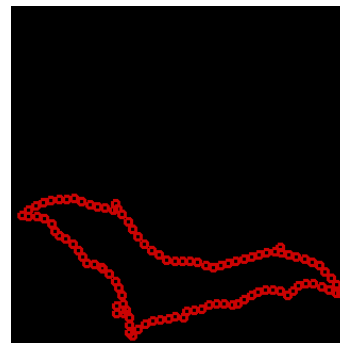
- <https://github.com/vlarobbyk/yolov5-and-yolov8-object-detection-OpenCV-C->
- Tutorial “**Object Detection using YOLOv5 OpenCV DNN in C++ and Python**” de LearnOpenCV: <https://learnopencv.com/object-detection-using-yolov5-and-opencv-dnn-in-c-and-python/>
- Artículo Científico “**A fine-tuned YOLOv5 deep learning approach for real-time house number detection**”: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10403189/>
- Tutorial “**Intersection over Union (IoU) for object detection**”: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- Vídeo tutorial (YouTube) “Train YOLOv8 object detection on a custom dataset”: <https://www.youtube.com/watch?v=m9fH9OWn8YM>

## PARTE 2

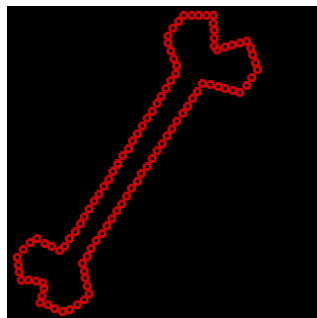
- Desarrollar un programa que obtener el descriptor Shape Signature del dataset de imágenes **Dataset-Shapes.zip**. Para ello deberá escoger 5 categorías diferentes y realizar la siguientes tareas:



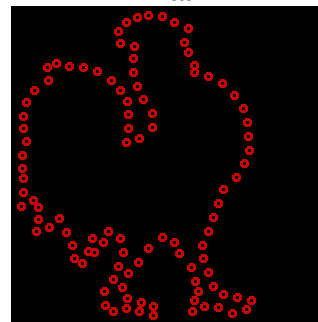
Apple



Bat



Bone



Chicken

**Ilustración 1.** Ejemplo de 4 categorías de imágenes del dataset “**Dataset-Shapes**”<sup>1</sup>

1. Seleccionar 10 categorías de imágenes (**ningún grupo puede tener las mismas imágenes**) del corpus **Dataset-Shapes**. El corpus tiene diversas imágenes que representan objetos como las que se muestran en la Ilustración 1:
2. Extraer el descriptor Shape Signature para todas las imágenes (tanto para las de entrenamiento como para las de pruebas). Estos descriptores deberán estar almacenados en archivos desde donde realizará la lectura y el análisis correspondiente.
3. Determinar el nivel de precisión para clasificar las imágenes de pruebas comparando el descriptor de cada imagen de entrenamiento con el descriptor de cada imagen de pruebas. Para ello, puede usar la **distancia Euclídea** (u otra medida de similitud que desee) vista en clase (en el tema de los momentos de HU).

<sup>1</sup>Imágenes generadas a partir de los archivos de descriptores del conjunto de datos “**The 2D Shape Structure Dataset**”: <http://2dshapesstructure.github.io/index.html>

4. Deberá generar un reporte que contenga la siguiente información:

- Código fuente de la solución planteada.
- Matriz de confusión de las categorías seleccionadas.
- Nivel de precisión del sistema planteado.
- Ejemplos gráficos de formas que confunde.
- Conclusiones y reflexión sobre la técnica aplicada.

**RESULTADO(S) OBTENIDO(S):**

Puede detectar templates en vídeos capturados con la cámara y almacenar fragmentos de vídeo.

**CONCLUSIONES:**

- Los estudiantes comprenden los principales aspectos relacionados con el entrenamiento y uso de redes neuronales de aprendizaje profundo para clasificación de objetos.
- Los estudiantes comprenden y extraen la transformada rápida de Fourier para generar descriptores de la forma de las figuras.
- El estudiante implementa sistemas básicos de clasificación de objetos usando deep learning.
- El estudiante implementa sistemas básicos de reconocimiento de formas con Shape Signature.

**RECOMENDACIONES:**

- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.
- Consultar con el docente las dudas que puedan surgir al momento de realizar la práctica.

**Docente / Técnico Docente:** Ing. Vladimir Robles, Bykbaev

**Firma:** \_\_\_\_\_