

### **TASK 3**

**Project Goal:** Develop a simple Flask API to detect and draw bounding boxes around memory modules on motherboards in images.

#### **Algorithm Choice:**

For this task, we recommend YOLOv5 (You Only Look Once version 5) for the following reasons:

YOLOv5 is a pre-trained object detection model known for its balance between real-time processing and good accuracy. This is ideal for your Flask API, which aims for responsiveness. Frameworks like TensorFlow or PyTorch offer pre-built implementations of YOLOv5, simplifying development. Makesense.ai Integration: Makesense.ai offers YOLOv5 integration, potentially speeding up the initial training phase by leveraging pre-trained weights.

#### **CPU vs. GPU Impact:**

It is always recommended to use a GPU for this task due to the computational intensity of object detection algorithms. While a CPU can technically run them depending on the type of algorithm that we are using, a GPU will significantly accelerate processing, especially for real-time applications like your Flask API.

#### **Video vs. Images:**

When dealing with videos instead of images, the approach would require adjustments:

Object Tracking algorithm will be needed to follow identified memory modules across video frames. Libraries like OpenCV offer such functionalities.

Frame Processing: The algorithm would need to process each frame of the video individually, potentially requiring more computational resources.

#### **Makesense.ai for Annotation:**

Makesense.ai is a suitable platform for annotating your images. Bounding Box Creation can be used to define a label for "memory module" and then draw bounding boxes around memory in each image. While some upload might be involved, makesense.ai emphasizes data privacy, potentially addressing any concerns about uploading project images.

**Additional Notes:**

Consider starting with a smaller dataset (less than 20 images) for initial training and testing to minimize training time. Experiment with different pre-trained weights for YOLOv5 to see which performs best on your specific motherboard imagery (weights are downloadable from the Ultralytics website).

## References:

YOLOv5 by Ultralytics: <https://www.ultralytics.com/>

TensorFlow Hub - YOLOv5: <https://www.zehntech.com/real-time-object-detection-using-yolov5-and-tensorflow/>

Makesense.ai Integrations: <https://www.makesense.ai/>

Why Use a GPU for Deep Learning: <https://developer.nvidia.com/deep-learning>

OpenCV Object Tracking:

[https://docs.opencv.org/4.x/d5/d54/group\\_objdetect.html](https://docs.opencv.org/4.x/d5/d54/group_objdetect.html)

Makesense.ai - Privacy Focus: <https://www.makesense.ai/>