## **Code Flow and Detailed Explanation**

This project performs **object detection on both images and videos** using OpenCV’s DNN module and a pre-trained **SSD MobileNet V3** model trained on the **COCO dataset**. It runs in **Google Colab** and uses **Google Drive** for file I/O.

### **Cell-by-Cell Code Flow Summary**

### **Cell 1 – Mount Google Drive**

Mounts your Drive to access files like:

* Input images/videos
* Model files
* Output results

### **Cell 2 – Install OpenCV**

Installs opencv-python, used for:

* Loading images/videos
* Running object detection
* Drawing boxes and labels

### **Cell 3–5 – Download and Extract Model Files**

* Downloads ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.tar.gz (contains .pb weights)
* Downloads .pbtxt config file (model structure)
* Extracts the tar file so that the .pb file is accessible

### **Cell 6 – Download COCO Labels**

Downloads coco.names, which contains 80 object class names (e.g., person, car, dog).

### **Cell 7 – Verify Label Loading**

Prints the number and first few class labels to confirm they were loaded correctly.

### **Cell 8 – Check Files**

Lists the current directory and extracted model folder to confirm that the necessary files exist:

* frozen\_inference\_graph.pb
* ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt

### **Cell 9 – Load the Model**

Loads the SSD MobileNet model using OpenCV's cv2.dnn\_DetectionModel, and configures:

* Input size (320x320)
* Scaling and normalization
* Color conversion (BGR → RGB)

### **Cell 10 – Run Object Detection on an Image**

* Reads an image from Drive
* Runs object detection
* Draws bounding boxes and labels
* Displays the annotated image using matplotlib

### **Cell 11 – Run Object Detection on a Video**

* Loads a video from Drive
* Reads video properties (resolution, FPS)
* Initializes a video writer
* Processes each frame:  
  + Detects objects
  + Draws bounding boxes and labels
  + Writes output frames to a video
* Saves the output to Drive