- **10 OBJECT DETECTION FUNCTION IN TENSORFLOW**
- **tf.keras.utils.get_file** or similar: Function used for downloading pre-trained model checkpoints (e.g., from the TensorFlow Model Zoo).
- **tf.saved_model.load**: Core function to load a pre-trained or fine-tuned object detection model into memory for inference.
- **tf.image.decode_image**: Used for decoding raw image data (e.g., from a file) into a TensorFlow tensor.

- **tf.image.resize** or similar preprocessing: Functions to resize the input image to the dimensions expected by the specific model architecture (e.g., 512x512 for an EfficientDet-D0 model).
- **tf.expand_dims**: A function to add a batch dimension to a single image tensor, as models typically expect a batch of images as input (e.g., changing shape from [height, width, channels] to [1, height, width, channels]).
- **model.predict()** or calling the model directly: The primary function for running the

inference process on the input image tensor to get raw detection outputs.

- **detection_boxes / detection_scores / detection_classes**: While not a single function, these are the key output dictionary keys/tensors from a detection model that contain the normalized coordinates of bounding boxes, confidence scores, and class indices for detected objects.
- **tf.image.non_max_suppression** (NMS): A crucial post-processing function that filters out redundant and overlapping

bounding boxes, ensuring only a single, high-confidence box per object instance is kept.

Helper Functions: Custom helper functions (e.g., using libraries like OpenCV or Matplotlib) are used to draw the final bounding boxes and labels onto the original image for human interpretation.

- **model.save()**: Function used to save a fine-tuned or newly trained model in the SavedModel format for future deployment.