# Algorithm:

1. Download Files:
   - Use Google Drive file IDs to download required files: `coco.names`, `yolov3-tiny.weights`, `yolov3-tiny.cfg`, and an example image (`bus.jpg`).

2. Import Libraries:
   - Import necessary libraries and modules: `cv2`, `numpy`, `matplotlib.pyplot`, and `sys`.

3. Read Object Names:
   - Open the `coco.names` file and read the object names into the `classes` list.

4. Search Object in Database:
   - Prompt the user to input an object name.
   - Check if the entered object exists in the database (`classes` list).
   - If found, proceed; otherwise, exit the program.

5. Load and Preprocess Image:
   - Load the example image (`bus.jpg`) using OpenCV.
   - Perform preprocessing:
     - Scale pixel values to the range [0, 1].
     - Resize the image to (320, 320).
     - Convert color channels from BGR to RGB.
     - Convert the image to a blob (4D numpy array).

6. Visualize Preprocessing:
   - Display the original and preprocessed images for comparison using `matplotlib.pyplot`.

7. Load YOLO Model:
   - Use OpenCV to load YOLOv3-tiny weights and configuration files.
   - Set the input image for the YOLO model.
   - Get the names of the output layers in the YOLO network.
   - Perform a forward pass through the YOLO network.

8. Extract Bounding Boxes:
   - Iterate through each output layer and detection.
   - Extract confidence scores, class IDs, and bounding box coordinates.
   - If confidence is above a certain threshold, add the detection information to respective lists.

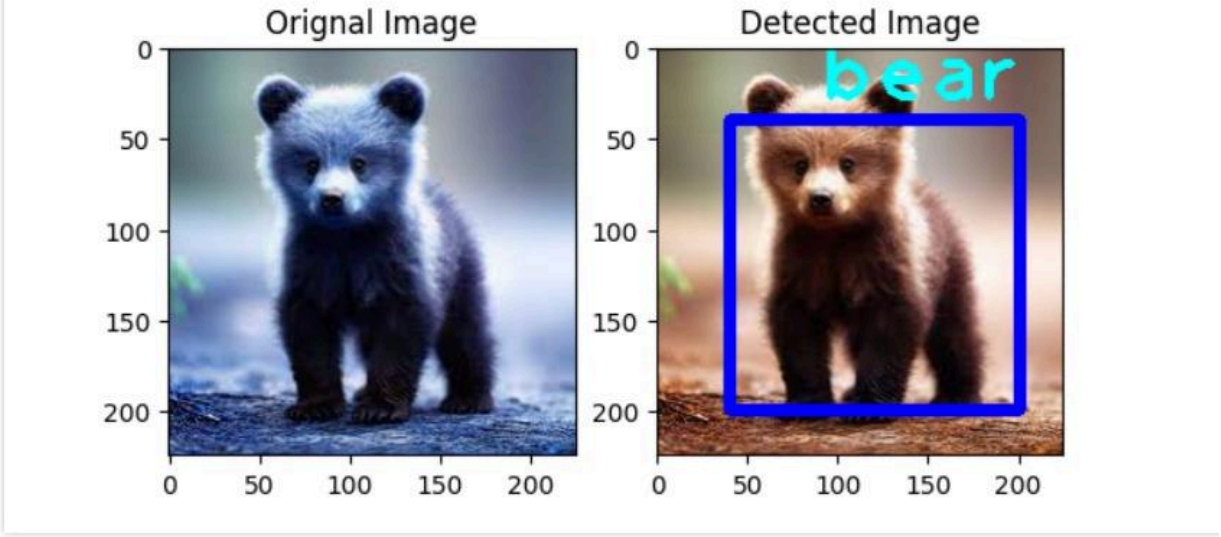
9. Non-Maximum Suppression (NMS):
   - Use OpenCV's NMSBoxes function to perform non-maximum suppression.
   - Filter out overlapping bounding boxes.

10. Draw Bounding Boxes:
   - Iterate through the remaining bounding boxes after NMS.
   - Draw bounding boxes and labels on the original image using `cv2.rectangle` and `cv2.putText`.

11. Visualize Results:
   - Display the original image with detected objects alongside the original image using `matplotlib.pyplot`.

Current Code Output (Code is uploaded in Git Hub Read me file)

```
plt.subplot(1,2,1)
plt.title("Orignal Image")
plt.imshow(finall)
plt.subplot(1,2,2)
plt.title("Detected Image")
plt.imshow(yolo_img)
plt.show()
```



```
plt.subplot(1,2,1)
plt.title("Orignal Image")
plt.imshow(finall)
plt.subplot(1,2,2)
plt.title("Detected Image")
plt.imshow(yolo_img)
plt.show()
```