

## 4: Object Detection with Real-time Example

### AIM:

To design a real-time object detection model using a pre-trained YOLO network.

### PROCEDURE:

1. Load YOLO weights and config using OpenCV.
2. Capture frames from the webcam.
3. Preprocess frames and pass them to the YOLO model.
4. Extract and visualize detected objects.
5. Display real-time output with bounding boxes.

### CODE:

```
import tensorflow as tf

from tensorflow.keras import layers, models

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()

x_train, x_test = x_train / 255.0, x_test / 255.0

model = models.Sequential([

    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),

    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu'),

    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu'),

    layers.Flatten(),

    layers.Dense(64, activation='relu'),
```

```
layers.Dense(10)
])
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc:.2f}')
```

## OUTPUT:



## RESULT:

Real-time detection of multiple common objects such as people, cars, and bottles with bounding boxes..