**Object Detection in Images using YOLOv5**

**A PROJECT FILE**

**for**

**INTRODUCTION TO AI (AI201B)**

**Session (2024-25)**

**Submitted by**

**Kiran Yadav 202410116100102**

**Gulsan Kumari 202410116100079**

**Dolly Prajapati**

**202410116100069**

**Khushi Kumari 202410116100101**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Mr. APOORV JAIN**

**Assistant Professor**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(APRIL- 2025)**

**TABLE OF CONTENT**

1. **Introduction**

**1.1 Background** 1.2 **Motivation** 3

**1.3 Objectives** 3

**1.4 Significance** **3**

1. **Overview** **4**
2. **Methodology** **5**
3. **Code** **6**
4. **Output** **7-8**
5. **Conclusion** **9**

# INTRODUCTION

1. **Introduction**

**1.1 Background**

Object detection is a fundamental task in computer vision that involves identifying and localizing objects within an image. It combines object classification with the task of drawing bounding boxes around the detected objects..

**1.2 Motivation**

With advancements in AI, object detection is becoming a vital tool in various fields such as self-driving cars, medical imaging, surveillance, and retail analytics. Traditional methods were slower and less accurate, hence deep learning models like YOLO (You Only Look Once) were developed for real-time object detection.

**1.3 Objectives**

* Implement object detection using YOLOv5.
* Apply a pretrained model to identify objects in a user-uploaded image.
* Visualize the detected objects along with bounding boxes.

**1.4 Significance**

This project highlights the capability of modern AI systems to perform real-time object recognition with high speed and accuracy, which is essential for automation, robotics, and safety systems.

# OVERVIEW

YOLOv5 (You Only Look Once version 5) is a state-of-the-art, real-time object detection model developed by Ultralytics. It is lightweight, fast, and pretrained on the COCO dataset, which includes over 80 different object categories.

The project workflow follows a simple structure:

1. Cloning the YOLOv5 repository.
2. Installing the necessary dependencies.
3. Uploading an image.
4. Running the detection script using pretrained weights.
5. Displaying the detected output.

The architecture of YOLOv5 includes a Backbone (CSPDarknet), a Neck (PANet), and a Head for making predictions. It takes a 640x640 image as input and outputs bounding boxes, labels, and confidence scores.

**METHODOLOGY**

1. **Clone Repository:**
   * Clone the official YOLOv5 GitHub repository from Ultralytics.
2. **Install Dependencies:**
   * Install required Python libraries including PyTorch, OpenCV, and others listed in requirements.txt.
3. **Upload Image:**
   * Upload an image file using Google Colab’s file upload feature.
4. **Run Detection:**
   * Use YOLOv5’s pretrained model (yolov5s.pt) to detect objects in the uploaded image.
   * Execute detection using the command-line interface with parameters for image size and confidence threshold.
5. **Display Results:**
   * Output is saved in the runs/detect/exp/ folder.
   * Display the image with bounding boxes using IPython’s display function.

# CODE

!git clone https://github.com/ultralytics/yolov5

%cd yolov5

!pip install -r requirements.txt

from google.colab import files

uploaded = files.upload()

import os

from pathlib import Path

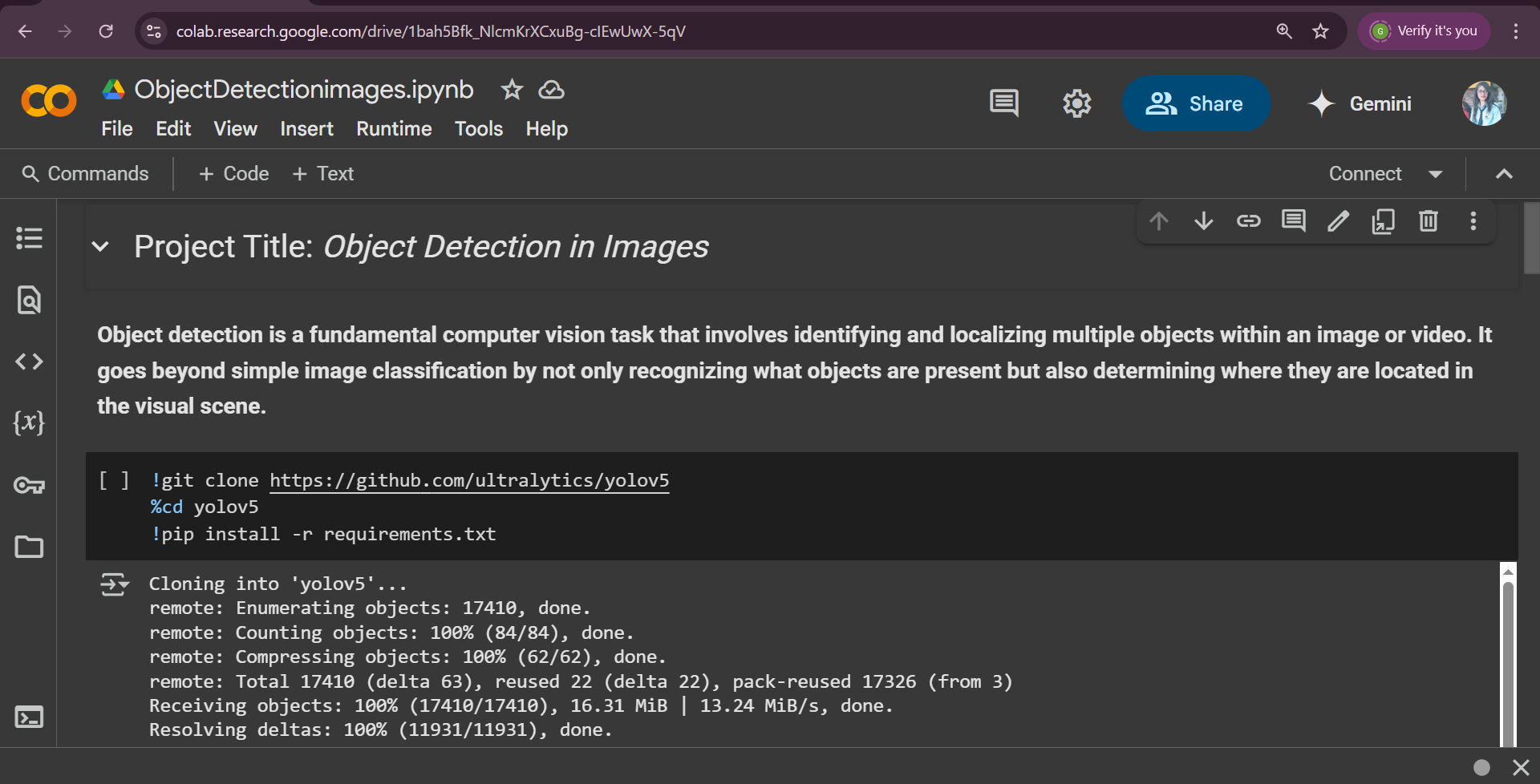
img\_path = list(uploaded.keys())[0]

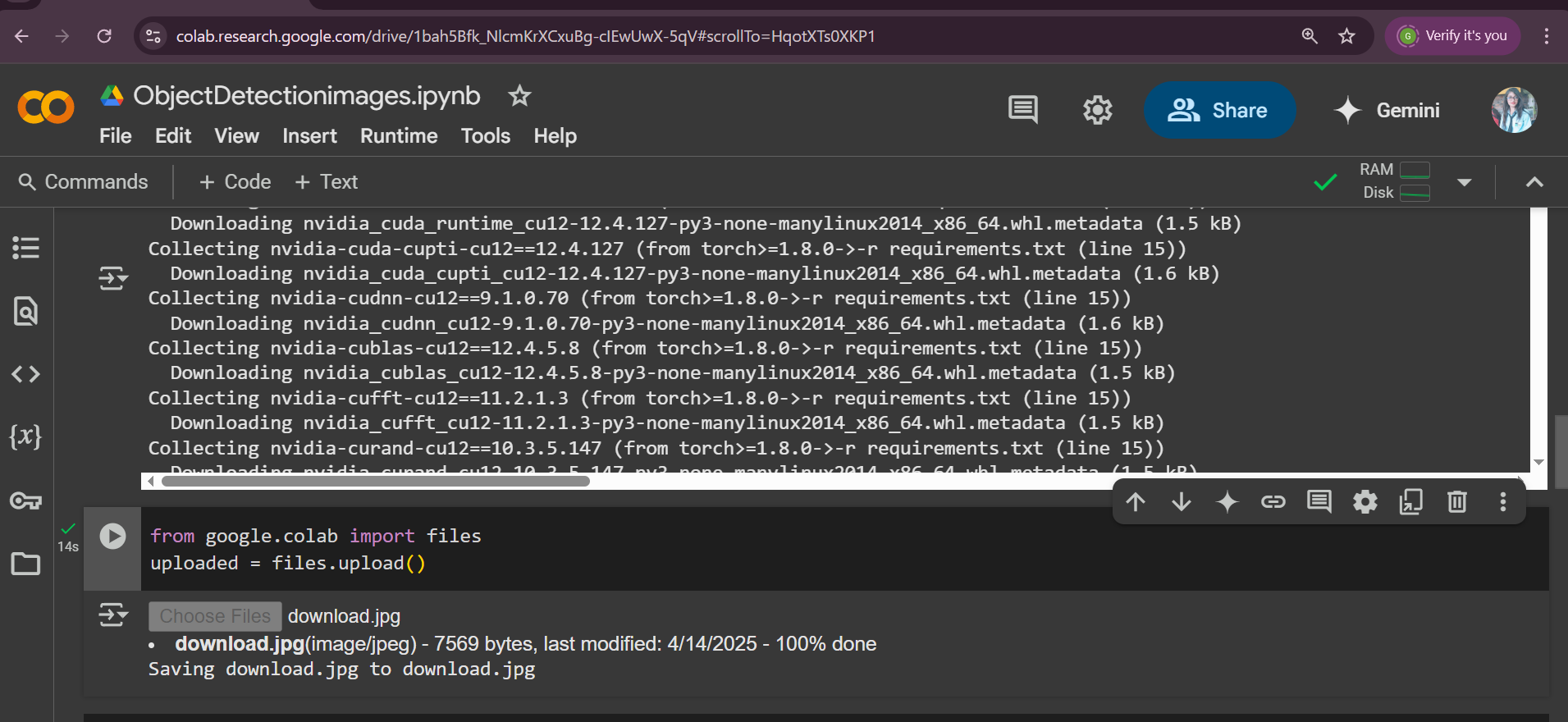
!python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source {img\_path}

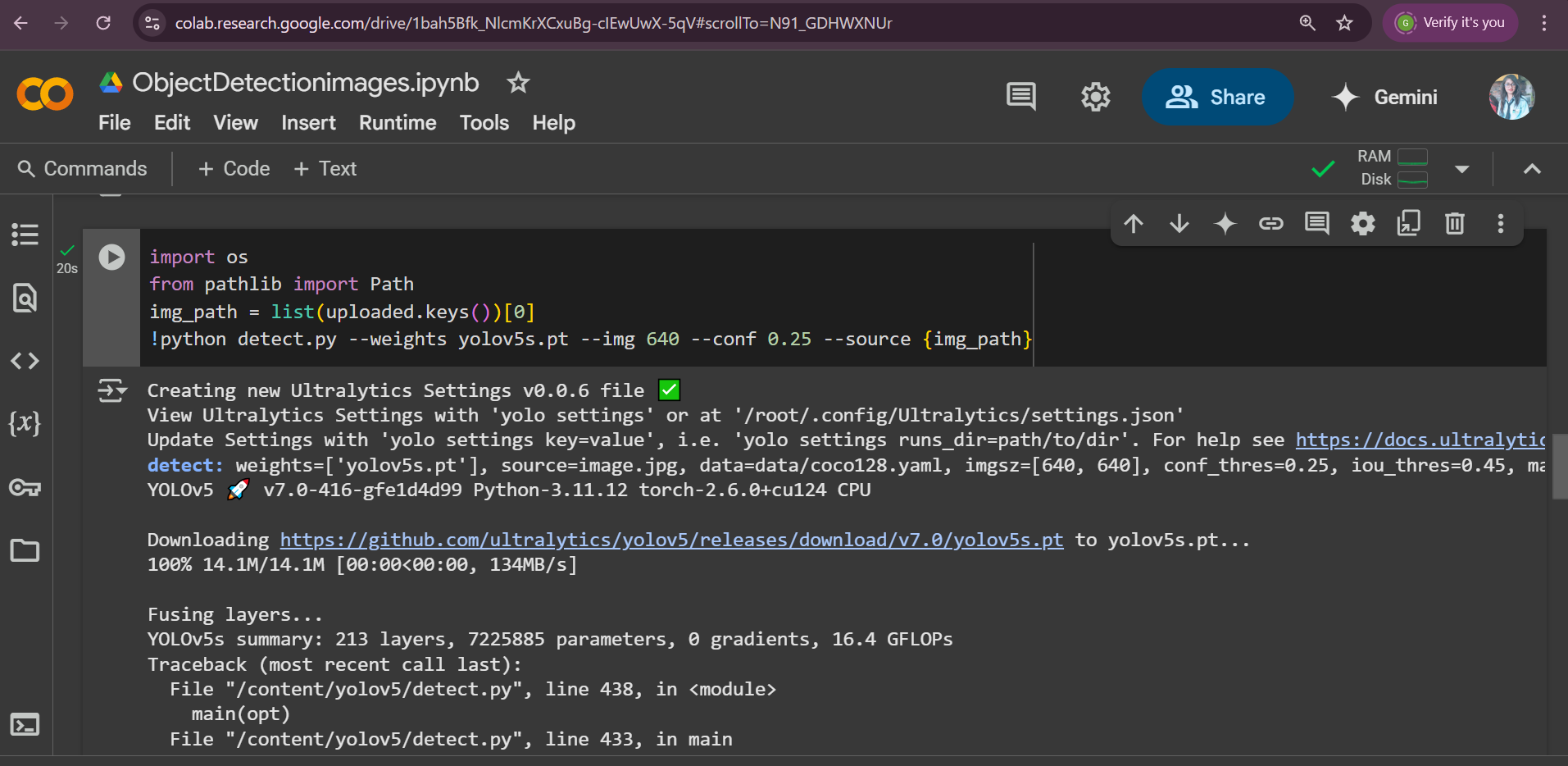
from IPython.display import Image, display

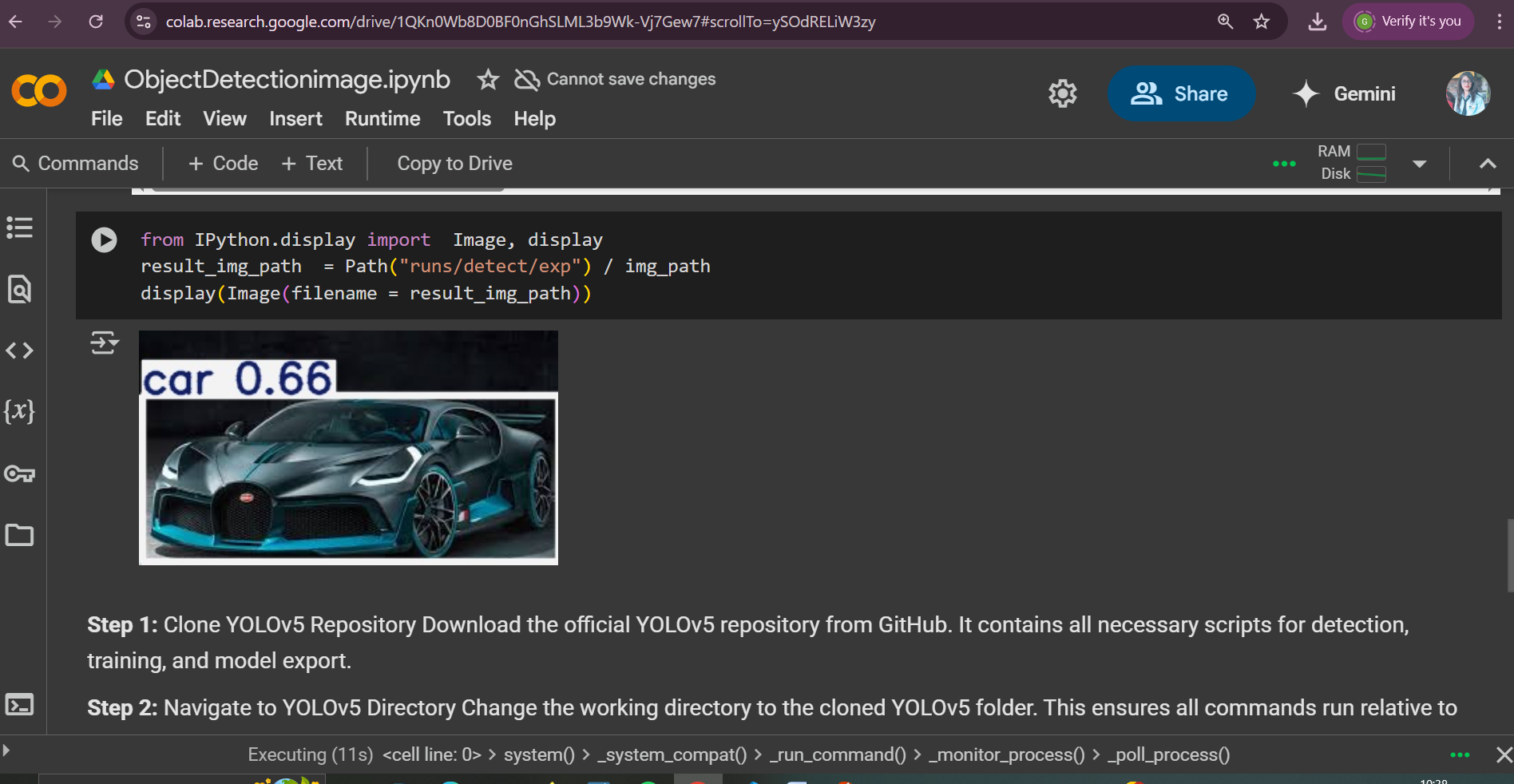
result\_img\_path = Path("runs/detect/exp") / img\_path

display(Image(filename=result\_img\_path))**OUTPUT**

****







# CONCLUSION

This project successfully demonstrates object detection using YOLOv5, a deep learning model that performs detection in real time with impressive speed and accuracy.

Using pretrained weights and simple Python scripts, we were able to identify and label multiple objects in an image. The implementation shows how YOLOv5 breaks down the image, analyzes visual features, and makes predictions efficiently.

This solution can be extended to real-world use cases like autonomous driving, security systems, healthcare, and retail monitoring. With the power of AI, object detection is becoming faster, more reliable, and easier to integrate into smart systems.

In conclusion, YOLOv5 provides a robust framework for real-time object detection and showcases the potential of AI in visual perception tasks.

.

Top of Form

Bottom of Form