



car 0.04 car 0.32 person 0
car 0.6 car 0.75



GROUP 1

PRESENTATION

By Faizan, Pasha, Eric, Frank

CONTENT

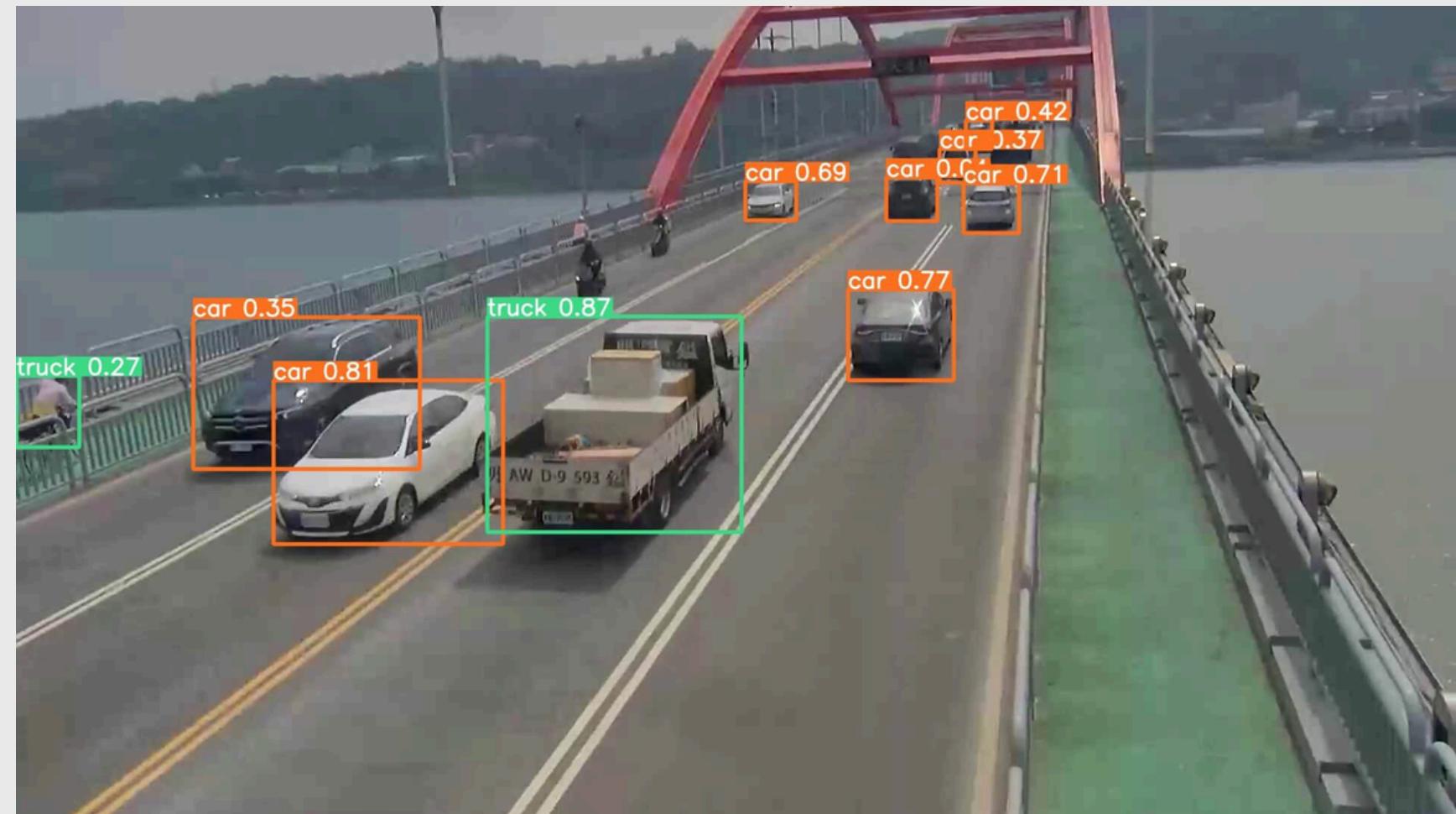
1. Steps to Build a YOLO Object Detection Model
2. What is Object Detection?
3. Why YOLOv8?
4. How We Made the Dataset
5. Training the Model
6. Training Progress
7. Various Curves
8. Confusion Matrix & Validation Predictions
9. Applications & Challenges
10. Future Work & Conclusion

STEPS TO BUILD A YOLO OBJECT DETECTION MODEL

- Make a diverse dataset with different angles, lighting, and backgrounds
- Label all images by drawing bounding boxes
- Save images and labels together so YOLO can match them correctly
- Split the dataset into train/val/test
- Upload the dataset to Google Colab (or mount Drive)
- Set up YOLO and run the training in Colab
- Download the trained model weights
- Test the model on photos, videos, and live camera
- Retrain or fine-tune if results need improvement

WHAT IS OBJECT DETECTION?

- Computer program that looks at pictures/videos and finds objects.
- It draws boxes and names the objects.
- Example: Detecting cars in traffic.



- using Yolo v8

WHY YOLOV8?

YOLO: You Only Look Once

- Fast → works in real time
- Accurate → finds most objects correctly
- Easy to use → ready code and tools

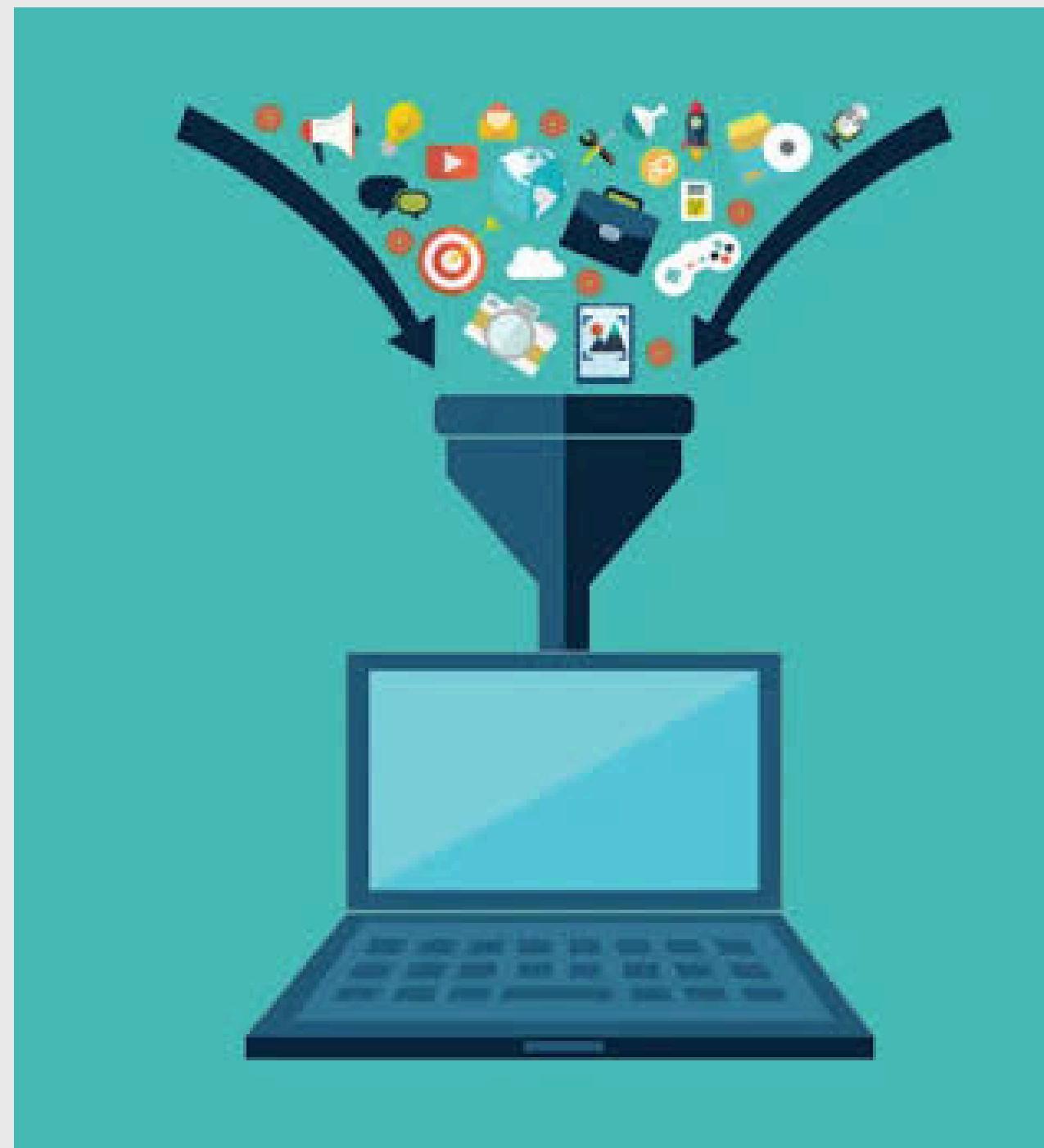


YOLOv8

HOW WE MADE THE DATASET

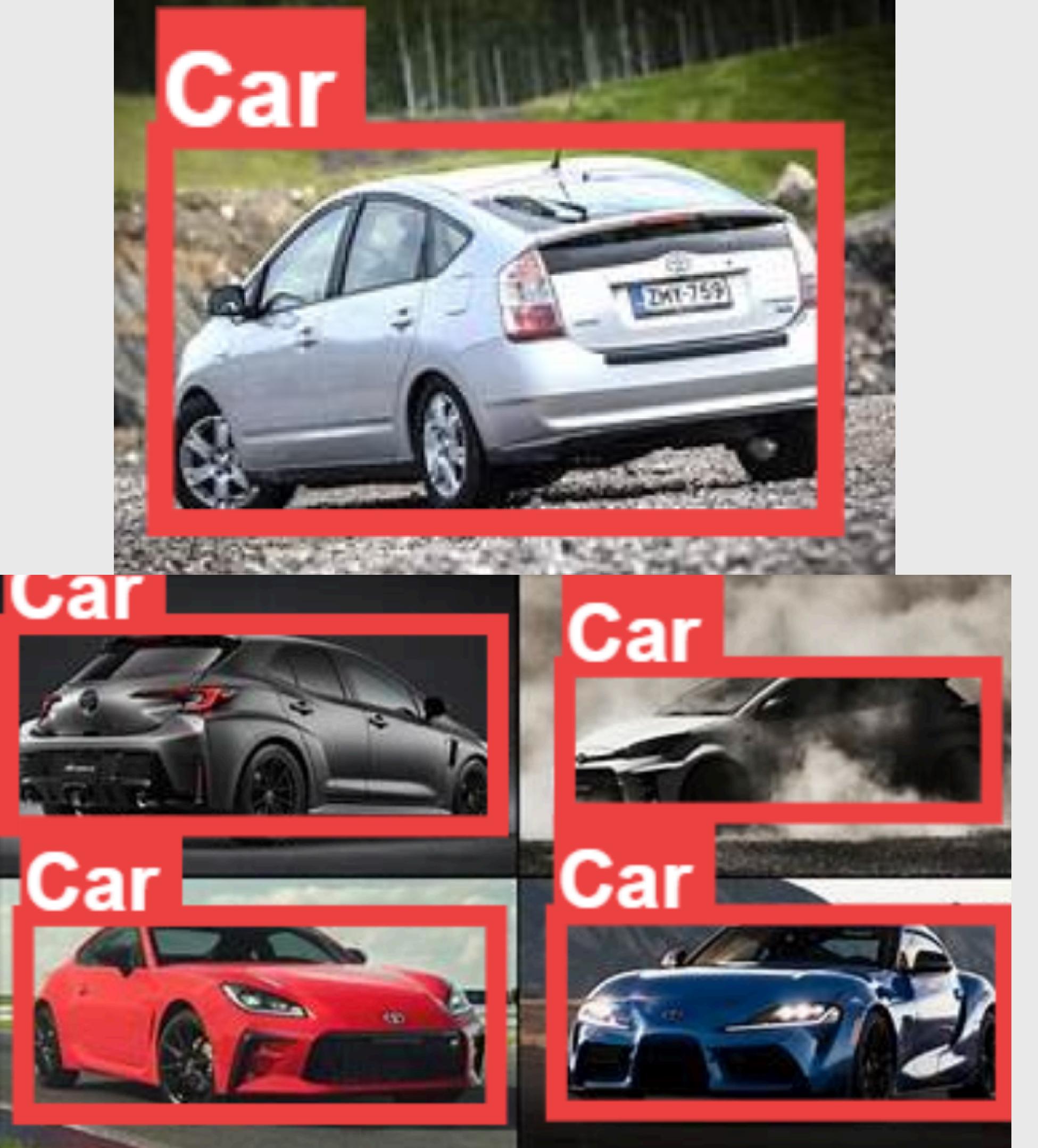
Step 1: Collect Images

- Collect images of cars of different type, from many angles, in different types of backgrounds.
- “The bigger the dataset, the better the model learns. The more variety in the dataset, the more accurate the model becomes.”



Step 2: Labeling Process

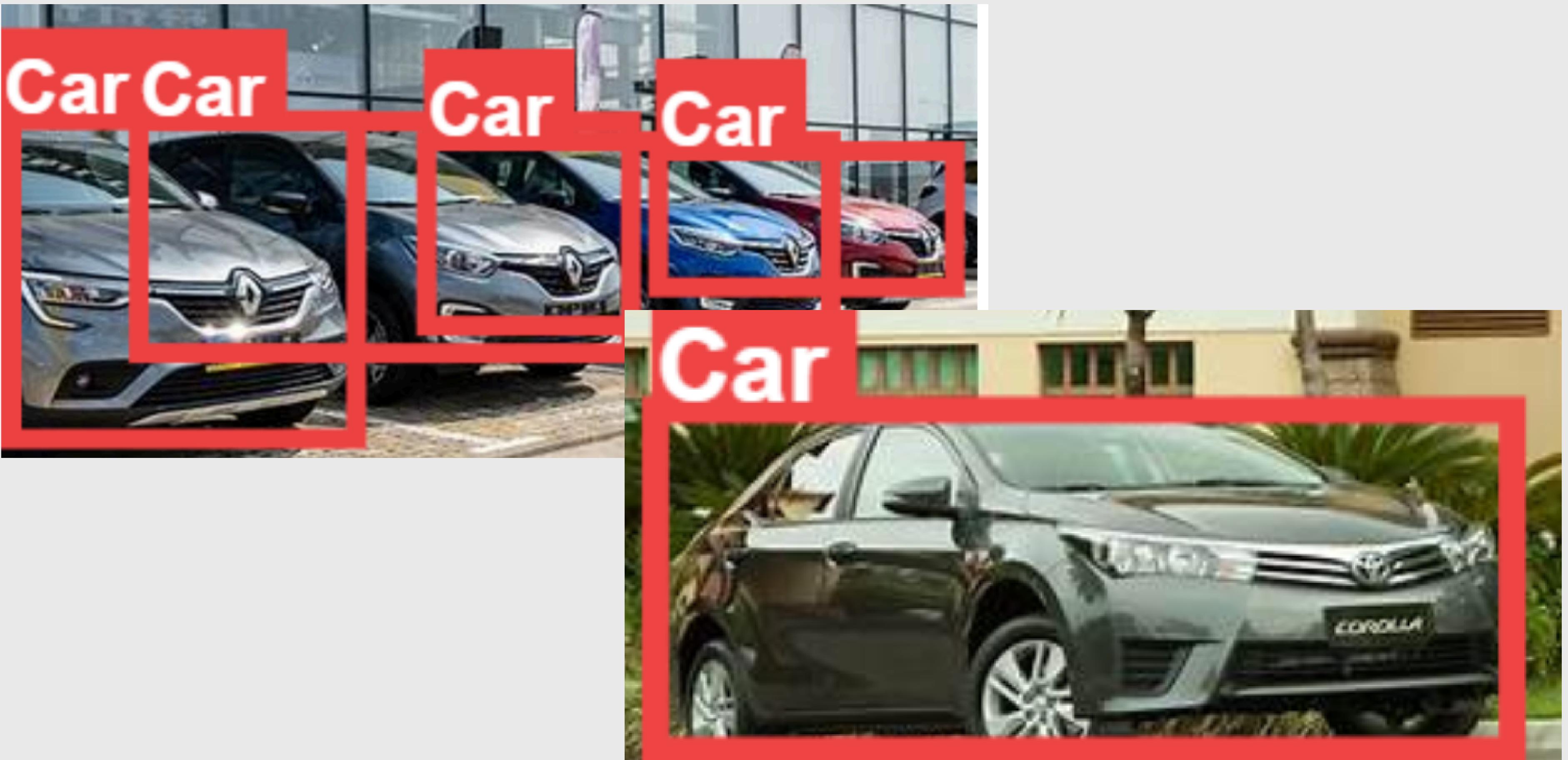
- Marking the area where your object is present. Those coordinates are saved and later that is used to train your YOLO model.
- It can be done in any labeling website like: Label Studio, Roboflow, YOLO labelling tool



HOW WE MADE THE DATASET



HOW WE MADE THE DATASET



TRAINING THE MODEL

Our Platform: Google Colab

- Colab gives us a GPU (special chip that makes training faster)
- Without GPU → training is very slow.
- Code we ran: `yolo detect train data=data.yaml model=yolov8n.pt epochs=50 imgsz=640`

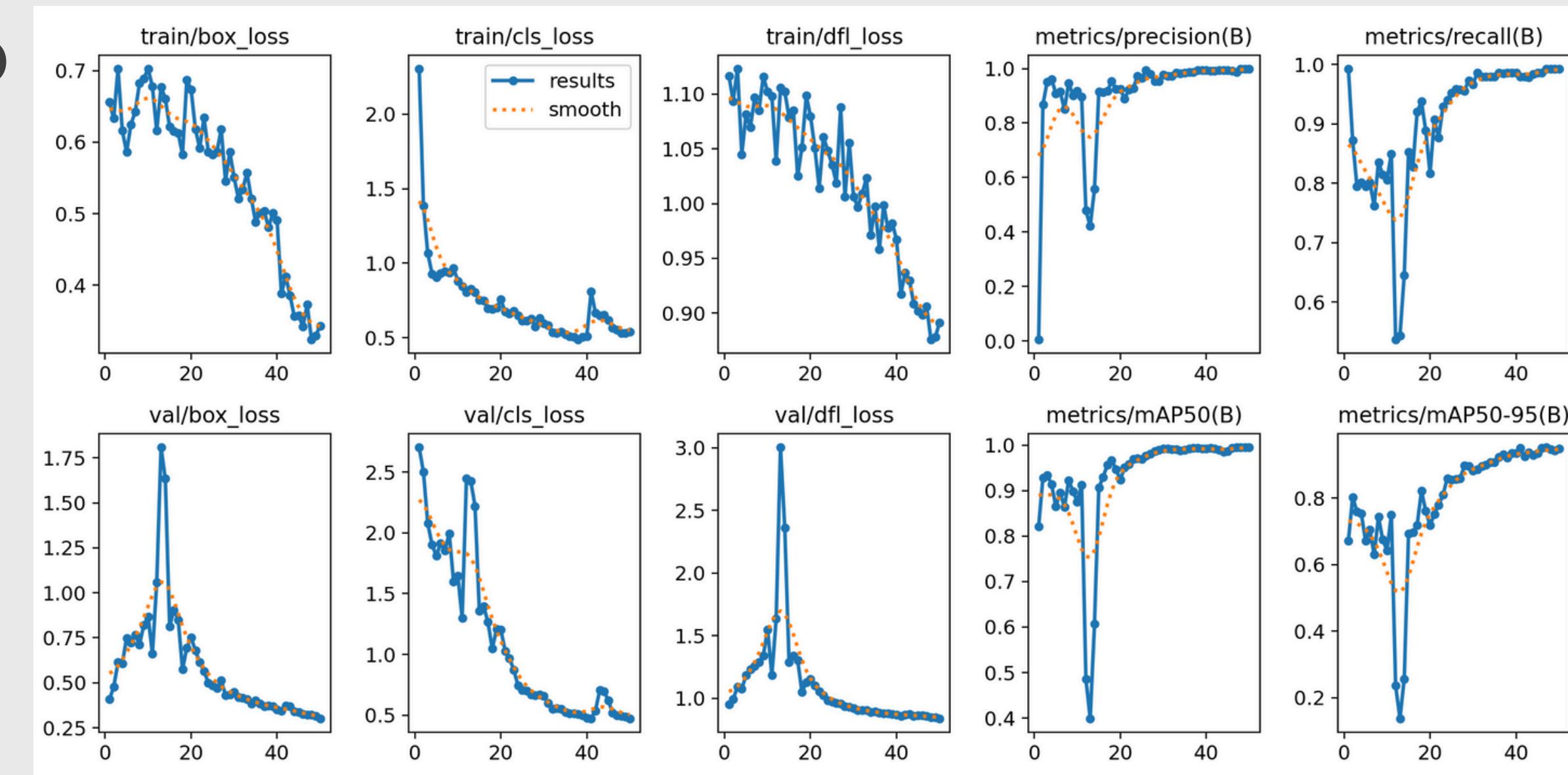
Google Colab



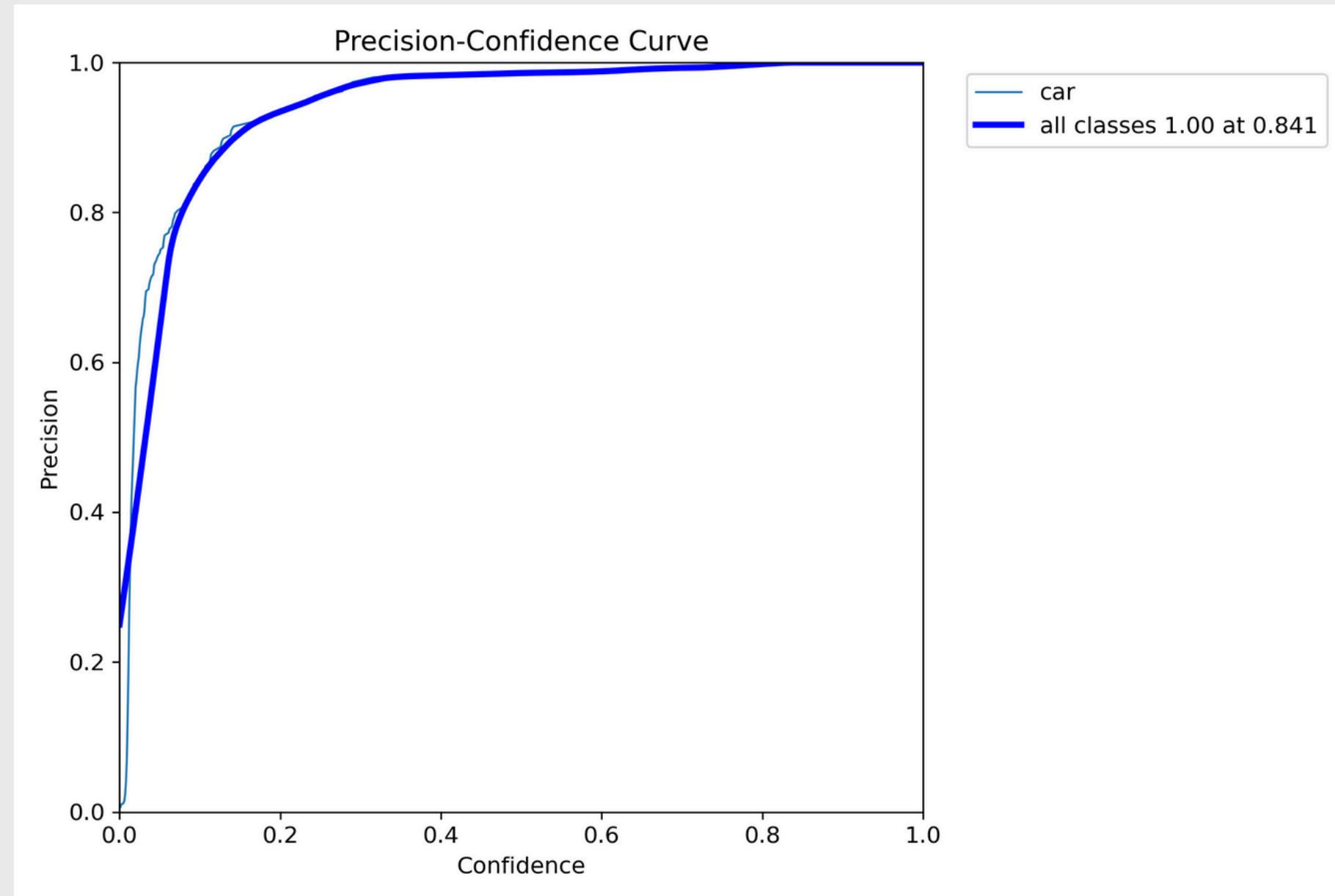
TRAINING PROGRESS

Model Learns Step by Step

- Loss goes down → means it's learning.
- Accuracy goes up → means it's getting better.

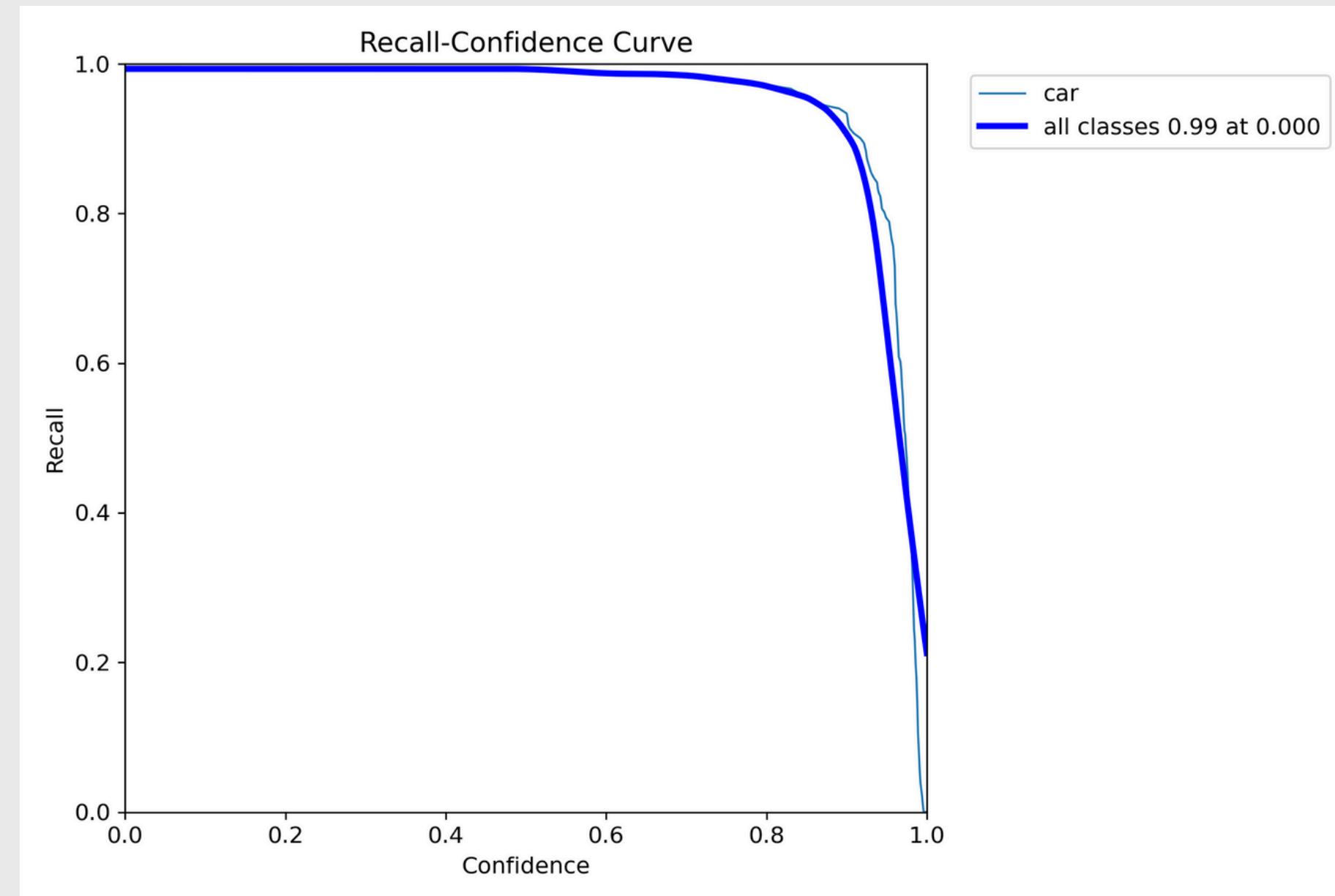


PRECISION CURVE



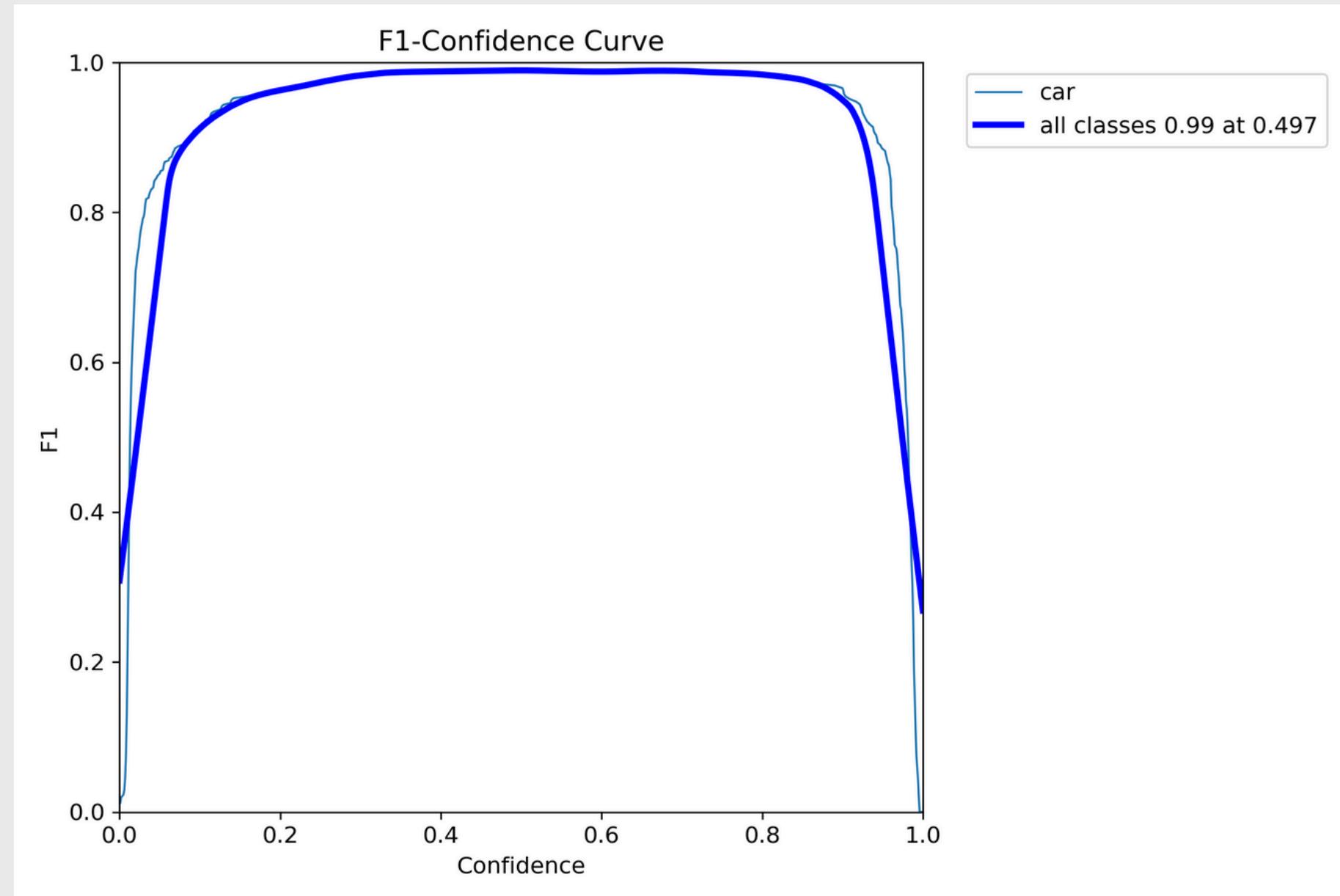
Precision: How many predictions are correct.

RECALL CURVE



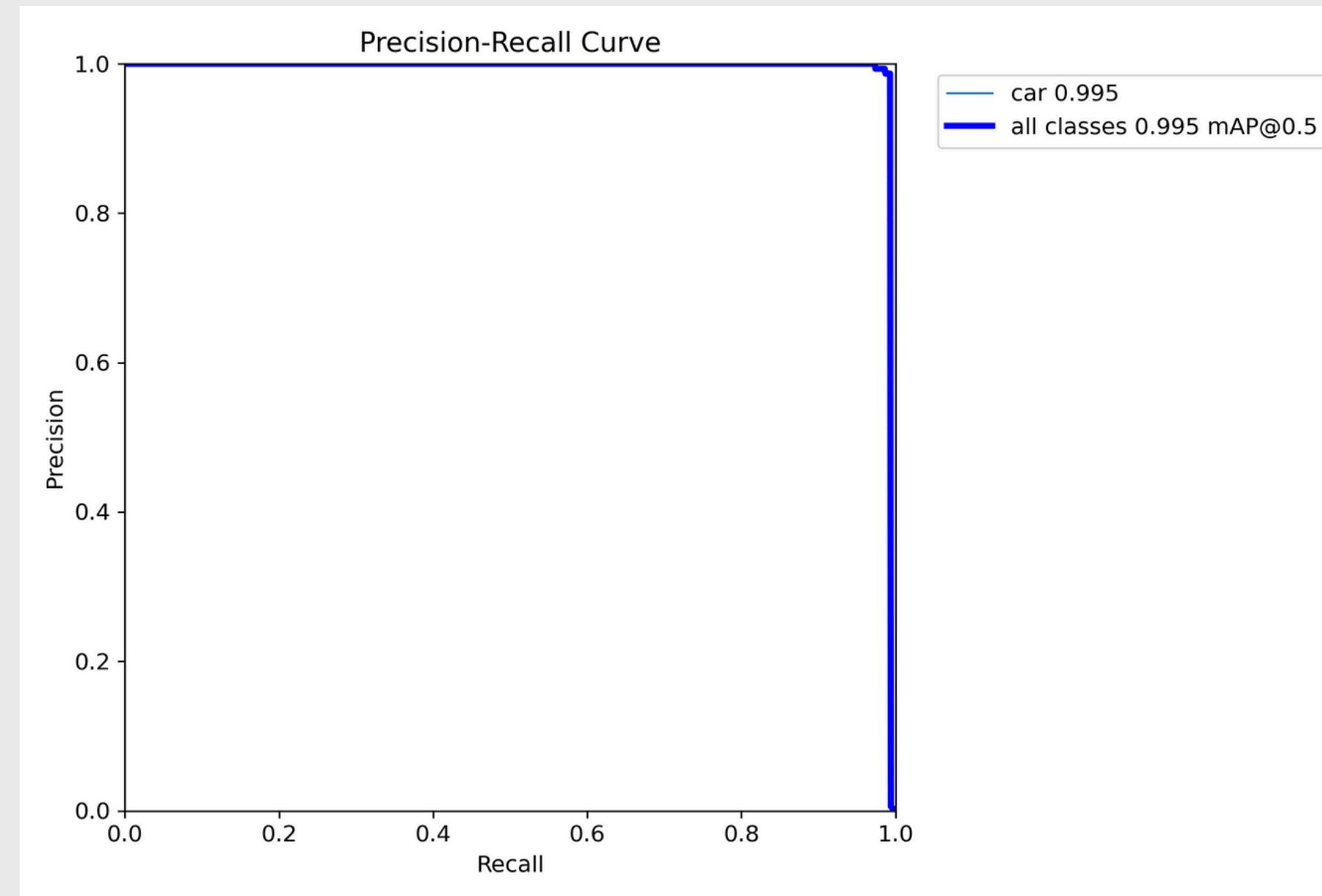
Recall: How many objects the model finds.

F1 SCORE CURVE



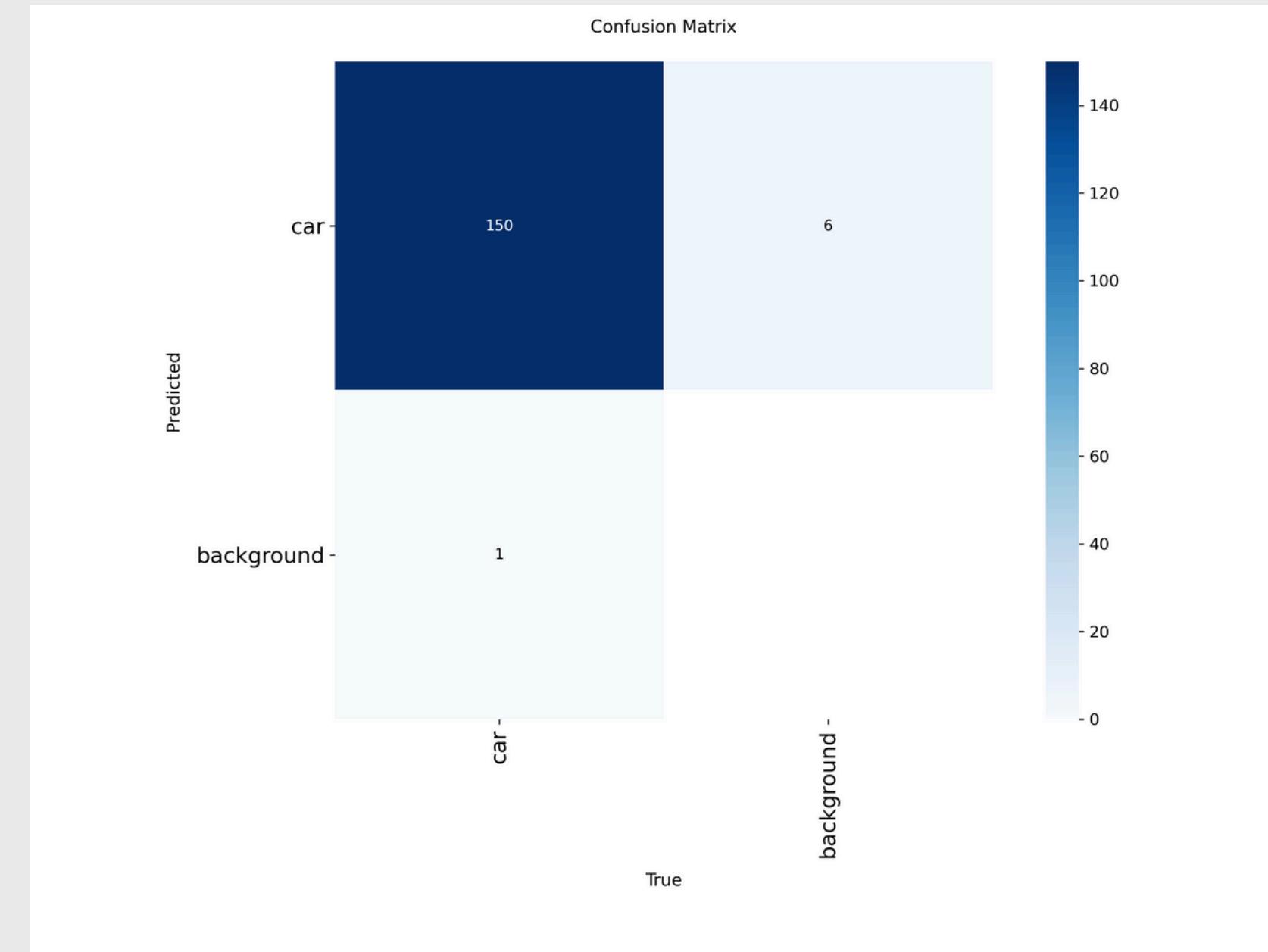
F1: Balance between precision and recall.

PRECISION-RECALL CURVE



Shows the trade-off between precision and recall.

CONFUSION MATRIX



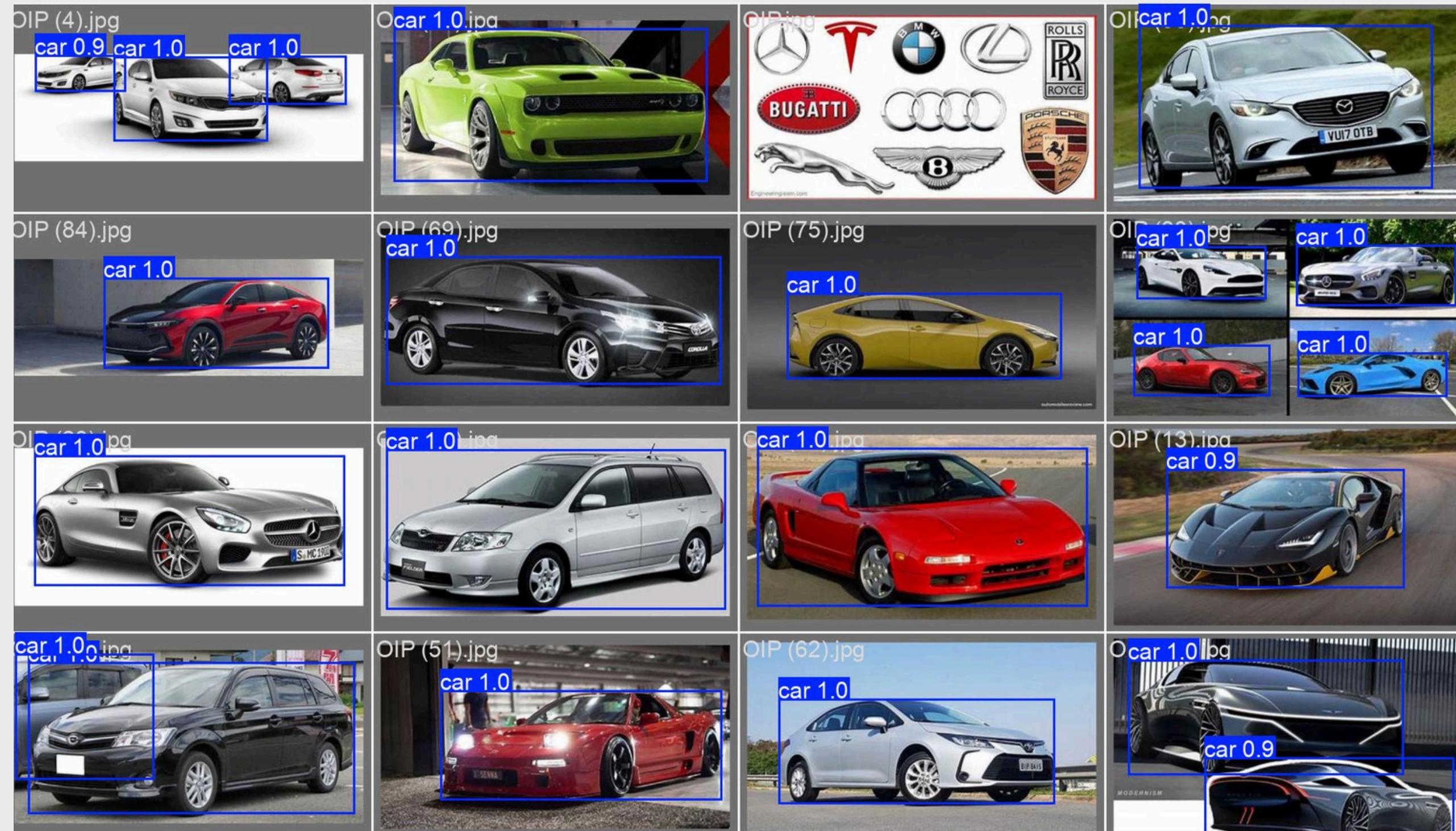
Shows correct vs wrong predictions for each class.

CODE WE USED FOR TESTING THE MODEL ON OUR PC

**YOLO PREDICT MODEL="RUNS/TRAIN/WEIGHTS/BEST.PT"
SOURCE="VIDEO.MP4" SHOW=TRUE SAVE=TRUE**

VALIDATION PREDICTIONS

Shows how the model predicts compared to the real labels.

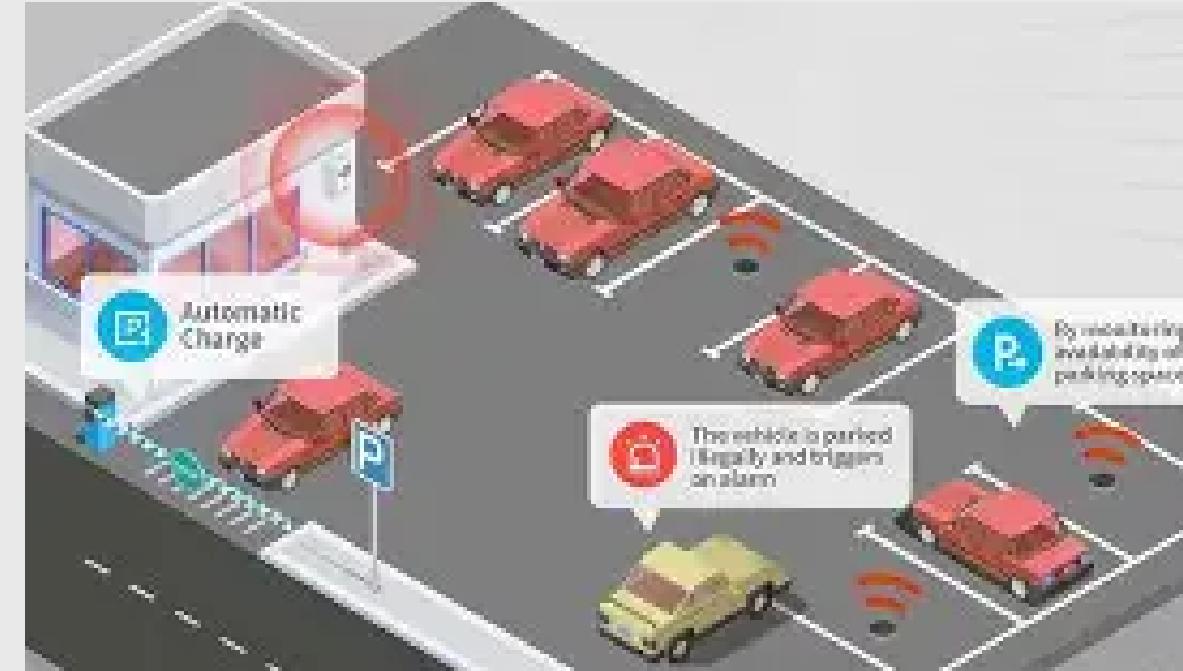


APPLICATIONS

Traffic monitoring



Parking systems



Self-driving cars

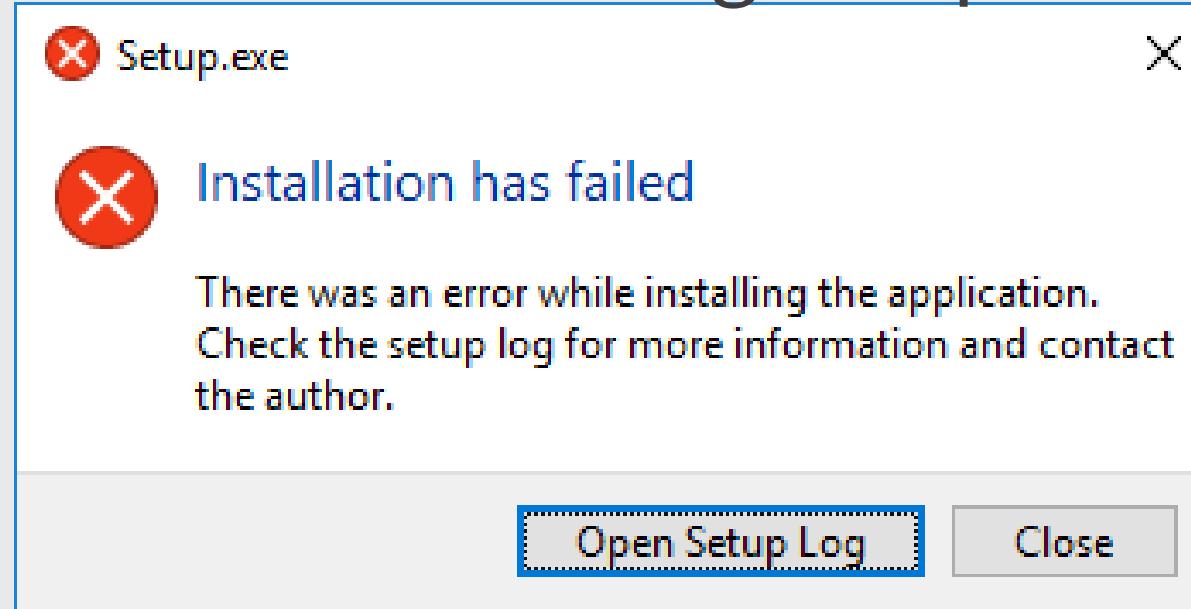


Security cameras



CHALLENGES

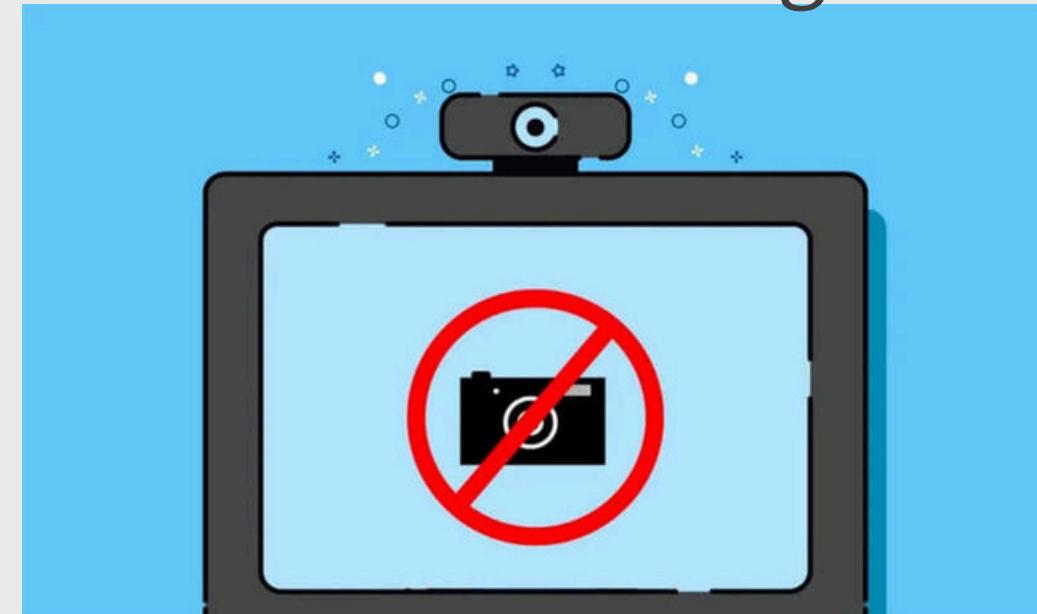
Errors during setup



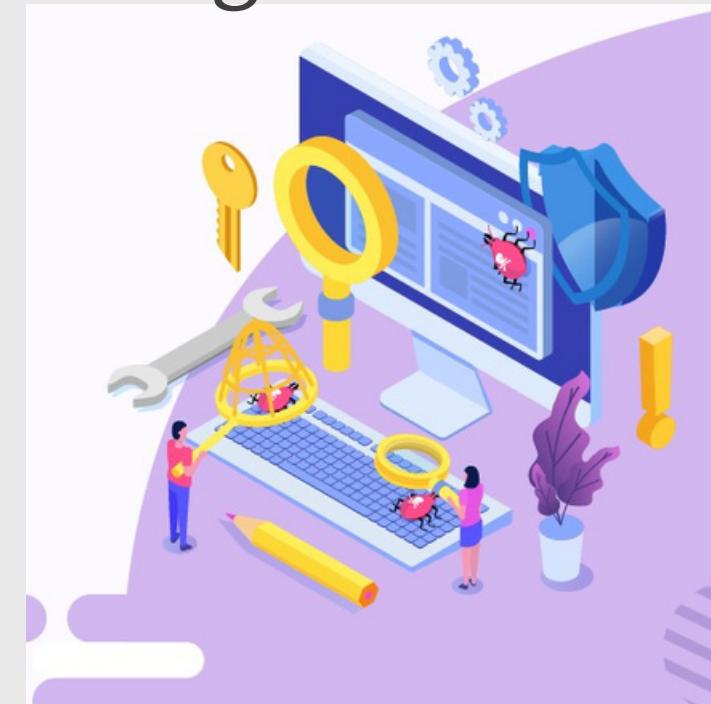
Dataset path issues



Webcam not working at first



Solved by using Colab and debugging



FUTURE WORK

- Train on bigger dataset
- Try larger YOLOv8 models (m, l) for better accuracy
- Deploy on mobile or edge devices

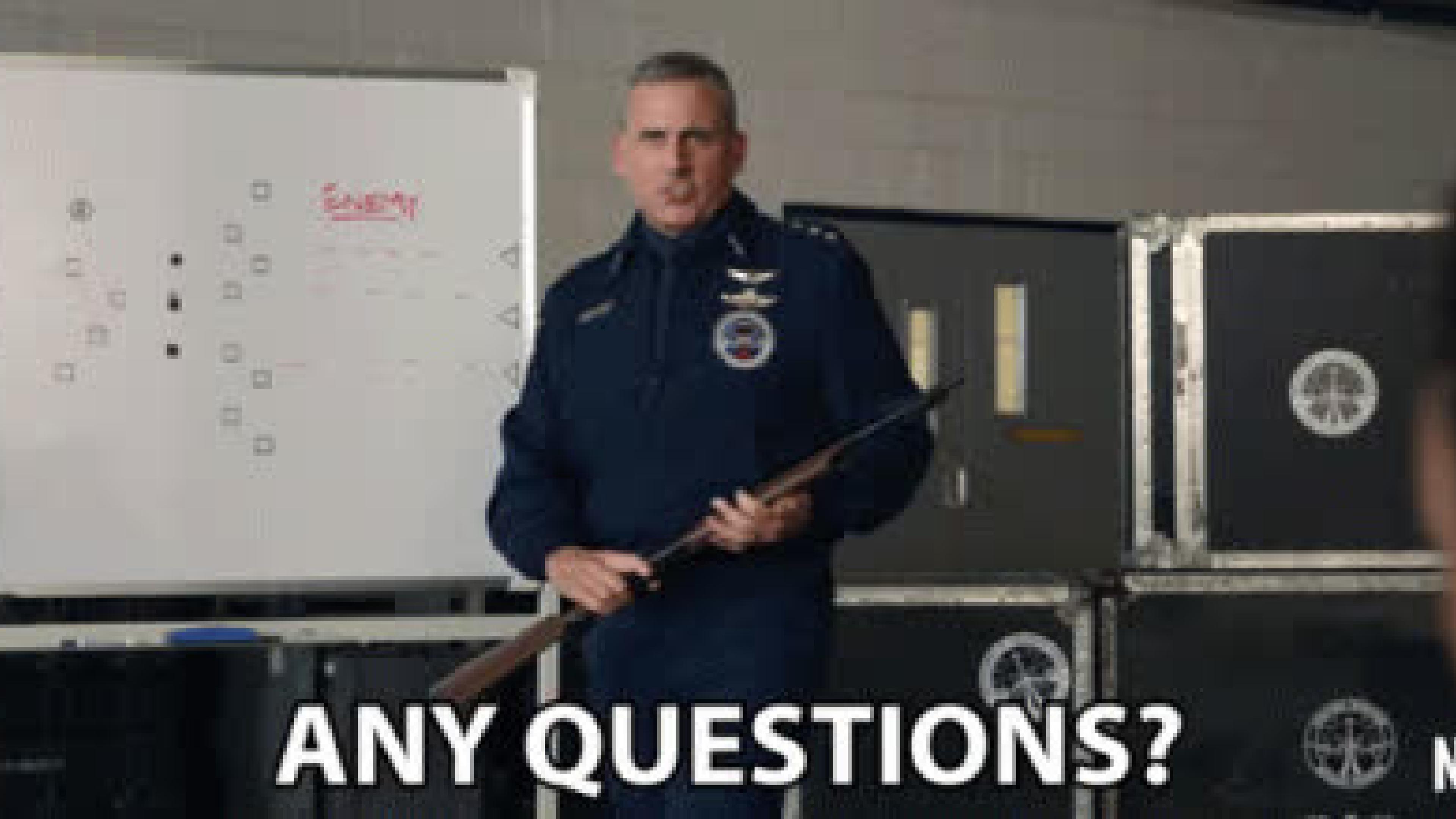


CONCLUSION

- We built a dataset, trained YOLOv8, and tested it.
- Model works well ($mAP > 0.9$).
- Ready for real-world use.



that's all for



ANY QUESTIONS?

THANK YOU....THANK YOU
VERY MUCH!

