LAB 9 REPORT

A PREPRINT

Daniele Cecca
Artificial Intelligence for Science and Technology
Milano Bicocca University
Supervised Learning

June 2, 2024

1 Introduction

This lab report details the training and analysis of YOLOv5s, a scaled-down version of YOLOv5, designed for object detection. Our aim was to assess the model's effectiveness in detecting objects within a custom video dataset.

2 YOLO

YOLO is an early object detection model based on convolutional networks.

In general, the input to the YOLO network is a 448x448 RGB image. This is passed through 24 convolutional layers that gradually decrease the representation size using max pooling operations while concurrently increasing the number of channels, similarly to the VGG network. The final convolutional layer is of size 7x7 and has 1024 channels. This is reshaped to a vector, and a fully connected layer maps it to 4096 values. One further fully connected layer maps this representation to the output.

The output values encode which class is present at each of a 7x7 grid of locations. For each location, the output values also encode a fixed number of bounding boxes. Five parameters define each box: the x- and y-positions of the center, the height and width of the box, and the confidence of the prediction. The confidence estimates the overlap between the predicted and ground truth bounding boxes. The system is trained using momentum, weight decay, dropout, and data augmentation. Transfer learning is employed; the network is initially trained on the ImageNet classification task and is then fine-tuned for object detection.

After the network is run, a heuristic process is used to remove rectangles with low confidence and to suppress predicted bounding boxes that correspond to the same object so only the most confident one is retained.

3 Dataset

The dataset contains images of 5 classes: 'Ambulance', 'Bus', 'Car', 'Motorcycle', 'Truck'.

4 Training

We train all the layers of the small model by using the following command:

```
!python train.py --data ../data.yaml --weights yolov5s.pt \
--img 640 --epochs {EPOCHS} --batch-size 16 --name {RES_DIR}
```

We have chosen the following hyperparameters for the training:

- Epochs: 25

- Batch size: 16

The images were of size 640x640.

5 Validation

To evaluate the validation set of images, which were used during training, we used some metrics like confusion matrix and F1-curve. Here we present only these two but many more were used.

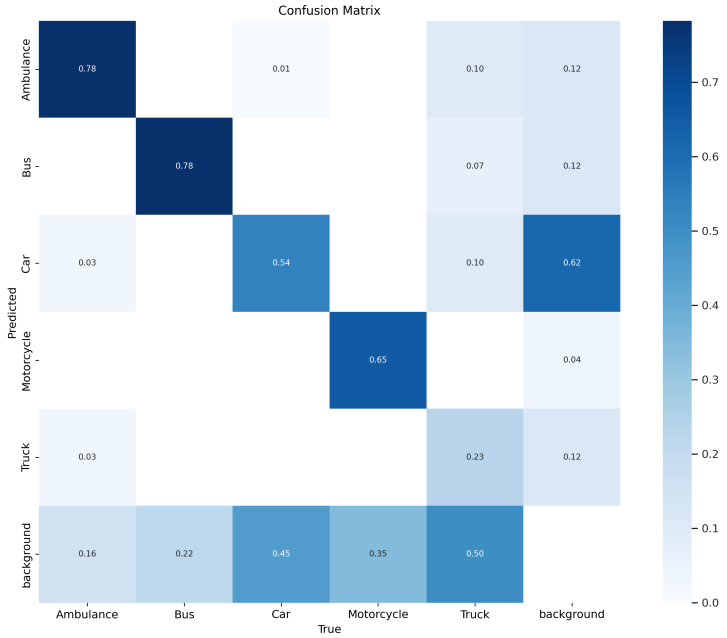

Figure 1: Confusion matrix

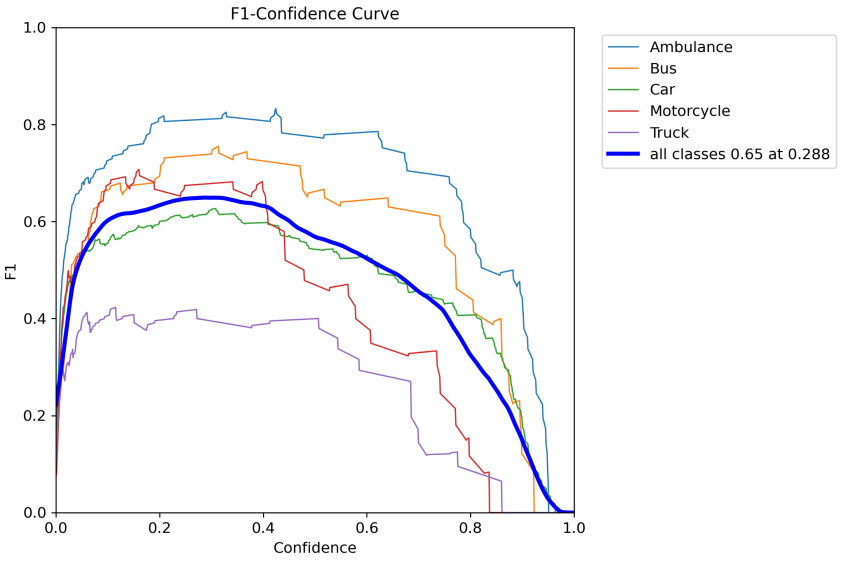

Figure 2: F1 curve

6 Test

We test the model both on images and videos and these are the results. Since in both cases we don't have true labels, we can only evaluate the results visually.

(a) Test 3

(b) Test 4

Figure 3: Test image 3-4

7 Test - Own Video

We also tested the model on my own video. Again, we can inspect the video visually.

As we can see from these images, the model was not able to identify the Motorcycle, perhaps because the Motorcycle's bounding box was too small compared to the cars.

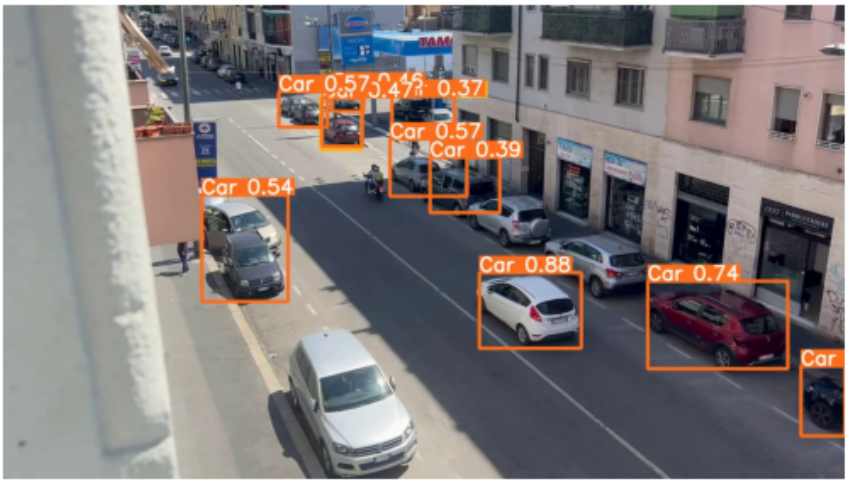

Figure 4: Motorcycle

The model identifies a bicycle as a motorcycle, which in a sense is correct because we didn't train our model on bicycles.

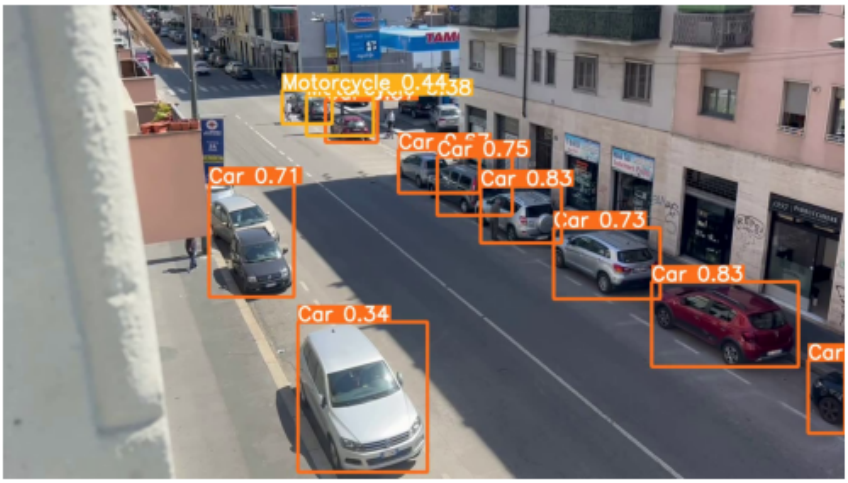

Figure 5: Bicycle

Not all objects were detected, so one limitation of this algorithm could be the number of vehicles.

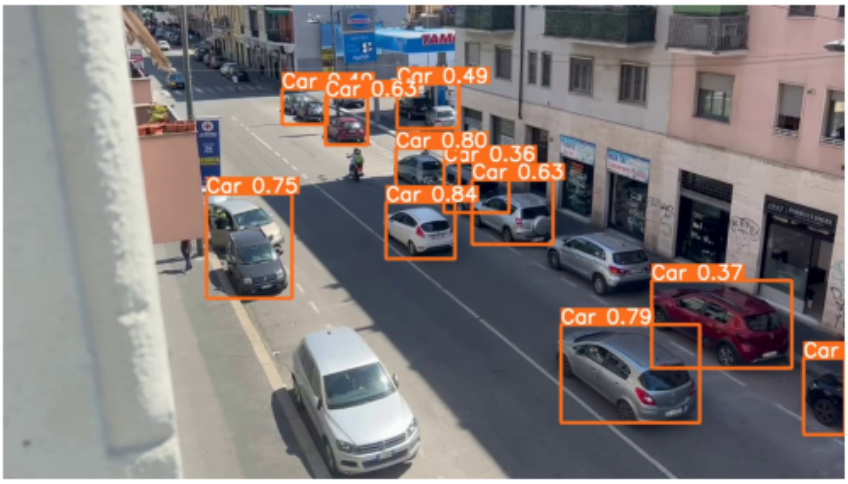

Figure 6: All objects

8 Conclusion

Our analysis of YOLOv5s for object detection reveals promising results, with the model demonstrating effectiveness in identifying and classifying objects in various scenarios. While the model performs well overall, some limitations were observed, suggesting the need for further refinement. Future enhancements could involve expanding the training dataset and fine-tuning hyperparameters to improve performance and robustness.

References

[1] Understanding Deep Learning, pag. 177 *YOLO - Understanding DL.* Available at: `https://udlbook.github.io/udlbook/`. Accessed 2 June 2024.