Name: Leama Tah

REPORT: OBJECT DETECTION USING UNSUPERVISED LEARNING.

Abstract

Recent technology innovation on vehicles have make a breakthrough in the artificial neural network domain. However, traffic safety and surveillance remained affected by rapid urban development. This project proposes a high-performance car detection system using unsupervised learning. I employ a modified YOLOv8 architecture, which is renowned for its speed and accuracy, to detect vehicles in various environmental conditions and from multiple camera perspectives.
The proposed system is trained on a comprehensive dataset comprising over 1178 images of dataset collected from Kaggle. It is a balance dataset and contain images with or without car.
The objective of this project is to first, achieve a high detection rate quantified by precision and recall metrics, aiming for over 95% accuracy in vehicle detection. Second, to ensure the model's real-time performance capability, I will train the model under different epochs and analyze the performance.
This project not only contributes a practical tool for vehicle detection but also extends the theoretical understanding of applying convolutional neural networks in dynamic and real-world environments. Future work will explore model training on different dataset to enhance model performance, proceed to project deployment and compare the performance of complex YOLO versus tiny YOLO.

Introduction

In today's rapidly urbanizing world, effective traffic management and enhanced vehicular safety have become primordial. With the development of smart cities and autonomous vehicles, the ability to accurately detect and identify cars in various environments is crucial. Traditional methods of vehicle detection, reliant on hardware such as inductive loops or pneumatic road tubes, are not only costly but also limited in scope and scalability. Consequently, there is a pressing need for more versatile, scalable, and cost-effective solutions.
The primary objective of this project is to develop a robust car detection system using state of the art object detection techniques. The system aims to accurately identify and track cars in real-time from video feeds, which is sourced from traffic cameras. This capability is vital for applications ranging from traffic flow management and surveillance to autonomous driving and parking space monitoring.
Implementing an efficient car detection system will significantly contribute to traffic management by improving traffic flow and reducing congestion by providing real-time data for traffic light control and route planning and safety enhancements by increasing road safety through enabling real-time accident detection, stolen vehicle recovery. The hypothesis is that the proposed model can detect and track multiple cars in diverse weather and lighting conditions with high accuracy.

Approach/ Methods

The objective of the project is to detect cars in real time from video feeds for traffic monitoring. To do so, the approach used will be break down into different stages. First step is data collection and data pre-processing then model selection, training and testing and finally model evaluation.
Dataset collected is Car-object-detection data from Kaggle. This dataset contains 1178 images including various weather condition. Then I proceed to data annotation. This step consists of labeling the data where cars appear in the images. Each car's position is annotated with a bounding box. The next step consists of data preprocessing. I resized images to the dimensions required by the YOLO model, and I normalize the pixel values.
Second step consists of model selection and training. I choose the model architecture of YOLOv8, for its balance of speed and accuracy. For training the model I used a machine with a suitable GPU for training. split the dataset into training, validation sets and testing. Then proceed to configuring the training parameters such as learning rate, number of epochs, and batch size. Next it is the validation step which consist of periodically test the model on a validation set during training to monitor its performance and adjust training parameters if necessary.
The final stage of the approach is model evaluation. I evaluated the model using metrics such as Precision, Recall, F1-score, and mean Average Precision. These metrics help in understanding both the accuracy and robustness of the model.

Experiment

Car-object detection dataset was used, and it contains 1178 of traffic surveillance cameras images. This dataset contains thousands of labeled images of cars in various environments and conditions. To increase the robustness of the model, I apply data augmentation techniques such as rotation and changes in brightness to simulate different driving conditions. Then I proceeded to split the dataset into two, 1001 for training and validation and 177 for testing. YOLOv8 was used because of its accuracy and speed. First, I pre-trained the model on car-object-detection dataset and then added two convolutional neural network layers in between the model architecture to modify model performance. I implemented the model using a deep learning framework which is PyTorch. I set the learning rate to 0.001 with batch size of 16. The stopping criteria is number of epochs which was set to 10, 20 and 50. To evaluate the model performance, I analyzed metrics such as precision, Recall, F1 Score, and mean Average Precision. And I set up Intersection over Union thresholds to determine the accuracy of the bounding boxes.

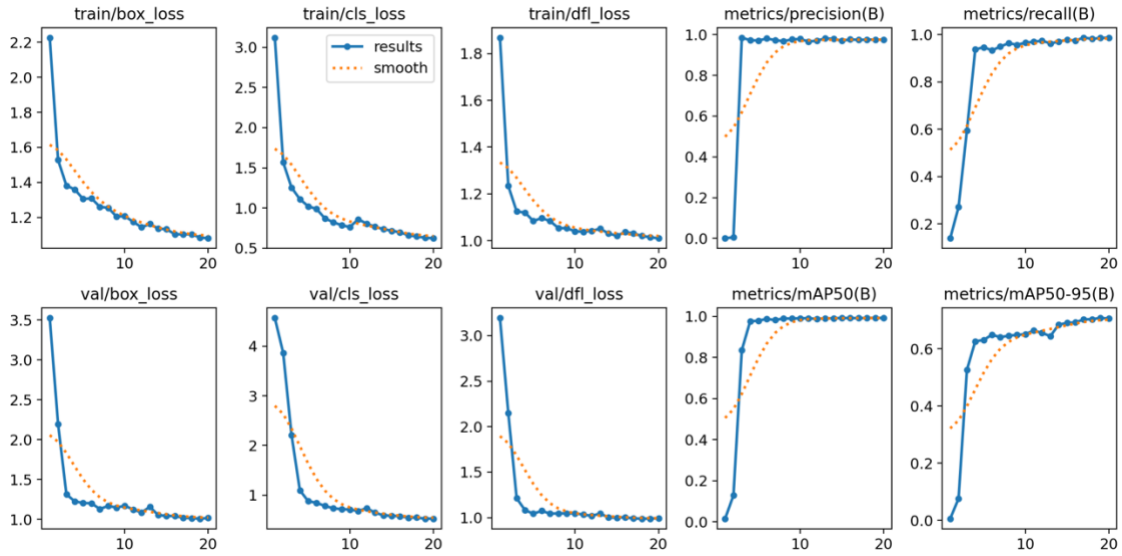
Sample of dataset images:

Training result

Training result with labels

Results

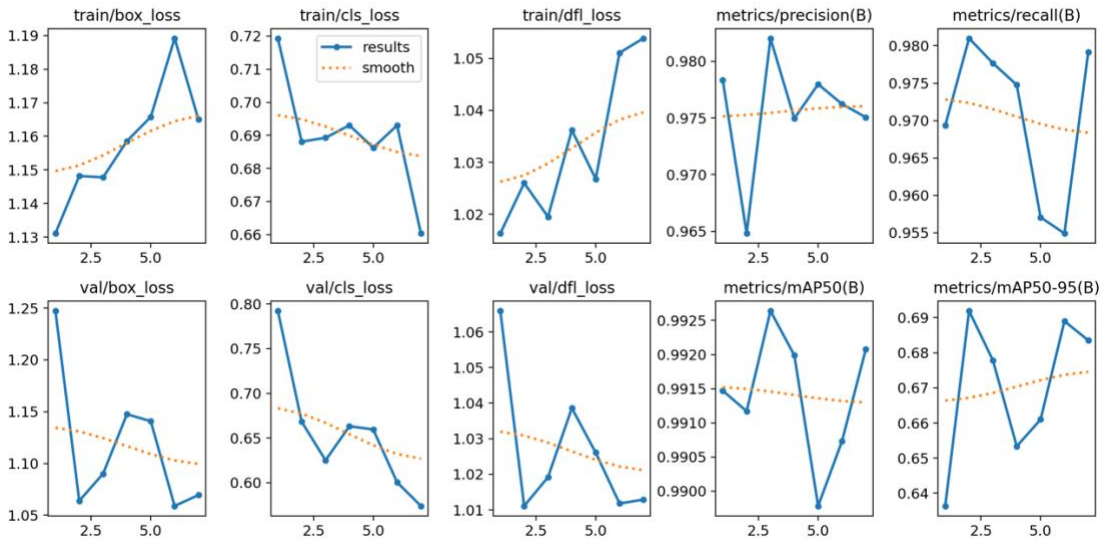
Result of model performance over 20 epochs:



On the above graph we can observe that box loss value starts at 2.2 and then decrease as the number of epochs increase. This indicates improvement of the model precision in localizing the object. Class loss measures the error in classifying the objects within the bounding boxes. As we can observe in the graph the value of class loss decreases with increasing number of epochs. Recall metric measures the ability of the model to identify all relevant instances within the dataset.
From the above graph we can conclude that the experiment was successful implemented.

Attempt for training the model over 50 epochs:
(it takes time to run 50 epochs on my computer)

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/50         0G      1.165     0.6604      1.054         10        640: 100%|██████████| 23/23 [03:27<00:00,  9.03s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 12/12 [00:31<00:00,  2.67s/it]
                   all        355        559      0.975      0.979      0.992      0.684
EarlyStopping: Training stopped early as no improvement observed in last 5 epochs. Best results observed at epoch 2, best model saved as best.pt.
To update EarlyStopping(patience=5) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.
```



Training stopped at after 7 epochs as no improvement was observed after 5 epochs.

Validation

Validation with labels

Validation with prediction

```
            0.38438,     0.38539,     0.38639,     0.38739,     0.38839,     0.38939,     0.39039,     0.39139,     0.39239,     0.39339,     0.3943
            0.40841,     0.40941,     0.41041,     0.41141,     0.41241,     0.41341,     0.41441,     0.41542,     0.41642,     0.41742,     0.41842
            0.43243,     0.43343,     0.43443,     0.43544,     0.43644,     0.43744,     0.43844,     0.43944,     0.44044,     0.44144,     0.44244
...
plot: True
results_dict: {'metrics/precision(B)': 0.975241705165211, 'metrics/recall(B)': 0.9865250312781522, 'metrics/mAP50(B)': 0.9926103662119635, 'metrics/
save_dir: PosixPath('runs/detect/train6')
speed: {'preprocess': 0.7155539284289723, 'inference': 76.79383452509491, 'loss': 4.1639301138864434e-05, 'postprocess': 0.6301443341752173}
task: 'detect'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Precision metric indicates the accuracy of the car detections, the results over 20 epochs is 97% precision.

```
path_best_weights="/Users/meme/Documents/ANN FINAL PROJECT/runs/detect/train6/weights/best.pt"
model = YOLO(path_best_weights)

metrics = model.val()
[41]  ✓ 29.7s                                                                                                    Python
...  Ultralytics YOLOv8.2.6 🚀 Python-3.12.1 torch-2.2.2 CPU (Apple M1)
Model summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GFLOPs
val: Scanning /Users/meme/Documents/ANN FINAL PROJECT/mydata/labels.cache... 355 images, 0 backgrounds, 0 corrupt: 100%|██████████| 355/355 [00:00<?
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 23/23 [00:24<00:00,  1.08s/it]
                   all        355        559      0.975      0.987      0.993      0.708
Speed: 0.5ms preprocess, 63.3ms inference, 0.0ms loss, 0.5ms postprocess per image
Results saved to runs/detect/val
```

Recall measures the model's ability to detect all relevant instances of cars in the dataset. The result is 98% recall.
F1 Score is the mean of precision and recall, providing a single score that balances both the precision and the recall the result is 92% F1 score.

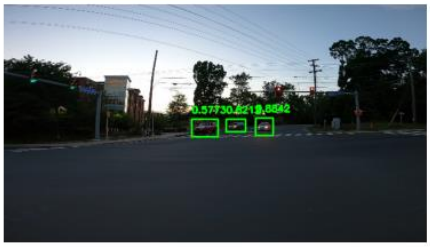
```
print(f'mean average precision @ .50: {metrics.box.map50}')
[42]  ✓ 0.0s
...  mean average precision @ .50: 0.9926103662119635
```

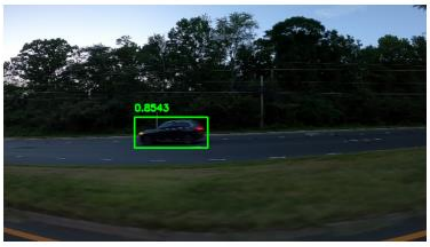
mean Average Precision is the mean of Average Precision across different classes or at different IoU thresholds. The result mAP50 is 99% and mAP50 -95 is 70% at IoU threshold of 0.5.

The object detection project demonstrated high effectiveness in detecting vehicles under varied urban conditions with good precision, recall, and real-time performance, making it suitable for real-time traffic monitoring and management applications.

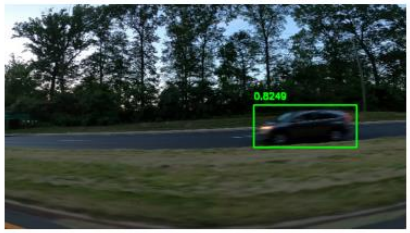
Testing

First random selection

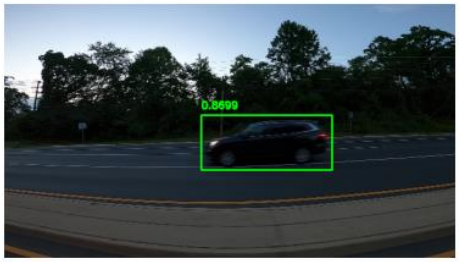




Second random selection



Third random selection



As observed on the three images above, the model was able to successfully identify all cars on the image. This prove how accurate is the proposed model.

Discussion

While comparing my study with precious work, I realized that there is a difference in the approach. Other paper first trained the model on large dataset and apply transfer learning to focus on a particular object detection. This method could be useful when we want to identify multiple objects on images. I added two convolutional neural networks layers to the model to enhance its performance. However, the pre-trained model itself was accurate. After training the model over 20 epochs, the performance was efficient. And my attempt to train the model on 50 epochs failed, there were not significant progress after 7 epochs. Perhaps, the dataset was not large enough to challenge the model or my computer could not support.

Conclusion

The objective of this car project was to develop a reliable and efficient system capable of identifying and localizing vehicles in various environments and under different conditions. By using the YOLOv8 object detection framework, I aimed to achieve real-time performance with high accuracy, ensuring the system's applicability in dynamic settings such as traffic monitoring and autonomous driving systems. I processed and analyzed thousands of images frames from the car-object-detection datasets. The model was trained on a dataset comprising 1178 images labeled with bounding boxes that accurately represent various types of vehicles under various lighting and weather conditions.

The final model achieved an impressive mean Average Precision of 99% at an IoU threshold of 0.5, demonstrating high reliability in vehicle detection. Despite the overall success, the project faced challenges in detecting vehicles at night and in heavy rain. These conditions often led to decreased accuracy due to poor visibility and reflections that confuse the detection algorithm.
The developed car detection system has significant potential applications in real-world scenarios. For instance, it can be deployed in smart city projects to enhance traffic management and safety. Additionally, it could serve as a critical component in the development of advanced driver-assistance systems, contributing to safer driving experiences.
In conclusion, this project has successfully demonstrated the feasibility and effectiveness of using advanced object detection techniques for car detection. The experience and insights gained pave the way for future innovations in the field of vehicle management systems.

Future Work

Future work consists of deployment by integrating the trained model into the target application or system and collect feedback from users. Train the model on lager set of data. Finally, compare the performance of complex YOLO and tiny YOLO.
Iteration

References

1. D. Gaifulina and I. Kotenko,"Selection of Deep Neural Network Models for IoT Anomaly Detection Experiments," 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Valladolid, Spain, 2021, pp. 260-265.

2. Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger (ed.), Advances in Neural Information Processing Systems 25 (pp.1097--1105)

3. N. Titarmare et al., "Hand Sign Language Detection - Using Deep Neural Network," 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1-4.

4. D. Arora, M. Garg and M. Gupta, "Diving deep in Deep Convolutional Neural Network," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 749-751.

5. E. Roopa and B. E. Reddy, "Predicting JNTUA CEA Student's Academic Performance Using Deep Neural Networks," 2023 Third International Conference on Advances in Electrical,Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2023, pp.1-6.

6. O. S. Amosov, S. G. Amosova, Y. S. Ivanov and S. V. Zhiganov, "The Use of Deep Neural Networks to Recognize Network Traffic Abnormalities in Enterprise Information and Telecommunication Systems," 2019 Twelfth International Conference "Management of large-scale system development" (MLSD), Moscow, Russia, 2019, pp. 1-5.