IE643: Deep Learning - Theory and Practice **July-Nov 2024**

Project Report: Vehicle Detection Model in Adverse Weather Conditions

Team Name: SelfVision *Team Members: Chappidi Preethi*

Abstract

This project report focuses on project which aims to detect vehicle in adverse weather conditions with limited dataset of adverse weather conditions. I used the existing YOLOv5m model for training and fine-tuning with mentioned datasets. I described main components of YOLOv5 architecture.

1 Introduction

This project aims to develop a Vehicle Detection Model that performs well in Adverse weather conditions using limited datasets of adverse weather conditions. This is useful for autonomous driving cars as they must navigate safely across diverse weather conditions in real world scenarios.

We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details on experiments in Section 6. A description of future work is given in Section 8. We conclude with a short summary and pointers to forthcoming work in Section 9.

2 Project Workflow

- **Literature Review**: Reviewed papers mentioned in Section 3 to gain insights into Object Detection models and Domain Adaptation Techniques
- **Data Collection and Processing**: Collected datasets from kaggle mentioned in Section 5 and performed data augmentation to solve issue of imbalanced dataset. Used RandomRain from Albumentations to simulate rainy weather datasets
- **Model Training**: Used the existing YOLOv5m model architecture from [4] to train my model on normal weather dataset
- **Model Fine-Tuning**: Fine-tuned the trained YOLOv5m model by freezing first 7,10,15,17,20 layers of yolov5m model and used combined datasets of foggy and rainy weathers
- **Model Evaluation**: Evaluated the model's performance on test datasets ,separately for normal, rainy and foggy weather datasets to analyze model performance across different conditions.
- **Deployment**: Used Google Colab to get the predictions of uploaded image by my trained model

3 Literature Survey

This project draws inspiration from a closely related work from paper 1 .In paper 1, the author describes the architecture of YOLOv5 model in detail. The paper specifies CSPDarknet as a backbone which helps in reducing number of parameters by maintaining accuracy. The paper specifies the use of PANet for feature fusion to enhance model's ability to detect objects at multiple scales. This architecture is effective for complex environment which makes the model suitable for adverse weather conditions

I took inspiration of fine-tuning from paper2. In paper 2 the author describes the challenge of limited dataset for fine-tuning and proposed to leverage data augmentation techniques to synthetically improve the dataset size and variations to improve model performance. With this inspiration i performed 10 augmentation per image also.

4 Proposed Approach or Approaches

I initially experimented with fine-tuning the model by alternating between the rainy and foggy datasets separately every 30-40 epochs ,but i observed that model's performance was not satisfactory, which i attributed to overfitting to the specific weather conditions of each datasets , to solve this issue, i fine-tuned the model suing combined datasets of both foggy and rainy weathers which improved model's adaptability and performance across varying weather conditions and all this is done with rainy and foggy weather dataset of 180 images in training but i later changed to few-shot setup by taking only 4 images in training and then used data augmentations to make them 64 images in total. With few shot weather datasets of rainy and foggy weathers i fine-tuned yolov5m model by freezing first 7,10,15,17,20 layers as starting layers focus on low-level features which are domain invariant so model can focus on learning domain variant features mainly to perform well in adverse weather conditions

4.1 Work done before prep-presentation review

I reviewed some papers to gain knowledge of Object Detection models and Domain Adaptation Techniques and collected Datasets of Normal and Foggy weathers

4.2 Work done after prep-presentation review

I trained the YOLOv5m model(didn't use other optimizer so by default it uses SGD using normal weather dataset during which i observed the model overfitted to highly domainant car class. To solve this issue i performed data augmentation after removing some images that had high number of the dominant car class from training dataset. I pre-processed both normal and foggy weather datasets to make them suitable to use them with YOLOv5m model. To get the rainy weather dataset i simulated rainy weather datasets using RandomRain augmentation method.

Later i fine-tuned the trained model using combined dataset of rainy and foggy weathers. Then i evaluated the model's performance using metrics like mAP@50, mAP@05-99, Precision, Recall, Confusion Matrix etc. I evaluated separately for normal ,foggy and rainy weather datasets.

The data augmentation was crucial to ensure model avoids overfitting to dominant class along with underfitting of underrepresented classes.

I reviewed few papers related to GANs as a part of domain adaptation technique exploration in case fine-tuning didn't give satisfactory performance

I utilized YOLOv5 neural network architecture for this project. Its main components are:

- **Backbone**: This is responsible for feature extraction from input images. This uses **CSPNet** network which uses residual blocks to divide features maps into 2 parts for better propagation

- **Neck**: This creates feature pyramids to enhance multi-scale detection capabilities. This uses **PANet** to combine features from different layers which allows model to leverage both high-level and low-level features for better detection performance

- **Head**: This has multiple convolution layers that generate predictions for bounding boxes and class probabilities from aggregated feature maps provided by Neck component allowing model to localize and classify detected objects

5 Data set Details

I used Image datasets along with their polygon labels which are like outlines of objects. My datasets are from Cityscapes LeftImg8bit dataset and from Cityscapes gtFine labels dataset and foggy image dataset from Foggy leftImg8bit dataset.

Table 1: Final Dataset

Class	Bicycle	Motorcycle	Car	Truck	Bus	Total Number of Images
Normal Train	4717	1161	5471	360	543	2710
Normal Val	1175	149	4667	93	98	489
Normal Test	646	126	4399	85	70	490
Foggy Train	32	16	67	16	16	64
Foggy Val	37	6	63	6	8	12
Foggy Test	78	44	416	19	20	52
Rainy Train	67	25	45	30	29	63
Rainy Val	12	7	37	7	10	18
Rainy Test	34	14	88	2	9	11

In Table 1, The numbers in columns of class names are number of instances of corresponding class and Total Number of Images is total number of images in corresponding dataset

In the corresponding labels set they ar in the form of outline co-ordinates so changed them to YOLOv5 Format by taking maximum and minimum of the x and y co-ordinates of the object annotation to form a box and then converting it YOLOv5 format which is 'class-id x-center y-center width height' in normalized form.

Initially the Normal weather train dataset was 27k cars, 3.7k cycles, 1.1k motorcycles and 700-800 trucks and buses. To avoid highly dominant car instances i removed the images with number of car classes greater than 7 and got a Dataset with around 5k cars, 1.7k cycles, 200 buses, 200 trucks, 350 motorcycles so i used data augmentations like horizontal flip, rotation, hues etc. and got the final Normal weather train dataset

For few-shot setup i selected 4 images from normal weather training dataset and used RandomRain to get training dataset of rainy weather and also used data augmentations and made them 64. Similarly i filtered 4 from foggy weather dataset and used data augmentations to make them 64

In the dataset i collected there are no labels for test dataset so for test dataset in each weather i made them using the images i removed(cars greater than 7) as these are not seen in train and validation. I tried few conditions on number of instances per class and got them.

6 Experiments

I performed data augmentations using ImageDataGenerator but when i tried to change the labels according to augmentation peroformed on image it didn't change labels accordingly so explored and found Albumentations for data augmentations

I have performed data augmentations on datasets as mentioned in section 5, and trained YOLOv5m model using normal weather dataset(60 epochs with batch size as 8 ,using TPU v2-8, with 334.6GB RAM and 225.3GB disk in g-colab) using class weights in loss function of train.py of YOLOv5m and then fine-tuned the trained model using combined datasets of foggy and rainy weathers (100 epochs with batch size 8). I didn't specify optimizer so by default YOLOv5m model uses SGD.

I tried to fine-tune the trained model by alternating between the rainy and foggy datasets separately every 30-40 epochs but it didn't give satisfactory performance so changed to combined datasets.(not few-shot for 180 images in rainy, foggy weather datasets)

I fine-tuned yolov5m model using combined rainy and foggy weathers datasets by freezing first 7,10,15,17,20 layers to make model focus on learning domain variant features instead of domain invariant features already learnt in first few layers.

I evaluated model's performance as mentioned in section 4.2

7 Results

Foggy Test Performance metrics :

Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**
all	52	576	0.58	0.484	0.553	0.366
bicycle	52	78	0.558	0.308	0.453	0.251
motorcycle	52	44	0.704	0.432	0.601	0.345
car	52	415	0.675	0.665	0.738	0.504
truck	52	19	0.25	0.263	0.247	0.191
bus	52	20	0.714	0.75	0.728	0.541

Table 2: Model Performance Results for a Smaller Dataset

Rainy Test Performance Metrics are:

Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**
all	11	147	0.447	0.219	0.316	0.17
bicycle	11	34	0.5	0.0882	0.29	0.0909
motorcycle	11	14	0	0	0	0
car	11	88	0.735	0.284	0.535	0.335
truck	11	2	0.5	0.5	0.375	0.15
bus	11	9	0.5	0.222	0.381	0.274

Table 3: Model Performance Results for a Small Dataset

Normal Weather Test Performance Metrics are:

Class	Images	Instances	P	R	mAP50	mAP50-95
all	490	5311	0.619	0.45	0.539	0.357
bicycle	490	634	0.698	0.333	0.501	0.284
motorcycle	490	126	0.69	0.389	0.555	0.312
car	490	4396	0.832	0.628	0.769	0.552
truck	490	85	0.325	0.376	0.312	0.215
bus	490	70	0.55	0.525	0.555	0.419

Table 4: Model Performance Results by Class

Weather wise performance metrics are in Normal Weather Performance metrics, Foggy Weather Performance metrics, Rainy Wether Performance metrics

From the metrics class wise to improve model's performance truck wise, should use data augmentation for trucks specifically. And low recall indicates model is not identifying all instances of classes effectively which is seen in confusion matrix also. mAP@50-99 is expected to be low as it shows model's ability to match predictions with ground truth labels at multiple IoU thresholds and must exceed 50 percent IoU with ground truth for a match. Moderate Precision shows that model is identifying good number of true positives among its predictions

8 Plan for Novelty Assessment

I plan to simulate better foggy for test dataset and i want to try GAN for domain adaption model if time permits and i plan to further use 40 epochs for training using normal weather dataset. I plan to work on improving models performance.

In future i want to try changing any other weather images to normal weather images and give this normal weather image as input and for yolov5m model trained on normal weather dataset.

9 Conclusion

This project is aimed to develop a Vehicle Detection Model in Adverse Weather conditions(foggy and rainy), using limited dataset. I trained YOLOv5m model using Normal weather dataset and fine-tuned the trained model using combined datasets of foggy and rainy weather datasets. Evaluated Model's performance separately for normal, foggy and rainy weathers to analyze weather wise. There is room for improvement in the model's performance by increasing dataset size of particular classes which are underrepresented. The model's performance for car is good and needs improvement for other classes.

References