

Vehicle Make and Model Recognition using Machine Learning

By

Hassan Rasheed	126900
Muhammad Ahsan Fahim	129416
Sikander Azam	133767



A project report submitted in partial fulfillment
of the requirement for the degree of
Bachelor of Science in Computer Science

Department of Computing

School of Electrical Engineering & Computer Science

National University of Sciences & Technology

Islamabad, Pakistan 2019

CERTIFICATE

It is certified that the contents and form of thesis entitled “**Vehicle Make and Model Recognition using Machine Learning**” submitted by Hassan Rasheed (NUST-2015-BSCS-126900), Muhammad Ahsan Fahim (NUST-2015-BSCS-129416), Sikander Azam (NUST-2015-BSCS-133767) have been found satisfactory for the requirement of the degree.

Advisor: _____

(Dr. Syed Ali Hassan)

Co-Advisor: _____

(Dr. Omar Arif)

DEDICATION

To Allah Almighty

&

To our Parents and Faculty

ACKNOWLEDGEMENTS

We are deeply thankful to our advisor and co-advisor, Dr Syed Ali Hassan, and Dr. Omar Arif for helping us throughout the course in accomplishing our final project. Their guidance, support and motivation enabled us to achieve the objectives of the project.

We pay lots of gratitude to our parents whose support, guidance, encouragement and prayers have been the backbone of our project.

And above all, we are extremely grateful to Almighty Allah, Who has carried us all the way to the finish line and has always been very kind on us in every field of life.

Table of Contents

ACKNOWLEDGEMENTS.....	4
Table of Figures.....	7
ABSTRACT.....	8
Introduction	9
What are Smart Cities?	9
What is Intelligent Transportation System?.....	9
Safe City Project in Islamabad (ITP and NADRA)	9
Crime reports	10
Road Accidents	11
Problem statement.....	11
Our Solution.....	11
Industrial Partners	11
Predictive policing	12
Working of Predictive Policing	12
Applications	14
Scope of the project	14
Literature Review.....	15
Previous work	15
Previous work in Pakistan.....	15
Algorithms	15
ResNet.....	15
Faster R-CNN	16
YOLO v3	16
Why we chose YOLO?	17
Functionality and Design.....	18
Google Instance Details	18
Data Gathering	20
What is Web scraping?	20
IntelliScraper.....	20
Dataset Generation.....	22
Data Augmentation	22
Dataset Annotation.....	23
Cars Models	26

Implementation and Result	27
Network Architecture Configuration	27
Hyperparameters Tuning	28
Results.....	28
Conclusion and Future Recommendations	32
Work division	33
References	34

Table of Figures

Figure 1 Road accidents figures.....	11
Figure 2 YOLOv3 on mAP-50 scale compared to other algorithms	16
Figure 3 YOLOv3 FPS comparison.....	17
Figure 4 Darknet-53 Architecture.....	19
Figure 5 Websites that we scraped images from.....	21
Figure 6 Dataset folders	22
Figure 7 Sample text file of an image Suzuki Wagon R.....	24
Figure 8 Code for annotation of data	24
Figure 9 General detection of the car before training	25
Figure 10 Algorithm printing out its predictions on the terminal	25
Figure 11 Anchor sizes for the custom dataset	27
Figure 12 Hyperparameters.....	28
Figure 13 Initial Training Results	29
Figure 14 Mid training results.....	29
Figure 15 Average Loss and mAP graph plotted against number of iterations	30
Figure 16 Final training results	31

ABSTRACT

A Vehicle Make and Model Recognition (VMMR) system can provide great value in terms of vehicle monitoring and identification based on vehicle appearance in addition to the vehicles' attached license plate typical recognition. A real-time VMMR system is an important component of many applications such as automatic vehicle surveillance, traffic management, driver assistance systems, traffic behavior analysis, and traffic monitoring, etc.

Vehicle Make and Model Recognition (VMMR) plays an important role in Intelligent transportation systems. Due to an increase in volume and diversity of vehicles on the road, traffic monitoring has become both important and difficult. Our solution is one of the modules of a much larger predictive policing project started by Center of Information Technology, Pakistan. We are developing an intelligent vehicle detection system for the Safe-City project which uses Deep learning to recognize the make and model of a car in real time.

The proposed VMMR system recognizes vehicles on the basis of make, model, and generation (manufacturing years) while the existing VMMR systems can only identify the make and model. Comparison with existing VMMR research demonstrates superior performance of the proposed system in terms of recognition accuracy and processing speed.

Introduction

What are Smart Cities?

A smart city is a designation given to a city that incorporates information and communication technologies (ICT) to enhance the quality and performance of urban services such as energy, transportation and utilities in order to reduce resource consumption, wastage and overall costs. The overarching aim of a smart city is to enhance the quality of living for its citizens through smart technology.

Smart Cities use Data and Technology to:

- Create efficiencies
- Provide better security
- Create economic development
- Enhance quality of life

What is Intelligent Transportation System?

An intelligent transportation system (ITS) is an advanced application which, without embodying intelligence as such, aims to provide innovative services relating to different modes of transport and traffic management and enable users to be better informed and make safer, more coordinated, and 'smarter' use of transport network. (European Union, 2010)

Safe City Project in Islamabad (ITP and NADRA)

The safe city model comprises of a host of features which unite to form a formidable technology driven approach in securing a specific region. The implementation of this project facilitated by the advent of Smart ID cards, Electronic RFID tags, Smart Vehicle Registration Cards and Centralized databases (primarily NADRA Citizen Database and Centralized Vehicle Database) for authentication of vehicles and individuals. The whole network of security structure is additionally supported by a myriad of Security cameras and CCTVs.

NADRA has already successfully implemented this project in Islamabad, the capital of Pakistan. Under the security plan as many as almost 2000 Security and

CCTV cameras and Intelligent Video Surveillance are installed in the city. NADRA is also planning to replicate this project in Karachi, which is the economic hub of Pakistan and the fifth largest city of the world in terms of population. (NADRA, 2019)

This is further complemented by installation of CCTV/IP cameras and RFID tag readers which are required to be placed at strategic locations across the city to create an effective monitoring and control system. The model is an amalgamation of technology and trained personnel who act together in fortifying the territorial space. Controlled through central command center, the population and their activities are monitored along with a vehicle tracking system that forms a preverbal umbrella hovering over the allocated access points and key locations which may include check posts, public access areas, sensitive installations and restricted areas. (Islamabad Police, 2018)

Safe City Project was envisioned to introduce Technology led policing to make the security of Capital City as a model of excellence. Under the project, Intelligence Video Surveillance and Vehicle Management System Cameras are installed in Islamabad with Automatic Number Plate Recognition facility which help with:

1. Aid in criminal investigation
2. Traffic Management
3. Implementation of a New patrolling plan
4. Tracing down criminal cases
5. Field surveillance cars and establishment of a situation room

Crime reports

Despite having a dedicated system to prevent vehicular and other crimes and installing thousands of CCTV cameras in the country there are still countless crimes being committed which are being quoted from the 'Crime & Safety report 2018' of Pakistan. (OSAC, 2018) The numbers show:

- 21,000 cars are stolen yearly which are worth almost PKR 5 Billion
- Around 395 people were killed during various criminal and terrorist activities in 2017
- Further 942 people were severely injured in 2017

Road Accidents

Furthermore, there have been over 3500 casualties due to road accidents as shown by the following table: (Islamabad Police, 2017)

Year	Total number of accidents	Accident		Persons		Total number of vehicles involved
		Fatal	Non-Fatal	Killed	Injured	
PAKISTAN						
2006-07	10466	4535	5931	5465	12875	11481
2007-08	10466	4610	5856	5615	12096	11456
2008-09	9496	4145	5351	4907	11037	10322
2009-10	9747	4378	5369	5280	11173	10496
2010-11	9723	4280	5443	5271	11383	10822
2011-12	9140	3966	5174	4758	10145	9986
2012-13 *	8988	3884	5104	4719	9710	9876
2013-14*	8359	3500	4859	4348	9777	9423
2014-15*	7865	3214	4651	3954	9661	8949
2015-16*	9100	3591	5509	4448	11544	10636

Figure 1 Road accidents figures

Problem statement

Countless hours of video feed generated from Safe city CCTV cameras is manually monitored by humans and these conventional surveillance methodologies result in an overall inefficient policing system.

Our Solution

A Machine learning based solution which uses Tuned-Darknet Architecture to accurately detect the Make and Model of a vehicle in real-time.

Industrial Partners

Our industrial partners include “CIT” and “E-Pakistan”. Information & Communication Technologies (ICTs) are fast changing the World around us. Pakistan Need to catch up with the upcoming changes in the region & globally. We in Pakistan have Challenges and Opportunities. As a Global best practice ICTs for

E-Education, E-Health, E-Agriculture, E-Governance is immediate solution to our Challenges .After going through the Process of E-Village, E-District now on 14th August 2015 we are announcing ‘ E-Pakistan Vision 2025 ‘which aims to all Public & private sector stakeholders of development on one page to create a National and International Impact. (E-Pakistan, 2019)

Vehicle make and model detection is one of the modules of a Mega-project started by E-Pakistan and CIT called Predictive Policing system.

Predictive policing

Predictive policing refers to the usage of mathematical, predictive analytics, and other analytical techniques in law enforcement to identify potential criminal activity. Predictive policing methods fall into four general categories: methods for predicting crimes, methods for predicting offenders, methods for predicting perpetrators' identities, and methods for predicting victims of crime.

Predictive policing uses data on the times, locations and nature of past crimes, to provide insight to police strategists concerning where, and at what times, police patrols should patrol, or maintain a presence, in order to make the best use of resources or to have the greatest chance of deterring or preventing future crimes.

Police may also use data accumulated on shootings and the sounds of gunfire to identify locations of shootings. The city of Chicago uses data blended from population mapping crime statistics, and whether to improve monitoring and identify patterns. (National Institute of Justice, 2014)

Working of Predictive Policing

Predictive policing tries to harness the power of information, geospatial technologies and evidence-based intervention models to reduce crime and improve public safety. This two-pronged approach, applying advanced analytics to various data sets, in conjunction with intervention models can move law enforcement from

reacting to crimes into the realm of predicting what and where something is likely to happen and deploying resources accordingly.

The predictive policing approach does not replace traditional policing. Instead, it enhances existing approaches such as problem-oriented policing, community policing, intelligence-led policing and hot spot policing.

Predictive policing leverages computer models such as those used in the business industry to anticipate how market conditions or industry trends will evolve over time for law enforcement purposes, namely anticipating likely crime events and informing actions to prevent crime. Predictions can focus on variables such as places, people, groups or incidents. Demographic trends, parolee populations and economic conditions may all affect crime rates in particular areas. Using models supported by prior crime and environmental data to inform different kinds of interventions can help police reduce the number of crime incidents. (Police One, 2019)

Vehicle Make and Model recognition (VMMR) can become a crucial part of the Smart Cities concept. A real-time VMMR system is a significant component of security applications in Intelligent Transportation Systems (ITS) and can help prevent and predict crimes and traffic violations.

Applications

1. Surveillance and investigation

An efficient VMMR system can help reduce car thefts. There are thousands of cars being stolen each year and VMMR system can be key to catching car thieves. It can further be used for the investigation of crimes in surveillance footages.

2. Law Enforcement agencies

The VMMR can be used by Law enforcement agencies including Police, Traffic police, ISI etc for security threat monitoring. A searchable database of processed data can aid in fighting crime and data analytics.

3. Traffic regulations

Our solution can be used in collaboration with the traffic police to help prevent traffic violations in the city. This in turn will help reduce traffic accidents on the road and save lives by preventing fatal and nonfatal accidents occurring on the roads due to traffic violations.

4. Distinguishing between LTV and HTV

Our system can be used to accurately detect between Light and Heavy transport vehicles and this can be used to restrict entrance of specific vehicles in high security areas. This can also be used for speed measurement for vehicles with different speed limits on highways and motorways.

Scope of the project

The scope of the project is to develop a VMMR system which is one of the modules of a Mega project called Predictive policing. Our project would be integrated into a much bigger system which consists of Number plate recognition, Human behavior detection and various other systems and with all of these combined the Mega system would help reduce crimes and road accidents and make smarter and safer cities.

Literature Review

Previous work

Various companies have come up with a Make and Model detection system including companies like EyeDea, Orpixon Computer vision but the recognition speed is not suitable for real-time application. These analysis are independent of number plate recognition systems but are not as accurate or fast as our solution.

Previous work in Pakistan

There has currently been no work in the field of Vehicle make and Model system in Pakistan so far. Under the Safe City project, Intelligence Video Surveillance and Vehicle Management System Cameras are installed equipped only with Automatic Number Plate Recognition facility. These cameras are also imported from other countries like China and cost millions of dollars. Every detection system is hardware dependent and rely a lot on the image or video quality that is being fed into it.

Algorithms

We tried a number of different object detection algorithms to use including ResNet and Faster R-CNN. The mentioned algorithms gave good accuracy but are not suitable for our system as they are not fast enough for real-time application.

ResNet

ResNet is a short name for Residual Network. As the name of the network indicates, the new terminology that this network introduces is residual learning. Deep convolutional neural networks have led to a series of breakthroughs for image classification. Many other visual recognition tasks have also greatly benefited from very deep models. So, over the years there is a trend to go more deeper, to solve more complex tasks and to also increase /improve the classification/recognition accuracy. But, as we go deeper; the training of neural network becomes difficult and also the accuracy starts saturating and then degrades also. Residual Learning tries to solve both these problems.

Faster R-CNN

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection.

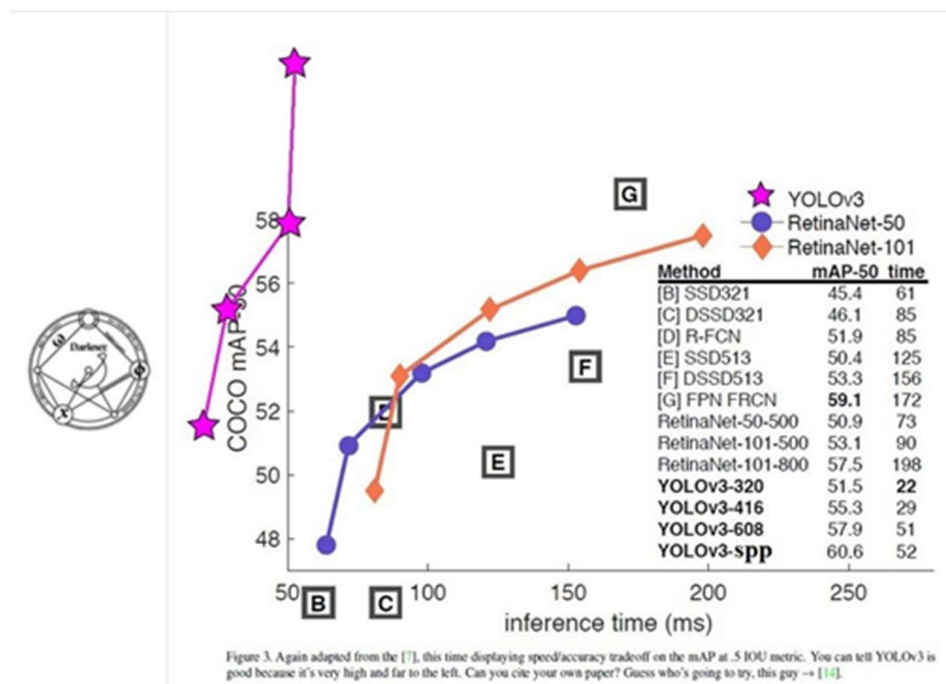


Figure 2 YOLOv3 on mAP-50 scale compared to other algorithms

YOLO v3

YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily trade-off between speed and accuracy simply by changing the size of the model, no retraining required. Prior detection systems repurpose classifiers or localizers to

perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

YOLOv3 uses a few tricks to improve training and increase performance, including: multi-scale predictions, a better backbone classifier, and more. (Joseph Redmon, 2018)

Why we chose YOLO?

YOLO v3 is the fastest real time detection algorithm while being on par in terms of precision. It provides great accuracy even at high FPS.

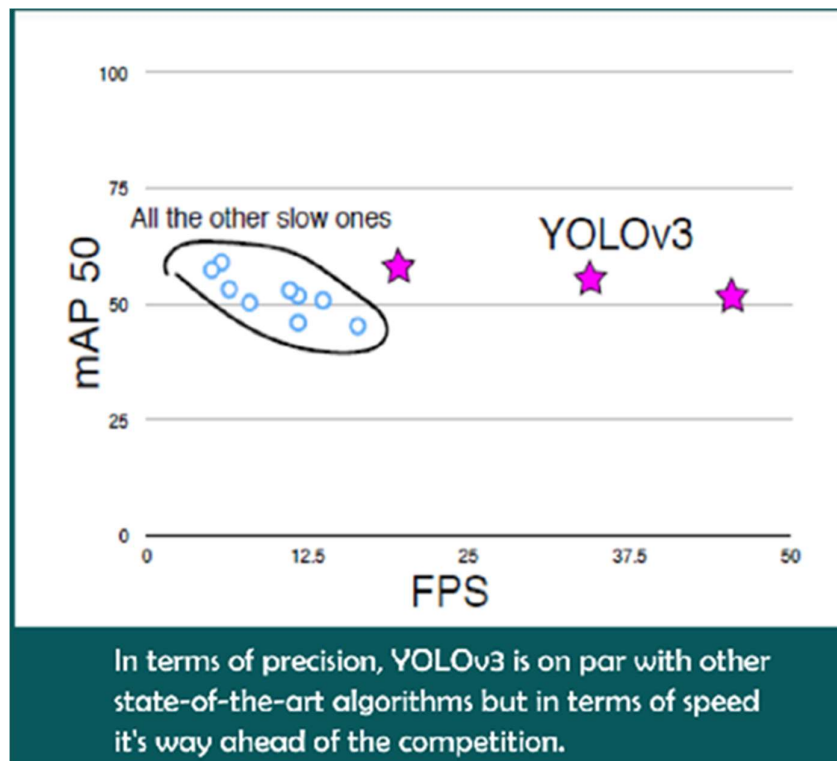


Figure 3 YOLOv3 FPS comparison

Functionality and Design

We used Google Cloud Platform (GCP) because of the compute power needed was not available in our computers.

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.

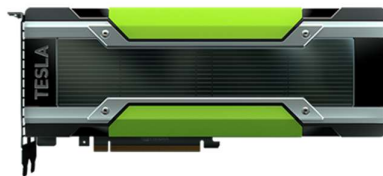
Google Cloud Platform provides Infrastructure as a service, Platform as a service, and Serverless computing environments. Google Cloud Platform is a part of **Google Cloud**, which includes the Google Cloud Platform public cloud infrastructure, as well as **G Suite**, enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

Google Instance Details

Since we did not have access to a real GPU, we did our complete project on Google cloud platform using a virtual system with the following specifications:

Virtual Memory: 15 GB

GPU: Tesla K80 with 12GB Video memory



The algorithm we chose as indicated earlier is the YOLOv3 algorithm due to its accuracy and speed. YOLOv3 is based upon the Darknet-53 architecture which

is written in C and CUDA because of which it is fast and computable on both CPU and GPU. We used the GPU version by compiling Darknet using CUDA, because GPU provides more compute power hence faster

YOLO v3 is a general object detection algorithm which can detect over a 1000 different classes of objects (Shown in Figure 4).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4 Darknet-53 Architecture

We modified the base Darknet architecture to our needs according to our requirements and trained it on our Pakistani cars dataset. Trained the modified network on 15 models of cars.

For each model we collected a dataset of around 1000 images and trained it on our modified network.

The network that we used was pretrained on COCO dataset which contained 80 general classes. So the network was pretty well versed in detecting general objects such as people, bicycles, dogs, cars etc.

We then used Transfer Learning to use the pretrained weights to help learn new weights specifically for our dataset of cars. For the transfer learning part we

changed the final YOLO Layer and the Convolutional Layer Parameters which will be later explained in the report.

Data Gathering

The product that we made is to be used as one of the module for a mega project as mentioned before. Our product right now is a prototype for that actual system. Our industrial partner is in talks with Safe City project and Islamabad Police to get the actual dataset from the CCTV cameras that are installed in Islamabad. Till then, we created our own Dataset to train our model on and prove our application and accuracy of the Make and Model detection system.

For this purpose we used an intelligent web scraper which uses Object detection to filter out incorrect search results from the images to be scraped.

What is Web scraping?

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis. (G Boeing, 2016)

IntelliScraper

To collect data, first of all we designed an intelligent web image scraper which not only downloads pictures in bulk from any website that you give its link to, but also takes a “tag” as an argument and before downloading the images, passes them from YOLO detection algorithm to verify if the picture being download is indeed the thing that we need. Basically it is an intelligent image scraper for the web that uses a simple python web scraper to generate images which are then refined

based on output from a YOLO v3 Deep Learning model that extracts only the required images of objects and stores them in a separate sub-folder.

We used our IntelliScraper to download over 10,000 images of Pakistani cars from different website including PakWheels and Olx.



Figure 5 Websites that we scraped images from

Dataset Generation

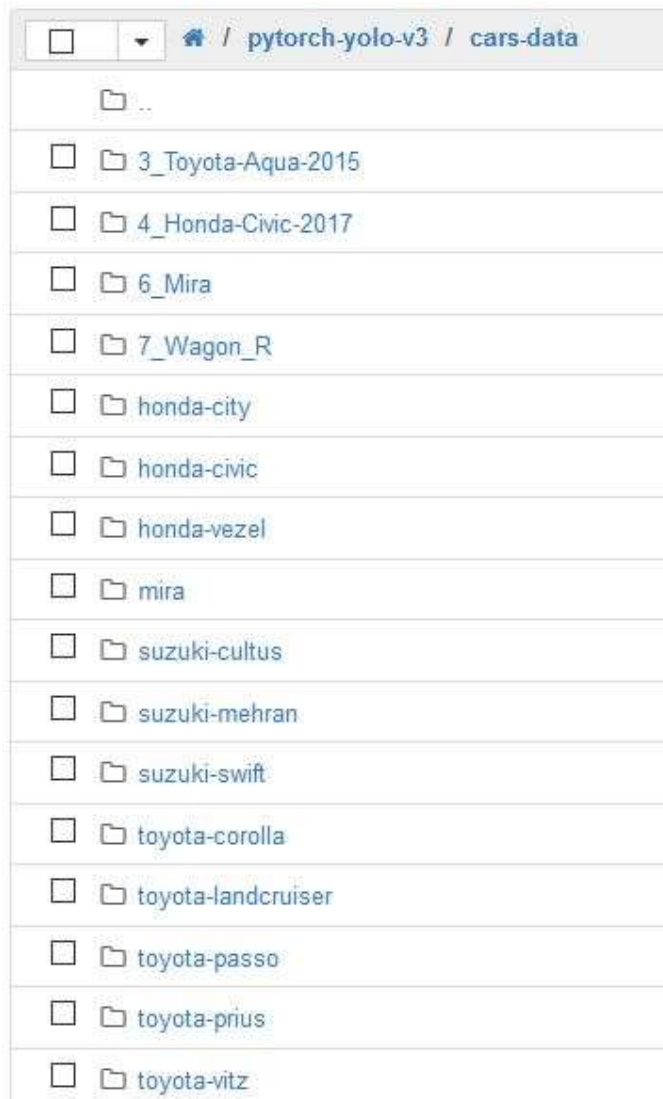


Figure 6 Dataset folders

Data Augmentation

Data augmentation adds value to base data by adding information derived from internal and external sources within an enterprise. Data is one of the core assets for an enterprise, making data management essential. Data augmentation can be applied to any form of data, but may be especially useful for customer data, sales patterns, product sales, where additional information can help provide more in-depth insight. (Techopedia, 2019)

So in order to increase the number of images we had for each car, we used Data Augmentation techniques including:

- Grey Scaling
- Rotate by 5-10 degrees
- Increase/decrease brightness
- Increase/decrease contrast
- Flipping images horizontally



Dataset Annotation

When training a computer vision or pattern recognition solution, humans are needed to identify and annotate specific data, such as outlining all the pixels containing trees or traffic signs in an image. Using this structured data, machines can learn to recognize these relationships in testing and production.

The Darknet required all the images to be labeled in the following format: <object-class> <x_center> <y_center> <width> <height>. For this purpose normally as explained earlier humans have to manually draw bounding boxes over each dataset image. To save time and avoid this tedious process, we used the general object detecting YOLOv3 algorithm to detect cars in our dataset images and print

their bounding boxes to a separate text file as required by the algorithm to prepare the dataset for training. (Murugavel, 2018)

We automated the annotation process by using the YOLOv3 trained on COCO dataset to print out the bounding box coordinates of the cars detected in images to the text files of training images in the dataset which was in the following form:

<object-class> <x_center> <y_center> <width> <height>

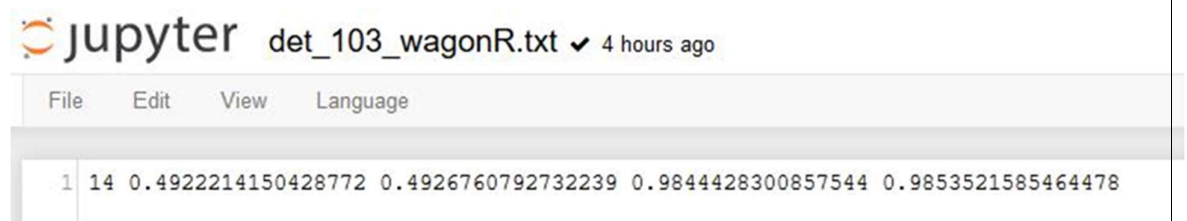


Figure 7 Sample text file of an image Suzuki Wagon R

The code loops through all the image files in the batch and for each image file draws bounding boxes of the objects (cars) detected in the images to a separate text file as required by YOLO for training of the network.

```
#List of image text file names
txt_names = writetext(imlist)
imgid=0
#for each text file name
for name in txt_names:
    file= open(name,"w")
    for i in range(output.shape[0]):
        #check if the id of the image matches the id of the output
        if(imgid==int(output.data.cpu().numpy()[i,0])):
            print("id matched")
            #if id is matched the data values are written into the corresponding image data text file
            norxs=(output.data.cpu().numpy()[i, [1]]/im_dim_list.data.cpu().numpy()[i,0])
            norxe=(output.data.cpu().numpy()[i, [3]]/im_dim_list.data.cpu().numpy()[i,0])
            norys=(output.data.cpu().numpy()[i, [2]]/im_dim_list.data.cpu().numpy()[i,1])
            norye=(output.data.cpu().numpy()[i, [4]]/im_dim_list.data.cpu().numpy()[i,1])
            norwidth=norxe-norxs
            norheight=norxe-norxs
            norx=((norxs+norxe)/2)
            nory=((norys+norye)/2)
            file.write("12 " +str(norx.item())+" "+str(nory.item())+" "+str(norwidth.item())+" "+str(norheight.item())+"\n")
        else:
            print("could not match id")
    imgid+=1
    file.close()
```

Figure 8 Code for annotation of data

Before training the algorithm only detects that there is a car in the image. Not specifically defining its make and model.



Figure 9 General detection of the car before training

```
-----
toyota-land-cruiser-zx-2-2012-23078998.jpg predicted in 0.166 seconds
Objects Detected: car car car truck
-----
toyota-prado-2013-22542050.jpg predicted in 0.173 seconds
Objects Detected: car car car car car
-----
toyota-land-cruiser-zx-2-2015-22546148.jpg predicted in 0.163 seconds
Objects Detected: car
-----
toyota-land-cruiser-zx-2-2015-22546146.jpg predicted in 0.172 seconds
Objects Detected: car
-----
toyota-prado-rx-2-7-3dr-2004-22076267.jpg predicted in 0.183 seconds
Objects Detected: car
-----
toyota-land-cruiser-cygnus-2003-23064255.jpg predicted in 0.156 seconds
Objects Detected: car car
-----
toyota-land-cruiser-ax-g-selection-2008-22971531.jpg predicted in 0.147 seconds
Objects Detected: car
-----
toyota-land-cruiser-ax-2-2013-19586936.jpg predicted in 0.161 seconds
Objects Detected: person car
-----
toyota-land-cruiser-vx-4-2d-2001-22929470.jpg predicted in 0.160 seconds
Objects Detected: car
-----
toyota-land-cruiser-ax-2-2013-19586918.jpg predicted in 0.164 seconds
Objects Detected: truck
-----
toyota-land-cruiser-vx-4-2d-2001-22929468.jpg predicted in 0.183 seconds
Objects Detected: car car truck
-----
```

Figure 10 Algorithm printing out its predictions on the terminal

As shown in the screenshot above, our YOLO system detects all the objects in the image and we automated the system to only take boxes which identified cars and no other objects which we did not require.

SUMMARY

```
-----  
Task                        : Time Taken (in seconds)  
Reading addresses          : 0.002  
Loading batch              : 1.506  
Detection (100 images)     : 35.794  
Output Processing          : 0.000  
Drawing Boxes              : 1.735  
Average time_per_img       : 0.390  
-----
```

Cars Models

We trained our model on 15 different kinds of car models that are common in Pakistan. These 15 models make up 80-90% of all the cars in Pakistan. The models include:

Suzuki Cultus	Suzuki Mehran	Suzuki Wagon R
Toyota Corolla	Toyota Vitz	Toyota Prius
Suzuki Swift	Land Cruiser	Toyota Passo
Honda City	Honda Civic	Honda Civic 2017
Toyota Aqua	Honda Vezel	Daihatsu Mira

Implementation and Result

Network Architecture Configuration

We used 53-layered Darknet architecture as base and then configured it to suit our needs. The original YOLO contains 80 classes in the YOLO layer which is the final prediction layer. We reduced these classes to 15 which is the number of classes we will need our network to print results for. (Joseph Redmon, 2018)

Another thing that we changed is the anchor sizes. YOLO needs to know what sizes of objects are we intending to detect, for that we need to calculate anchors for our dataset. The anchors that were originally used were too small for our dataset of cars which contained only one car per image. So we recalculated the anchor boxes to suit our data.

Anchors are calculated by comparing the annotated dataset with the image sizes and then multiplying it by the bounding box size of the object detected in the image which gives the size of the bounding box that is created for each image. Then by averaging out all the anchors in the dataset, new anchors are calculated which allow quicker and more accurate detection of objects. Following are the recalculated anchors for our dataset.

anchors = 127,142, 305,338, 490,298, 348,488, 451,428, 576,385, 474,539, 555,479, 564,566

Figure 11 Anchor sizes for the custom dataset

Another change that was made was changing the number of filters in the last Convolutional layer before the final YOLO layer. Filters are the number of kernels applied to the weights of the previous layer. The number of filters was recalculated to 60. The calculation was done according to the following formula. (Murugavel, How to train YOLOv3 to detect custom objects, 2018)

$$(\text{classes} * \text{predictions}) * \text{channels}$$

For 15 classes, 5 predictions are needed each and the images contain 3 channels. So,

$$[15 \text{ classes} + 5 (\text{class, x center, y center, width, height})] * 3 (\text{number of channels}) \\ = 60$$

Hyperparameters Tuning

Hyperparameters are the parameters that do not directly affect the data but they improve the quality of training and results nonetheless. The two hyperparameters that we adjusted were the learning rate and the resolution of the network.

The learning rate was set to 0.001 with 0.1 decrease at 80% and 90% iterations each.

The original YOLOv3 algorithm has been trained on a resolution of 416x416 but we increased our network's resolution to 608x608 for better accuracy. This improved our results significantly.

```
8 width=608
9 height=608
10 channels=3
11 momentum=0.9
12 decay=0.0005
13 angle=0
14 saturation = 1.5
15 exposure = 1.5
16 hue=.1
17
18 learning rate=0.001
```

Figure 12 Hyperparameters

Results

Initially the results were not very convincing as mean average precision (mAP) was only 19.3% with average precision for Honda Civic 2017 being the highest at 54.49%.

```

detections_count = 24595, unique_truth_count = 1146
class_id = 0, name = suzuki cultus, ap = 8.47% (TP = 0, FP = 0)
class_id = 1, name = honda civic, ap = 7.96% (TP = 0, FP = 0)
class_id = 2, name = toyota vitz, ap = 19.13% (TP = 0, FP = 0)
class_id = 3, name = toyota corolla, ap = 2.84% (TP = 0, FP = 0)
class_id = 4, name = suzuki mehran, ap = 20.61% (TP = 0, FP = 0)
class_id = 5, name = toyota landcruiser, ap = 14.29% (TP = 0, FP = 0)
class_id = 6, name = suzuki swift, ap = 6.05% (TP = 0, FP = 0)
class_id = 7, name = honda city, ap = 5.49% (TP = 0, FP = 0)
class_id = 8, name = toyota prius, ap = 3.79% (TP = 0, FP = 0)
class_id = 9, name = honda vezel, ap = 22.42% (TP = 3, FP = 1)
class_id = 10, name = toyota passo, ap = 18.50% (TP = 0, FP = 0)
class_id = 11, name = honda civic 2017, ap = 54.49% (TP = 40, FP = 14)
class_id = 12, name = toyota aqua, ap = 54.18% (TP = 0, FP = 0)
class_id = 13, name = daihatsu mira, ap = 19.98% (TP = 0, FP = 0)
class_id = 14, name = suzuki wagon r, ap = 31.52% (TP = 8, FP = 2)

for thresh = 0.25, precision = 0.75, recall = 0.04, F1-score = 0.08
for thresh = 0.25, TP = 51, FP = 17, FN = 1095, average IoU = 51.82 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.193142, or 19.31 %
Total Detection Time: 139.000000 Seconds

```

Figure 13 Initial Training Results

After a few thousand steps the mAP improved significantly with mAP at 95.57% and ap for Honda Civic 2017 at 100%.

```

class_id = 0, name = suzuki cultus, ap = 99.66% (TP = 54, FP = 6)
class_id = 1, name = honda civic, ap = 97.78% (TP = 49, FP = 11)
class_id = 2, name = toyota vitz, ap = 98.50% (TP = 97, FP = 36)
class_id = 3, name = toyota corolla, ap = 92.25% (TP = 28, FP = 5)
class_id = 4, name = suzuki mehran, ap = 89.82% (TP = 71, FP = 20)
class_id = 5, name = toyota landcruiser, ap = 92.91% (TP = 75, FP = 6)
class_id = 6, name = suzuki swift, ap = 81.42% (TP = 48, FP = 18)
class_id = 7, name = honda city, ap = 92.41% (TP = 40, FP = 29)
class_id = 8, name = toyota prius, ap = 98.67% (TP = 40, FP = 8)
class_id = 9, name = honda vezel, ap = 99.57% (TP = 75, FP = 0)
class_id = 10, name = toyota passo, ap = 98.74% (TP = 89, FP = 10)
class_id = 11, name = honda civic 2017, ap = 100.00% (TP = 119, FP = 0)
class_id = 12, name = toyota aqua, ap = 96.71% (TP = 97, FP = 3)
class_id = 13, name = daihatsu mira, ap = 96.86% (TP = 103, FP = 16)
class_id = 14, name = suzuki wagon r, ap = 98.21% (TP = 111, FP = 0)

for thresh = 0.25, precision = 0.87, recall = 0.96, F1-score = 0.91
for thresh = 0.25, TP = 1096, FP = 168, FN = 50, average IoU = 65.07 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.955673, or 95.57 %
Total Detection Time: 77.000000 Seconds

```

Figure 14 Mid training results.

Loss had significantly decreased as well. With mAP in the 90s.

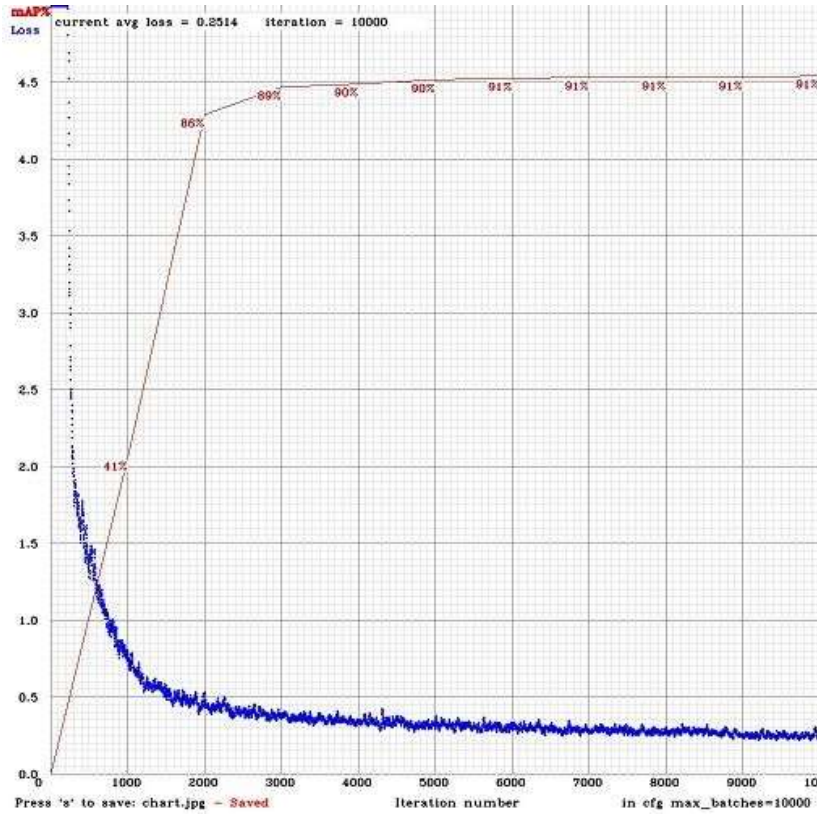


Figure 15 Average Loss and mAP graph plotted against number of iterations

In the end the average loss was at 0.07383 and final mAP was at 99.5% which is very high considering the quality and amount of data that we had. The following images show the stats.


```

class_id = 0, name = suzuki cultus, ap = 100.00% (TP = 55, FP = 0)
class_id = 1, name = honda civic, ap = 100.00% (TP = 52, FP = 0)
class_id = 2, name = toyota vitz, ap = 100.00% (TP = 98, FP = 0)
class_id = 3, name = toyota corolla, ap = 97.30% (TP = 36, FP = 1)
class_id = 4, name = suzuki mehran, ap = 95.96% (TP = 74, FP = 4)
class_id = 5, name = toyota landcruiser, ap = 100.00% (TP = 79, FP = 0)
class_id = 6, name = suzuki swift, ap = 100.00% (TP = 54, FP = 0)
class_id = 7, name = honda city, ap = 100.00% (TP = 42, FP = 0)
class_id = 8, name = toyota prius, ap = 100.00% (TP = 40, FP = 0)
class_id = 9, name = honda vezel, ap = 100.00% (TP = 77, FP = 1)
class_id = 10, name = toyota passo, ap = 100.00% (TP = 89, FP = 0)
class_id = 11, name = honda civic 2017, ap = 99.17% (TP = 120, FP = 0)
class_id = 12, name = toyota aqua, ap = 100.00% (TP = 99, FP = 0)
class_id = 13, name = daihatsu mira, ap = 100.00% (TP = 112, FP = 0)
class_id = 14, name = suzuki wagon r, ap = 100.00% (TP = 113, FP = 0)

for thresh = 0.25, precision = 0.99, recall = 0.99, F1-score = 0.99
for thresh = 0.25, TP = 1140, FP = 6, FN = 6, average IoU = 88.80 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.994951, or 99.50 %
Total Detection Time: 81.000000 Seconds

```

Figure 16 Final training results

Our trained model gave 99.5% accuracy with a real-time detection speed (i.e. more than 20fps), and average IOU at 88.8%. Following are some cars that we detected using our trained algorithm.



Conclusion and Future Recommendations

Our network gives great results on our dataset but the quality and diversity of our dataset will be hindrance in using this implementation for any high security and high stake environment. Even though the network is more than capable to detect vehicles in such environments, it needs more quantity and quality of data. It also needs more computer power and more time to train on a larger dataset. After these conditions are met the accuracy that we achieved on our (easier) dataset can be achieved on even more diverse and difficult dataset.

We intend to train our model on a much larger and more suitable dataset (Safe city CCTV cameras data) in the future so that our model can be integrated into the mega-project and can actually be implemented to help reduce vehicular crimes and make safer and smarter cities.

Furthermore, a paper can also be written upon YOLO and Darknet implementation that detects vehicles in real-time. But that can only be done after a good quantity and quality of data is used.

Work division

- Literature Review (All Members)
- Data Gathering and Augmentation (Sikander Azam)
- Data Preprocessing and Annotation (Muhammad Ahsan Fahim)
- Network Training and Testing (Hassan Rasheed)

References

- E-Pakistan. (2019). *E-Pakistan and CIT*. Retrieved from E-Pakistan.org: <https://www.e-pakistan.org/Home#event>
- European Union. (2010). DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010. *Official Journal of the European Union*. Retrieved from <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF>
- G Boeing, P. W. (2016). New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research*. Retrieved from <https://arxiv.org/abs/1605.05397>
- Islamabad Police. (2017). *Crime Statement 2017*. Retrieved from <http://islamabadpolice.gov.pk/ipwe/crime-statistics.php>
- Islamabad Police. (2017). *Islamabad police*. Retrieved from <http://islamabadpolice.gov.pk/ipwe/crime-statistics.php>
- Islamabad Police. (2018). *Safe City Project*. Retrieved from <http://islamabadpolice.gov.pk/ipwe/safecity>
- Joseph Redmon, A. F. (2018). YOLOv3: An Incremental Improvement. *Tech report*. Retrieved from <https://pjreddie.com/media/files/papers/YOLOv3.pdf>
- Murugavel, M. (2018). How to train YOLOv2 to detect custom objects. Retrieved from https://medium.com/@manivannan_data/how-to-train-yolov2-to-detect-custom-objects-9010df784f36
- Murugavel, M. (2018). How to train YOLOv3 to detect custom objects. *Medium*. Retrieved from https://medium.com/@manivannan_data/how-to-train-yolov3-to-detect-custom-objects-ccbcafeb13d2
- NADRA. (2019). *Safe City*. Retrieved from <https://www.nadra.gov.pk/solutions/security-solutions/safe-city-solutions/>
- National Institute of Justice. (2014). *Predictive Policing*. Retrieved from <https://www.nij.gov/topics/law-enforcement/strategies/predictive-policing/Pages/welcome.aspx>
- OSAC. (2018). *Pakistan 2018 Crime & Safety Report: Islamabad*. Retrieved from <https://www.osac.gov/Pages/ContentReportDetails.aspx?cid=24120>
- Police One. (2019). *Predictive Policing*. Retrieved from Police One: <https://www.policeone.com/predictive-policing/>
- Techopedia. (2019). Data Augmentation. *Techopedia*.