



REAL TIME ROAD TRAFFIC DETECTION USING COMPUTER VISION

¹.GAYATHRI,Y.DINESH KUMAR,M.ASWINI KUMAR

ASSISTANT PROFESSOR, ASSISTANT PROFESSOR, ASSISTANT PROFESSOR
COMPUTER SCIENCE AND ENGINEERING,
¹SITAM, VIZIANAGARAM,INDIA

Abstract: We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the MAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is far less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and RCNN, by a wide margin when generalizing from natural images to artwork on both the Picasso Dataset and the People Art Dataset.

Index Terms – YOLOv3, Decision tree, SVC, CNN.

I. INTRODUCTION

YOLO is refreshingly simple: see Figure 1. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems.

II RESEARCH METHODOLOGY

The implementation process started with data cleaning, preprocessing and algorithms are presented with a short summary of these issues at the end of the chapter.

1. Tools Used

Like every other researches and studies this thesis utilized some applications and tools for creating models and experiments.

1.1 Python

Python is a general purpose programming language which is created by Guido van Rossum, it is used for different platforms like mathematic, computer GUI, web and so many large scientific applications.

1.2 Matplotlib

Plotting is growing in all fields to visualize the data and to make it understandable. Therefore Matplotlib is being used as a plotting library to create different kind of graphs and figures for variety of aims.

1.3 Jupyter Notebook

Jupyter Notebook is a web based application which makes us able to create and modify live codes, equations, plaintexts and visualizations.

1.4 Datasets

Udacity website equips students with the great resources for training the classifiers. Vehicles and non-vehicles samples of the KITTI vision benchmark suite have been used for training.

2.Design

We implement this model as a convolutional neural network and evaluate it on the PASCAL VOC detection dataset [9]. The initial convolutional layers of the network extract features from the image while the fully connected layers predict the output probabilities and coordinates. Our network architecture is inspired by the Google Net model for image classification [33]. Our network has 24 convolutional layers followed by 2 fully connected layers. However, instead of the inception modules used by Google Net we simply use 1×1 reduction layers followed by 3×3 convolutional layers, similar to Lin et al [22].

2.1 Convolution layers in YOLOv3

- It contains 53 convolutional layers which have been, each followed by batch normalization layer and Leaky ReLU activation.
- Convolution layer is used to convolve multiple filters on the images and produces multiple feature maps
- No form of pooling is used and a convolutional layer with stride 2 is used to downsample the feature maps.
- It helps in preventing loss of low-level features often attributed to pooling.

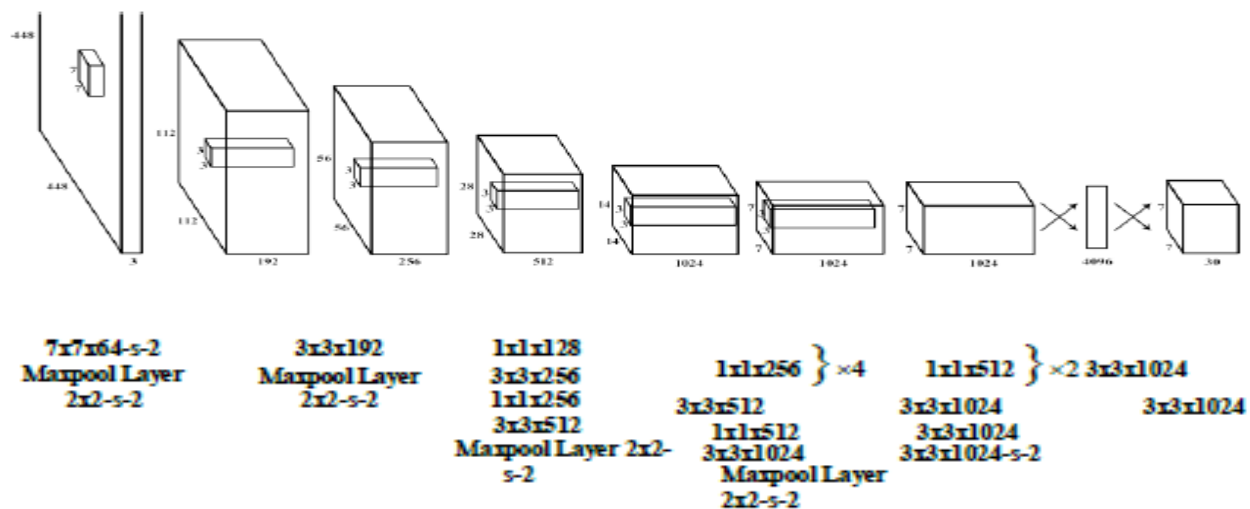


Figure:1

2.2 ARCHITECTURE

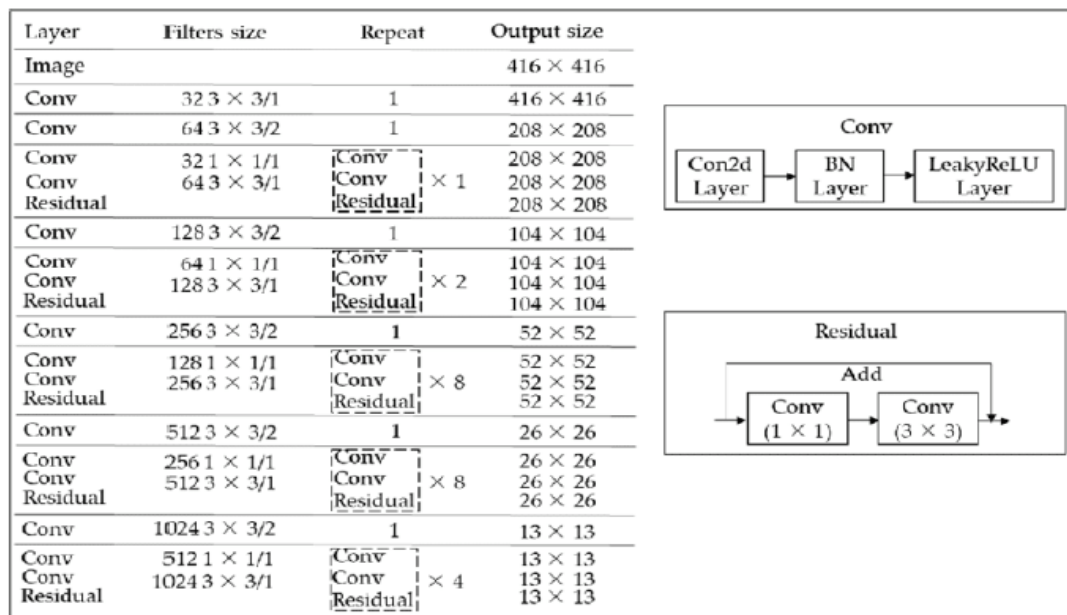
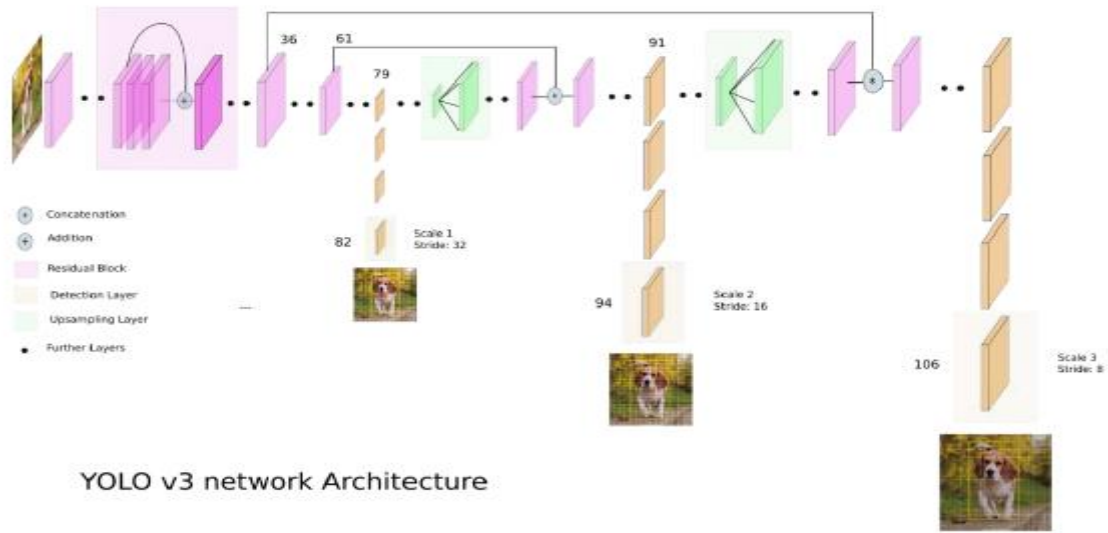


Figure:2

2.2 Description of Architecture

Steps for object Detection using YOLO v3:

- The inputs is a batch of images of shape (m, 416, 416, 3).
- YOLO v3 passes this image to a convolutional neural network (CNN).
- The last two dimensions of the above output are flattened to get an output volume of (19, 19, 425):
- Here, each cell of a 19 x 19 grid returns 425 numbers.
- $425 = 5 * 85$, where 5 is the number of anchor boxes per grid.
- $85 = 5 + 80$, where 5 is (pc, bx, by, bh, bw) and 80 is the number of classes we want to detect.
- The output is a list of bounding boxes along with the recognized classes. Each bounding box is represented by 6 numbers (pc, bx, by, bh, bw, c). If we expand c into an 80-dimensional vector, each bounding box is represented by 85 numbers.
- Finally, we do the IoU (Intersection over Union) and Non-Max Suppression to avoid selecting overlapping boxes.



YOLO v3 network Architecture

Figure:3

AT SCALE ONE

2. AT SCALE TWO



Detection at scale 1, we see somewhat large objects are picked. But we don't detect a few cars.

No detections

3. AT SCALE THREE



Detection at the largest scale (3). Look how only the small objects are picked up, which weren't detected

Figure:4

3. METHODOLOGY

The YOLO framework (You Only Look Once) on the other hand, deals with object detection in a different way. It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes. The biggest advantage of using YOLO is its superb speed – it's incredibly fast and can process 45 frames per second.

Intersection over Union and Non-Max Suppression.

How can we decide whether the predicted bounding box is giving us a good outcome (or a bad one)?

This is where Intersection over Union comes into the picture. It calculates the intersection over union of the actual bounding box and the predicted bounding box.

Consider the actual and predicted bounding boxes for a car as shown below:

$\text{IoU} = \text{Area of the intersection} / \text{Area of the union}$, i.e.

$\text{IoU} = \text{Area of yellow box} / \text{Area of green box}$

If IoU is greater than 0.5, we can say that the prediction is good enough. 0.5 is an arbitrary threshold we have taken here, but it can be changed according to your specific problem.

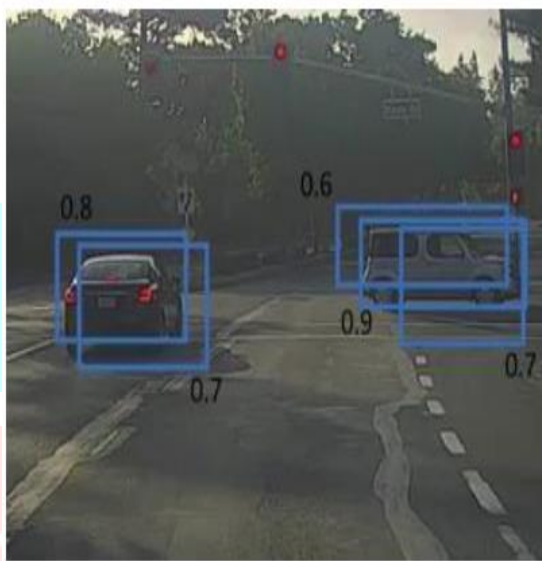


Figure:5

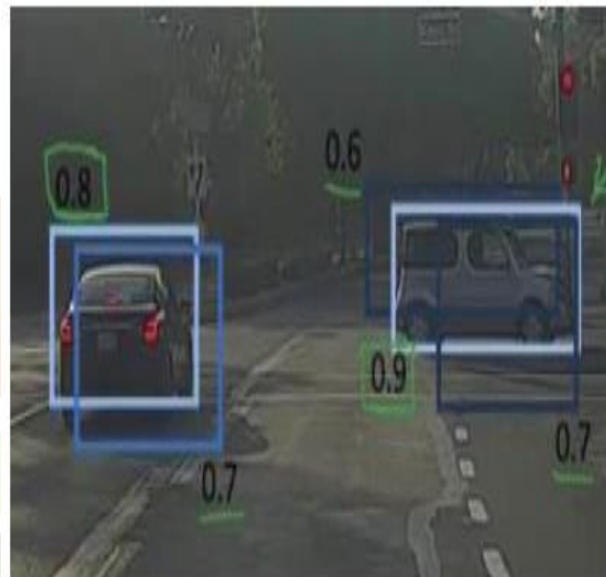


Figure:6

4. Implementation

4.1 Training

The input for training our model will obviously be images and their corresponding y labels. Let's see an image and make its y label:

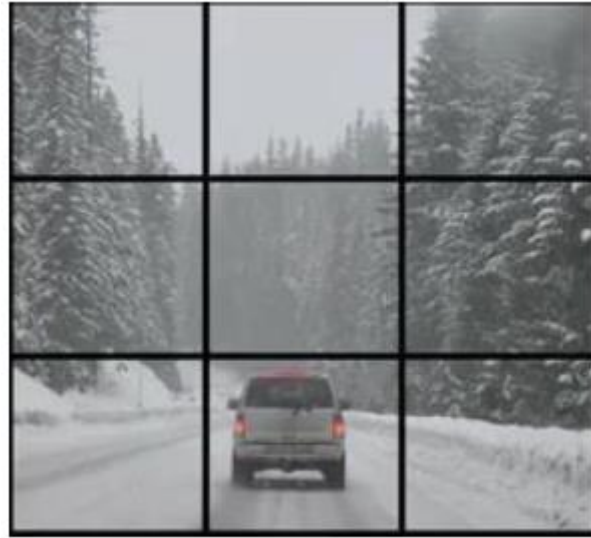


Figure:7

Consider the scenario where we are using a 3 X 3 grid with two anchors per grid, and there are 3 different object classes. So the corresponding y labels will have a shape of 3 X 3 X 16. Now, suppose if we use 5 anchor boxes per grid and the number of classes has been increased to 5. So the target will be 3 X 3 X 10 X 5 = 3 X 3 X 50. This is how the training process is done – taking an image of a particular shape and mapping it with a 3 X 3 X 16 target (this may change as per the grid size, number of anchor boxes and the number of classes).

4.2 Testing

The new image will be divided into the same number of grids which we have chosen during the training period. For each grid, the model will predict an output of shape 3 X 3 X 16 (assuming this is the shape of the target during training time).

The exact dimensions and steps that the YOLO algorithm follows:

- Takes an input image of shape (608, 608, 3)
- Passes this image to a convolutional neural network (CNN), which returns a (19, 19, 5, 85) dimensional output
- The last two dimensions of the above output are flattened to get an output volume of (19, 19, 425):
- Here, each cell of a 19 X 19 grid returns 425 numbers
- $425 = 5 * 85$, where 5 is the number of anchor boxes per grid
- $85 = 5 + 80$, where 5 is (pc, bx, by, bh, bw) and 80 is the number of classes we want to detect
- Finally, we do the IoU and Non-Max Suppression to avoid selecting overlapping boxes

5. Experimental Setup

While developing Machine learning algorithms there are many options to choose for instance MATLAB, Python, and R programming language. Each option has its own advantages and privileges. Because of that the researcher has decided to choose python for the development due to the easiness and rich libraries available for all kind of tasks. SVMs algorithm are used mostly for classification tasks. These models are working based on discovering a hyperplane concept which actually perfectly divides the data into two classes (Bambrick, 2016).

- SVM works perfect while dealing with unknown data.

- It is efficient when the data is semi-structured or unstructured like texts, trees, and images.
- SVM properly measure dimensional data
- In SVM usually there are no danger of over fitting due to having generalization in practice

5.1 SVM implementation

Like mentioned earlier, there are bunch of libraries that can help us implement the models easily. First these libraries are needed to be imported in the notebook. Then the dataset needed to be read and load into arrays for further processing. Numpy have been used for these job. For the testing purpose and to understand whether the data is loaded properly or not, few examples of the dataset showed in Figure .

5.2 Results of Descriptive statistics

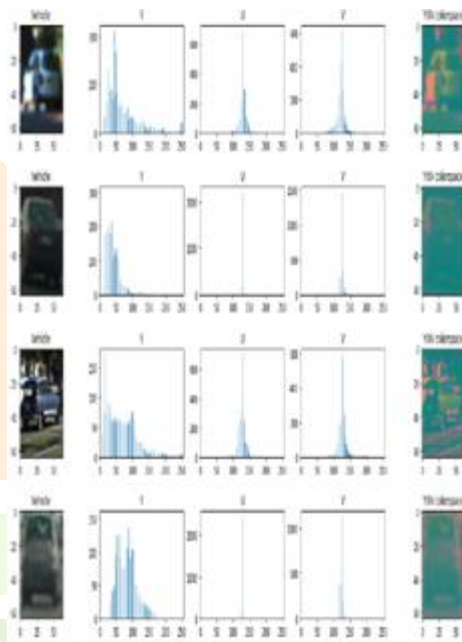


Figure: 8 Histogram and color feature of vehicles.

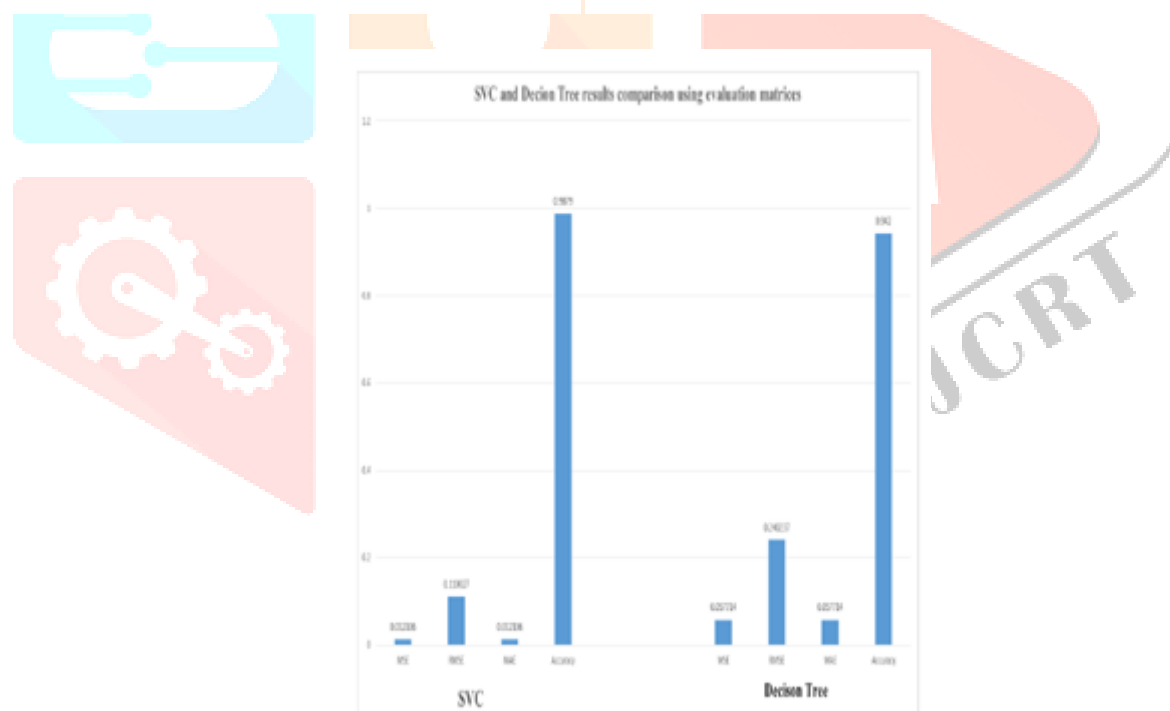
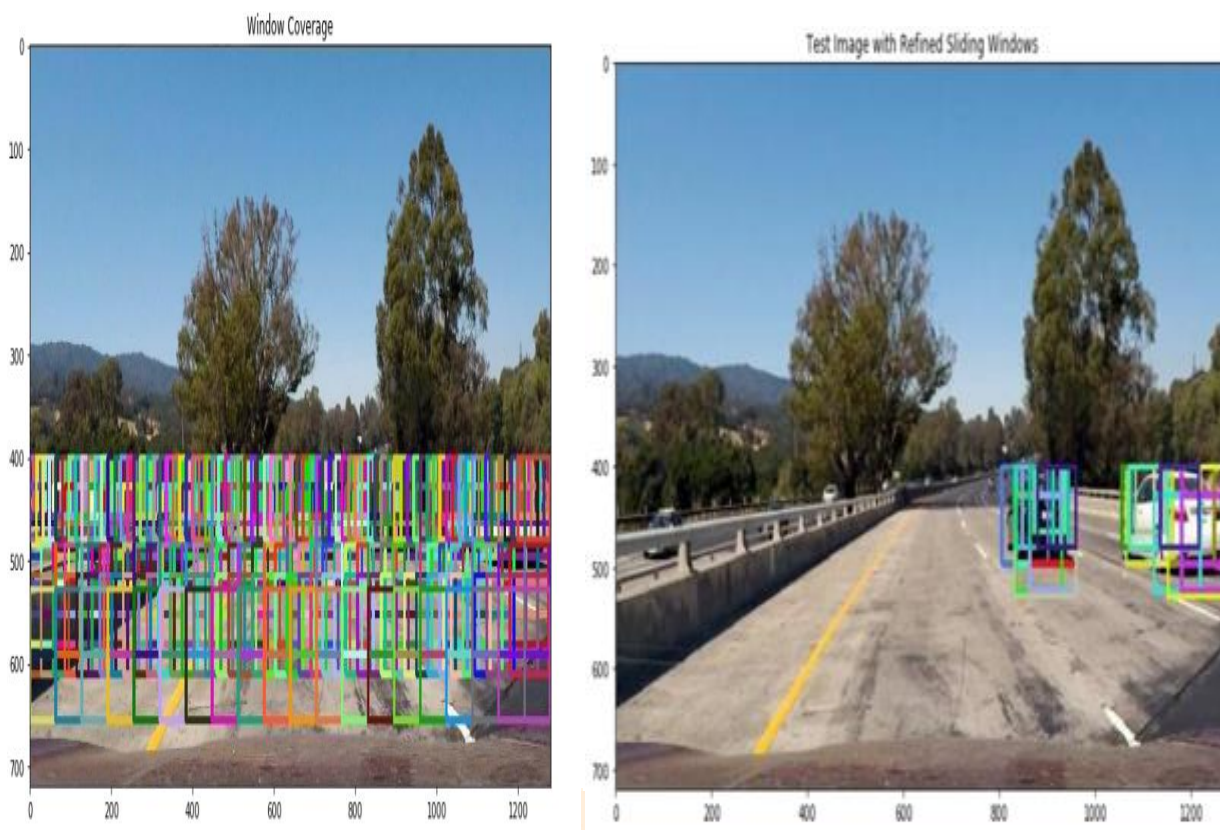


FIGURE:9 SVC and Decision tree comparison result

6.CONCLUSION

By rapid development in car and traffic industries, at the same the growth of population in the world brought the needs for different tools and techniques specially technology solutions in order to manage traffics in cities and populated areas. Object detection can be used in industries, digitized cities, government, research, academia, environment etc. Vehicle detection and tracking is part of the object detection which is used in traffic, cities etc. the importance of the topic is growing larger. That being said this research is intended to contribute the improvement of the accuracy of these algorithms and models via available techniques and tools. This Thesis developed two classifier algorithms to detect and track vehicles. These two models are Support Vector Machine (SVM) and Decision Tree. The algorithm selection was based on

various studies in literature review. The most suggested models by other researchers were these two models. Therefore, the author decided to choose these models and compare them in order to specify the best model among these two. Many techniques have been deployed to increase the accuracy level and to make the best result possible.

II. ACKNOWLEDGMENT

REFERENCES

- [1] Andrew, W. M. and Victor, M. (2003), *Handbook of International Banking (London: Edward Elgar Publishing Limited)*, 350-358
- [2] Baha, N. (2014). Real-Time Obstacle Detection Approach using Stereoscopic Images. *International Journal of Information Engineering and Electronic Business*, 6(1), 42.
- [3] Bahrepour, M., Akbarzadeh-T, M. R., Yaghoobi, M., & Naghibi-S, M. B. (2011). An adaptive ordered fuzzy time series with application to FOREX. *Expert Systems with Applications*, 38(1), 475-485.
- [4] Bambrick, N. (2016, June 24). *Support Vector Machines for dummies; A Simple Explanation*. Retrieved April 28, 2018, from AYLIEN | Text Analysis API | Natural Language Processing: <http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/pp.1-13>.
- [5] Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing-Letters and Reviews*, 11(10), 203-224.
- [6] Berni, J. A., Zarco-Tejada, P. J., Suárez, L., & Fereres, E. (2009). Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3), 722-738.
- [7] BIS. (2016). Triennial central bank survey: Foreign exchange turnover in April 2016 Monetary and Economic Department.
- [8] Chen, X., & Meng, Q. (2015, November). Robust vehicle tracking and detection from UAVs. In *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)* (pp. 241-246). IEEE.
- [9] Chen, X. (2015), *Automatic Vehicle Detection and Tracking in Aerial Video*, Doctor of Philosophy Thesis in Loughborough University.
- [10] Kalghatgi, M. P., Ramannavar, M., & Dr. Sidnal, N. S. (2015). A Neural Network Approach to Personality Prediction based on the Big-Five Model. *International Journal of Innovative Research in Advanced Engineering*, 2(8), 56-63
- [11] Kanistras, K., Martins, G., Rutherford, M. J., & Valavanis, K. P. (2015). Survey of unmanned aerial vehicles (UAVs) for traffic monitoring. *Handbook of unmanned aerial vehicles*, 2643-2666.
- [2] Basu, S. 1997. The Investment Performance of Common Stocks in Relation to their Price to Earnings Ratio:
- [12] Test of the Efficient Markets Hypothesis. *Journal of Finance*, 33(3): 663-682.
- [3] Bhatti, U. and Hanif. M. 2010. Validity of Capital Assets Pricing Model. Evidence from KSE-Pakistan. *European Journal of Economics, Finance and Administrative Science*, 3 (20).