OBJECT DETECTION AND CLASSIFICATION THROUGH DEEP LEARNING APPROACHES

Deepika Solanki, Mr. Shrwan Ram
MBM engineering college, Jodhpur

ABSTRACT-In this paper, we implemented the image classification and object detection. This paper presents a deep learning approach for traffic light detection in adapting a single shot detection(SSD) approach and image classification of two categories of bicycle by retraining inceptionv3 model both using an open source tool called TensorFlow Object Detection API. We reviewed the current literature on convolutional object detection and tested the implementability of one of the methods and discovered that convolutional object detection is still evolving as a technology despite that convolutional object detection has outranked other object detection methods. To implement object detection and image classification there is free availability of datasets and pre-trained networks it is possible to create a functional implementation of a deep neural network without access to specialist hardware.

KEYWORDS-Object detection, Deep learning, Convolutional neural network, TensorFlow Object Detection API, SSD model, InceptionV3, InceptionV2.

I. INTRODUCTION

Classification of objects into their specific categories is often been vital tasks of machine learning. In recent years, deep learning has been utilized in image classification, object detection. To increase the performance of image classification deep learning uses a neural network with more than one hidden layer. For image classification and object detection one of the most frequently used deep learning neural network with more number of hidden layers is the convolutional neural network (CNN). CNN information gets directly from the image, so it eliminates manual feature extraction.There is a common problem in classifying image with deep learning, is lower performance because of over-fitting. To increase performance, and to prevent over-fitting we use large datasets. CNN have fewer connections and hyper parameter that make CNN model easy to train and perform slightly worse than other models [5].

Robotized driving on roadways is an effectively looked into issue which has prompted the rise of numerous driver help frameworks. Urban territories give another arrangement of difficulties which require more advanced calculations in different zones running from observation over conduct intending to impact shirking frameworks. One essential piece of recognition is the identification and classification of traffic lights. Traffic lights exhibit a testing issue because of their little size and high vagueness with different items introduce in the urban condition, for example, lights, beautifications, and reflections [7], [14].

Recent enhancements in object detection area unit driven by the success of convolutional neural networks (CNN). They're able to learn rich features outperforming hand stitched options. So far, research in traffic light detection mainly focused on hand-crafted features, admire color, shape or brightness of the traffic light bulb. In this research work we present a deep learning approach for traffic light detection in adapting a single shot detection (SSD) approach. SSD performs object proposals creation and classification using a single CNN. The initial SSD struggles in sleuthing terribly tiny objects, which is essential for traffic light detection. By our variations it's potential to find objects a lot of smaller than 10 pixels while not increasing the input image size. As a result, we have a tendency to reach high accuracy [13].

In this paper, we performed two separate experiments, for the first experiment we are classifying images of bikes, we are taking two different categories of bike images for e.g. mountain bikes and road bikes and for the second experiment traffic light detection in an image and its classification. We have used TensorFlow object detection API to train and evaluate convolutional neural network, one of the most popular Python programming language libraries for deep learning.The flow diagram of Proposed Methodology is shown in Figure 1 and 2.

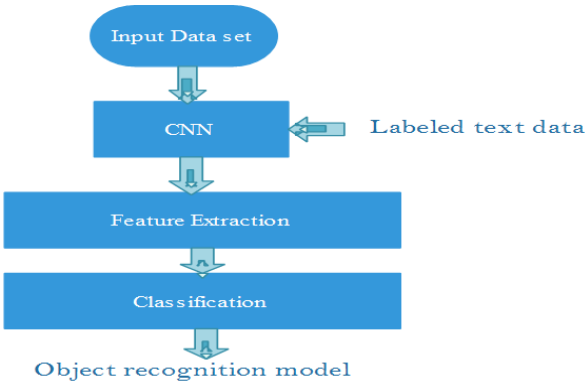

Figure 1:Training dataset

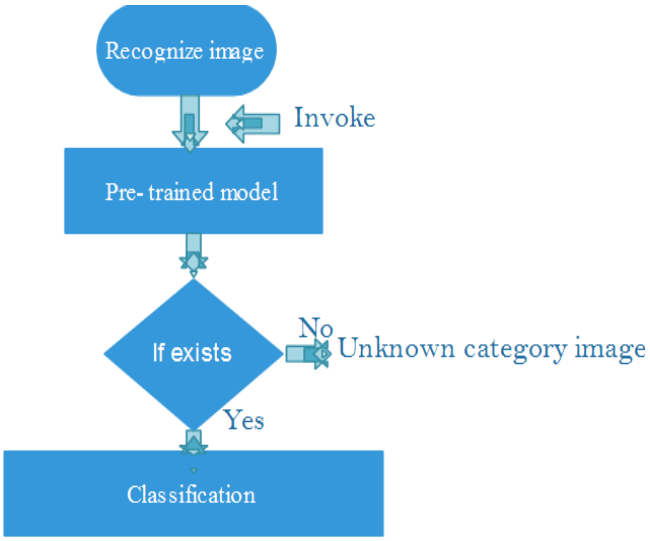

Figure 2:Testing dataset

II. RELATED WORK

Past takes a shot at traffic light detection and classification use spotlight detection and color thresholding, template matching and map information.Every one of these frameworks make solid suppositions [6], [7], [8]. Generally, they require the traffic lights to be no less than a specific size for the calculation to take a shot at an unmistakable foundation, for example, suspended traffic lights before the sky or expect the presence of maps that contain earlier information about the areas of all traffic lights in the earth. With the ongoing advances and execution of deep neural network, significant changes were made in a few fields of machine learning and particularly computer vision. Deep learning has been utilized for image classification end-to-end object detection, pixel-precise object segmentation, and different applications. The disadvantage of deep neural network as of now is theamount of needed training data.With rising achievement of CNNs, additionally object proposal generation was performed by sharing a base network together with classification. Ordinarily, those systems are prepared by a confidence and localization loss, ensuring exact bouncing boxes with respect to the overlap metric intersection over union. One key issue of those approaches is the discovery of small objects. It is for the most part caused by pooling tasks, which increment the receptive field and decrease the computational effort. Nonetheless, pooling additionally diminishes the image resolution prompting difficulties for precise localization of small objects.

III. SSD

The Single Shot MultiBox Detector [10] (SSD) takes integrated detection even further. The method does not generate proposals at all, nor does it involve any resampling of image segments. It generates object detections using a single pass of a convolutional network.Somewhat resembling a sliding window method, the algorithm begins with a default set of bounding boxes. These include different aspect ratios and scales. The object predictions calculated for these boxes include o set parameters, which predict how much the correct bounding box surrounding the object differs from the default box.

The algorithm deals with different scales by using feature maps from many different convolutional layers (i.e. larger and smaller feature maps) as input to the classifier. Since the method generates a dense set of bounding boxes, the classifier is followed by a non-maximum suppression stage that eliminates most boxes below a certain confidence threshold.

IV. TENSORFLOW

Tensorflow[3] is a machine learning library produced by Google that was released in 2015 after they upgraded their system from Disbelief, which is the platform GoogLeNet was developed in. Tensorflow was built around scalability and the ability to perform a wide variety of tasks. It allows for symbolic differentiation to reduce compute time, was written in C++ for easy deployment, scales across many machines, and is compatible with both CUDA and cuDNN. Tensor-flow relies on computational graphs to represent all operations and sessions to store the computational graph's variable's values. There are a multitude of features that Tensorflow includes such as an execution timeline, model optimizations, and powerful network debugging tools such as Tensorboard. Due to the power and scale of Tensorflow, it has become the most widely used CNN developing tool.

V. EXPERIMENTS

To perform the first experiment we are classifying images of mountain bikes and road bikes. We use TensorFlow to train and evaluate convolutional neural network, one of the most popular Python programming language libraries for deep learning and retrain the Inception v3 image classifier architecture on ImageNet data set. For training image set we are using 105 labelled mountain bike images and 106 labelled road bike images.Finally we train the system with training dataset and test the validation of the system with test dataset.

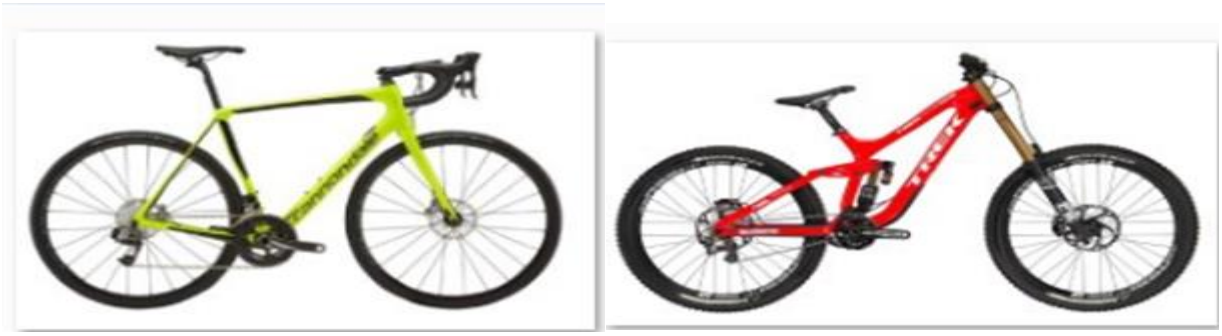

Figure 3: Input labelled data - Mountain bikes and Road bikes.

We are using the 'inception_v3' model which is the most accurate, but also the slowest. In this experiment we are giving 500 training steps, with a learning rate to use when training as 0.01 and calculating Train accuracy, cross entropy and validation accuracy in every training step.

Accuracy simply measures how often the classifier makes the correct prediction. It's the ratio between the number of correct predictions and the total number of predictions (the number of data points in the test set)

$$Accuracy = \frac{correct\ predictions}{total\ data\ points}$$

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0. Below are the predicted results which we obtained.

Fig 5: Test results of the set of 2 category (class)

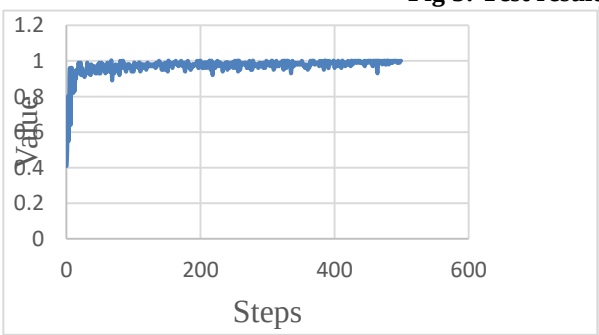

Figure 6: Train Accuracy graph

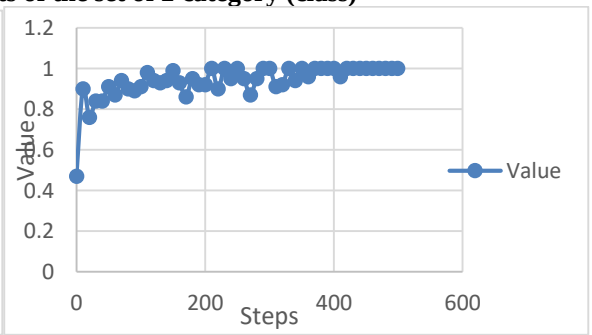

Figure 7: Validation Accuracy graph

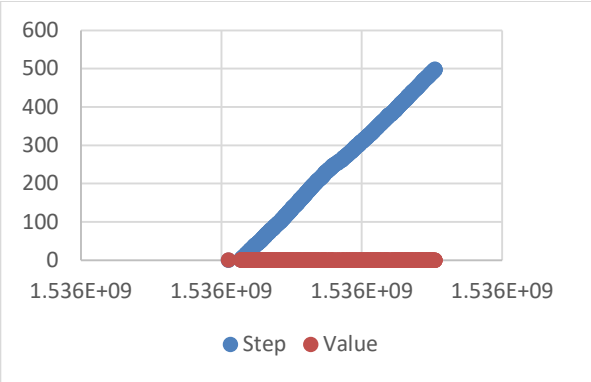

Figure 8: Train cross entropy graph

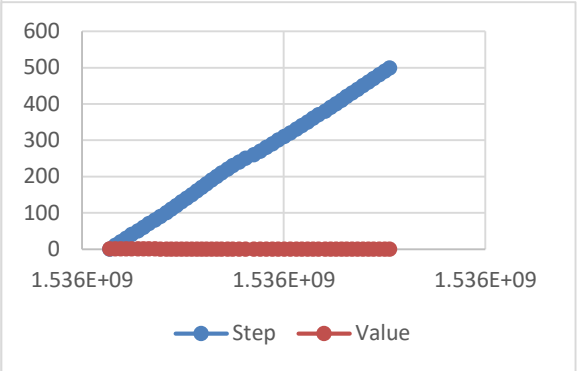

Figure 9: Validation cross entropy graph

To perform the second experiment we detect a traffic light in an image and classify it.We use TensorFlow to train and evaluate convolutional neural network, SSD Single shot multi box detection architecture is used for prediction.

In the training image set for all the images of traffic lights, we drawn a box around each instance of an object we like to train, which is called the Bounding box. We usedLabelImg software to complete this task. We are using 127 images which has traffic lights in it to train the network.

Figure 10: Input image with the Bounding box details

The pre-trained SSD model (ssd_inception_v2_coco) was tested with 127 pictures of traffic lights. The resulting tacked images were visually checked.

Training set: The dataset includes over 127 images containing traffic lights with very specific tags such as "traffic_light". For practical usage of a traffic light detection system,mainly front orientated traffic lights, i.e. traffic lights facingthe vehicles road have to be detected. Traffic lights for pedestrians,trams or turned traffic lights are negligible.

Evaluation set/ Test set: For evaluation we use the proposed splitof the dataset. Thus, the evaluation set contains around onethird of all annotations. It contain only 10 images.

Limitations of the Evaluation Set: Varying label rules isone key problem of the dataset for the purpose of evaluation.The majority of the images are annotated with front-facingtraffic lights only. A small part also contains annotated turnedtraffic lights (e.g. for pedestrians or oncoming traffic). Detectionson those traffic lights are counted as false positives.

Figure 11: Test results of detecting traffic lights

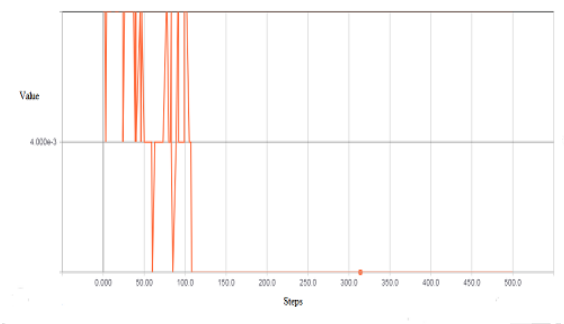

Figure 12: Learning rate of the model graph

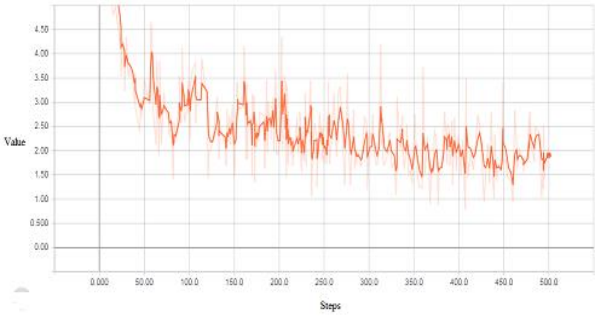

Figure 13: Classification loss graph

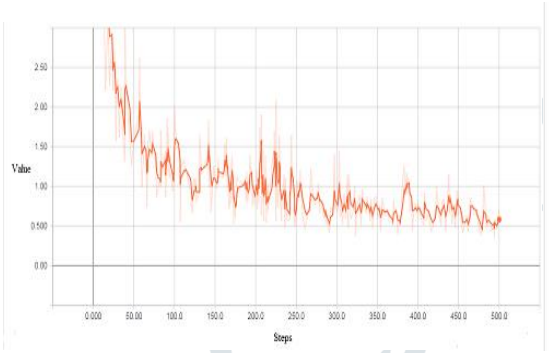

Figure 14: Localization loss graph

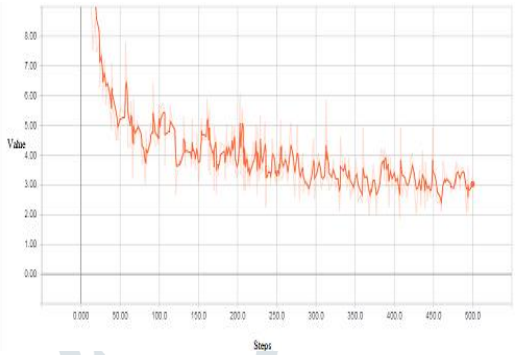

Figure 15: Total loss graph

VI. CONCLUSION

We evaluated different operating points differing in the number of false positives per image. One common statement about CNNs is that more data automatically helps to improve generalization and the overall results. To investigate the impact of the amount of data on the recall. We showed, that more data leads to better results and the network depth has to be chosen carefully. Recall values up to 95 percent even for small objects were reached, values increase up to 98-100 percent for larger objects.

A model with another type of architecture such as Faster R-CNN could be better suited for detecting small objects.Image pre-processing could improve detection results.A larger and more variable datasets used for training and testing the models. With a more efficient hardware setup such as using graphics processing units. GPUs are more efficient for the taskthan traditional CPUs and provide a relatively cheap alternative to specialist hardware. Today, researchers typically use high-end consumer graphiccards, such as NVIDIA Tesla K40 which will result in a better results in less execution time.

VII. REFERENCES

[1]. LeCun, Yann & Bengio, Y & Hinton, Geoffrey, "Deep Learning". Nature. 521, pp 436-44, 2015. 10.1038/nature14539.

[2]. P. N. Druzhkov and V. D. Kustikov, " A survey of deep learning methods and software tools for image classification and object detection Pattern Recognition and Image Analysis", Volume 26, Issue 1, pp 9–15, 2016.

[3]. William G. Hatcher and Wei Yu " A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends", DOI 10.1109/ACCESS.2018.2830661, IEEE Access.

[4]. Bo Zhao, Jiashi Feng, Xiao W and Shuicheng Yan, "A Survey on Deep Learning-based Fine-grained Object Classification and Semantic Segmentation", International Journal of Automation and Computing DOI: 10.1007/s11633-017-1053-3.

[5]. Zhuwei Qiny," How convolutional neural networks see the world A survey of convolutional neural network visualization methods", American Institute of Mathematical Sciences Volume 1, Number 2, pp. 149-180, 2018. doi:10.3934/mfc.2018008.

[6]. Christian Szegedy, Alexander Toshev and Dumitru Erhan," Deep Neural Networks for Object Detection", NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Page 2553-2561, 2013.

[7].Müller, Julian; Dietmayer and Klaus "Detecting Traffic Lights by Single Shot Detection" eprint arXiv:1805.02523 ,05/2018.

[8]. D. P. Sudharshan and S. Raj, "Object recognition in images using convolutional neural network," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, pp. 718-821, 2018. doi: 10.1109/ICISC.2018.8398912.

[9]. Lin TY. et al, Fleet D., Pajdla T., Schiele B. and Tuytelaars T "Microsoft COCO: Common Objects in Context", (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham.

[10]. Liu W. et al. Leibe B., Matas J., Sebe N. and Welling M. " SSD: Single Shot MultiBox Detector", (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham.

[11]. J. Huang et al., "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 3296-3297, 2017. doi: 10.1109/CVPR.2017.351.

[12]. Pathak A.R., Pandey M., Rautaray, Pattnaik P., Rautaray S., Das H. and Nayak J. S. "Deep Learning Approaches for Detecting Objects from Images: A Review", (eds) Progress in Computing, Analytics and Networking. Advances in Intelligent Systems and Computing, vol 710, 2018. Springer, Singapore

[13].T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(, Beijing, pp. 721-724, 2017. doi: 10.1109/ICBDA.2017.8078730

[14].K. Behrendt, L. Novak and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 1370-1377, 2017.doi: 10.1109/ICRA.2017.7989163

[15]. M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, "Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset," 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, pp. 2341-2345, 2015.doi: 10.1109/ITSC.2015.378.