International Journal of Research in Engineering, Science and Management
Volume-3, Issue-4, April-2020
www.ijresm.com | ISSN (Online): 2581-5792

Lane Detection Using Tensorflow

Arushi Jaitley[1], Anshika Singhal[2], Apoorva Singh[3], Ayushi Agaarwal[4], Arpesh Kumar Singh[5]
[1,2,3,4]*Student, Department of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut, India*
[5]*Professor, Department of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut, India*

***Abstract*: The area of computer vision is emerging continually with the increasing interaction and development to provide a comfortable interaction between human and machines. One of the key aspects in the process of computer vision is object detection. Either objects can be identified partially or close to the original objects. Traditional algorithms can't recognize objects as efficiently due to its limitations. Whereas the deep learning models require large amount of data for training the dataset, which has more resource and labour intensive in nature. The algorithm determines its precision in object detection as well as its reliability. The recognition and classification of object begins with preparing dataset followed by splitting the dataset into training dataset and test dataset. The task of training the dataset can be assisted by both traditional as well as modern deep neural networks. The loss per step or epoch is calculated on the training dataset to signify the efficiency and accuracy of the model. In this model, we have achieved a maximum accuracy of about 85.18% after training the dataset used.**

***Keywords*: Object detection, Deep neural network, Image segmentation, Computer vision, Tensorflow object detection API, Self-driving vehicle.**

1. Introduction

Object detection is the blooming research area in the field of computer vision. The ability to identify and recognize objects either in single or more than one image frame can gain extreme importance in various ways as while driving the vehicle, the driver cannot identify objects properly due to the dearth of attention, reflection of light, anonymous objects etc. which may lead to fatal accidents. In order to overcome such perceptible problems, autonomous vehicles and Advanced Driver Assistance System (ADAS) took the generous task of object detection and classification. The task of computer vision is performed in the following steps:

1. Classification of object in image
2. Localization of object in image
3. Object detection
4. Segmentation of image

The application of object detection can be found in advanced robotics, defense systems, surveillance systems, space research, face recognition and many more. The idea of self-driving vehicles has been evolving with the advancement in techniques related to the task of identifying and extracting features from the objects. Object detection for self-driven vehicles is a non-trivial task in order to navigate on the road. The evolution of deep neural networks have changed the aspect of computer vision over the traditional methods. Conventional machine learning and computer vision models plays a prevalent role in the process of object classification, however the industry now heavily relies on the deep learning based classifiers. The emergence of graphical processing units (GPU) has led to more efficient and convenience in achieving the task of object classification through deep neural network models. These models try to learn important features corresponding to each class that are inspired from the biological structure of neurons in humans. Google's Tensorflow is one such machine learning framework which works on dataflow programming among a range of tasks. Nodes in TensorFlow represent mathematical operations and the graph edges represent multidimensional arrays called as Tensors. Tensorflow object detection API is capable in detecting objects in an image with good accuracy it is also able to detect objects in live streaming video with a good degree of precision in which speed of frames is about 25–30 frames per second. We propose the use of Tensorflow object detection API for our dataset to train and test the dataset in order to detect objects successfully for an autonomous vehicle.

The main challenge of this project lies in three aspects: acceleration, compression and accurateness, where accuracy would be the trade-off for the first two aspects. We evaluate the models based on three metrics: mAP metric, detection speed (fps) and model size.

2. Methodology

The below image is an example of illustrating how a lane detection algorithm works. Each object in the image, have been located and identified with a certain level of precision.

We began by collecting our dataset of the test bench that we have created for the navigation of the demo test. The test data was collected from COCO dataset. The rest process is followed in the steps as follows:

A. Preparing dataset

COCO stands for Common Objects in Context. As hinted by the name, images in COCO dataset are taken from everyday scenes thus attaching "context" to the objects captured in the scenes. COCO is a large-scale object detection, segmentation, and captioning dataset. This version contains images, bounding boxes " and labels for the 2019 version, with a downloadable size of 19.57 GiB.

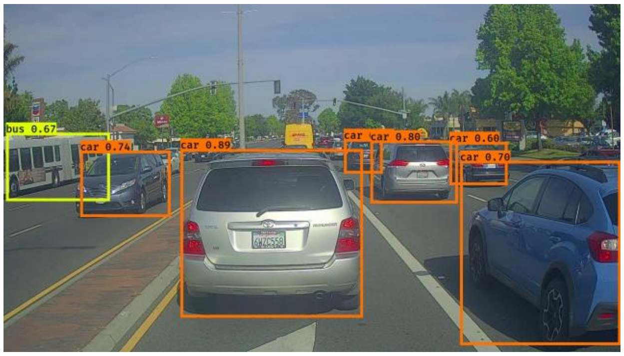

Fig. 1. Sample image showing lane detection

B. *Creating bounding box*

For creating the bounding box around the test images, the image's height, width and each class with parameters like xmin, xmax, ymin, ymax are required. The bounding box captures exactly the class of the object in the image. This follows the task of creating labels for the test images. Labels are created by using 'labelImg' tool. The labels are stored into individual xml label for each image which further need to be converted into csv file for training.

C. *Converting csv file into Tensorflow Record (TFRecord)*

For each training and testing dataset, a csv file is obtained which is further converted into TFRecord. The TFRecord is a format for storing the sequential structured data into binary strings.

D. *Selecting a model*

SSD model along with MobileNet neural network is selected as it provides moderate efficiency and the rate of result production is faster. The MobileNet is a light weight neural network as it consumes low processing power.

Fig. 2. Graph displaying relationship between the loss per step vs number of steps

E. *Retraining the model with data*

A file containing records of all the classes with their attributes is created and stored in the training directory. The configuration file for the selected model is executed such that the training of dataset starts showing the losses and checkpoints at step-wise.

F. *Generating Loss graph*

The proper working of the module can be estimated when the loss per step is under 3. The lower loss per step implies to greater accuracy. In our model the loss per step is 2.73. The loss per step decreases on increasing the number of steps thus ultimately increasing the number of images in the dataset.

3. Object detection task

Image classification involves assigning a class label to an image, whereas object localization involves drawing a bounding box around one or more objects in an image. Object detection is more challenging and combines these two tasks and draws a bounding box around each object of interest in the image and assigns them a class label. Together, all of these problems are referred to as object recognition.

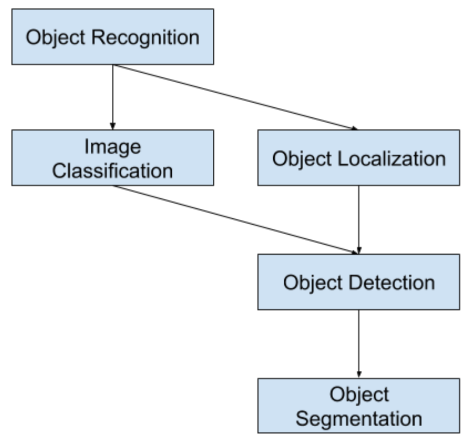

Fig. 3. Overview of object recognition computer vision tasks

Object recognition is refers to a collection of related tasks for identifying objects in digital photographs.

1) *Image Classification:* Predict the type or class of an object in an image.
   - *Input:* An image with a single object, such as a photograph.
   - *Output:* A class label (e.g. one or more integers that are mapped to class labels).
2) *Object Localization:* Locate the presence of objects in an image and indicate their location with a bounding box.
   - *Input:* An image with one or more objects, such as a photograph.
   - *Output:* One or more bounding boxes (e.g. defined by a point, width, and height).
3) *Object Detection:* Locate the presence of objects with a bounding box and types or classes of the located objects in an image.
   - *Input:* An image with one or more objects,

International Journal of Research in Engineering, Science and Management
Volume-3, Issue-4, April-2020
www.ijresm.com | ISSN (Online): 2581-5792

such as a photograph.

- *Output:* One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

Intuition of RCNN: Region-Based Convolutional Neural Networks, or R-CNNs, are a family of techniques for addressing object localization and recognition tasks, designed for model performance.

Instead of working on a massive number of regions, the RCNN algorithm proposes a bunch of boxes in the image and checks if any of these boxes contain any object. RCNN uses selective search to extract these boxes from an image (these boxes are called regions). Below is a summary of the steps followed in RCNN to detect objects:

1) We first take a pre-trained convolutional neural network.
2) Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected.
3) The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the CNN input size.
4) After getting the regions, we train SVM to classify objects and background. For each class, we train one binary SVM.
5) Finally, we train a linear regression model to generate tighter bounding boxes for each identified object in the image.

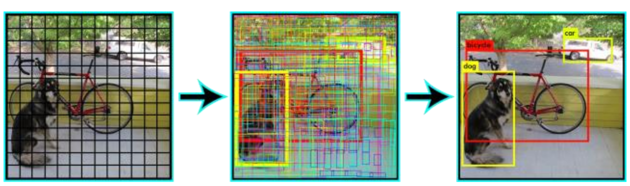

Fig. 4. Sample image processing of the model

You only look once object detection model: You Only Look Once, or YOLO, is a second family of techniques for object recognition designed for speed and real-time use.

Separate components of object detection have been combined in a single neural network. Every information in the image is used to predict the boundaries of each object. The image is first separated in an AxA matrix. The cell where the centre of the object falls in is responsible for identifying the object along with the confidence scores. These scores explain the confidence level of the model in identifying an object. Below is the architecture of the model,

The dataset used to train the model was the COCO dataset. The first 20 convolutional layers of the model were used that were followed by average-pooling layer and a fully connected layer.

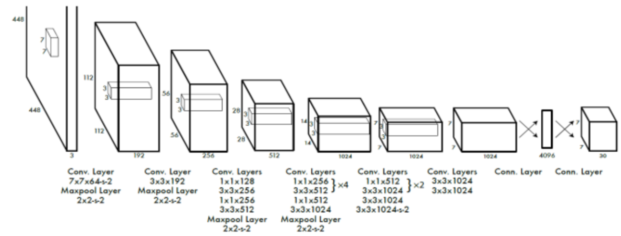

Fig. 5. Architecture of the model

4. Hough Transform

Most lanes are designed to be relatively straightforward not only as to encourage orderliness but also to make it easier for human drivers to steer vehicles with consistent speed. Therefore, our intuitive approach may be to first detect prominent straight lines in the camera feed through edge detection and feature extraction techniques. We will be using OpenCV, an open source library of computer vision algorithms, for implementation. The following diagram is an overview of our pipeline.

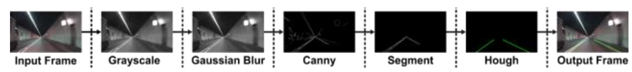

Fig. 6. Detection Pipeline

5. Conclusion

In this work we leveraged the task of object detection for self-driving vehicle by using TensorFlow API followed by MobileNet neural network. The efficiency in detection for objects is about 85.18%, which is above average, but the rate of result production is very fast. The loss per step or epoch is 2.73 (under 3) that supervises the reliability of the model. As for now we have tested the model on the dataset prepared from the test bench. The model works fine in identifying object in an image but for multiple objects in an image the bounding box shifts from one object to another inconsistently. Such inconsistency can be overcome by increasing the computation cost as well as dataset. We are planning to augment the model on the actual electric vehicle for performing object recognition and classification.

References

[1] Shetty, J., Jogi, P.S., Pandian, D., et al. (eds.): Study on Different Region Based Object Detection Models Applied to Live Video Stream and Images Using Deep Learning. Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB). Lecture Notes in Computational Vision, 2018.
[2] Fomin, I., Gromoshinskii, D., Bakhshiev, A., Kryzhanovsky, B., et al. (eds.): Advances in Neural Computation, Machine Learning, and Cognitive Research. SCI, vol. 736.
[3] Lin, C., Li, L., Luo, W., Kelvin, C.P., Wang, J.G.: Transfer learning based traffic sign recognition using inception-v3 model.
[4] Saha, S., Tairin, S., Khaled, M.A.B., Saha, S., et al.: An efficient traffic sign recognition approach using a novel deep neural network selection architecture. In: Proceedings of IEMIS 2018, vol. 3, 2018.
[5] Sudarshan, D. P., Raj, S, "Object recognition in images using convolutional neural network," in 2nd International Conference on Inventive Systems and Control (ICISC), 2018.