Modern Management based on Big Data II and Machine Learning and Intelligent Systems III
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210286

Object and Traffic Light Recognition Model Development Using Multi-GPU Architecture for Autonomous Bus

Jheanel ESTRADA [a, b 1], Gil OPINA JR [b] and Anshuman TRIPATHI [b]
[a] *Technological Institute of the Philippines-Manila, Philippines*
[b] *Energy Research Institute @ Nanyang Technological University, Singapore*

Abstract. The autonomous vehicle is both an exciting yet complex field to dig in these past few years. Many have ventured out to develop Level 4 Autonomous Vehicle but up to this point, many issues were still arising about its safety, perception and sensing capabilities, tracking, and localization. This paper aims to address the struggles of developing an acceptable model for object detection in real-time. Object detection is one of the challenging areas of autonomous vehicles due to the limitations of the camera, lidar, radar, and other sensors, especially during night-time. There were various datasets and models available, but the number of samples, the labels, the occlusions, and other factors may affect the performance of the dataset. To address the mentioned problem, this study has undergone a rigorous process of scene selection and imitation to deal with the imbalance dataset, applied the state-of-the-art YOLO architecture for the model development. After the development process, the model was deployed in a multi-GPU architecture that lessens the computational load on a single GPU structure and was tested on a 12-meter fully electric autonomous bus. This study will lead to the development of a usable and safe autonomous bus that will lead the future of public transportation.

Keywords. autonomous driving, autonomous vehicle, intelligent transport system, object detection, real-time, daytime, night time.

1. Introduction

Intelligent Transport System (ITS) is a combination of technologies, techniques, and processes that can improve public transit or the whole transportation system management [1]. The public sector has greatly been addressed as one of the predominant drivers of ITS. Early improvements show changes in the traffic management center in the urban areas, automated toll collection, traffic signal controls, and satellite-based dispatching systems. Recent improvements in ITS include autonomous vehicle which is a combination of multi-disciplinary fields such as robotics, embedded systems and circuits, sensors, machine learning, artificial intelligence, and others.

Due to the recent advances in these fields, a combination of highly calibrated cameras, lidars, and radars made this possible to monitor the vehicle and the things around its environment. Autonomous Vehicles (AV) are gradually becoming capable to mimic some human driver actions such as maneuver and defining routes. Recently, many

[1] Corresponding Author, Jheanel Estrada, Computer Science Department, Technological Institute of the Philippines, Manila, Philippines; E-mail: jheanelestrada29@gmail.com.

car companies announced that autonomous vehicles will be available to society in the upcoming years [2]. Due to its promising benefits, this attracts many stakeholders to invest in the automotive industry. This predicts that in the year between 2020 and 2025, self-driving cars will be seen on the road [3]. However, due to the complexity-ty and factors such as safety, legal issues, and social acceptance, technical issues, and its capability were questioned [4]. In a natural human driver setting in the road, according to the World Health Organization, nearly 1.25 million people die each year relative to traffic crashes and road accidents [5]. Now, autonomous vehicles' priority is to ensure the safety of both the driver and other road elements. An autonomous vehicle is expected to do the same level of safety as a human driver or even exceeding it. To achieve this, a higher level of sensing capabilities, perception, decision, motion, and mission planning is required.

As part of this global mission to improve the trust and safety of autonomous vehicles, this study aims to develop an acceptable object detection and recognition model by developing an acceptable dataset that covers day time, night time and rain conditions; to deploy the model in a real-time manner that prevents huge frame loss and lastly to utilize a multi-GPU architecture.

2. Related Literature and Studies

To stay in-sync with ongoing research and innovation regarding object detection and recognition, a literature survey was deployed investigating the present-day movements in this field. Image processing, machine learning and map-based techniques are some of the identified methods for this task. The most classic way of this task is through image processing, however, though image processing approach is quite straight and uncomplicated compared to other tasks, it requires critical phases such as thresholding and filtering. One of the undesirable conditions in object detection is the fact that even slight miscalculations or deviations from these standard techniques may result to ambiguous outcomes which is highly sensitive to this task. To somehow addressed this problem, machine learning based methods in combination with some algorithms and ample processing techniques will be helpful to prune the misleading directions.

Object Detection is one of the crucial capabilities of an autonomous vehicle together with the adaption of Deep Neural Network (DNN). All the sensors available in the use of AV are irreplaceable especially cameras since they gather widely usable data or texture information regarding an object in the surroundings of an AV. In literature, there is a wide range of studies about the use of cameras and DNN in the object detection process.

To date, object detection using Deep Neural Network (DNN) can be divided into two (2) strategies namely one-stage detection which is Single Shot Detection (SSD) [6] and You Only Live Once (YOLO) [7], and two-stage detection which includes Spatial Pyramid Pooling (SPP) [8] and Region Convolutional Neural Network (R-CNN) [9]. One stage detection mainly focuses on higher-speed object detection and works on a real-time basis. However, when it comes to high-precision object detection, this type of detection shows a slight disadvantage since it gives out regression detection on the pixel feature map of the CNN and gives the classification and detection results. However, in two-stage detection, aside from obtaining the feature map of an image, it generates

proposals through an RPN through a Region-of-Interest (RoI) then gives the detection and classification result [10].

After identifying the type of deep neural network, the next crucial step is to identify the dataset. Today, numerous datasets were available in the market. Each dataset offers advantages over the other in different terms such as the number of classes, labeling techniques, and others. This includes the following:

- Pattern Analysis, Statistical Modelling, and Computational Learning Visual Object Classes (PASCAL VOC) [11];
- Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago Object Detections (KITTI) [12];
- Common Object in Context (COCO) [13];
- and ASTAR3D Dataset that contains 3X the size of nuScenes dataset [13].

One of the crucial steps to obtain optimized results is the collection of large training and testing datasets and the significant amount of time and resources needed to train a good and acceptable model. Moreover, to address the imbalance samples due to inadequate amount of data collected and to provide solutions for image recognition problems, this study will be able to tries to experiment and conduct hardware-based and software-based solutions.

3. Methodology

A good object detection model relies on the preparation of dataset and pre-processing methods. Our dataset is comprised of different scenarios taken to imitate the real-world environment. Once the dataset was completed and all the scenarios were generated and imitated, data augmentation processes come in. In the figure 1 below, the over-all process starts with data preparation which includes scenario identification, data gathering, and data analytics and statistics. For scenario identification, risky situations that involve pedestrian and other objects were imitated in a real-world scenario. The examples are the following: pedestrian on the side of the road, pedestrian crossing the road, a pedestrian on the side of the road that does not have the intention to cross, overtaking vehicles, and many more. These scenarios were imitated, staged, and recorded for both daytime and night time. Aside from these scenarios, other common and usual scenarios on the road were recorded. After data gathering, the data that were collected will then be analyzed in terms of statistics and usability. At this point, we are ensuring that all the classes have the same portion of samples. Afterward, the dataset will be pre-processed by region cropping. Region Cropping is a technique we applied to get the usable part of the frame eliminating the top and bottom part of the image. This method is important to lessen the processing time taken for each image since not all the image part is beneficial in the recognition process (see Figure 1 below).

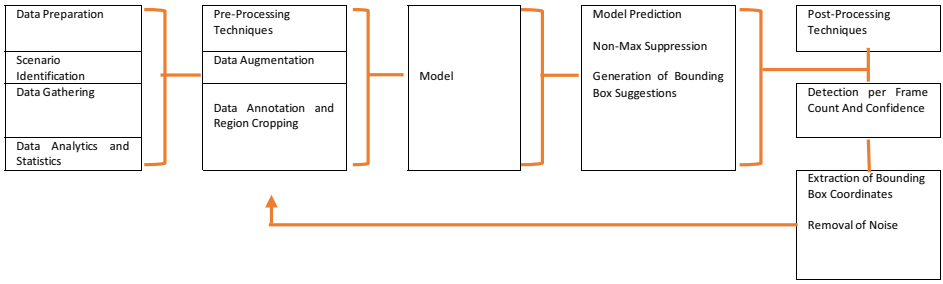

Figure 1. Process Diagram from Dataset Preparation to Post-Processing

3.1. Dataset

The dataset for this study is comprised of seventeen (17) labels (see Figure 1). The distribution of these labels was also shown in Figure 2. A total of images and approximately 700,000 annotations across labels. The strength of this dataset compared to other famous datasets is it consists of high-density images, heavy occlusions, and a large number of night-time images.

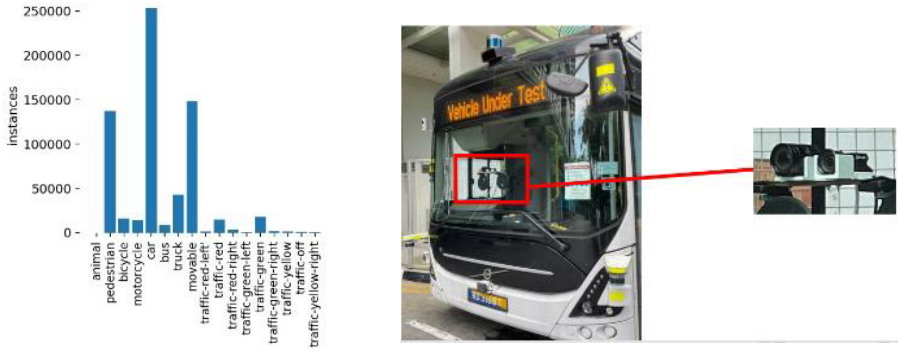

Figure 2. Distribution of Samples and Camera Setup

The dataset was developed using 6x FLIR Blackfly BFLY-PGE-31S4M-C mounted on the front side of the bus (see Figure 2 right image). The recording took for a month during both day time and night time along Nanyang Technological University Route. Using Robotic Operating System (ROS), these recorded were stored in a rosbag and extracted ten(10) frames per second. These extracted frames were manually annotated using labelImg tool.

3.2. Darknet Framework

Darknet is an open source framework to train neural networks. It is a framework used to train YOLO networks compatible with Robotics Operating System (ROS) for self-driving cars. YOLO is a powerful neural network in classifying an object using a bounding box around the detected object. It can classify the object in a single pass compared to other implementations which are why it is fast, and performant compared

to other implementations undergo several tasks to classify. Additionally, YOLO works on a real-time basis which is relevant in identifying and recognizing objects on a real-world scenario.

YOLO works by putting and splitting an image into n grid cells (usually 19x19). For each cell that represents a certain part of an object, there will be predicted bounding boxes, confidence scores, and class probabilities. The confidence is calculated using an IOU (intersection over union) metric that measures the overlapping of detected objects with the ground truth as a fraction of the total area of detection. There were many publicly available versions of YOLO. The YOLOv1 was released May 2016, YOLOv2 was released December of the same year, YOLOv3 was released April 2018, YOLOv4 was released May 2020. All versions offer strength on the time they were released. As shown in Figure 3 below, YOLOv3 uses a variant of Darknet which has 53 layers. Because of the task of detection, 53 layers were added on top of the original number of layers that sums up to 106 layers fully convolutional. Aside from these advantages of YOLOv3, we conducted an experiment on our dataset which uses both YOLOv3 and YOLOv4 and it shows that YOLOv3 has a higher mAP (mean average precision) than YOLOv4.

YOLO aims to predict a class of an object in an image by the use of a bounding box (see Figure 3 below). Each bounding box has four (4) descriptors namely:

Center of bounding box (bxby) or (x, y)
Width (bw)
Height (bh)
C = class or label

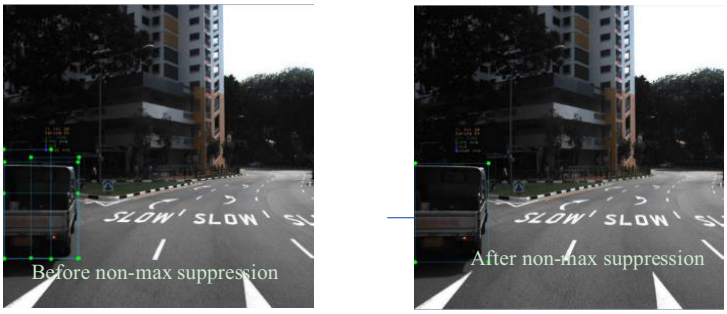

Figure 3. YOLO Bounding Box

YOLO is predicting that there is an object in the image instead of searching for regions of interest. It is splitting an image S x S grid cells. Each cell is responsible for predicting n number of bounding boxes. Each grid cell predicts a bounding box alongside with a confidence value. If a grid cell does not contain a bounding box, its confidence value must be zero. Most of these cells do not contain the object, therefore YOLO will predict a value that will remove boxes with low object probability and bounding boxes with highest. This process is called non-max suppression. If the center of an object falls into a grid cell, that cell is the main responsible for the detection.

Since YOLOv3 is a fully convolutional network, it applies 1x1 kernel on a feature map of three different sizes at three different places in the network. It follows the:

1 x 1 x (B x (5+C))

Where:
B = number of bounding box a cell on the feature map can predict
C = number of classes

In the case of this study, we used B=3 and C=17, so the kernel size is 1 x 1 x 66.

4. Results and Discussion

For training purposes, the study used PyTorch in multi-GPU architecture. This dataset used 70-30 distrubution. Seventy (70) percent of the dataset was used for training and the remaining thirty (30) percent was used for validation set. The training and validation were deployed using TITAN PC with 2 GPUs available. The list of classes and actual data was listed below.

```
# number of classes
nc: 17

# class names
names: ['animal', 'pedestrian', 'bicycle', 'motorcycle', 'car', 'bus', 'truck', 'movable',traffic-red-left', 'traffic-red',
'traffic-red-right', 'traffic-green-left', 'traffic-green', 'traffic-green-right', 'traffic-yellow', 'traffic-off', 'traffic-
yellow-right']
```

Using a pre-trained model at the beginning, we started to train the dataset using 2 GPUs.

After a series of iterations, the training stops when it meets certain criteria as follows:

- Sufficient iterations of at least 2000 for each class or at least less than the number of training images; and
- When the average loss no longer decreases (the lower, the better).

The study stopped at 25700 iterations and generated weights file indicated in the path written on the data file. Usually, YOLO generates the best and last weight files which show the highest mAp (see Figure 4). As seen in the Figures 4, shows f1 curve at 0.81 at 0.394 and all the classes stated in the confusion matrix shows acceptable kappa percentages.

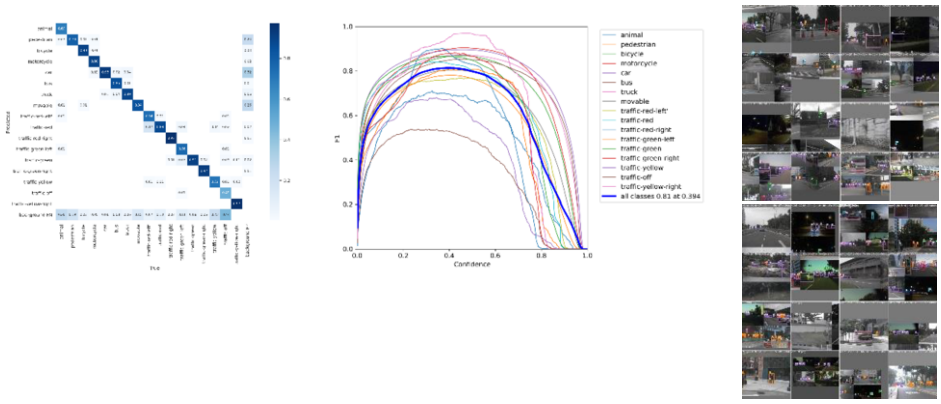

Figure 4. F1 Curve and Confusion Matrix

This model was initially tested offline using recorded mp4 files and we found out some issues which arise because of the possible reasons (see Figure 5):

- imbalance number of samples for objects and traffic lights; and
- imbalance number of samples for daytime and night time.

Figure 5. Inconsistent Traffic Light Recognition

Based on the Figure 5, left image shows only 2 (two) traffic light signals and instead of four (4). Additionally, one of the two (2) detected traffic light signal should be green right (see the 3rd red circle), it only detected green. Another issue arise in the right image which shows the bursting of the lights during night time although the model is capable of some of it but the green right was not detected (see the red circle) since it is too close to the green traffic light signal.

To solve this, we deploy a calibration file and a long-range camera dedicated to the traffic light signals. The calibration file will adjust the parameters of the camera such as exposure, frames per second, and aperture for both daytime and night time (see the results below).

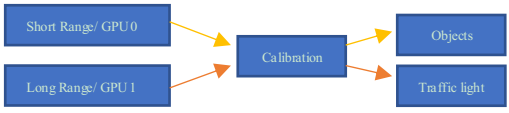

Figure 6. A more reliable and efficient traffic light recognition

Figure 6 shows the process we deploy to minimize the inconsistency that we encountered on the recognition. The two (2) cameras which deployed the same model has a dedicated GPU ID and a calibration file. The short range camera will be dedicated to react to objects only while the long range camera will be allotted to react to traffic light signals only.

Since an acceptable model was generated with an mAP of ~90%, this model will then be integrated into the Autonomous Bus using ROS (Robotic Operating System). For this a vision_darknet_detect node was utilized. The model was implemented in a 50m Autonomous Bus within Nanyang Technological University route with a Nuovo PC 6108GC. The two (2) cameras as mentioned earlier, each of them is connected to a dedicated GPU using an ID. A GPU ID is dedicated for each camera during runtime. Once an object or traffic light signal was detected, the bus will depend its movement on the confidence level and the coordinates of the bounding box on the frame and the

consistency of the recognitions based on the number of frames. This follows the algorithm below:

Object Recognition

detect objects
recognize object
- *check confidence level*
- *check coordinates (calculate the distance)*
- *check consistency of the frames*

Traffic Light Recognition

detect traffic light signals
recognize traffic light signals
- *check confidence level - > check confusion matrix*
- *check conditions for each road structure - > check the route of the bus*
- *check coordinates - > calculate for the coordinates on the frame*
- *check consistency of the frames*

Figure 7. Object and Traffic Light Recognition Algorithm

Since the bus route is very complex which comprises of one (1) roundabout, three (3) zebra crossings and five (5) bus stops and one (1) traffic light signals in the junction (see Figure 8 below). In the case of the junction, it must detect the greenright traffic signal from afar since the bus must be on the rightmost lane before approaching the junction.

Figure 8. Object and Traffic Light Recognition Algorithm

As the figure 8 shows from the leftmost image, the bus using a long-range camera and following the bus designated route, it detects and recognize the traffic light signal from afar and goes into the rightmost lane (the route is to turn right at the junction). While it is still at redright, the bus will stop before the allotted stopping line and continues to detect the traffic light signal and when it satisfies the conditions set for confidence level, and the consistent detection of frames, once it recognizes greenlight, the bus will turn right at the junction.

5. Conclusion

This study was able to develop an acceptable model and deploy it on 50meter Autonomous Bus operating in day time and night time conditions. It was able to recognize objects in a real-time basis and generate a command to the bus. Though this study produced an acceptable and usable result, this study recommends to do sensor fusion incorporating lidar data together with the camera data for a higher safety consideration.

References

[1] Shaheen S, Finson R. Intelligent Transportation Systems. 2013 10.1016/B978-0-12-409548-9.01108-8.
[2] Abbeel P, Ng AY. Apprenticeship learning via inverse reinforcement learning. In Proceedings of ICML, 2004.
[3] Business Insider News: http://www.businessinsider.com/companiesmaking-driverless-cars-by-2020-2017-1/#tesla-recently-made-a-bigmove-to-meet-its-goal-of-having-a-fully-self-driving-car-ready-by-2018-1 - accessed on March 24th, 2017.
[4] Campbell K, Shia V, Bajcsy R. Decisions for autonomous vehicles: integrating sensors, communication and control. HiCoNS'14, April 15–17, 2014, Berlin, Germany. ACM 978-1-4503-2652-0/14/04.http://dx.doi.org/10.1145/2566468.2576851.
[5] World Health Organization Website:<http://www.who.int/mediacentre/factsheets/fs358/en> - accessed on March 10th.
[6] Liu W, Anguelov D et al. "ssd: Single shot multibox detector," In ECCV, 2016.
[7] Redmon J, Farhadi A "Yolov3: An incremental improvement,"arXiv:1804.02767 [cs.CV], 8 Apr 2018.
[8] He K, Zhang X et al., "Spatial pyramid pooling in deep convolutional networks for visual recognition," In CVPR, 2014
[9] Girshick R "Fast r-cnn," In ICCV, 2015.
[10] Ren S, He K et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2017.
[11] Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes challenge: A retrospective," International J. Comp. Vision, 2015; 111(1): pp. 98–136.
[12] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite, in CVPR, 2012
[13] Lin TY, Maire M, Belongie S et al. Microsoft COCO: Common objects in context. in European Conference on Computer Vision, Oral, Jan. 1, 2014.