RBE/CS549: Project 3
Einstein Vision

Using 1 late day

Prasanna Natu
M.S. Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
pvnatu@wpi.edu

Peter Dentch
M.S. Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
pdentch@wpi.edu

***Abstract*—This project presents Visualisation as a Human-Robot Interaction (HRI). The goal of Tesla's technology is to improve quality of life by making it easier and more efficient with every update to go around. At the same time, the key component to any product which a human has to interact with is beautiful visualisations for building a trust-worthy autonomous machine. In this project, a series of data has been used for object identification and depth detection. Multiple networks including Faster R-CNN, YOLOV5x, and Yolo3D have been used for object identification. Whereas, the Midas depth identification model is used for depth detection. This project highlights the approach towards the problem, the effectiveness and the prettiness of the visualisations.**

***Index Terms*—Faster R-CNN, YOLOV5x, YOLOPv2, YOLO3D, HRI, Midas**

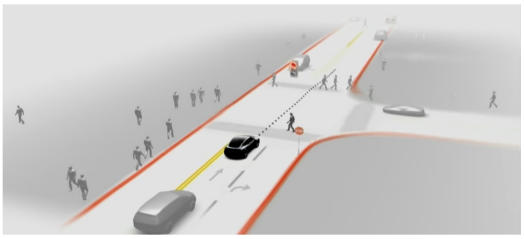

Fig. 1. Tesla's Frontend Visualisation in Autonomous Mode

I. Introduction

A. Object Detection

Object detection is the process of identifying objects in an image or video frame. The images or videos captured by the camera are processed by a neural network that is trained to identify and classify objects in the scene. The neural network is trained on a large dataset of labelled images that includes a variety of objects and lighting conditions.

This is the general structure for object detection:
Input - Backbone - Neck - Head - Output

- Backbone: refers to the feature extracting network, which is used to recognize several objects in a single image and provides rich features on the information of objects.
- Detection Head (head): After the feature extract, it gives us a feature map representation of the input.
- Neck: The neck is between the backbone and head, it is used to extract some more elaborate features.

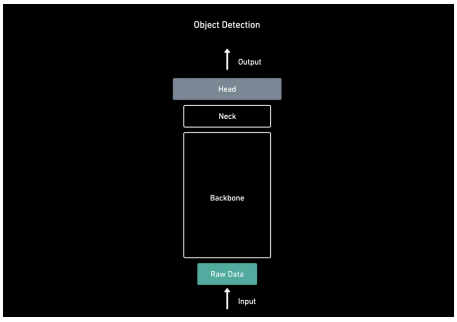

Fig. 2. Object Detection Structure

The various networks used for Object detection are as follows:

- Faster R-CNN:
  The Faster R-CNN architecture consists of the RPN as a region proposal algorithm and the Fast R-CNN as a detector network.

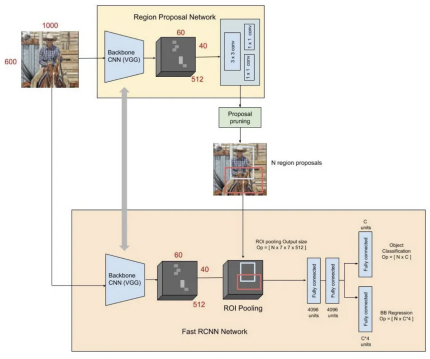

Fig. 3. The RPN for region proposals and Fast R-CNN as a detector in the Faster R-CNN detection pipeline

Following are the summarised steps followed by a Faster R-CNN algorithm to detect objects in an image or a video:

- Take an input image and pass it to the ConvNet which returns feature maps for the image.
- Apply Region Proposal Network (RPN) on these feature maps are get object proposals.
- Apply ROI pooling layer to bring down all the proposals to the same size.
- Finally, pass these proposals to a fully connected layer in order to classify any predict the bounding boxes for the objects.

- YOLOV5x
  YOLO is short for You Only Live Once. It is a family of single-stage deep learning-object based detectors. They are more capable of more than real-time object detection with state-of-the-art accuracy.

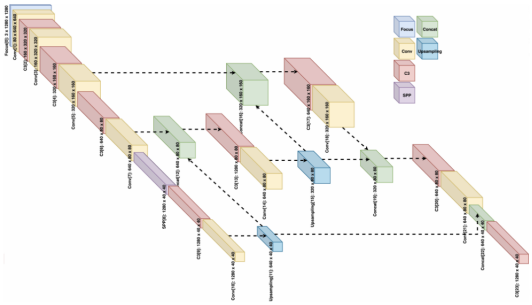

Fig. 4. Architecture of YOLOV5x

It is the largest among the five models and has the highest mAP among the 5 models of YOLOV5. Although it is slower compared to the others and has 86.7 million parameters.

- YOLO3D The complex-YOLO network takes a bird-eye-view RGB-map as input. It uses a YOLO CNN architecture to detect the 3D objects in real-time. The translation from 2D to 3D is done by a predefined height based on each object class.

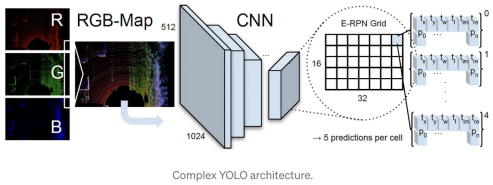

Fig. 5. Architecture of YOLOV5x

The YOLO Network divides the image into a grid (16 X 32) in this case and then, for each grid cell, predicts 75 features.

B. *Depth Identification*

Depth estimation of an image predicts the order of objects from the 2D image itself. It is used to estimate the distance of objects around a car helping in navigation.

- MiDas Depth Estimation:
  MiDas is a machine learning model that estimates depth from an arbitrary input image.
- MiDas uses multiple datasets for training.
- Therefore, it can estimate the depth of images in various conditions and environments.

The architecture of the network is based on ResNet.

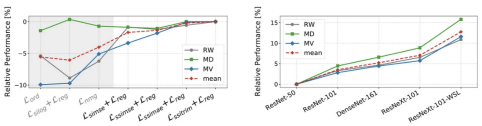

Fig. 6. Architecture of MiDas Depth Detection

Fig. 7. Output of MiDas Depth Detection

II. Checkpoint 1

Implementation of basic features for a self-driving car.

- Lanes - The lane detection is a computer vision task that involves identifying the boundaries of driving lanes in a video or image of a road scene.
  The goal is to accurately locate and track the lane markings in real-time, even in challenging conditions such as poor lighting, glare or complex road layouts. It provides information about the road layout and the position of the vehicle within the lane, which is crucial of navigation and safety.
  The image below shows Identifying and showing the different kinds of lanes on the road.
  Output for Lane Detection is given below: The network used was YOLOPv2 which was specifically trained on road lane data with provided ground truth. The results are the best we found and work well in different lighting or weather conditions.
- Vehicles The objective is to identify vehicle from a Tesla 5's dashboard video.

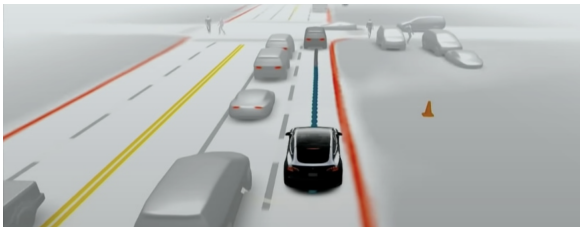

Fig. 8. Lane detection

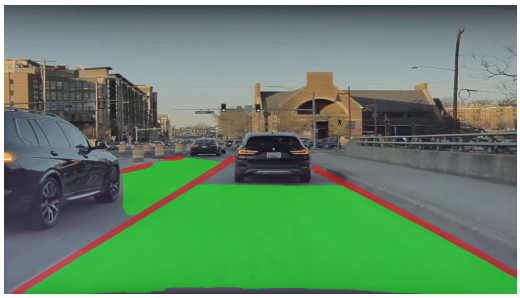

Fig. 9. Output of Lane detection

The process of detecting the vehicles on the road is summed up in the following steps:
The outputs of vehicle detection are shown below:

- Pedestrians Pedestrian detection or person detection in streets and footpaths is an essential part.
- Traffic Lights Detecting and recognising the status of traffic lights is one of the most important applications and of great significance.
- Road Signs Detecting Traffic Signs in real time is the building blocks of automated cars.

III. Checkpoint 2

Checkpoint 2 involved detecting objects in the scene such as dustbins, traffic poles, and traffic cones. A network specifically trained on traffic cone data was used to detect cones. An image with bounding boxes detecting cones and their rendered output in Blender is shown.

IV. Problem faced and Conclusion

The main issue we faced while is the exact depth identification and getting the mispredicted output to filter out. There were scenario in which the car was detected both as car and truck. The pedestrian are well placed but sometimes even they are getting misplaced. The another main task was getting the pose of the car. We found some networks working on getting human pose but not on getting the car orientation. Also most of the models for the traffic sign identification were trained on Europian dataset .There were some models trained on United States dataset but they were glitchy and working at some instances and not on other. A model was found which only detect crosswalk sign at very close distance. yolo works better than fasterRCNN if you want to avoid false prediction for the same output.

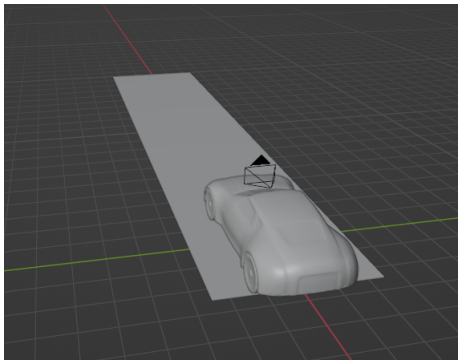

Fig. 10. 3D output of Lane detection

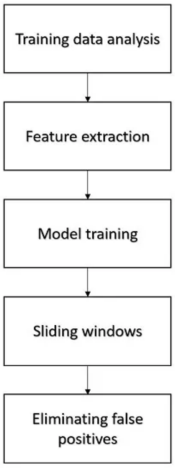

Fig. 11. Vehicle Detection model

References

[1] URL:https://saneryee-studio.medium.com/deep-understanding-tesla-fsd-part-1-hydranet-1b46106d57
[2] URL:https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46
[3] URLhttps://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/
[4] Dadboud, F., Patel, V., Mehta, V., Bolic, M., Mantegh, I. (2021). Single-Stage UAV Detection and Classification with YOLOV5: Mosaic Data Augmentation and PANet. 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). https://doi.org/10.1109/avss52988.2021.9663841/
[5] URL:https://learnopencv.com/custom-object-detection-training-using-yolov5/Models-Available-in-YOLOv5
[6] URL:https://becominghuman.ai/complex-yolo-3d-point-clouds-bounding-box-detection-and-tracking-pointnet-pointnet-lasernet-62e4fc2b6938
[7] URL:https://pyimagesearch.com/2022/01/17/torch-hub-series-5-midas-model-on-depth-estimation/
[8] URL:https://medium.com/axinc-ai/midas-a-machine-learning-model-for-depth-estimation-e96119cc1a3c
[9] URL:https://paperswithcode.com/task/lane-detection
[10] https://towardsdatascience.com/vehicle-detection-and-tracking-using-machine-learning-and-hog-f4a8995fc30a
[11] https://data-flair.training/blogs/pedestrian-detection-python-opencv/

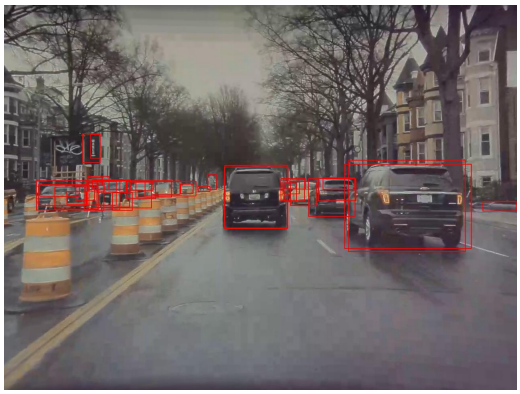

Fig. 12. Vehicle detection

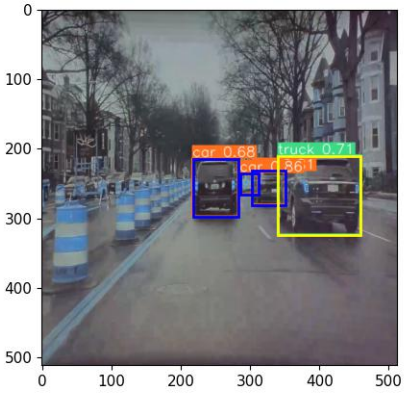

Fig. 13. Vehicle Detection using YOLO

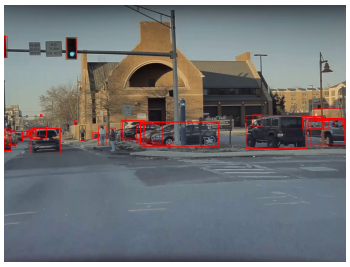

Fig. 14. Pedestrians and Traffic Light detected

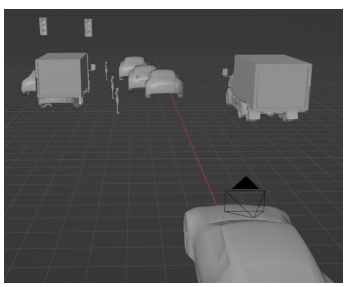

Fig. 15. Pedestrians and Traffic light Rendered

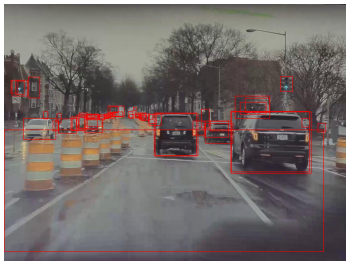

Fig. 16. Traffic Lights

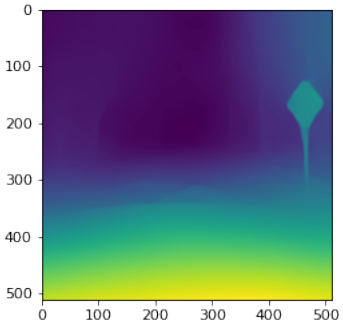

Fig. 17. Road Signs MiDas Depth

Fig. 18. Rendered traffic cones

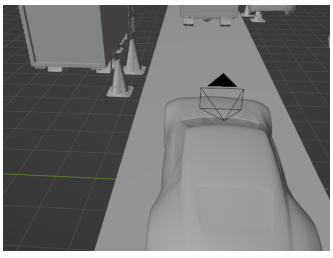

Fig. 19. Rendered traffic cones