International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 04 | April -2020 ISSN: 2582-3930

Self Driving Car

Ashutosh Priyadarshi
Department of Information Science and Engineering
The National Institute of Engineering
Mysore-570008

Supriya M
Department of Information Science and Engineering
The National Institute of Engineering
Mysore-570008

Abstract—This paper would present our research in the area of image processing and machine learning to implement a self driving car. The features we would like to investigate are collision avoidance, map navigation, and best route detection. To investigate the practical feasibility we would use Nvidia Jetson nano, logitech webcam, L289n motor driver with set of tt motors and wheels to create the prototype. The amount of real time computation as well as its feasibility on small GPU will also be analyzed. In this paper we would also investigate collision rate, efficiency, and success rate based on appropriate metrics.

Keywords—self driving car; collision avoidance; Nvidia Jetson nano; L289n motor driver.

1. Introduction

The technology behind image classifications has im-proved leap and bounds in the last few years. With the current technology of neural networks and stronger GPU, making predictions from a given image has not just become faster but also more accurate. Which has led to the advance-ment of new products which were simply not possible a few years ago, one of the major example of that is self-driving cars.
In this paper, we would be analyzing the feasibility of such machines using the latest machine learning algorithms and would be using a Nvidia jetson nano chipset to conduct experiments to find out real-world feasibility.
The main intention behind this paper is to find out whether we can make a prototype of a self-driving vehicle that can function in a satisfactory way with moderate speed and note its shortcomings and make a decision about its feasibility in the real world.

2. Methods and materials

For training our image classification model we are using fastai library. Fastai's convolution neural network models are builded upon TensorFlow's model and consistently provide world-class results in various problems and models.

To run our model and make the predictions in real-time we would be using Nvidia jetson nano chipset.NVIDIA Jetson Nan Developer Kit is a small, powerful computer that lets us run multiple neural networks in parallel for applications like image classification, object detection, seg-mentation, and speech processing. The Nvidia Jetson nano has 128 core maxwell GPU and Quad-core ARM A57 @ 1.43 GHz as CPU.

For taking the image we are using raspberry pi camera version 2.

To run our prototype we are using a pair of tt motor, and l289 motor driver to control them.

Figure 1. The Nvidia Jetson nano

3. Datasets

For training our model we are using Camvid: Motion-based Segmentation and Recognition Dataset(CamVid). This dataset contains. The CamVid is the first collection of videos with object class semantic labels, complete with metadata. The data in CamVid are captured from the perspective of a driving automobile. The driving scenario increases the number and heterogeneity of the observed object classes.

For each image in the data in the training subset of the CamVid dataset there is also available mask for every pixel present. The pixel values in the mask varies between 1-32, each being a code for the classes of objects available in the image.

The available classes in CamVid are: 1. Animal 2. Pedestrian 3. Child 4. Rolling cart/luggage/pram 5. Bicyclist 6. Motorcycle/scooter 7. Car (sedan/wagon) 8. SUV / pickup truck 9. Truck/bus 10. Train 11. Misc 12. Road == drivable surface 13. Shoulder 14. Lane markings drivable 15. Non-Drivable 16. Sky 17. Tunnel 18. Archway 19. Building 20. Wall 21. Tree 22. Vegetation misc. 23. Fence 24. Sidewalk 25. Parking block 26. Column/pole 27. Traffic cone 28. Bridge 29. Sign / symbol 30. Misc text 31. Traffic light 32. Other

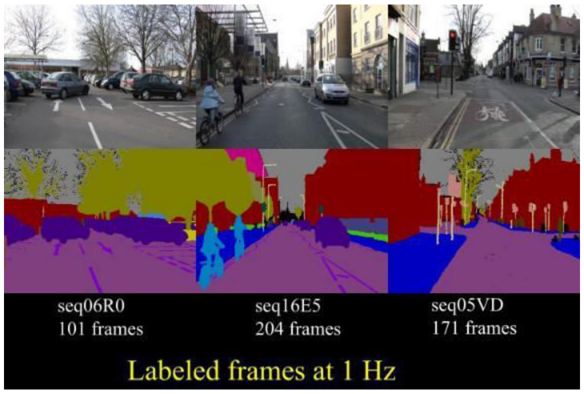
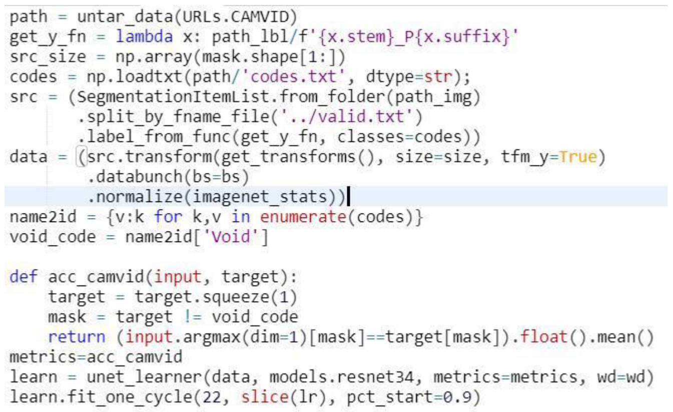


Figure 2. The images in the CamVid dataset along with their mask representation.

4. Data and results

The CamVid dataset were first arranged into the databunch by using datablock api of fastai. Code:



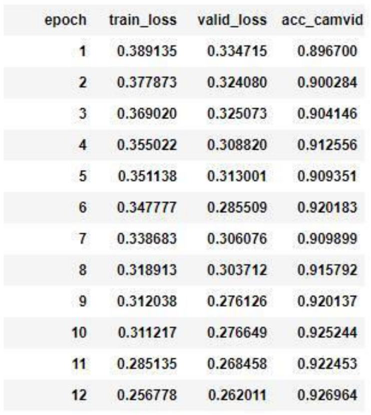
```
path = untar_data(URLs.CAMVID)
get_y_fn = lambda x: path_lbl/f'{x.stem}_P{x.suffix}'
src_size = np.array(mask.shape[1:])
codes = np.loadtxt(path/'codes.txt', dtype=str);
src = (SegmentationItemList.from_folder(path_img)
       .split_by_fname_file('../valid.txt')
       .label_from_func(get_y_fn, classes=codes))
data = (src.transform(get_transforms(), size=size, tfm_y=True)
        .databunch(bs=bs)
        .normalize(imagenet_stats))
name2id = {v:k for k,v in enumerate(codes)}
void_code = name2id['Void']

def acc_camvid(input, target):
    target = target.squeeze(1)
    mask = target != void_code
    return (input.argmax(dim=1)[mask]==target[mask]).float().mean()
metrics=acc_camvid
learn = unet_learner(data, models.resnet34, metrics=metrics, wd=wd)
learn.fit_one_cycle(22, slice(lr), pct_start=0.9)
```

Figure 3. Source code

After running our CNN for 24 epochs we got an accuracy of about 92percentage. From this model we can predict whether there is a drivable road in front of our vehicle.

5. Results and discussion

The inference rate for the nvidia jetson nano, for the unet model we used in our learner, was around 10 fps. If we look in real life situations it is low and would be prone to accidents. But it is satisfactory enough for other usecases were there is not a lot of risks involved.



epoch	train_loss	valid_loss	acc_camvid
1	0.389135	0.334715	0.896700
2	0.377873	0.324080	0.900284
3	0.369020	0.325073	0.904146
4	0.355022	0.308820	0.912556
5	0.351138	0.313001	0.909351
6	0.347777	0.285509	0.920183
7	0.338683	0.306076	0.909899
8	0.318913	0.303712	0.915792
9	0.312038	0.276126	0.920137
10	0.311217	0.276649	0.925244
11	0.285135	0.268458	0.922453
12	0.256778	0.262011	0.926964

Figure 4. Train loss , validation loss as well as accuracy in the last 12 epochs of our CNN model.

6. Conclusions

In this paper we tried to test a feasibility of a self-driving vehicle using affordable and comparatively weak hardware. Our finding showed that we can actually achieve a self driving which would work on satisfactory speed. In future we would like to improve our model by utilizing the gpu providied by the nvidia jetson nano to the fullest.

References

[1] Nvidia jetson nano chipset https://www.nvidia.com/en-in/autonomous-machines/embedded-systems/jetson-nano/

[2] Fastai's convolution neural network model https://docs.fast.ai/

[3] Camvid motion based segementation based recognition dataset http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid

[4] The inference rate for the nvidia jetson nano https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks