CSC420 Project Proposal

Project Summary

Group Members

Yaolong Wang
Jiahuang Lin
Zilun Zhang

Project And Goals

We are going to use sample project 4, we will skip some detail since most of it is provided under the sample project file. Autonomous driving is one of the major research venues these days, and we are doing this project to learn how to build a 3D model and using deep learning to detect objects such as road, other cars and pedestrian. Then do some further operation on the model we get.

This problem is important because 3d-model, object detection and accident estimation are basic aspects of auto-driving. Furthermore, it is the problem that we can try to solve by using what we studied. Maybe solution is simple, but it is a good practice, a great learning process (especially when reading the paper) and it could enhance our understanding of knowledge we studied so far.

This kind of problem has been done before, absolutely. So far, the best method for detecting car is iDST-VC, the best method for detecting car is SWC and the best method for detecting car in urban marked road is UNV.

Final Outcome

Our final outcome will be a video labeled with all the cars and pedestrian. For pedestrians we label them in different colors, indicating different dangerous level. For example, the pedestrians who are not on the lane are labeled in yellow, while if a pedestrian is right on the road, we say they might be in danger and should be labeled in black, if the pedestrian become too dangerous (say he/she will be hit under car's current track and speed), it will be labeled as red.

Cars are similar with pedestrians but model will be more complex in further operation and simpler in classification.

Paper

We need to read some paper, mainly about models:

Yamaguchi, Koichiro, David McAllester, and Raquel Urtasun. "Efficient joint segmentation, occlusion labeling, stereo and flow estimation." European Confe- rence on Computer Vision. Springer International Publishing, 2014 Paper and code.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." eprint arXiv:1506.01497

Kaiming He, Georgia Gkioxari, Piotr Dollr, Ross Girshick. "Mask R-CNN." eprint arXiv:1703.06870

We might also use some other paper in the process of doing project

Datasets

The data used in this project is provided by the KITTI Vision Benchmarking Suite[1] which is widely used in robotics for testing machine learning algorithms. The Road/Lane Detection Evaluation (2013) dataset contains images (with individual pixel color intensity) of urban scenes taken from the top of a vehicle, with ground truth labels for roads. There are three different scene categories: urban unmarked roads (uu), urban marked roads (um) and urban roads with multiple marked lanes (umm).
If KITTI doesn't work well, we may use Cityscapes dataset.

Overall

0. 3D Model Construction

Explained in handout.

1. Road Detection

To simplify the task, we could detect road by first detect lane, since lane is the boundary of road and it's easier to be recognized. We will tackle this part using neural network. The first option is CNN, which is generally used in lane detection. If it doesn't work well, we might try on some other types of neural network.

2. Car Detection

Because of the similarity of car detection and pedstrian detection, we will try the similar network structure first (i.e. CNN based). If it doesn't work well, we may combine them by using one neural network with different classfier in the end of the structure to output bounding box and probability such as faster RCNN. An important difference between Car detection and pedestrian detection is the different assumption. To simplify the model, we ignore the pedestrian's speed, but we cannot make this assumption for Car detection because car's speed is a lot faster than prdestrian. We will detect the most (potentially) dangerous car on the road and mark it.

3. Pedestrian Detection

In this section, we will detected pedestrians by trying to use CNN-based neural network first. If the result doesn't satisfy us, we will try to use other type of neural networks such as faster RCNN or mask RCNN. After detect the pedestrians, we will classified those pedstrians into 3 classes. The most closest one will be in a black box with imaginary line, it is the most dangerous one. Ones that on the road will in a yellow box with imaginary line. Others will in a white box with imaginary line. After that we will calculate if the car in current speed will hit the most dangerous person. This will be achieved by calculating distance between car and the most closest pedestrian and current speed of car (displacement (z axies displacement in real world coordinate metioned above) * fps). If result is yes, we will change the bounding box from black imaginary line to red full line.

[1] `http://www.cvlibs.net/datasets/kitti/`

Work Distribution

Yaolong will work on part 0, 3D model construction, and part 1, road detection. Jiahuang will work on car detection while zilun will work on pedestrian detection.

Milestones

1. The First Week

Learn and get up to speed with literature
Find a few articles that address this problem
Set up environment
Find some source code and learn how to build a neural network

2. The Second Week(Reading Break)

Finish all the parts that is not using neural network (speed calculation, 3D model construction)
Build the neural network
Train the data

3. The Third Week

Continue working on the the neural network.
Try to optimize the outcome.
Debug the project.

4. The Last Week

Further debugging and Write report.

Expansion

Since we have three people in the group, we will expand the given sample project by adding a pedestrian detection part and building an independent neural network for it (Or change the model, then combine the car detection part and pedestrian detection part into one network structure).