# Self-Driving Car Expert System

Shriyash Shingare[1],Ranjeet Jagtap[2], Prasad Rawal[3],Sohail Kazi[4]

*Abstract*- **Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make machines "smart”. At present we use the term AI for successfully understanding human speech, competing at a high level in strategic game systems (such as Dota2 and Alpha Go), self-driving cars, and interpreting complex data. Some people also consider AI a danger to humanity if it progresses unabatedly. An extreme goal of AI research is to create computer programs that can learn, solve problems, and think logically. In practice, however, most applications have picked on problems which computers can do well. Searching data bases and doing calculations are things computers do better than people. On the other hand, "perceiving its environment" in any real sense is way beyond present-day computing. In the modern era, the vehicles are focused to be automated to give human driver relaxed driving. In the field of automobile various aspects have been considered which makes a vehicle automated. Google, the biggest network has started working on the self-driving cars since 2010 and still developing new changes to give a whole new level to the automated vehicles. In this project we have focused on two applications of an automated car, one in which two vehicles have same destination and one knows the route, where other don't. The following vehicle will follow the target (i.e. Front) vehicle automatically. The other application is automated driving during the heavy traffic jam, hence relaxing driver from continuously pushing brake, accelerator or clutch. The idea described in this project has been taken from the Google car, defining the one aspect here under consideration is making the destination dynamic. This can be done by a vehicle automatically following the destination of another vehicle. Since taking intelligent decisions in the traffic is also an issue for the automated vehicle so this aspect has been also under consideration in this project .**

## I. INTRODUCTION

Recent progress in computer vision has been driven by high-capacity models trained on large datasets. Image classification datasets with millions of labeled images support training deep and highly expressive models.Following their success in image classification, these models have recently been adapted for detailed scene understanding tasks such as semantic segmentation. Such semantic segmentation models are initially trained for image classification, for which large datasets are available, and then fine-tuned on semantic segmentation datasets, which have fewer images. We are therefore interested in creating very large datasets with pixel-accurate semantic labels. Such datasets may enable the design of more diverse model architectures that are not constrained by mandatory pre-training on image classification. They may also substantially increase the accuracy of semantic segmentation models, which at present appear to be limited by data rather than capacity. (For example, the top performing semantic segmentation models on the PASCAL VOC leaderboard all use additional external sources of pixel wise labeled data for training.) Creating large datasets with pixel wise semantic labels is known to be very challenging due to the amount of human effort required to trace accurate object boundaries

## II. PROBLEM DEFINATION

Comparatively, self-driving cars are already precisely that much safer. Some studies show that self-driving cars may get in more accidents at low speeds (4mph or below) because they follow traffic laws more carefully, but the consequences are nearly irrelevant.

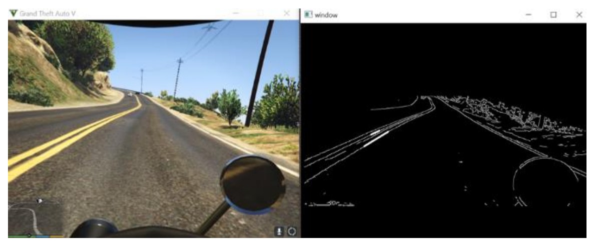
## III. EXISTING SYSTEM

Synthetic data has been used for decades to benchmark the performance of computer vision algorithms. The use of synthetic data has been particularly significant in evaluating optical flow estimation due to the difficulty of obtaining accurate ground-truth flow measurements for real-world scenes .Most recently, the MPI-Sintel dataset has become a standard benchmark for optical flow algorithms and has additionally yielded ground-truth data for depth estimation and bottom-up segmentation. Synthetic scenes have been used for evaluating the robustness of image features. and for bench marking the accuracy of visual odometry .Renderings Object Models have been used to analyze the sensitivity of convolutional network features. In contrast to this line of work, we use synthetic data not for bench marking but for training a vision system, and tackle the challenging problem of semantic segmentation. Rendered depth maps of parametric models have been used prominently in training leading systems for human pose estimation and hand tracking. 3D object models are also increasingly used for training representations for object detection and object pose estimation. Renderings of 3D object models have been used to train shape-from-shading algorithms. And convolutional networks for optical flow estimation. Renderings of entire synthetic environments have been proposed for training convolutional networks for stereo disparity and scene flow estimation. Our work is different in two ways. First, we tackle the problem of semantic segmentation, which involves both recognition and perceptual grouping. Second, we obtain data not by rendering 3D object models or stylized environments, but by extracting photorealistic imagery from a modern open-world computer game with high-fidelity content. Computer games – and associated tools such as game engines and level editors – have been used a number of times in computer vision research. Development tools accompanying the game Half Life 2 were used for evaluating visual surveillance systems . These tools were subsequently used for creating an environment for training high-performing pedestrian detectors . And an open-source driving simulator was used to learn mid-level cues for autonomous driving . In contrast to these works, we deal with the problem of semantic image segmentation an demonstrate that data extracted from an unmodified off-the-shelf computer game with no access to the source code or the content can be used to substantially improve the performance of semantic segmentation systems. Somewhat orthogonal to our work is the use of indoor scene models to train deep networks for semantic understanding of indoor environments from depth images . The training data synthesized in these works provides depth information but no appearance cues. The trained models are thus limited to analyzing depth maps. In contrast, we show that modern computer games can be used to increase the accuracy of state-of-the-art semantic segmentation models on challenging real-world benchmarks given regular color images only.

## IV. PROPOSED SYSTEM

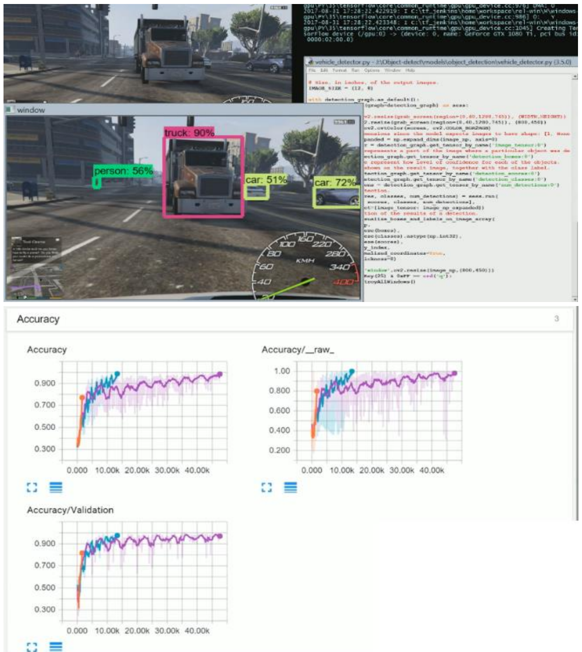
As part of our project to open source self-driving car, we faced multiple challenges. One the Critical problem was to predict the steering angle. Predicting steering angle can be challenging based on different road types, traffic as well as weather. Our goal is to develop a deep learning model that learns from Datasets and by feeding it various series images and their corresponding records .Once the model is trained, it should predict how to drive itself as close as possible to a human driver.

## V. FIGURES AND TABLES



Now that we're reading frames, and can do input, we're back on the task of trying to do some self-driving. In order to do this, a common goal is to be able to detect lanes. We can use lane detection to both create a self-driving AI that works based on simple rules based on these lanes and also to train an AI that we hope could later generalize to more scenarios. Code up to this point: put horizon img edge detection

First, find the main lines. Next, find the groups of lines that are similar to each other (by comparing slope and bias), and save these as "the same line." Next, take the two most common lines, and assume these must be our lanes. After we've done ROI, the next most likely "line" just simply is almost certain to be the lanes. That's the hypothesis anyway! We can begin to use this data for all sorts of artificial intelligence purposes




For training data, we need both features and labels. The feature sets will be the pixel data from the screen data. The labels are the key inputs we want the AI to make.

We already have the pixel data, but we do not have a way of collecting inputs.

We made inputs, but we need a way to capture our own inputs.Resizing the game data to 80x60. This is so that the game data more easily fits into our Convolutional Neural Network when it comes time to train.

*AbbreviationsandAcronyms*

| | |
|---|---|
| ABS | Abstract |
| AI | Artificial Intelligence |
| BGR | Blue Green Red |
| CVT | Convert |
| CPY | Copy |
| ConvNet | Convolutional Neural Network |
| GTA | Grand Theft Auto |
| GMM | Gaussian Mixture Model |
| GPS | Global Positioning System |
| ImShow | Image Show |
| RGB | Red Green Blue |
| NP | Numpy(Numerical Python) |
| MTS | Mesh Texture Shader |
| TF | TensorFlow |
| PY | Python |

## VI. CONCLUSION

We presented an approach to rapidly producing pixel-accurate semantic label maps for images synthesized by modern computer games. We have demonstrated the approach by creating dense pixel-level semantic annotations for 25 thousand images extracted from a realistic open-world game. Our experiments have shown that data created with the presented approach can increase the performance of semantic segmentation models on real-world images and can reduce the need for expensive conventional labeling.

## VII. ACKNOWLEDGMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Mr. S.Y.Divekar for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.I also take this opportunity to express a deep sense of gratitude to Harrison Sentdex, Consultant, Pythonprogramming.net, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.I am obliged to staff members of AISSMS's Second Shift Polytechnicfor the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment. Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this assignment would not be possible.

## REFERENCES

[1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB (1994)

[2] Akenine-M¨oller, T., Haines, E., Hoffman, N.: Real-Time Rendering. A K Peters, 3 edn. (2008)

[3] Appleby, A.: Murmurhash, https://github.com/aappleby/smhasher

[4] Aubry, M., Maturana, D., Efros, A.A., Russell, B.C., Sivic, J.: Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In: CVPR (2014)

[5] Aubry, M., Russell, B.C.: Understanding deep features with computer-generated imagery. In: ICCV (2015)
[6] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. IJCV 92(1) (2011)
[7] Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. IJCV 12(1) (1994)
[8] Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters 30(2) (2009)
[9] Butler,D.J.,Wulff,J.,Stanley,G.B.,Black,M.J.:An aturalisticopensourcemovieforoptical flow evaluation. In: ECCV (2012)
[10] Chen, C., Seff, A., Kornhauser, A.L., Xiao, J.: DeepDriving: Learning affordance for direct perception in autonomous driving.