Car Steering Angle Prediction Using Deep Learning

Dr. M. Senthil Kumaran[1], Kaniganti Siva Sankar[2], Kodakandla Srikar[3]

[1]Associate Professor & HOD, Department of Computer Science, Sri Chandra Sekhendra Saraswathi Viswa Mahavidyalaya , Kanchipuram, India
[2]Student, B.E, Department of Computer Science, Sri Chandra Sekhendra Saraswathi Viswa Mahavidyalaya , Kanchipuram, India
[3]Student, B.E, Department of Computer Science, Sri Chandra Sekhendra Saraswathi Viswa Mahavidyalaya , Kanchipuram, India

---***---

ABSTRACT: Self-driving cars have become very popular in the last few years. Udacity has released a database containing, among the data, a collection of images with a direct angle taken during the drive. The Udacity challenge aimed at predicting the directional angle only supports the given images.

We are testing two different models to make high quality assumptions of support angles using different in-depth learning strategies including Transfer Learning, 3D CNN, LSTM and ResNet. If the Udacity challenge persisted, both of our models would have included in the top ten of all entries

Keywords: Deep Learning, Neural Network, Convolution neural network

INTRODUCTION

Self-driving cars have had a huge impact on the economy over the next decade. Creating models that meet or exceed the capacity of a human driver can save thousands of lives each year. Udacity has an ongoing challenge to create an open source for self-driving vehicles [19]. In their second challenge Udacity released a database of photographs taken while driving and the corresponding directional angle as well as the corresponding sensor data for the training set (left, right, and central angles with integrated angles supported by the camera angle). The goal of the challenge was to find a model, based on the image taken while driving, would reduce the RMSE (root mean square error) between the prediction of the model and the actual directional angle generated by the human driver. During this project, we explore a variety of strategies that include 3D convolutional neural networks, common neural networks using LSTM, ResNets, etc. to extract the directional angle predicted by numerical values.

The aim of the project is to remove the requirement for handwriting rules and create a program that learns how to drive visually. Predicting the directional angle is an important part of the self-driving end of a car and will allow us to test the full potential of the neural network. for example, using a directional angle only because the training signal, deep neural networks can automatically extract features to help set the road to create prediction.

The two models to be discussed are models that use 3D dynamic layers followed by duplicate layers using LSTM (short-term memory). This model will test how temporary information is used to predict the directional angle. Both 3D conversion layers and continuous layers use temporary information. The second model to be discussed uses transfer learning (using the lower layers of the pre-trained model) using the high quality model trained in CNN

Pictures of two straight center shifts can be taken from left and right camera. Additional switching between cameras and all rotation mimics the transformation of the image view from a nearby camera. Conversion of precision view requires 3D space information that is not available. We are therefore measuring change by assuming that all points below the horizon are flat and that all points above the horizon are very far away. This works well on a flat surface but introduces distortions of objects that stick to the surface, such as cars, poles, trees, and buildings. Fortunately this distortion does not pose a major problem in network training. Transformed photo control label is adapted to one that can return the car to its desired position and position in two seconds.

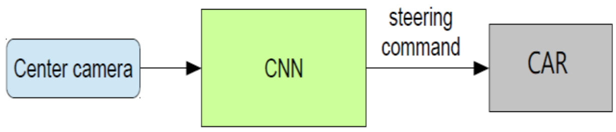

Figure 1: The trained network is used to generate steering wheel angle given input an image

A diagram of our training system block is shown in Figure 2. The images are uploaded to CNN and include the proposed direction. The proposed command is compared to the required command of that image and the CNN weights are adjusted to bring the CNN output to the desired output. Weight adjustment is achieved using a back distribution as used in the Torch 7 machine learning package.

Once trained, the network can generate direction from the video images of the central camera.

1. LITERATURE REVIEW

This The use of the neural network of autonomous vehicles was initiated by Pomerleau (1989) who created the Autonomous Land Vehicle during the Neural Network (ALVINN) program. The structure of the model was simple, incorporating a fully connected network, small by today's standard. The network predicts actions from pixel inputs used in simple driving situations with few obstacles. However, it has shown the power of neural networks for automatic end-to-end navigation. Last year, NVIDIA released a paper about the same idea you gained from ALVINN. within the paper, the authors have used the relatively basic structures of CNN to extract features in driving frames. The layout of the buildings can be seen in Figure 1. The addition of information collected was found to be important. The authors used the work shifts and rotation training set. Left and right cameras with integrated directional angles are also included. This framework has succeeded in simple real-world situations, such as following highways and driving on flat, unobstructed courses. More recently, additional efforts to use deep CNNs and RNNs to address the challenges of video fragmentation, scene resolution, and object acquisition have led to the use of complex CNN architecture in autonomous driving. "Spatiotemporal Learning Features With 3D Convolutional Networks" introduces how to create 3D conversion networks to capture spatiotemporal features in the sequence of images or videos. "Without Short Captions: Advanced Video Editing Networks" describes two methods that include using LSTM to view videos 1 arXiv: 1912.05440v1 [cs.CV] 11 Dec 2019 Figure 1. CNN architectures used in. The network contains approximately 27 million connections and 250 thousand parameters. . "In-depth Learning of the Image Censor Remnants" and "Most Connected Networks" describe techniques for building residual interactions between different layers and making it easier to train deep emotional networks. In addition to CNN and / or RNN methods, there are other research programs that use in-depth learning strategies in automated driving challenges. Another line of staff manages auto-navigation as a video predictor function. Comma.ai [14] has proposed the acquisition of a driving simulation system that includes a Variational Auto-encoder (VAE) and a Generative Adversarial Network (GAN). Their approach is in a state of constant predicting video that looks realistic for a few pre-supported frames despite a modified version developed without value function within a pixel space. In addition, deep reinforcement learning (RL) has also been applied to automatic driving. RL was not successful in automotive applications until some recent activity demonstrated in-depth reading ability to learn good environmental representation. This has been demonstrated by reading games like Atari and Elapse Google DeepMind. Encouraged by these activities, they have developed a framework for self-driving using deep RL. Their framework is expandable to include RNN information integration, which enables the vehicle to handle less visible situations. The framework also integrates attention models, utilizing viewing networks and actions to direct CNN characters to input file areas that are compatible with the driving process.

2. IMPLEMENTATION

We train the weight of our network to reduce the rate of square error between the output of the network direction command and consequently the human driver command, or adjusted steering command for non-central and circular images. Our specification is shown in Figure 2. The network consists of 9 layers, including a standard layer, 5 convolutional layers and 3 fully connected layers. The inserted image is split into YUV aircraft and transferred to a network.

The first layer of the network makes the image familiar. The normalizer has a solid code and is not fixed within the learning process. Normal performance within the network allows the custom system to be changed by spec and accelerated by GPU processing.

Convolutional layers were designed to perform feature removal and were randomly selected by a series of tests that altered the layout setting. We use convolutions with lines between the first three layers with a 2 × 2 stride and a 5x5 kernel and a consistent convolution with a 3 × 3 kernel size between the last two layers.

We follow the five layers of convolution with three fully integrated layers that lead to the amount of output control i.e. the rotating radius. Fully connected layers are designed to act as a controller, but we realize that by training the system end-to-end, it is not possible to make a clean break between which parts of the network serve primarily as a feature output and which function control out.

2.1 Training Details

2.2.1 Data Selection

The first step in training the neural network is to choose the frames you will use. Our collected data is based on road type, space event, and driver activity (stay on the highway, change lanes, turns, etc.). to train CNN so that we can try to navigate by following only select data where the driving force stays on the highway and discard the rest. Then we sample that video at 10 FPS. a better rate can lead to the installation of very similar images and thus not provide much useful information.

2.2.2 Argumentation

After selecting the final set of frames we increase the information by adding activity shifts and rotations to show the network how to get through the wrong area or shape. The magnitude of that disturbance is randomly selected from the standard distribution. Distribution has zero meaning, so the difference doubles the quality deviation we have measured with human drivers. Adding information automatically does not add unwanted artifacts as the size increases.

2.2.3 Simulation

Before examining the trained CNN, we first evaluated the performance of the simulation networks. A simplified diagram of the simulation system is shown in Figure 3.

The template captures video footage of a camera on the front facing of a man-made vehicle and produces almost identical images if CNN, instead, was directing the car. These test videos are time-synchronized with recorded directional instructions created by the human driver.

Since human drivers do not always drive in the middle of a track, we measure the center of the route associated with each video frame used by the template. This position is called the "basic truth".

The simile converts the first images into narratives from a lower reality. Note that this change includes any differences between the way a person operates and the basic truth. Conversions are carried out in the same manner as described in Section 2.

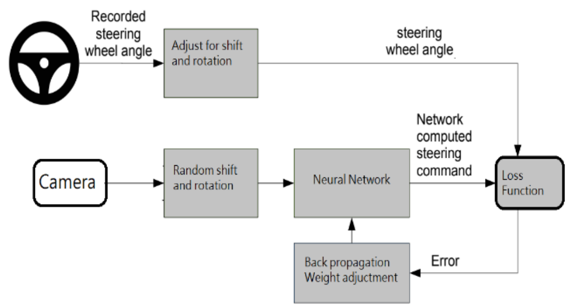

Figure 2: Overview of Process

The template accesses a recorded test video and syncing instructions that occur while the video is being shot. The template sends the main frame of the selected test video, optimized anywhere from the low reality, to the CNN-trained input. CNN then returned the directing order of that framework. CNN's directional instructions are in addition to the fact that the driver's recorded instructions are inserted into the flexible model [8] of the vehicle to update the position and position of the simulated vehicle.

The template then adjusts the next closure in the test video so that the image looks like the car is in place which caused it to follow the instructions from CNN. This new image was then submitted to CNN and the process is repeated.

The template records the non-central distance (distance from the car to the center of the track), yaw, hence the distance traveled by the visible vehicle. If the distance in the center exceeds one meter, the visual human intervention is initiated, so the visual vehicle area and shape are reset to match the lower reality of the corresponding frame of the original test video.

2.2.4 Evaluation

Testing our networks is done in two steps, first by imitation, and then by road tests.

In simulation we have networks that provide directional instructions in our simulation on a combination of recorded test routes corresponding to a total of three hours and 100 miles of driving in Monmouth County, NJ. The test data was taken from a variety of light and weather conditions and included highways, local roads, and residential lanes.

3. CONCLUSION

We have strongly demonstrated that CNN is able to learn all the work of the track and track without having to go through a lot of detection of road or route marking, semantic abstraction, route planning, and control. A small amount of training data from less than 100 hours of driving is enough to train a car to operate in a variety of conditions, on highways, on local roads and in residential areas with sunny, flexible, and rainy weather. CNN is able to learn the correct road features in a limited training signal (direction only).

The program learns for example to find a roadmap without the need for clear labels during training.

More work is needed to improve network durability, find ways to ensure durability, and improve the visibility of internal network processing steps.

4. FUTURE SCOPE

The proposed fingerprint based voting system project aims at reducing illegal activities during the election time. This system will also provide accurate results. We can implement this system in real time elections to reduce rigging and to conduct free and fair elections. In this project we are using the fingerprints of the voters so that more security is provided as no two individuals will have the same fingerprint patterns. This system also doesn't allow a person to vote twice. In future this project can be implemented in real time. The additional features that can be implemented in this project are we can use fingerprint scanner to capture the fingerprints of the voter. We can also connect aadhar database to the system in real time implementation. The main aim of this project is to conduct free and fair elections by using the biometrics of the voters.

5. REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backprop- agation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, Winter 1989. URL: http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and

K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks. pdf.

[3] L. D. Jackel, D. Sharman, Stenard C. E., Strom B. I., , and D Zuckert. Optical character recognition for self-service banking. AT&T Technical Journal, 74(1):16–24, 1995.

[4] Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/ challenges/LSVRC/.

[5] Net-Scale Technologies, Inc. Autonomous off-road vehicle control using end-to-end learning, July 2004. Final technical report. URL: http://net-scale.com/doc/net-scale-dave-report.pdf.

[6] Dean A. Pomerleau. ALVINN, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, 1989. URL: http://repository.cmu.edu/cgi/viewcontent. cgi?article=2874&context=compsci.

[7] Wikipedia.org. DARPA LAGR program. http://en.wikipedia.org/wiki/DARPA_LAGR_ Program.

[8] Danwei Wang and Feng Qi. Trajectory planning for a four-wheel-steering vehicle. In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, May 21–26 2001. URL: http:

//www.ntu.edu.sg/home/edwwang/confpapers/wdwicar01.pdf.

[9] DAVE 2 driving a lincoln. URL: https://drive.google.com/open?id= 0B9raQzOpizn1TkRIa241ZnBEcjQ.