Convolutional Neural Networks for AI Cars and Object Recognition

Soumya Mishra
Raajdhani Engineering College, Bhubaneswar
soumyamishra @rec.ac.in

Abstract

Convolutional Neural Networks were utilized to forecast the vehicle's behavior while it was being driven. It combines the challenges of supervised reversing with the inability to discern between the steering orientations of the car and things in front of it in road pictures. Those pictures were shot from three separate camera angles: the car's left, center, and right. The car simulator was used to test the model when it was constructed.

Keywords:- *Augmentation, behavioral cloning, convolutional neural network, openCV, validation (key words)*

INTRODUCTION

AI brings a new perspective to the automotive world. Global car manufacturers are facing a revolution at the moment when cars have access to advanced computer systems, Internet access, and advanced display / communication hardware. Generally, car manufacturers have little or no knowledge of software development.**AI CYBORG CAR** is a vehicle that is capable of sensing its environment and moving safely with no human inputs.

An autonomous vehicle, also known as a robotic or random vehicle such as a non-motorized or self-driving vehicle, is a private vehicle capable of realizing the human mobility of a traditional car [3]. Like an autonomous vehicle, it can sense its own environment and navigate on its own. One can choose the destination, but it is not necessary to perform any car operation. Advanced control systems translate data to identify appropriate roaming routes, as well as appropriate obstacles and signs. Private cars update their maps based on sensor insertion, so they can travel in unknown locations.

I. LITERATURE REVIEW

Self-driving vehicles, also known as private cars, non-motorized vehicles, or robot vehicles, are vehicles that can sense their environment and travel safely with little or no human input. [1] [4]

In another activity that accepts CIFAR-10 databases [2], a more comprehensive and in-depth network construction, combined with GPU support to reduce training time. On popular databases, such as MNIST handwritten digits, Chinese characters, and CIFAR-10 images, near human functionality are available. Extremely low error rates beat previous most artistic results. It should be noted, however, that the network used for the CIFAR-10 database has 4 specific layers with 300 maps each, 3 layers of integration, and 3 fully connected 6 | P a g e extraction layers. As a result, even though the GPU was used, the training period was a few days.

In 2010, the introduction of the annual Imagenet [4] challenge expanded research into image classification and a large collection of labeled data has been widely used in literature since then. In the recent work of Krizhevsky et al. [10], a network with 5 convolutional, 3 pooling max, and 3 fully connected layers trained with 1.2 million high resolution images in the Imagenet LSVRC-2010 competition. After applying the reduction techniques, the results are promising compared to the previous condition of the variety. In addition, tests are being performed to reduce network size, suggesting that the number of layers can be significantly reduced while performance decreases slightly.

Based on a number of self-driving papers / textbooks related to automotive or automotive automation, there are a number of substandard artificial intelligence methods such as some of the following abstract-based literature: Independent movement of robotic vehicles is achieved through continuous communication between vision, intelligence and performance [4]. Navigating automatic robotic vehicles in hotspots requires the discovery and use of real-time sensors. Effective independent autonomous control algorithms, should mimic the way people operate controlled or similar vehicles. Fuzzy logic is a type of mind with a lot of value or possible logic; it is about imaginary thinking rather than straightforward and straightforward [5]. Contrary to the traditional view, where binary sets have two values: true or false, unreasonable logical variables can have a true value between the scale of 0 and 1.

Yim and Oh [14] developed a three-dimensional algorithm based on line detection. The factors used determine the position, position and amount of stiffness. In the first step, Sobel operator is used to get the details on the edge. The boundary of the routes is represented as a vector containing these three elements. The current line vector is calculated based on the input image and the vector of the front row model. Two windows, one by one, are used for left and right borders. Taking a pixel N to each horizontal line, l lane vector candidates are produced. The best selected is selected based on the lowest distance from the vector of the previous row using the weighted metric distance. By measure each element is assigned a different weight. Then a line trending system is used to predict a new line vector. If the width of the path suddenly changes, the calculated current vector is discarded and the previous one is taken as the current vector.

II. METHODOLOGY

We are building a model which is best suited for the autonomus driving.

First, Images will be collected on a dacity car display with appropriate steering angles and the corresponding number of pressures. These Images will be further divided into a training set and a test set. The model will be built with the appropriate number of categories listed. This model will then be trained in the received training images. Once the model is trained, this model will be tested for accuracy with pre-set test images. Depending on the accuracy of some of the changes that will be made to the model will be made. If this is a case of overwork Model accuracy will be measured The images obtained in the test set. When the model works well on the set test images, the model will be used in the vehicle simulator.

Model accuracy will be measured The images obtained in the test set. When the model works well in the set test images, the model will be used in the car simulator.

Using Udacity car simulator the CNN model built will be de- ployed and the performance will be tested.

The application will be set in Autonomous mode and the mo- del will be deployed using python flask api.

The api will receive multiple frames of images. These images are sent to our model by the api. The model gives predictions appropriate to the images obtained.

The model performance will be tested with a completely different „test track‟ where the car will be made to operate independently. If the performance of the model is good in the new track then the model is a success and can be implemented in used cases. If the performance is not upto the mark then the following tweeks have to be made to the model and has to be trained again.

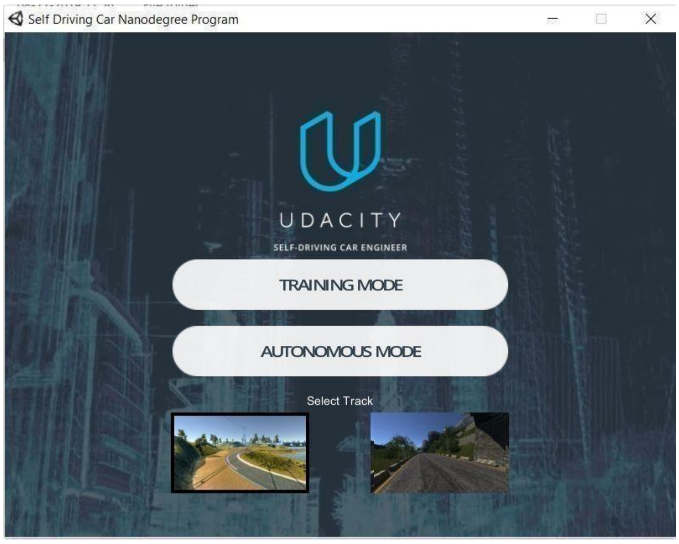
The main take away from the project is to create a model that is best suited for autonomous driving and this can be implemented in any environment scenario.

Manipulation. This major flexibility our model will learn to adjust is the steering of the car at any time.

IV SYSTEM DESIGN

A. Software Used

We have used software to simulate our model and test its performance. This is an open source software provided by Udacity .


Self Driving Car Nanodegree Program
UDACITY
SELF-DRIVING CAR ENGINEER
TRAINING MODE
AUTONOMOUS MODE
Select Track

Two modes are present in the application Training and Autonomous. From the training mode we obtain the steering angles, throttle and the Image associated with it.

B. Model Architecture Design

The model is designed using CNN (Convolutional Neural Networks) for end -to-end self driving. The model was custom built and tested as the model accuracy would be very high.

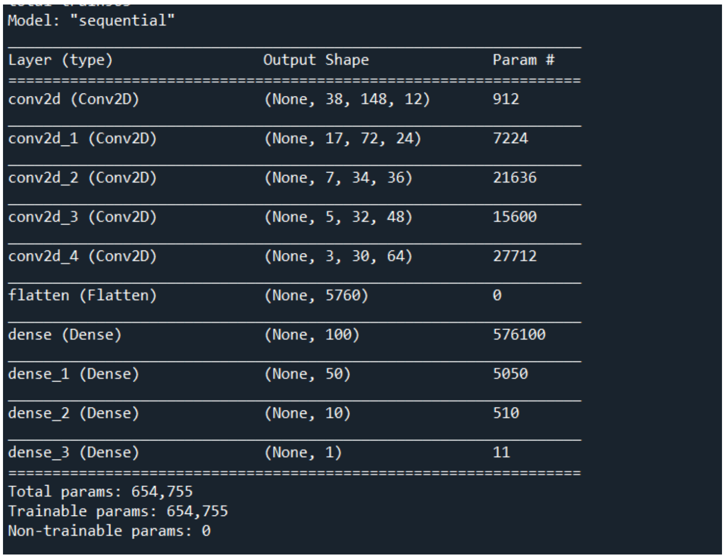
It is a flexible neural network that works well with controlled image editing / retrieval problems. The model is designed in such a way that the model avoids the challenges of overcrowding and inadequacy.

The model is built in the following way:

- The first layer of the model is a Convolutional layer provided with an input image of size 80X300 RGB images.
- The first layer will be using 12 5x5 filter with with (2,2) shride in it. In each of the layer elu activation is used.
- The second layer will be using 24 5x5 filter with with (2,2) shride in it. In each of the layer elu activation is used.
- The third layer will be using 36 5x5 filter with with (2,2) shride in it. In each of the layer elu activation is used.
- The fourth layer will be using 48 3x3 filter with with (1,1) shride in it. In each of the layer elu activation is used.
- The fifth layer will be using 64 3x3 filter with with (1,1) shride in it. In each of the layer elu activation is used.

- The next layer will be a flattened layer 5760 neurons followed by dense layers with 100,50,10.
- Final layer will have one neuron to predict the steering angle.

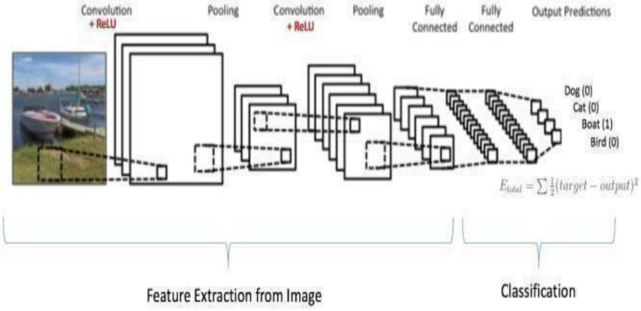
Finally, the model looks like this:



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 38, 148, 12)       912
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 17, 72, 24)        7224
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 7, 34, 36)         21636
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 5, 32, 48)         15600
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 30, 64)         27712
_________________________________________________________________
flatten (Flatten)            (None, 5760)              0
_________________________________________________________________
dense (Dense)                (None, 100)               576100
_________________________________________________________________
dense_1 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_2 (Dense)              (None, 10)                510
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 11
=================================================================
Total params: 654,755
Trainable params: 654,755
Non-trainable params: 0
```

C. *Convolutional Neural Network*

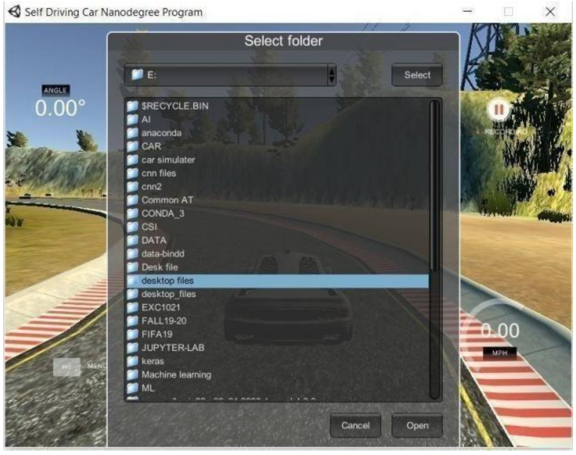
Convolutional Neural Networks are essentially the same as conventional Neural Networks from the previous phase: they contain neurons that carry loads and tendencies. Every neuron acquires a few sources of information, plays a tragic thing and discards it with non-linearity [13] [11]. The whole system still extends the function of isolation alone: from raw pixels of images to one side to schools of another. After all they have a bad job (for example SVM / max max) in the final layer (completely related) with all the tips / traps we have developed to study Neural Networks still apply



V IMPLEMENTATION

A. *Collecting Data*

First, we have to drive the car on the Training track and we have to collect the details by recording in the simulator. Ordinary Arrow keys are used to drive the car. When finished viewing our data in the selected folder before recording.



Producing Excel sheet of data and IMAGE.

The excel sheet contains a lot of data. And a sample of it is shown below

B. *Image Pre-processing (Image Sizing)*

- Images are cut so that the correct view of the road can be seen in the photo.
- Images enlarged to 80X300 (3 YUV channels) according to our model.
- Images are of standard size as mentioned in the above architecture section, this is to prevent overcrowding and make the gradients work better.

The images were taken by a Car with three front-facing cameras. Center, left, right image. Below are photos of all three cameras at once.

C. Training and Validation Split

Images are separated by training and development sets to measure performance at each epoch. Tests are performed using a simulator.

For the training set,

- Mean squared error was chosen as the loss function.
- Adam optimizer was used for optimization of the model having a learning rate of 1.0e-4.
- Model Checkpoint from Keras was used to save the model only if the confirmation loss is improved which is checked each time.

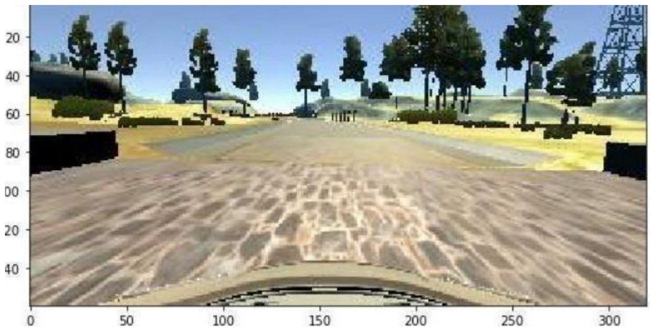
D. Model Training (Image Augmentation)

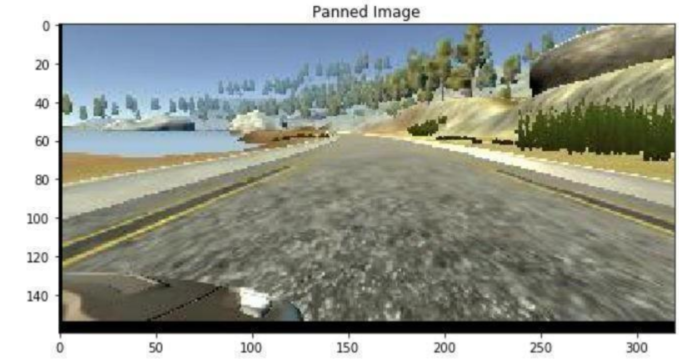
For training,The collected images were first augmented using the open CV library in python and the datasets were normalized:

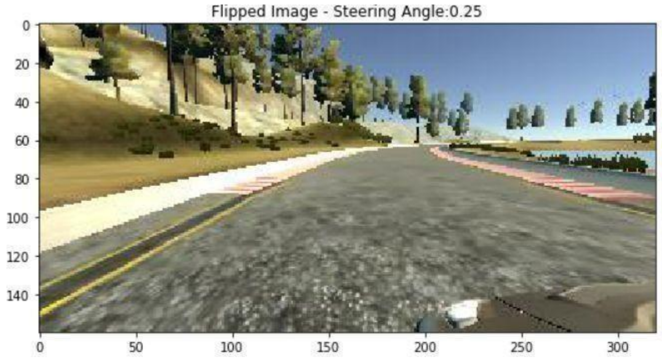
- Randomly select images to the right, left or center.
- In the left image, the directional angle is adjusted by + 0.2
- To get the correct image, the directional angle is adjusted by -0.2
- Flip image random left / right
- Randomly translate image horizontally with adjustment angle adjustment (0.002 per pixel shift)
- Indirectly translate image vertically
- Randomly added shadows
- Random change of image brightness (bright or dark

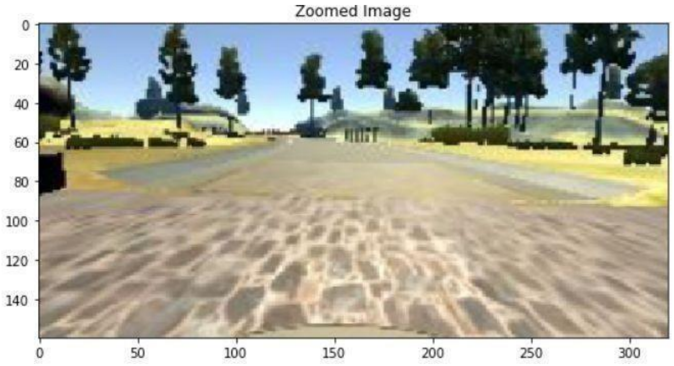
Using the left / right images helps to train the driving mode for recovery. Horizontal translation is helpful for heavy curve handling (e.g. the one behind the bridge).

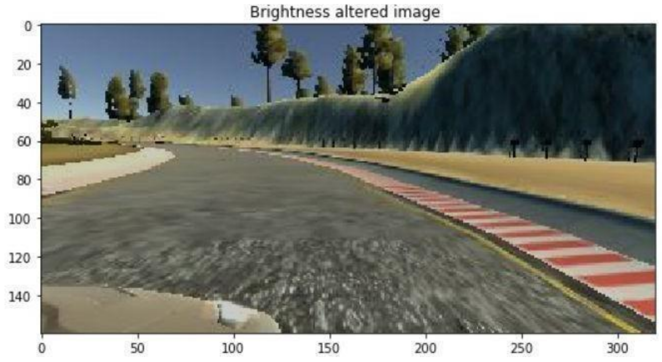
The Augmented Images used are shown below.



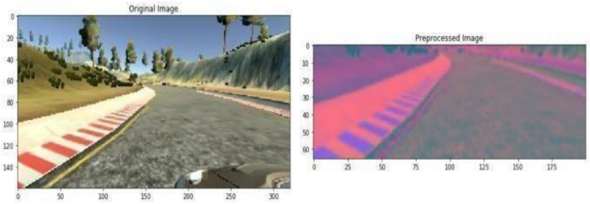








Below is the preprocessed image which is actually used in training to train the car whether if its raining or any climatic changes are there.



Now we have done Image preprocessing which is to be in training process.

C. Training, verification and evaluation

Images are divided into training and validation set to measure performance at all times. Tests are performed using a simulator.
As for training,

- MSQ error is used for the loss function to measure ho w close the model predicts to the directional angle of each image
- The Adam optimizer is used as a good performance with a reading rate of 1.0e-4.
- Model Checkpoint from Keras was used to save the model only if the confirmation loss is improved which is checked each time..

Lake Side Track:

Since there may be an unlimited number of extended photos, We place samples per epoch for 20,000. I tried from 1 to 200 epochs but found 5-10 epochs good enough to produce a well-trained model of the track next to the lake. The 40-bit batch size was chosen as that is a large size that does not cause memory error on my Mac with NVIDIA GeForce GT 650M 1024 MB.

VI RESULTS

The model is a Convolutional Neural Network, feature extraction of Continuous Regression. This was deployed on the Udacity car simulator an open source simulator for testing the deep learning models.

Once the model was created, tested and reworked the model was the deployed in the car simulator to obtain the performance. Depending on the performance the tweeks for the model were made and the final version of the model was deployed.
In the training set, the accuracy was calculated to be 93 percent and 91 percent in the test set.
Once the final tweaks were made the model was deployed in a new track and a good performance of the model was observed.

VII CONCLUSION AND FUTURE SCOPE

Artificial Intelligence is a method by which human cognitive abilities can be captured and reproduced in a computer program. As a person‟s study develops that ability, his actions are recorded along with the situation that created the action.

Self-driving cars are ready to transform the transport industry. There have been many significant changes in the automotive industry since the start of automotive production nearly eighty years ago, but the basic formula of a man driving a car using a steering wheel and pedals held well at that time. That changes quickly. New cars already have automated features such as parking and collision crashes, and [9] [8] Automotive and technology companies are working hard to bring cars that can move smoothly without having to install a human driver.

VIII REFERENCES

[1] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall **"Activation Functions: Comparison of trends in Practice and Research for Deep Learning"**

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi "**You Only Look Once: Unified, Real-Time Object Detection**" [*Submitted on 8 Jun 2015 (v1), last revised 9 May 2016 (this version, v5)]*

[3] M. Bojarski et al. End-to-End Learning for Self-Driving Cars. arXiv:1604, 2016.

[4] 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. 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/wdwicar0

1. pdf.[9] DAVE 2 driving a lincoln.

[5] Viorel Stoian "A Control Algorithm for Autonomous Electric Vehicles by Fuzzy Logic" Advanced Engineering Forum Submitted: 2017- 01-31ISSN: 2234-991X, Vol. 27, pp. 103-110 revised:2017-05 31doi:10.4028/www.scientific.net/AEF.27.103Acc epted: 2017-06-07© 2018 Trans Tech Publications, Switzerland system performance.

[6] Campbell, Mark, Magnus Egressed, Jonathan P. How, and Richard M. Murray. "Autonomous driving in urban environments: approaches, lessons and challenges." Transactional Philosophy of Royal Society A: Mathematical, Physical and Engineering Sciences 368, no. 1928 (2010): 4649-467.

[7] An introduction to convolutional neural networks[Online]availableat:http://white.stanford.edu/teach/i n dex.php/An_Introduction_to _CNN

[8] Sun woo, M. K. Jo, Dongchul Kim, J. Kim, and C. Jang." Development of Autonomous Car–Part I: Distributed System Architecture and Development Process."(2014): 1-1.

[9] Journal of Field Robotics 25(9), 569–597 (2008) Cˇ 2008 Wiley Periodicals, Inc. Published online in Wiley InterScience

[10] S. Liu et al., Creating Autonomous Vehicle Systems, Morgan Claypool Publishers, 2017.

[11] E. Coelingh, J. Nilsson, J. Buffum, "Driving tests for self-driving cars", IEEE Spectrum, vol. 55, no. 3, pp. 40-45, March 2018.

[12] E. Ackerman, "Lidar that will make self-driving cars affordable [News]", IEEE Spectrum, vol. 53, no. 10, pp. 14- 14, October 2016.

[13] M. Montgomery, The new big 4 of the auto world: Tesla Google Apple and Uber, Nov. 2015

[14] A. Rassõlkin, L. Gevorkov, T. Vaimann, A. Kallaste, R. Sell, "Calculation of the traction effort of ISEAUTO self- driving vehicle", 2018 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives (IWED), pp. 1-5, 2018.