International Journal of Scientific Research in Science, Engineering and Technology (www.ijsrset.com)
© 2020 IJSRSET | Volume 7 | Issue 3 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
DOI : https://doi.org/10.32628/IJSRSET207323

Convolutional Neural Network Computation for Steering Angle Prediction Based on Road Direction

Aires Da Conceicao[1], Dr. Shashang Degadwala[2]

[1]U.G. Scholar, Computer Engineering Department, Sigma Institute of Engineering, Vadodara, Gujarat, India

[2]Associate Professor, Computer Engineering Department, Sigma Institute of Engineering, Vadodara, Gujarat, India

ABSTRACT

Self driving vehicle is a vehicle that can drive by itself it means without human interaction . This system shows how the computer can learn and the over the art of driving using machine learning techniques. Therefore for a car achieving the autonomous ability it must show the control of human activities while driving. Those activities include control of steering wheel. There exist different techniques to control the steering angle and one of them is CNN. In this article we are going to see how CNN can be used to predict the steering angle.

Keywords : Self-driving car, CNN , Steering angle, CNN for steering angle.

I. INTRODUCTION

Nowadays the world is moving so fast concern about the new technologies. The automotive industry is also one of the most affect by this up coming technology. Self driving vehicle is an example of that. This kind of vehicle is no longer an imagination nowadays it turned into a reality and may enhance the future systems where computers take over the art of driving.

Autonomous car uses different kinds of technologies such as GPS to help with navigation and use sensors like RADAR or LIDAR that are used to avoid collisions. There are a huge interest in the development of autonomous vehicle in recent years. However one of the most important aspect to take in consideration while driving is controlling the steering angle. The autonomous car has the ability to predict the steering angle of the wheel based on the road direction. This ability can be achieved by different techniques one of the is CNN. CNN means Convolutional Neural Network it it a Deep Learning technique where fully layered neurons are connected together in order to perform the computation. The data to be passed to CNN need to preprocessed for that some segmentation methods are applied.

II. RELATED WORKS

End-to-End Self-Driving Using Deep Neural Networks with Multi-auxiliary Tasks Dan Wang, Junjie Wen, Yuyong Wang, Xiangdong Huang and Feng Pei in their article Networks with Multi-auxiliary Tasks they intend to compare various end-to-end multi-task deep learning networks using deep convolutional neural network combined with long short-term memory recurrent neural network (CNN-LSTM), which could obtain not only the visual spatial information but also the dynamic temporal information in the driving scenarios, and improve steering angle and speed predictions.

Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks by Rodolfo Valiente, Mahdi Zaman, Sedat Ozer and Yaser P. Fallah in their research they proposed a soluction of the problem generated while predicting steering angles directly from the raw input images. When temporal dependencies between the image frames is ignored. They presented and studied a new deep architecture to predict the steering angle automatically by using Long-Short-Term-Memory (LSTM) in the deep architecture. The deep architecture is an end-to-end network that utilizes CNN, LSTM and fully connected (FC) layers and it uses both present and futures images (shared by a vehicle ahead via Vehicle-to-Vehicle (V2V) communication) as input to control the steering angle.

III. BACKGROUND THEORY

1) Gaussian Blur

A Gaussian blur technique is used to convolve with the image and effectively reduce the effects caused by the noise and. The equation for a Gaussian blur used is :

$$Hij = \frac{1}{2\pi\sigma^2} \exp\left(\frac{(l-(k+1))^2 + (l+(k+1))^2}{2\sigma^2}\right); 1 \le i,j \le (2k+1) \quad (1)$$

2) Histogram Equalization

Histogram Equalization is a image processing technique used to improve the contrast of the image. It is accomplished by spreading out the most frequent intensity values of the image or stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of image with lower local contrast to gain a higher contrast.

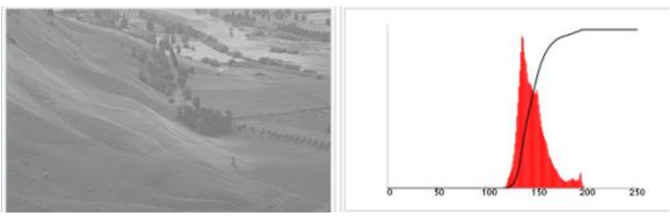


Fig. 1: Image before Histogram Equalization

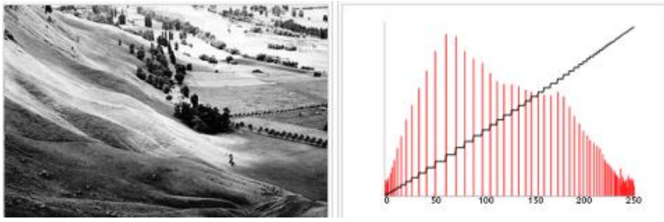


Fig. 2: Image after Histogram Equalization

3) Panning

Panning is a image processing technique. It is a combination of a slow shutter speed and camera motion to create a sense of speed around a moving object. It is a way of keeping your subject in focus while blurring the background. Panning is typically applied on a subject moving horizontally, such as a moving car.

4) Feature Extraction

Hough Transform is a feature extraction method used to identify different types of shape based on voting occurrence. In this case is used to detect the lane line on the incoming frames from the camera. The line is detected by analyzing all points that are passed through the line and this points are represented as a sinusoidal waves passing through a common line as shown in the image below. The line formula is :

$$x = mx + y \quad (2)$$

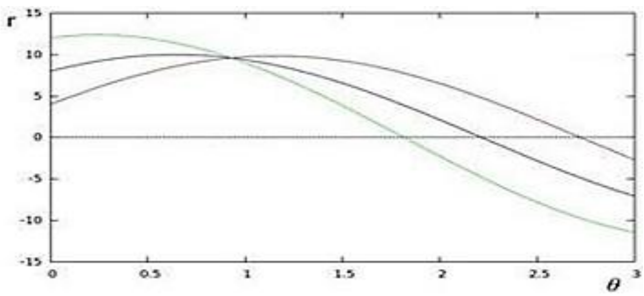


Fig 3: Hough Transform.

Feature extraction is a technique of reducing the dimensionality of the processed data for manage it better.

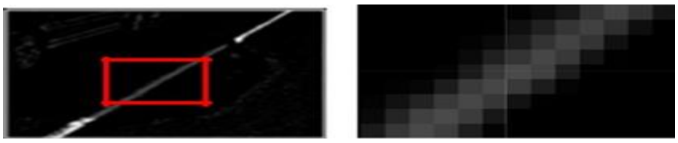


Fig 4: Feature Extraction

5) Activation Function

Activation function is responsible for activating or deactivate the neurons in order to get de desired output. It defines as neurons operation, the inputs of activation are the bias weights. Activation function is responsible to decide in any neural network if the receiving information is relevant or not. ELU means (Exponential Linear Unit) it is a function that converge values to zero faster and generating more accurate results. The difference between ELU and other activation functions is ELU has an extra alpha constant which should be positive value. ELU is very similar to RELU but ELU converges to negative inputs different from RELU. They are identity function form for non-negative inputs. On the other hand, ELU performs smooth slowly until its output equal to -α whereas RELU sharply smoothies.

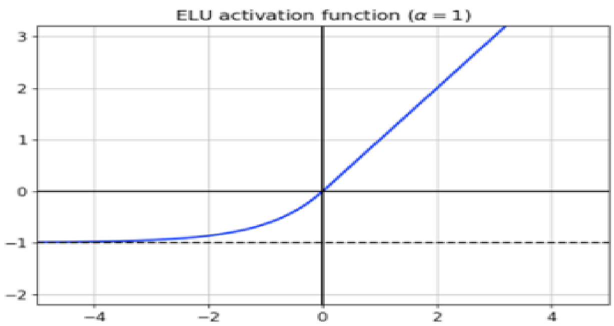


Fig.5: ELU Activation Function

IV. CNN COMPUTATION

1) Data Collection

The first step for computing steering angle prediction is to get a stream of good quality image of the road. More images leads to high accuracy but for this article we have used 580 labeled images. All the images must be of the same size and dimension.

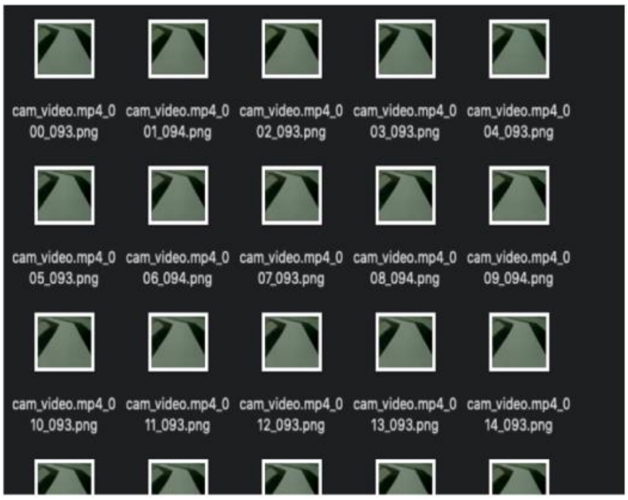


Fig.6: Image Set

While collecting the images we have to make sure to collect the image in all direction of the road that include (Right, Center and Left).

2) Data Plot

The below image shows the angles labeled in the images. All the images in our data set are labeled with the steering angle. While collecting the image the steering angle will be assigned at each frame based of the road direction it means the values less than 80 are considered left, values between 80 to 120 are considered center and values greater than 120 are considered Right.

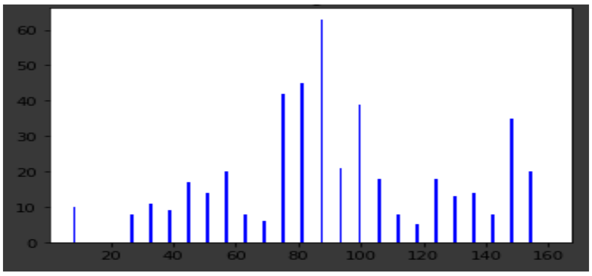


Fig.7: Data Representation

2) Histogram Equalization

After collecting the image data some data may come with a bad contrast due to position of the light so to enhance the quality of those images we have to use the histogram equalization. Histogram Equalization is used to enhance the quality of image by spreading its dynamic range.

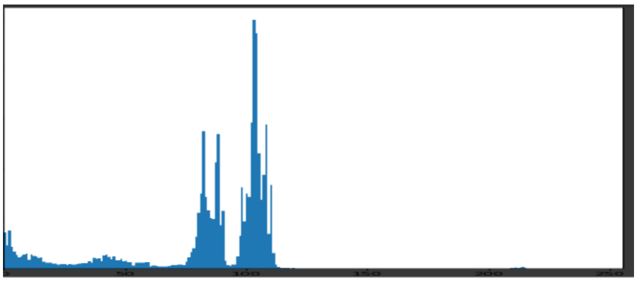


Fig.8: Histogram of Image

3) Trained and validation data

After collecting the image data we have to separate two data set. Training data and validation data. Training data will be passed into CNN convolution for generate our model and validation data is used to validate the generated model.

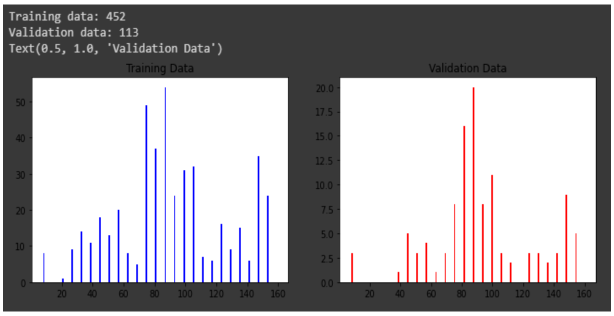


Fig.9: Image Set

4) Blurred Image

For our model there was no need to spread to much the dynamic range this much was enough to extract the feature perfectly. After that we used Gaussian blur technique to reduce the noise on image.

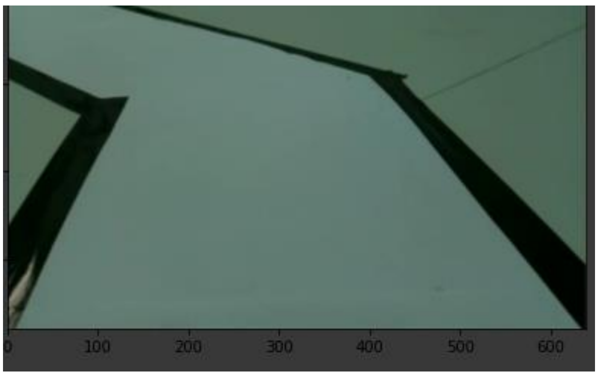


Fig.9: Blurred Image

5) YUV Image

Based on Nvidia model the YUV Image has proven to be very efficient for image feature extraction so we have converted our data set into YUV image.

Now we have to set the region of interest that is the lane only we do not want to capture everything so

We have to resize the image with (200,66) dimension based on Nividia model.

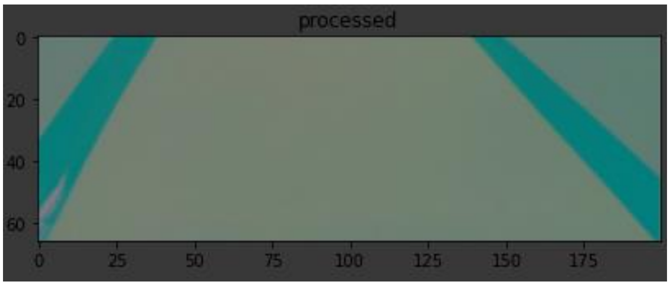


Fig.10: YUV Image

6) Nvidia Architecture

The NVIDIA architecture is based on 4 fully connected layer and 5 convolution operation. The first 3 convolution operation is made by 5x5 kernel or filter and the last 3 convolution is made by 3x3 filter. The activation function used in our model is ELU.

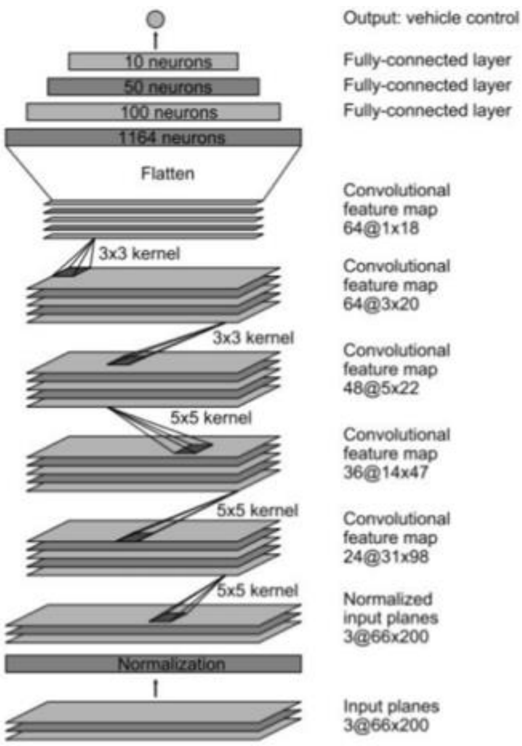


Fig 11: Nvidia Architecture.

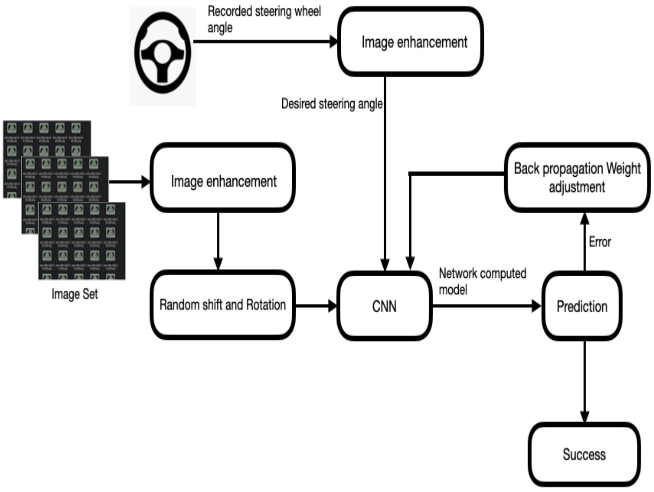
1) Structure Code

Our neural network code based on Nvidia model

```
model = Sequential(name='Nvidia_Model')

# elu=Expenential Linear Unit, similar to leaky Relu
# skipping 1st hiddel layer (nomralization layer), as we have normalized the data

# Convolution Layers
model.add(Conv2D(24, (5, 5), strides=(2, 2), input_shape=(66, 200, 3), activation='elu'))
model.add(Conv2D(36, (5, 5), strides=(2, 2), activation='elu'))
model.add(Conv2D(48, (5, 5), strides=(2, 2), activation='elu'))
model.add(Conv2D(64, (3, 3), activation='elu'))
model.add(Dropout(0.2)) # not in original model. added for more robustness
model.add(Conv2D(64, (3, 3), activation='elu'))

# Fully Connected Layers
model.add(Flatten())
model.add(Dropout(0.2)) # not in original model. added for more robustness
model.add(Dense(100, activation='elu'))
model.add(Dense(50, activation='elu'))
model.add(Dense(10, activation='elu'))

# output layer: turn angle (from 45-135, 90 is straight, <90 turn left, >90 turn right)
model.add(Dense(1))

# since this is a regression problem not classification problem,
# we use MSE (Mean Squared Error) as loss function
optimizer = Adam(lr=1e-3) # lr is learning rate
model.compile(loss='mse', optimizer=optimizer)
```

Fig 12: Architecture code

V. PROPOSED SYSTEM



V. RESULT

After completely acquire the 580 images where 480 image was used for data training and the rest of the images for data validation we have applied four fully connected layer and 5 convolution operation to getting the proper rule or command, for that we used supervised learning by passing the feature and the label on the algorithm. By using the ELU it could produces negative outputs, which helps the network nudge weights and biases in the right directions. After the CNN operation we got 95% accuracy and about 0.5 error rate.

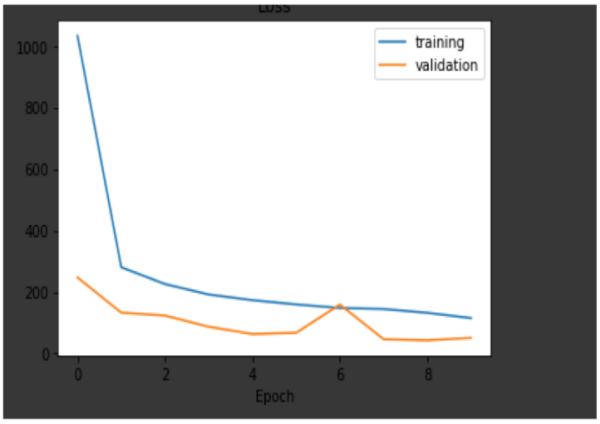


Fig 10: CNN Result

VI. CONCLUSION

There are different techniques with their own advantages and disadvantages to work in autonomous car computation . The capital aim of this analysis assignment is to propose techniques that enhance the operation and techniques for a better steering angle prediction. However for obtaining a better result the set of frames should be well processed or enhanced We could see that CNN can be effectively used for prediction of steering angle. The obtained rate of the experience results generated using actual driving images is 95% .

VII. REFERENCES

[1]. Zhilu Chen and Xinming Huang, :"End-To-end learning for lane keeping of self-driving cars", IEEE Intelligent Vehicles Symposium Proceedings, 2017.

[2]. Rodolfo Valiente ; Mahdi Zaman ; Sedat Ozer ; Yaser P. Fallah: Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks

[3]. Chen, M., Jochem, T., & Pomerleau, D. “AURORA: A vision-based roadway departure warning system,” Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995

IEEE/RSJ International Conference on Vol. 1, pp. 243-248.

[4]. Deng, J., & Han, Y. “A real-time system of lane detection and tracking based on optimized RANSAC B-spline fitting,” Proceedings of the 2013 Research in Adaptive and Convergent Systems , pp. 157-164.

[5]. Wang, Y., Teoh, E. K., & Shen, D. “Lane detection and tracking using B-Snake,” Image and Vision computing, 22(4), 2004, pp. 269-280.

[6]. Aly, M. “Real time detection of lane markers in urban streets,” Intelligent Vehicles Symposium, 2008 IEEE pp. 7-12.

[7]. Liu, W., Zhang, H., Duan, B., Yuan, H., & Zhao, H. “Vision-based realtime lane marking detection and tracking,” Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on pp49-54.

[8]. Revilloud, M., Gruyer, D., & Pollard, E. “An improved approach for robust road marking detection and tracking applied to multi-lane estimation,”Intelligent Vehicles Symposium (IV), 2013 IEEE, pp. 783- 790.

[9]. Jang, H. J., Baek, S. H., & Park, S. Y. “Lane Marking Detection in Various Lighting Conditions using Robust Feature Extraction,” International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, 2014.

[10]. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.

[11]. M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911, 2017.

[12]. A. El Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving.

[13]. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Autonomous Vehicles and Machines, Electronic Imaging, 2017. 14C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence, 35(8):1915–1929, 2013.

Cite this article as :

Aires Da Conceicao, Dr. Shashang Degadwala, "Convolutional Neural Network Computation for Steering Angle Prediction Based on Road Direction", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 7 Issue 3, pp. 290-295, May-June 2020. Available at doi : https://doi.org/10.32628/IJSRSET207323
Journal URL : http://ijsrset.com/IJSRSET207323