International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2062112

Convolutional Neural Network Computation for Autonomous Vehicle

Aires Da Conceicao[1], Dr. Sheshang D. Degadwala[2]

[1]U.G. Student, Sigma Institute of Engineering, Sigma Institute of Engineering, Vadodara, Gujarat, India

[2]Associate Professor, Sigma Institute of Engineering, 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. This technique includes line lane tracker, robust feature extraction and convolutional neural network.

Keywords: Self-Driving Car, Drive-Less Car, Intelligent Car.

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 upcoming 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. The interest are to make autonomous vehicle be able to drive in urban area and highways at such high-speed. So during this process some challenge is being faced that is how to monitor the road in different weather and outdoor conditions that affect the road appearances and those environmental condition include rain, snow and less illumination. The perception for autonomous vehicle has been approached with a wide variety of computer vision techniques. Among the methods used are cascade classifier, feature extraction, tracking methods, image flow methods and neural network

.

II. SEGMENTATION

1) Gaussian Blur

Normally the results of canny method are affected by image noise, we have to filter it out to prevent from false detection caused by image noise. A Gaussian blur technique is used to convolve with the image and effectively reduce the effects of caused by the noise on the canny method results. The equation for a Gaussian blur used is :

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

1) Canny edge detection

Edges are a set of connected pixels that form a boundary between two different region. They are representation of the change of the pixel intensity function. Canny edge detection is a technique that processes the variation of brightness on set of pixel. After reducing the noise of image edge detection algorithm is used to find the variation of gray level on the noiseless image.

The criteria for edge detection include:

- Before apply edge detection algorithm we have to convert the image from RGB into a grayscale image because RGB has 3 components and has 24 bit whereas grayscale image is 8 bit only.
- The image should be noise free by using Gaussian blur technique. In some case it is compulsory to be noise free.

Fig. 1: Grayscale Image

III. FEATURE EXTRACTION

1) Hough Transform

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 :

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

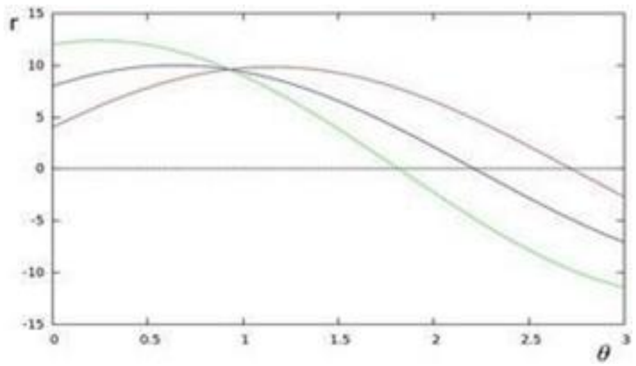

Fig 2: Hough Transform.

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

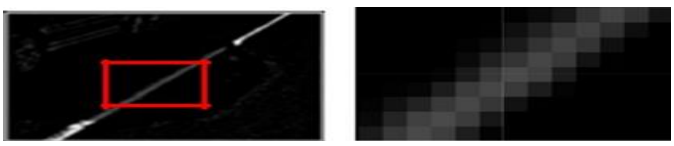

Fig 3: Feature Extraction

After finding out Hough lines from Region of Interest images and then drawn on original images.

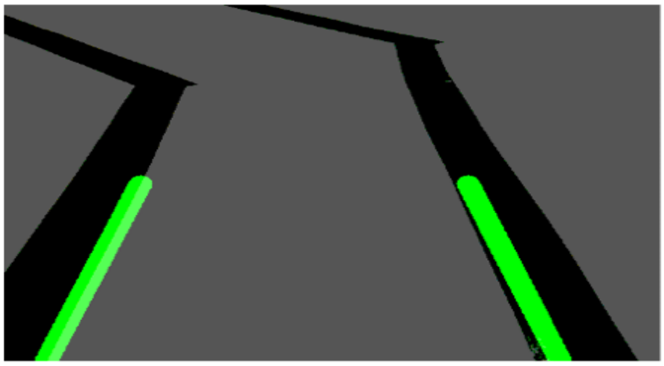

Fig 4: Cascade process.

IV. STEERING ANGLE PREDITION

Steering techniques for self-driving vehicles show how to compute the steering angle on the autonomous vehicle based on Convolutional neural network or CNN.

1) Data Collection

We have to collect a set of image for training our model , in this research we used 500 images for training purpose.

Fig 5: Image set

2) Structure

Our neural network design was based on NVIDIA architecture

```python
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 6: Architecture code

The NVIDIA architecture is based on 4 fully connected layer and 5 convolution operation

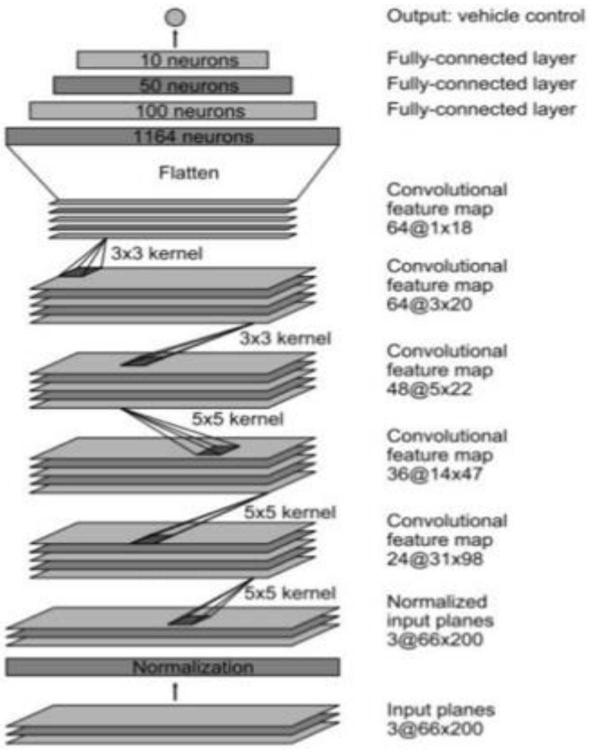

Fig 7: NVIDIA Architecture.

V. CASCADE CLASSIFIER

Cascade classifier is a technique to train the object detection algorithm. Using cascade classifier techniques you can effectively create an algorithm to detect any type of object of your wish. In this case we are working to develop an algorithm to detect the traffic signs. First of all we have to acquire a set of positive image that are the image that contain the traffic sign that we desire to detect then we have to acquire a set of negative image that are the image that does not contain the traffic sign that we want to detect. Then we need to extract features from it. For this. Each feature is a unique.

Fig 8: Image set.

1) Cascade Architecture

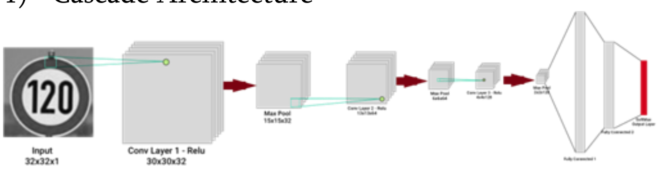

Fig 9: Cascade process.

VI. RESULT

After completely acquire the required set of positive images and set of negative images we apply a 30 layer cascaded classifier for training the algorithm to detect traffic signs. To build the detector, a set of traffic signs and non traffic signs images were used. The process includes of getting 50 labeled traffic signs image scaled to a base resolution of 24X24 pixels. Following the above process we got an output of 85% of accuracy.

For steering angle prediction we used four fully connected layer and 5 convolution operation to getting the proper rule, for that we used supervised learning by passing the feature and the label on the algorithm. By using CNN we got 95% accuracy and about 0.2 error rate

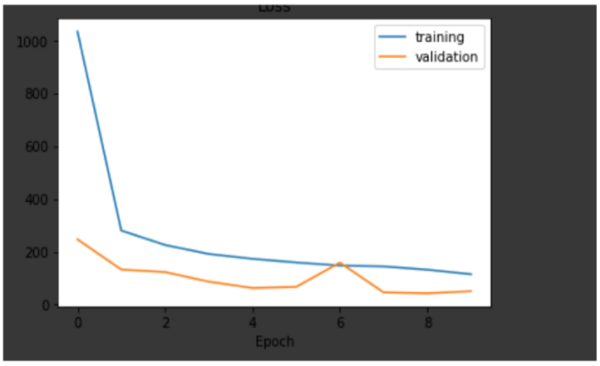

Fig 10: CNN Result

VII. 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 lane line detection at different location as highways or cities. This method obtain curved line candidates by first using the feature extraction technique to detect the lane line. Then we apply the CNN for better prediction of steering angle. The obtained rate of the experience results generated using actual driving images is 95%.

VIII. REFERENCES

[1]. Want, J.S. and Knipling, R.R. Single-Vehicle Roadway Departure Crashes: Problem Size Assessment and Statistical Description National Highway Traffic Safety Administration Technical Report DTNH-22-91-C-03121

[2]. Dagan, E., Mano, O., Stein, G. P., & Shashua, A. "Forward collision warning with a single camera," Intelligent Vehicles Symposium, 2004 IEEE, pp. 37-42.

[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]. Autonomous Vehicles and Machines, Electronic Imaging, 2017. 14 C. 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.

[14]. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672– 2680, 2014.

Cite this article as :

Aires Da Conceicao, Dr. Sheshang D. Degadwala, "Convolutional Neural Network Computation for Autonomous Vehicle", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 2, pp. 368-372, March-April 2020. Available at doi : https://doi.org/10.32628/CSEIT2062112

Journal URL : http://ijsrcseit.com/CSEIT2062112