Design and Optimization of a Deep Neural Network Architecture for Traffic Light Detection

Tomohide Fukuchi∗, Mark Ogbodo Ikechukwu, and Abderazek Ben Abdallah

1Adaptive Systems Laboratory, Graduate School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu, Japan

Abstract. Autonomous Driving has recently become a research trend and efficient autonomous driving system is difficult to achieve due to safety concerns. Applying traffic light recognition to autonomous driving system is one of the factors to prevent accidents that occur as a result of traffic light violation. To realize safe autonomous driving system, we propose in this work a design and optimization of a traffic light detection system based on deep neural network. We designed a lightweight convolution neural network with parameters less than 10000 and implemented in software. We achieved 98.3% inference accuracy with 2.5 fps response time. Also we optimized the input image pixel values with normalization and optimized convolution layer with pipeline on FPGA with 5% resource consumption.

1 Introduction

Recently, neural network have been applied in many fields such as farm monitoring [1], autonomous driving [2], natural image processing [3] and character recognition [4]. Traffic light detection is one of the important factors to prevent traffic accidents in advance [5]. To realize safety autonomous driving system, we developed a system for traffic light detection. 11,652 [6] traffic accidents due to ignorance or violation of traffic light signals were recorded in Japan in 2019. However, unrecorded traffic accidents are clearly more. Thus, traffic light detection to prevent violation of traffic light signals in advance is one of the big advantages of autonomous driving system.

Improving recognition accuracy contributes to improving safety and realizing efficient autonomous driving system. However, there are factors to consider which include the maneuverability and the power constraints of autonomous driving car environment. Thus, implementation with real time processing in compact circuit size and extremely high accuracy are necessary. Traffic light detection is also object detection, we can realize high accuracy with high speed using well known object detection algorithm such as VGG [7] base Single Shot Multibox Detector SSD [8] and You Only Look Once (Yolo) [9]. However, as we mentioned above, since this detection algorithm requires large architecture and also high power, these object detection algorithms are not suitable for traffic light detection in a power constrained autonomous driving car environment [5]. In this work, we proposed an architecture for traffic light detection. This paper contributes the design of optimized neural network for traffic light detection system both in software on CPU and hardware on FPGA. In next chapter, we introduce basic knowledge for neural network.

1.1 Artificial Neuron

Artificial neuron is a mathematical model of biological neuron. It abstracts the synapse connection, internal potential and neuron firing mechanism of biological neuron. Figure 1 shows a simple biological neuron model and equation (1) shows the mathematical model of the biological neuron model. Strength of synapse connection is represented by \( w_k \), input signal by \( x_k \), membrane potential of each axon by \( y_m \), and final output signal \( y_m \) is obtained by multiplying Input \( x_k \) and weight \( w_k \) then the bias \( b \) is added. \( n \) is number of input.

Figure 1. Biological Neuron Model
\[ y_m = \sum_{k=1}^{n} x_k w_k + b \]  

(1)

1.2 Multilayer Perceptron

Figure 2 shows an example of multilayer perceptron [5]. Multilayer perceptron has input layer, more than two hidden layer and output layer. Input image is given as one dimensional array \( x_k \) and two dimensional array weight \( w_k \). As we mentioned previously, multiplication between input \( x_k \) and weight \( w_k \) is performed. And the product is sent to the activation function. The result from the activation function is then sent to the next layer. The activation function could be, Identity activation function, used in regression problem, sigmoid function used in binary classification problem [10], or softmax activation function used in multi class classification problem [10]. In Figure 2, final classification result is represented by probability.

1.3 Convolutional Neural Network

Convolutional neural network is one of the multi layer perceptron. Figure 3 is an example of convolutional neural network LeNet-5 [11]. Convolutional neural network consists of mainly convolution layer, pooling layer and fully connected layer.

1.3.1 Convolutional Layer

Convolution layer which we developed has three channel RGB color image for traffic light image, and background image is also given as input. There are green, yellow, red and background. In convolution layer, input image is extracted as feature using convolution filter. The output image size after it is extracted using convolution filter is given by following formula (2) and (3).

\[ W_{out} = \frac{W_{in} + 2 \times \text{padding} - \text{filter}}{\text{stride}} + 1 \]  

(2)

\[ H_{out} = \frac{H_{in} + 2 \times \text{padding} - \text{filter}}{\text{stride}} + 1 \]  

(3)

Each variable used in the formula, \( W_{out}, H_{out}, W_{in}, H_{in}, \) stride, padding and filter correspond to width of output image, height of output image, row of input image, column of input image, stride size, padding size and filter size respectively. In this work, we used two convolution layer.

1.3.2 Pooling Layer

In pooling layer, row and column of image is shrinked. Pooling is robust to microscopic gap because result of pooling will be same result to small gap of image. In this work, we used max pooling layer which extract a maximum pixel value after convolution layer.

1.3.3 Fully Connected Layer

Fully Connected Layer performs flatten operation on the output from convolution layer or pooling layer is given as input. Final recognition accuracy is output as probability.

2 System Overview and Optimizations

Figure 4 [5] shows the overview of the proposed traffic light detection system. The image acquired from the camera is fed as input to the system. In region proposal phase, input image is divided into some appropriate size image blocks using slide window for image classification. In classification phase, image is classified as one of four classes; green, yellow, red and background using convolution neural network (CNN). In detection phase, traffic light is detected using trained parameters.

2.1 pragma PIPELINE

pragma HLSPIPELINE in Figure 5, is used to increase the parallelism and calculation speed of the loops required for multiplication of weight and input values. In the case of this experiment, SDx Pipeline is used in convolutional layer.

2.2 Normalization

Normalization in Figure 6 is used to make CNN work faster. In this experiment case, training and testing image values are normalized between -1 to +1 by dividing by 255.0.

3 Evaluation

3.1 Traffic Light Detection on CPU

Table 1 shows the proposed neural network architecture for traffic light detection which performed training and detection in software, using CPU on Ubuntu 16.04. We used two convolution layers and two dropout layers. In dropout layer, connected edges to neuron are chosen randomly, then the randomly chosen weight value will be zero and trained. The number of trained weights and bias is 7884. This neural network architecture is lightweight because the number of parameter is less than 10000. It is suitable for hardware implementation in terms of hardware complexity. The inference accuracy achieved 98.3 % with the 2.5 fps response time for detection. The result of prediction accuracy is preliminarily acceptable for autonomous driving system.

3.2 Hardware Complexity

In this experiment, Zynq Ultrascale MPSoC+ FPGA is used for implementation. Figure 7 shows a demonstration of traffic light image classification on FPGA. Table 2 presents the result of evaluated hardware complexity of optimized convolutional layer using Xilinx SDx. Resource consumption of BRAM18K, DSP48E, FF and LUT are respectively 50, 41, 4859 and 5866. We optimized convolution layer for traffic light detection using 5% FPGA resource.
Figure 2. Multilayer Perceptron [5]

Figure 3. Convolutional Neural Network [11]

Figure 4. Proposed System Architecture for Traffic Light Detection [5]
Table 1. Neural Network Structure for Traffic Light Detection

| Layer Name       | Image Size |
|------------------|------------|
| Input            | 64x64x3    |
| Conv2d₁          | 62x62x20   |
| MaxPooling2d₁    | 12x12x20   |
| Conv2d₂          | 10x10x20   |
| MaxPooling2d₂    | 3x3x20     |
| Dropout          | 3x3x20     |
| Flatten₁         | 180        |
| Dense₁           | 20         |
| Dropout          | 20         |
| Dense₂           | 4          |
| Activation Function | Relu     |
| Params           | 7884       |
| Accuracy (%)     | 98.3       |

Table 2. Hardware Complexity of Convolution Layers

| Resource | Utilization |
|----------|-------------|
| BRAM     | 50          |
| DSP      | 41          |
| FF       | 4859        |
| LUT      | 5886        |

4 Conclusion and Future Works

In this work, we designed and evaluated neural network for traffic light detection with different types of activation function achieving 98.3% inference accuracy. We also implemented, optimized and evaluated the hardware complexity of the traffic light classification on FPGA using only 5% of FPGA resources. In the future, we optimize execution time per image for detection with Spiking Neural Network (SNN).

5 Discussion

The classification accuracy is 98.3% with ReLu activation function and 97.9% with sigmoid activation function. In Table 3, accuracy is improved 0.4% with ReLu compared to sigmoid function. There is no big difference between the two result, ReLu activation function is suitable for hardware implementation because it is easier to implement.
Table 3. Neural Network Structure for Traffic Light Detection

| Layer Name        | Image Size | Image Size |
|-------------------|------------|------------|
| Input             | 64x64x3    | 64x64x3    |
| Conv2d1           | 62x62x20   | 62x62x20   |
| MaxPooling2d1     | 12x12x20   | 12x12x20   |
| Conv2d2           | 10x10x20   | 10x10x20   |
| MaxPooling2d2     | 3x3x20     | 3x3x20     |
| Dropout1          | 180        | 180        |
| Dense1            | 20         | 20         |
| Dropout2          | 20         | 20         |
| Dense2            | 4          | 4          |
| Activation Function | Relu    | Sigmoid    |
| Params            | 7884       | 7884       |
| Accuracy (%)      | 98.3       | 97.9       |

References

[1] R. Murarami, Y. Okuyama, A.B. Abdallah, Animal Recognition and Identification with Deep Convolutional Neural Networks for farm Monitoring, in Information Processing Society Tohoku Branch Conference, Koriyama, Japan (Feb 10, 2018)

[2] J. Kim, G. Lim, Y. Kim, B. Kim, C. Bae, Deep learning algorithm using virtual environment data for self-driving car, in 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIC) (2019), pp. 444–448

[3] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014), pp. 1724–1734

[4] T.H. Vu, R. Murakami, Y. Okuyama, A.B. Abdallah, Efficient Optimization and Hardware Acceleration of CNNs towards the Design of a Scalable Neuro inspired Architecture in Hardware, in 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (2018), pp. 326–332

[5] Y. Murakami, Y. Okuyama, A.B. Abdallah, SRAM Based Neural Network System for Traffic Light Recognition in Autonomous Vehicles, in Information Processing Society Tohoku Branch Conference, Koriyama, Japan (Feb, 10, 2018)

[6] Road traffic accidents in japan (2019), e-stat.go.jp/en/stat-search/files?page=1&layout=datalist&toukei=00130002&tstat=000001027457&cycle=7&year=20190&month=0

[7] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in International Conference on Learning Representations (2015), pp. 1–14, https://arxiv.org/abs/1409.1556

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single Shot MultiBox Detector, in Computer Vision – ECCV 2016 (2016), pp. 21–37

[9] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), pp. 779–788

[10] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation Functions: Comparison of trends in Practice and Research for Deep Learning (2018), pp. 1–20, http://arxiv.org/abs/1811.03378

[11] Y. Lecun, L. Bottou, Y. Bengio, P. Haﬀner, Gradient-based learning applied to document recognition, in Proceedings of the IEEE (1998), pp. 2278–2324