

## Light weighted Convolutional Neural Network for License Plate Recognition

Vibha Pandey<sup>1</sup>, J.P. Patra<sup>2</sup>, Siddhartha Choubey<sup>1</sup> Abha Choubey<sup>1</sup>

<sup>1</sup>Shri Shankaracharya Technical Campus, Bhilai, Chattisgarh, India

<sup>2</sup>S.S.I.P.M.T., Bhilai, Chattisgarh, India

**Article History:** Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 May 2021

**Abstract:** To achieve significant outcomes in license plate recognition is always a challenging task for the researchers. This paper proposes a light weight convolutional neural network without initial segmentation of characters. Mainly, this work is inspired by current revolutions in deep neural networks. Further, it also does a good job in real-time with accuracy of recognition fit for 96% for Indian plates on GPU (Graphics Processing Units) and Multicore CPUs. For the training point of view endwise, light weighted convolutional neural network model have been proposed. Main advantage of this model is that it doesn't use RNN (Recurrent Neural Network). The proposed method further can be implemented to produce embedded solutions for license plate recognition systems that feature high level precision on challenging Indian license plates as well.

**Keywords** – License plate recognition, Deep Neural Network, Deep Models, Convolution Neural Network.

### I. Introduction

With increased number of vehicles on the road, in such modern world of population and lack of space for vehicles movement in the road became genuine global problems of traffic management system. So that automation of this issues is highly required. In this current era of technologies, to relieve from this problem in a great extent, automatic licence plate recognition is a key computer vision solution in traffic surveillance. The other areas of LPR include traffic management, parking management, highway toll collection, vehicle recognition, security issues etc [1]. Now-a-days, the smart cities in the urban areas installed network of cameras in different road junctions to capture the vehicle movement. But the big issues to capture the license plate includes small area of licence plate, varying of size, fonts, colour, dimension and number of lines in the licence plate with various background clutters as shown in Fig. 1 [1,2].

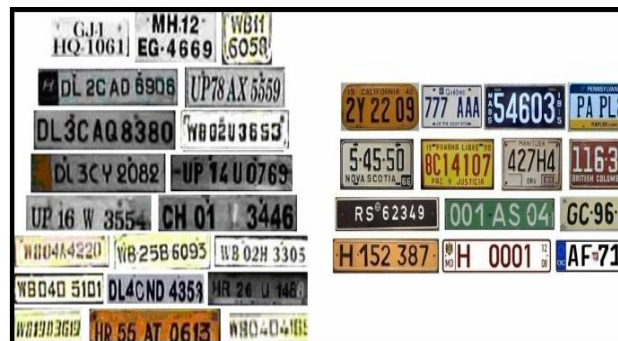


Fig. 1. Samples of licence plate with varying in size, font, colour, dimension and number of lines in the licence plate. Left side of India and right side of other countries. (Image source: researchgate.net)

To overcome these issues, deep convolutional neural network goes further steps ahead, due to its trending and current success in various domains for object recognition and classification. This LPR system also suffers from some false positive cases due to variation in licence plate which can be significantly reduced by different CNN based classifier. We have deployed CNN for both from detection of license plate to recognition of characters present in it. Mainly, this task is an intricate delinquent due to numerous issues such as inadequate conditions of lighting, blurry images, inconsistency of numbers of the license plates, physical impact (deformations), weather conditions and many more. The robust LPR system wishes to manage with a diversity of environments although preserving accuracy. In different words we can say that a system should work well in natural circumstances. In this paper, a light weighted model has been implemented, it is designed in such a way that pre segmentation and subsequent recognition of characters are not needed. Further, license plate detection problem has not considered and in our case, it can be implemented by LBP-cascade. As in [3], authors have implemented LPRNet which is a light weighted deep

convolutional network. We have also focused on such type of network. If we talk about LPRNet as mentioned by [3], Tomakeasingleforwardpass it receiptsonly0.34GFLOps. In our implemented model this is real-time on Intel Core i7 CPU processor with good precision on license plates of India and further it can be trained endwise. Additionally, this light weighed implemented may be in part ported on FPGA. That can make available power of CPU for further slices of the pipeline. Our core implemented contributions can be concise as follows: In this paper we have developed light weight convolutional neural network that works on character independent variable length Indian license plates. Pre segmentation is not needed in our case. Further, the model is trainable endwise from the scratch for different Indian plates. Also, it does not use RNN (Recurrent Neural Network) and this model's light weight characteristics make it to run on different platform.

Further, this paper is organized as follows section II discusses about the literature survey implemented on LPR in the recent years. Section III describes about our implemented model. Results have been discussed in section IV. Lastly, conclusion is given in the section V.

## II. Literature Survey

Till now, several researches have been done based on automatic license plate recognition such as in [1] authors proposed CNN based methodology for automatic LPR system. They classified their methodology mainly into two parts namely LP Detection and LP Recognition and both the parts follow CNN based deep neural network for detection and recognition respectively. In each section of their systematic research work, visibly, related work, proposed method, experimental results also follow this structure. They created a newly dataset of Indian vehicles, from the diverse form of Indian license plates. For LP detection they got 99.36% accuracy for license plate and 99.68% for non-license plate. Similarly, for LP recognition, the accuracy for single line and double line license plate are 92% and 93% respectively.

The research study in [2], states that the whole LPR proposed work can be solved by basically in two ways, first, vehicle region detection followed by candidate region generation, in which they try to localize the LP from the each vehicle region. They follow a deep learning based faster R-CNN algorithm for object detection i.e. vehicle region detection and CNN based classifier for candidate region generation. In their experimental result, they use Caltech Cars (Real) 1999 dataset and the proposed method got precision of 98.39 and recall of 96.83 respectively.

The work in [4] suggests the use of image contour properties and deep learning technologies. The detection of LP is based on morphological processing and Gaussian smoothing, adaptive thresholding and filtering contour properties of the characters in the lower part of the plate. The recognition of LP is carried out by multilayer Convolution Neural Networks (CNN) deep learning model. The algorithm showed good performance in processing various types of images such as rainy, rotated, different illuminations and low contrast.

A cloud based deep learning model is proposed on [5], in which a heuristic based convolution image manipulation technique is adapted for detection. The extracted license plate is binarized with several binarization techniques like Savoula for improving the image contrast that is beneficial for accuracy of number recognition. There are some patterns like street sign, window-based pattern which make most of LPDS to get failed in detection. This proposed algorithm detect these patterns however does not recognize them as license plate which is one of the superiority of the introduced scheme in compare with other LPDS. Further, in [6] the pipeline contains of classification of characters and character segmentation stages. For extraction and segmentation of characters unrelated combining projections, handmade algorithms, connectivity and contour-based image components have been implemented which receipts a binary images or transitional depiction as input so segmentation of characters quality is extremely exaggerated by their input image noise, low resolution, blur or deformations. For character classification, typically optical character recognition methods have been used. Subsequently classification methods trail the character segmentation, endwise recognition quality hang on profoundly on the applied segmentation method. In this paper, to solve the issue of character segmentation CNNs based solutions have been considered taking the entire image as input and constructing the output sequence of characters. In [7], authors have implemented segmentation free model with flexible length sequence mainly determined by connectionist temporal classification (CTC) loss [8,9]. Basically, this uses LBP based features on a binary image as input to produce characters. That is applied to entirely input image locations by the mechanism of sliding window method which makes the input sequence for the bi-directional Long-Short Term Memory (LSTM) [10] based decoder. In [11], authors have applied the same model as described in [7] apart the sliding window mechanism is replaced by CNN. [12] implemented the CNN-based model for the whole license plate image to generate the global license plate embedding that is decoded to a length of 11 character sequence by 11 fully connected model. Every head is further trained to classify the target string character; hence the whole recognition task can be implemented in a solitary feed-forward pass. Further, It also exploits the STN (Spatial Transformer Network) [13] to decrease the outcome of input image deformations. The method in [14] tries to resolve

both license plate detection and license plate recognition problems by using sole Deep Neural Network. In [15], author tries to develop synthetic data approach based on Generative Adversarial Networks [16] for generation of data procedure. In our model, we dodged using extracted features over a binary image - as an alternative we have considered raw RGB pixels of license plate as input of CNN. Further, the LSTM built sequence decoder employed on outcomes of a sliding window CNN has substituted with a FCNN (fully convolutional neural network) model which outcome is construed as probabilities of characters sequence for CTC loss training and greedy or prefix search string inference. For improved performance point of view, the pre-decoder midway feature map was amplified by the global context embedding as discussed in [17]. Further, as the backbone CNN model was abridged expressively using the truncated computation cost basic building block mainly inspired by SqueezeNet as suggested in [18] and Inception model as given in [19, 20, 21]. Moreover, Batch Normalization [22] and Dropout [23] techniques have implemented for regularization. Size of input license plate image input disturbs both the recognition quality and the computational cost [24]. Hence, there is a trade-off among using moderate [12, 7] or high [11] resolution.

### III. Model Architecture

Here we are going to discuss the implemented architecture. In the current literatures we have found that different CNN architectures are frequently used such as Rednet, VGG 13, VGG 16, GoogleNet and many more. Basically they are based on transfer learning techniques. In this paper we have applied some significant modification in transfer learning method to make efficient architecture.

Our implemented CNN model consists of the following attributes. For the input layer dimension is like input  $x$  height  $x$  width features map, for convolution layers we have used different window size such as  $1 \times 1$ ,  $1 \times 3$ ,  $3 \times 1$  with different stride and padding. Further, output is again having output  $x$  height  $x$  width feature map. Mainly it is stimulated with SqueezeNet Fire Blocks and Inception block used by researchers in [18,19,20,21]. Further we have used batch normalization as well and activation function ReLU after every convolution layer. Overall structure of our model design consist of lightweight CNN, per position character classification head, different procedure for post filtering, probabilities for further sequence decoding. As suggested by the researcher in [13], Spatial transformation have been implemented as the preprocessing steps. Mainly it is done for the better features for recognition. To evaluate optimal transformation parameters LocNet architecture have been implemented in which RGB images have been taken, for average pooling  $3 \times 3$  window size with stride 2, for convolution 32 filters with size  $3 \times 3$ , concatenation have been done channel wise, dropout ratio is 0.5.

Basically, the pillar network model is stated in the following figure.

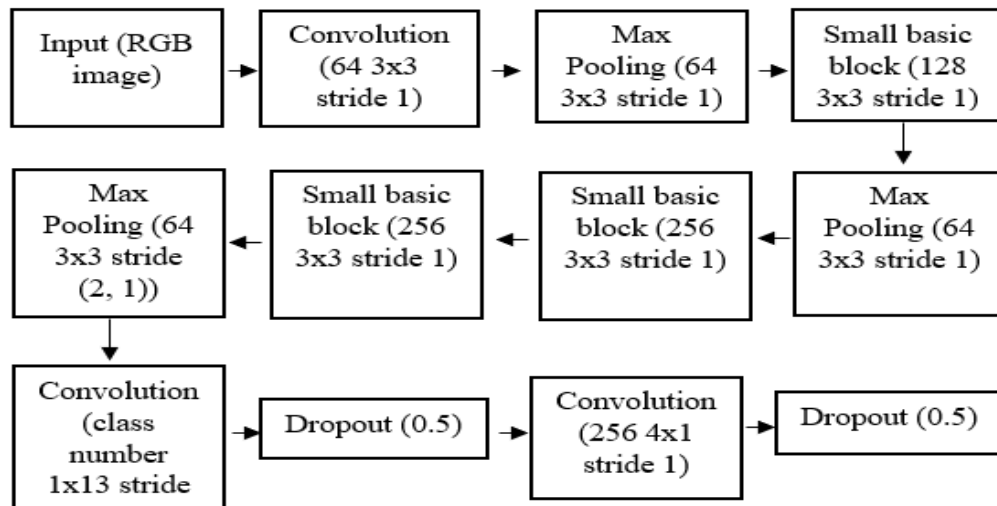


Fig.2. Network Architecture

In this model a raw RGB images have been taken as input and further it computes spatially dispersed rich features. Extensive convolution i.e. with the size of  $1 \times 13$  employs local context in its place of using LSTM-based RNN. If we consider sub network outcome, it can be considered as a sequence of probabilities of character whose length resembles to the pixel width of the input image. We have applied the method of CTC loss as [25] for segmentation free end on training because output of decoder and the length of target character sequence are of diverse length.

Basically CTC loss is the method mainly used if the input and output are having variable length and skewed. Furthermore, it provides an well-organized method to move with probabilities at each time step to the output's sequence probability.

As defined in [17], the pre-decoder in-between feature map have been amplified by the global context entrenching for the performance improvement. Further, it is calculated by the FCNN over the output, overlaid to the desired size. Then it is concatenated with the backbone output. For the adjustment of the deepness of the feature map to the character class number extra  $1 \times 1$  convolution operations have been applied.

To implement decoding method, we have used two methods beam and greedy search. In greedy search, maximum of class probabilities has been taken in each position. While in the case of beam search, it maximizes the over all probability of the output sequence [8,9].

Further, for the post-filtering an efficient task-oriented model has implemented as the set of the templates of target country license plates. This post-filtering method implemented with beam search. Also this method grows topmost plausible sequences found by originate through beam search and returns the initial one which matches the set of predefined models that depends on regulations of country license plates.

### Training details

All training has been implemented with the help of TensorFlow. We have used 'Adam' optimizer using batch size of 32 for training point of view, initial learning rate 0.001 has been taken and further gradient noise scale of 0.001. We have thrown down the learning rate after every 100k iterations with the factor of 10 and further train our network for 250k iterations in whole. In our experiments we have also used augmentation of data via different random affine transformations, such as scaling, rotation, translation and shift. It is valuestating, that application of the architecture LocNet from the starting of training primes to deprivation of outcomes, since LocNet model can't get rational gradients from a recognizer which is characteristically too weedy for the initialsomerepetitions. So, in our work, we have turned LocNet on only after 5k repetitions. Further, All other different hyper-parameters have been taken by cross validation over the target dataset.

### IV. Results Obtained

Our implementation is encouraged by the researchers worked in [3, 7] i.e. LPRNet baseline network and we have worked with different models and architectures. Specifically, it is based on different inception blocks tracked by a bidirectional long short-term memory (LSTM) decoder further that is trained with CTC loss. First, we have performed few experiments for replacing substituting biLSTM by biGRU cells, however we did not perceive any strong profits of using biGRU over biLSTM. Further, we mainly intensive on eradicating of the intricate biLSTM decoder, since maximum current embedded devices still are not having adequate memory and compute to proficiently perform biLSTM. In our case LSTM has been applied to a spatial sequence relatively to a temporal one. Thus, entirely LSTM inputs are identified upfront both at the at the inference stage as well as training stage. Hence, we trust that Recurrent neural network(RNN) can be substituted by spatial type of convolutions without a noteworthy drop in precision. If we talk about RNN-less model with some backbone alterations is referenced as LPRNet which has been described in details in above sections. Further, we have also modified LPRNetbasic model to improve runtime performance by using  $2 \times 2$  strides for all different pooling layers. This kind of alteration decreases the size of middle feature maps and total inference computational cost expressively. We can refer it into the table 4.

### Indian License Plates dataset

We have performed our method on the Indian License Plate data set. Firstly, this data set was run over the detector to find bounding boxes for every license plate. Further, all license plates have been labeled by hand. Over and all, total dataset is having 15636 cropped LP (license plate) images. Then data set have been fragmented as 9:1 into training and validation subsets correspondingly.

For training point of view, automatically cropped license plate images have been taken. Mainly, to develop the network extra strong to detect present noise and artifacts. Detection of the artifacts is necessary in some cases plates are cropped with some background edges. Whereas in additional scenario images are cropped moreover adjacent to edges with no background at all or event with few parts of the missing license plate.

Following table 1 displays accuracies of recognition attained by different models.

Method	Recognition Accuracy, %	GFLOPs
Network baseline	94.6	0.72
Network basic	95.2	0.44
Network reduced	93.9	0.152

Table 1. Results on Indian License Plates.

For the study of abstraction during the study, following table 2 demonstrate different architecture approaches and their effect on accuracy basically this study has been done to identify correlation among various enhancement implemented and their corresponding improvements.

Approach	Model Architecture							
Data augmentation	Y	Y	Y		Y	Y	Y	Y
Beam Search			Y				Y	Y
Global Context				Y	Y	Y	Y	Y
STN-alignment		Y	Y			Y	Y	Y
Post-filtering			Y					Y
Accuracy, %	52.3	59.1	58.263	692.793	895.3	96.0		

Table 2. Paraphernalia of various tricks on the implemented model.

As we can perceive from the table that the highest gain in accuracy obtained by using the global context. Further, the techniques of different augmentation of data also support to progress accuracy expressively. Deprived of using global context and the data augmentation, it’s difficult to train the model from the scratch.

The STN-based alignment sub network delivers conspicuous perfection of 2.7-5.4%. Further, Beam Search with post filtering progresses recognition precision by 0.3-0.7%.

**Performance Analysis**

This light weight model has been implemented to numerous hardware architecture such as platforms including Central Processing Unit (CPU), Graphics Processing Unit(GPU) and FPGA(A field-programmable gate array). The results obtained are given below in the Table 3.

Hardware Architecture	Processing time
GPU + Deep Neural Network Library(cuDNN)	4millisecond
CPU (using Caffe as given in [27])	12-15 millisecond
CPU + FPGA (using DLA as given in [28])	3.8millisecond (reduced model used)
CPU (using IE from Intel OpenVINO as given in [29])	1.6millisecond

Table 3. Analysis of Performance

Here GPU is nVIDIA GeForce940M, CPU is Intel Core i7 Processor, FPGA is Intel Arria10 and IE is implicated Inference Engine from Intel OpenVINO.

**5. Conclusions and Future Work**

In this work, light weighted CNN (Convolutional Neural Network) has been instigated for automatic license plate recognition system. This light weighted model can be used for challenging data and we have achieved accuracy upto 96%. Further, in this paper model’s architectural details, its impetus and the ablation study have been shown.

Further, we presented that this kind of architecture may accomplish implication in real-time on a diversity of different hardware designs such as GPU, CPU and FPGA. Also, such model can achieve real-time performance better on additional dedicated embedded devices with low-power. This light weighted model can prospective be trampled using current different methods of quantization and pruning, that would theoretically support to decrease the further computational complexity. For the future point of view, such light weighted model can be prolonged by

integration of light weighted CNN model-based recognition part into our method, so that both detection and recognition errors will be appraised as a sole network.

#### References

1. Zhang, X., Gu, N., Ye, H., & Lin, C. (2018). Vehicle license plate detection and recognition using deep neural networks and generative adversarial networks. *Journal Of Electronic Imaging*, 27(04), 1. doi: 10.1117/1.jei.27.4.043056
2. Kim, S., Jeon, H., & Koo, H. (2017). Deep-learning-based license plate detection method using vehicle region extraction. *Electronics Letters*, 53(15), 1034-1036. doi: 10.1049/el.2017.1373
3. Zherzdev, S. and Gruzdev, A. (2018). Lprnet: License plate recognition via deep neural networks. arXiv preprint arXiv:1806.10447.
4. Abedin, Md Zainal, et al. "License plate recognition system based on contour properties and deep learning model." *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2017.
5. Polishetty, R., Roopaei, M., & Rad, P. (2016). A Next-Generation Secure Cloud-Based Deep Learning License Plate Recognition for Smart Cities. 2016 15th IEEE International Conference On Machine Learning And Applications (ICMLA). doi: 10.1109/icmla.2016.0054
6. Anagnostopoulos, C., Anagnostopoulos, I., Psoroulas, I., Loumos, V., & Kayafas, E. (2008). License Plate Recognition From Still Images and Video Sequences: A Survey. *IEEE Transactions On Intelligent Transportation Systems*, 9(3), 377-391. doi: 10.1109/tits.2008.922938
7. Li, H., Wang, P., You, M., & Shen, C. (2018). Reading car license plates using deep neural networks. *Image And Vision Computing*, 72, 14-23. doi: 10.1016/j.imavis.2018.02.002
8. Graves, A. (2014). *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin: Springer Berlin.
9. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376. 2, 3
10. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735
11. T. K. Cheang, Y. S. Chong, and Y. H. Tay, "Segmentation free Vehicle License Plate Recognition using ConvNet- RNN," arXiv:1701.06439 [cs], Jan. 2017, arXiv: 1701.06439. 2
12. Jain, V., Sasindran, Z., Rajagopal, A., Biswas, S., Bharadwaj, H. S., & Ramakrishnan, K. R. (2016). Deep automatic license plate recognition system. *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing - ICVGIP 16*. doi: 10.1145/3009977.3010052
13. M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," arXiv:1506.02025 [cs], Jun. 2015, arXiv: 1506.02025.
14. Li, H., Wang, P., & Shen, C. (2019). Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3), 1126–1136. doi: 10.1109/tits.2018.2847291
15. X. Wang, Z. Man, M. You, and C. Shen, "Adversarial Generation of Training Examples: Applications to Moving Vehicle License Plate Recognition," ArXiv e-prints, Jul. 2017.
16. Rocca, J. (2019, August 25). Understanding Generative Adversarial Networks (GANs). Retrieved from <https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29>.
17. W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking Wider to See Better," arXiv:1506.04579 [cs], Jun. 2015, arXiv: 1506.04579
18. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size," arXiv:1602.07360 [cs], Feb. 2016, arXiv: 1602.07360.
19. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," arXiv:1602.07261 [cs], Feb. 2016, arXiv: 1602.07261.
20. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," arXiv:1409.4842 [cs], Sep. 2014, arXiv: 1409.4842.
21. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," arXiv:1512.00567 [cs], Dec. 2015, arXiv: 1512.00567.
22. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167 [cs], Feb. 2015, arXiv:1502.03167.

23. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach.*
24. Agarwal, S., Tran, D., Torresani, L., & Farid, H. (2017). Deciphering Severely Degraded License Plates. *Electronic Imaging*, 2017(7), 138–143. doi: 10.2352/issn.2470-1173.2017.7.mwsf-337
25. Hannun, A. (2017). Sequence Modeling with CTC. *Distill*, 2(11). doi: 10.23915/distill.00008
26. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," arXiv:1603.04467 [cs], Mar. 2016, arXiv: 1603.04467.
27. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014
28. U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An OpenCL(TM) Deep Learning Accelerator on Arria 10," arXiv:1701.03534 [cs], Jan. 2017, arXiv: 1701.03534.
29. Admin. (2019, November 20). Intel® Distribution of OpenVINO™ Toolkit. Retrieved from <https://software.intel.com/en-us/opencvino-toolkit>.