International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

Object Detection from Images using Convolutional Neural Network based on Deep Learning

Md. Mehedi Hasan Naim[1], Rohani Amrin[2], Md. Romzan Ali[3], Abdullah Al Zubaer[1] , Md. Ariful Islam[3]
[1]Lecturer, Department of Computer Science and Engineering, Rabindra Maitree University, Kushtia, Bangladesh
[2]Lecturer, Department of Information and Communication Technology, Rabindra Maitree University, Kushtia, Bangladesh
[3]Lecturer, Department of Electrical and Electronic Engineering, Rabindra Maitree University, Kushtia, Bangladesh

***Abstract*: According to the object detection definition object detection can be defined by identifying different objects automatically from image files. Implementing by multiple deep learning technique, many problems which occur frequently and disturb the accuracy can be improved. Convolutional neural network are currently the state of the art solution for object detection. To improve and test object detection system is the main task of this project. This system is applied for images based on convocational neural network. In this arena there are two parts. From theoretical part, relevant literature and how convocational neural network improved computer vision are studied and from the experimental part how easily a convocational neural network can be implemented for object detection will be shown.**

Keyword:* *Convocational neural network, Deep learning, Cifar-10, Dataset description, Object detection.

I. INTRODUCTION

There is a huge amount of image data in the world, and the rate of growth itself is increasing. Before around 2012, a dataset was considered relatively large if it contained 100+ images or videos. Now, datasets exist with numbers ranging in the millions. Many of these images are stored in cloud services or published on the Internet. Over 1.8 billion images were uploaded daily to the most popular platforms, such as Instagram and Facebook.We need to have some effective ideas about its contents to manage all of this data [2]. Automated processing of image contents is useful for a wide variety of image-related tasks. For computer systems, this means crossing the so-called semantic gap between the pixel level information stored in the image and the human understanding of the same images. Computer vision attempts to bridge this cap [3]. Object detection from repository of images is challenging task in the area of computer vision.

Lately, a lot of work has been employed in the object detection. CNN's have a high computational cost in terms of memory and speed in the learning stage, but can achieve some degree of shift and deformation invariance [5]. Nowadays, this approach became more feasible thanks to the hardware evolution and the capable of using the GPU processors to perform convolutions and the large amount of available data that allows the learning of all CNN's parameters [13]. This network type has demonstrated being able to achieve high recognition rates in various image recognition tasks like character recognition, handwritten digit recognition; object detection, and facial expression recognition [6]. Although there are many methods in the literature, some aspects still deserve attention, for example, accuracy is somewhat low in and validation methods could be improved and the recognition time could be a little improved to be performing in general [8].

II. LITERATURE REVIEW

I. R. J. Cintra, S. Duffner, C. Garcia, and A. Leite [1] (2018): We present an approach for minimizing the computational complexity of trained Convolutional Neural Networks (ConvNet). The idea is to approximate all elements of a given ConvNet and replace the original convolution filters and parameters (pooling and bias coefficients; and activation function) with efficient approximations capable of extreme reductions in computational complexity.

II. A. Dundar, J. Jin, B. Martini, and E. Culurciello [4] (2017): Deep convolution neural networks (DCNNs) have become a very powerful tool in visual perception. DCNNs have applications in autonomous robots, security systems, mobile phones, and automobiles, where high throughput of the feed forward evaluation phase and power efficiency is important.

III. C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao [7] (2015): Today, there are two major paradigms for vision-based autonomous driving systems: mediated perception approaches that parse an entire scene to make a driving decision, and behavior reflex approaches that directly map an input image to a driving action by a regressor. In this paper, we propose a third paradigm: a direct perception approach to estimate the affordance for driving.

IV. C. Wojek, P. Dollar, B. Schiele, and P. Perona [9] (2012): Pedestrian detection is a key problem in computer vision, with several applications that have the potential to positively impact quality of life. In recent years, the number of approaches to detecting pedestrians in monocular images has grown steadily. However, multiple data sets and widely varying evaluation protocols are used, making direct comparisons difficult. To address these shortcomings, we perform an extensive evaluation of the state of the art in a unified framework.

V. P.F. Felzenszwalb, R.B. Girshick, D. Mcallester, and D. Ramanan [11] (2010): We describe an object detection system based on mixtures of multi scale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL data sets. Our system relies on new methods for discriminative training with partially labeled data.

III. METHODOLOGY

Back propagation is defined by the way the computer is able to adjust its filter values (or weights) by applying a filter process. Among different ways, a common loss function is MSE (mean squared error) which means 1/2 times (actual predicted) squared.

$$L = \sum \frac{1}{2}(a_p - y_p)^2$$

Where,

L=Total Error

A_p=Target probability

Y_p=outcome probability

For reducing loss the main task is trying to adjust the weights. If the loss is increasing, the derivate of the loss with respect to the weight will be computed. dl/dw is the mathematical equivalent of this where w are the weights at a particular layer. Now the process of loss decreasing is applying by the backward pass through the network which determines which weights contribute most of the losses and then finding ways to adjust them. After completing the computation of the derivative, the last step called weight update can be moved by us. Not only this but also for changing the gradient in the opposite direction all the weights of the filter are taken and updated.

$$w = w_i - \mu\left(\frac{dL}{dw}\right)$$

Where,

W= weight

W_i =Initial weight

μ=learning rate

One of the parameter is learning rate that is chosen by the programmer. When bigger steps are taken in the weight update it is called high learning rate. As a result for making the model to converge on an optimal set of weights it may take less time. Moreover, jumping can be the result where the learning rate is too high and large and not explicit enough to reach the optimal point [10].

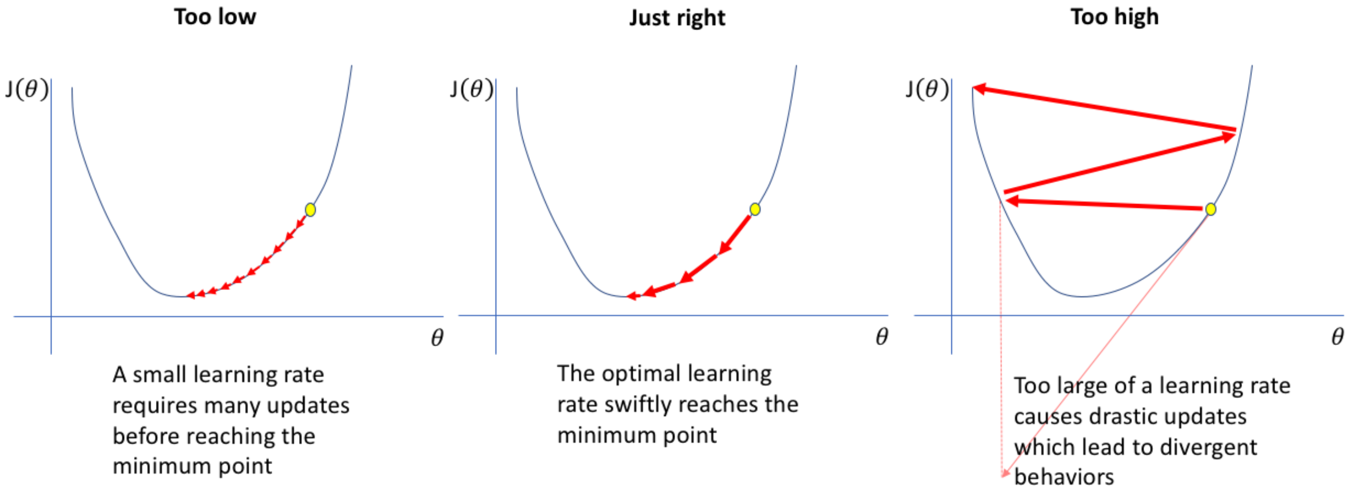

Figure-1: Selecting optimal learning rate for gradient decent

The overall training process of CNN can be summarized as below: Step in the training process of CNN

Step 1: Initialize all filters and parameters/weights with random values.

Step 2: The network takes a training images as input, goes through the forward propagation step (convolution, Relu and pooling operations along with forward propagation in the fully connected layer) and finds the output probabilities for each class.

Step 3: Calculate the total error at the output layer

Total error = ∑ ½ (target probability –output probability)^2

Step 4: Use Back propagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent to update all filter values / weights and parameter values to minimize the output error.

Step 5: Repeat steps 2-4 with all images in the training set.

IV. MODEL DESIGN AND IMPLEMENTATION

In this method, CIFAR 10 dataset is trained by us, through using convolution neural network. 10 unique data-classes are contained by CIFAR 10 dataset. That is why this model is trained only for these 10 objects. The dimension of the image of our dataset is 32*32. For extracting data from input and fitting the model different convolution layers are used.

Dataset description

For training the model CIFAR-10 data set is used by me which contains 32×32 color images in 10 classes where per class have 6000 images. As a result in total it has 60000 images. Not only is this but also their 50000 training images and 1,000 test images. Five training batches are the divisions of the data set where each batch contains 10,000 images. Here, exactly 1000 randomly selected images are contained by the test batch and the remaining images are contained by the training batches in random order.

Figure-2: CIFAR-10 Dataset

The one more thing here is that more images from one class then another can be contained by some training batches. But among them exactly 5,000 images are contained by the training batch from each class.

The network

Deep CNN with four convolution layers and two fully connected layers are trained. The first convolution layer had 32 32 32 filters, the second one had 64 16 16 filters, and the last one had 128 4 4 filters.

Layer(type)	Output Shape
conv2d_1	(32, 32, 32)
max_pooling2d_1	(32,16,16)
dropout_1	(32, 16, 16)
conv2d_2	(32, 16, 16)
max_pooling2d_2	(32, 16, 16)
dropout_2	(64, 8, 8)
conv2d_3	(128, 8, 8)
max_pooling2d_3	(128, 4, 4)
dropout_3	(128, 4, 4)
flatten_1	(2048)
dense_1	(80)
dropout_4	(80)
dense_2	(10)

Table -1: Different layer and their dimensions used on this project

From the entire convolution layers, we have a stride of size 1, batch normalization, dropout, max pooling and relu as the activation function. 80 neurons are contained by the hidden layer in the first FC layer. Batch normalization, dropout and relu are also used in FC layer which are also similar in the CN layers. Additionally we also used soft max as our loss function. Table-1 shows the architecture of this deep network.

V. RESULTS

To show the performance of the deep CNN model, we plotted the loss history and the obtained accuracy for the model. Figures-3 and 4 exhibit the results. As seen in Figure-3, the deep network validation accuracy is 84.89 %.

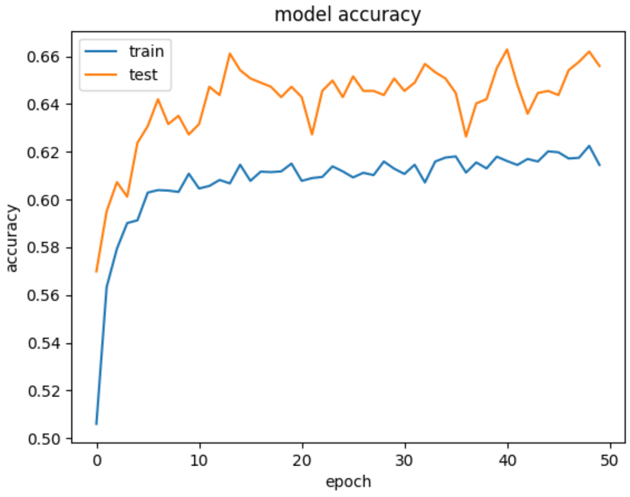

Figure-3: Accuracy of training and validation data set

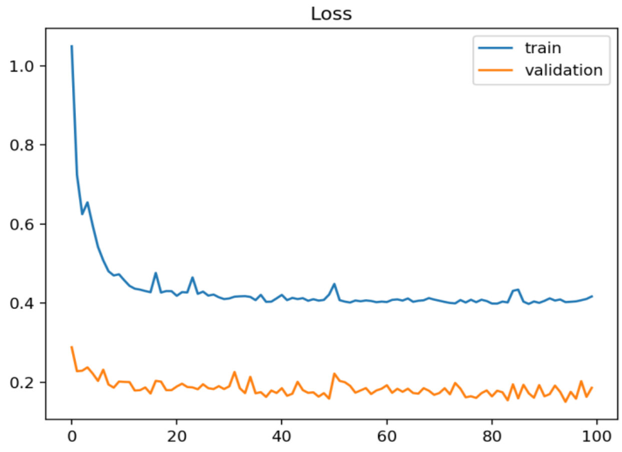

Figure-4: loss history of training and validation data set

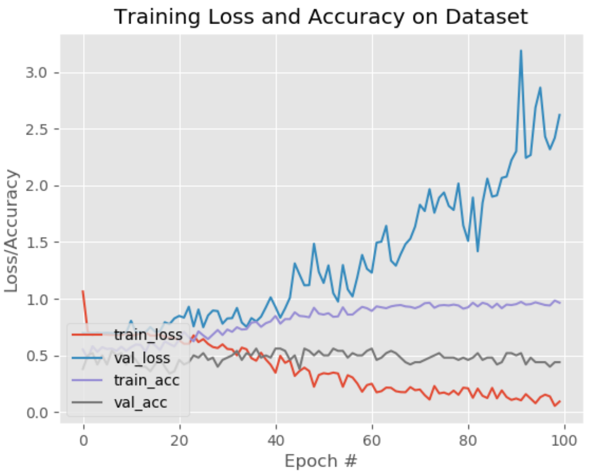

Figure-5: Training Loss and Accuracy on data set

Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing. We computed the confusion matrices for our deep CNN. Table-2 presents the visualization of the confusion matrices. As demonstrated, the deep network results in higher true predictions for most of the labels.

Null	**biman**	**car**	**chicken**	**cat**	**deer**	**dog**	**toad**	**ghora**	**boat**	**bus**
airplane	600	5	78	88	55	120	12	9	117	170
automobile	14	690	10	59	15	27	70	60	35	75
bird	35	0	528	155	13	89	31	12	8	0
cat	10	1	19	662	91	175	19	19	3	0
deer	7	0	38	125	811	51	21	33	9	0
dog	5	0	18	231	52	701	9	11	9	1
frog	1	1	30	88	100	49	647	3	6	1
house	6	0	18	120	120	106	5	647	4	0
ship	44	9	25	86	13	18	6	4	791	7
truck	30	29	5	65	31	29	2	19	59	706

Table-2: Confusion Matrix for object detection system

VI. CONCLUSION

From the above description both an experimental and theoretical experiences are revealed about the development of a object detection system The most precious matter is that GPU has shown better performances by applying deep convolution network trained on it except the higher cost factor of it. Learning rate, the number of layers, the type of each layer, the area size of convolution and pooling and so on hyper parameters are optimized by the deep architecture. On the other sides, grid search and manual search are used for hyper parameter optimization. Furthermore, for ensuring the main motive of object detection above model can also be utilized.

VII. FUTURE WORK

In the future work, we would like to plan to use more advanced network that will be helpful to
Train deep architectures and allow us to investigate the accuracy of our object detection system. Image localization was left outside the scope of the project due to not having GPU.

REFERENCES

[1] R. J. Cintra, S. Duffner, C. Garcia, and A. Leite, vol. PP, no. 99, pp. 1–12, 2018, "Low-complexity approximate convolutional neural networks," IEEE Trans. Neural Netw. & Learning Syst.

[2] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Shovel, and R. Togneri, vol. PP, no. 99, pp. 1–15, 2017, "Cost-sensitive learning of deep feature representations from imbalanced data." IEEE Trans. Neural Netw. & Learning Syst.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, 2017, "Realtime multi-person 2d pose estimation using part affinity fields," in CVPR.

[4] A. Dundar, J. Jin, B. Martini, and E. Culurciello, vol. 28, no. 7, pp. 1572–1583, 2017 ,"Embedded streaming deep neural networks accelerator with applications," IEEE Trans. Neural Netw. & Learning Syst.

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, 2017 , "Multi-view 3d object detection network for autonomous driving," in CVPR.

[6] Z. Yang and R. Nevatia, 2016, "A multi-scale cascade fully convolutional network face detector," in ICPR.

[7] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, 2015 "Deepdriving: Learning affordance for direct perception in autonomous driving," in ICCV.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, in ACM MM, 2014, "Caffe: Convolutional architecture for fast feature embedding".

[9] C. Wojek, P. Dollar, B. Schiele, and P. Perona, vol. 34, no. 4, p. 743, 2012, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. Intell.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 2012 , "Imagenet classification with deep convolutional neural networks," in NIPS.

[11] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, vol. 32, no. 9, p. 1627, 2010, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell.

[12] Felzenszwalb, P. F., and Huttenlocher, D. P. E_2 (2004), 167-181, cient graph based image segmentation. International journal of computer vision 59.

[13] K. K. Sung and T. Poggio, vol. 20, no. 1, pp. 39–51, 2002, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell.

[14] H. Kobatake and Y. Yoshinaga, vol. 15, no. 3, pp. 235–245, 1996, "Detection of spicules on mammogram based on skeleton analysis." IEEE Trans. Med. Imag.