Convolutional Neural Network (CNN) for Image Detection and Recognition

Conference Paper · December 2018

DOI: 10.1109/ICSCCC.2018.8703316

3 authors, including:

Rahul Chauhan
Graphic Era Hill University
18 PUBLICATIONS **141** CITATIONS
SEE PROFILE

R. C. Joshi
250 PUBLICATIONS **2,952** CITATIONS
SEE PROFILE

Convolutional Neural Network (CNN) for Image Detection and Recognition

Rahul Chauhan
Graphic Era Hill University
Dehradun, India
chauhan14853@gmail.com

Kamal Kumar Ghanshala
Graphic Era University
Dehradun, India
kamalghanshala@gmail.com

R.C Joshi
Graphic Era University
Dehradun, India
rcjoshi.geu@gmail.com

***Abstract*- Deep Learning algorithms are designed in such a way that they mimic the function of the human cerebral cortex. These algorithms are representations of deep neural networks i.e. neural networks with many hidden layers. Convolutional neural networks are deep learning algorithms that can train large datasets with millions of parameters, in form of 2D images as input and convolve it with filters to produce the desired outputs. In this article, CNN models are built to evaluate its performance on image recognition and detection datasets. The algorithm is implemented on MNIST and CIFAR-10 dataset and its performance are evaluated. The accuracy of models on MNIST is 99.6 %, CIFAR-10 is using real-time data augmentation and dropout on CPU unit.**

Keywords- Deep Learning, Handwritten digit Recognition, Object Detection, Convolutional neural networks, MNIST, CIFAR-10, Dropout, Overfitting, Data Augmentation, Relu

I INTRODUCTION

Image Recognition and detection is a classic machine learning problem. It is a very challenging task to detect an object or to recognize an image from a digital image or a video. Image Recognition has application in the various field of computer vision, some of which include facial recognition, biometric systems, self-driving cars, emotion detection, image restoration, robotics and many more[1]. Deep Learning algorithms have achieved great progress in the field of computer vision. Deep Learning is an implementation of the artificial neural networks with multiple hidden layers to mimic the functions of the human cerebral cortex. The layers of deep neural network extract multiple features and hence provide multiple levels of abstraction. As compared to shallow networks, this cannot extract or work on multiple features. Convolutional neural networks is a powerful deep learning algorithm capable of dealing with millions of parameters and saving the computational cost by inputting a 2D image and convolving it with filters/kernel and producing output volumes.

The MNIST dataset is a dataset containing handwritten digits and tests the performance of a classification algorithm. Handwritten digit recognition has many applications such as OCR (optical character recognition), signature verification, interpretation and manipulation of texts and many more[2,3]. Handwritten digit recognition is an image classification and recognition problem and there have been recent advancements in this field [4]. Another dataset is CIFAR-10 which is an object detection datasets that classifies the objects into 10 classes and detects the objects in the test sets. It contains natural images and helps implement the image detection algorithms.

In this paper, Convolutional neural networks models are implemented for image recognition on MNIST dataset and object detection on the CIFAR-10 dataset. The implementation of models is discussed and the performance is evaluated in terms of accuracy. The model is trained on an only CPU unit and real-time data augmentation is used on the CIFAR-10 dataset. Along with that, Dropout is used to reduce Overfitting on the datasets.

The remaining sections of the paper are described as follows: Section 2 describes a brief literature survey; Section 3 describes the classifier models with details of the techniques implemented. Section 4 evaluates the performance of the model and describes the results. Section 5 summaries the work with future works.

II. LITERATURE SURVEY

In recent years there have been great strides in building classifiers for image detection and recognition on various datasets using various machine learning algorithms. Deep learning, in particular, has shown improvement in accuracy on various datasets. Some of the works have been described below:

Norhidayu binti Abdul Hamid et al. [3] evaluated the performance on MNIST datasets using 3 different classifiers: SVM (support vector machines), KNN (K-nearest Neighbor) and CNN (convolutional neural networks). The Multilayer perceptron didn't perform well on that platform as it didn't reach the global minimum rather remained stuck in the local optimal and couldn't recognize digit 9 and 6 accurately. Other classifiers, performed correctly and it was concluded that performance on CNN can be improved by implementing the model on Keras platform. Mahmoud M. Abu Gosh et al. [5] implement DNN (Deep neural networks), DBF (Deep Belief networks) and CNN (convolutional neural networks) on MNIST dataset and perform a comparative study. According to the work, DNN performed the best with an accuracy of 98.08% and other had some error rates as well as the difference in their execution time.

Youssouf Chherawala et al. [6] built a vote weighted RNN (Recurrent Neural networks) model to determine the significance of feature sets. The significance is determined by weighted votes and their combination and the model is an application of RNN. It extracts features from the Alex word images and then uses it to recognize handwriting. Alex krizhevsky [7] uses a 2-layer

Convolutional Deep belief network on the CIFAR-10 dataset. The model built classified the CIFAR-10 dataset with an accuracy of 78.90% on a GPU unit. Elaborative differences in filters and their performance is described in the paper which differs with every model.

Yehya Abouelnaga et al. [8] built an ensemble of classifiers on KNN. They used KNN in combination with CNN and reduce the Overfitting by PCA (Principal Component Analysis). The combination of these two classifiers improved the accuracy to about 0.7%.

Yann le Cunn et al. [1] give a detailed introduction to deep learning and its algorithms. The algorithms like Backpropagation with multilayer perceptron, Convolutional neural networks, and Recurrent neural networks are discussed in detail with examples. They have also mentioned the scope of unsupervised learning in future in Artificial intelligence.

Li Deng [10] details a survey on deep learning, its applications, architectures, and algorithms. The generative, discriminative and hybrid architectures are discussed in detail along with the algorithms that fall under the respective categories. CNN, RNN, Autoencodes, DBN's, RBM's (Restricted Boltzmann machines) are discussed with their various applications.

III. CLASSIFIER MODELS

A. Datasets

MNIST is the dataset used for image recognition i.e. for recognition of handwritten digits [11]. The dataset has 70,000 images to train and test the model. The training and test set distribution is 60,000 train images and 10,000 test images. The size of each image is 28x28 pixels (784 pixels) which are given as input to the system and has 10 output class labels from (0-9). Fig.1 shows a sample picture from MNIST dataset [13].

CIFAR-10 is the dataset used for object detection which is labeled a subset of 80 million tiny images [12]. The dataset has 60,000 32x32 pixel color images with 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). Each class has 6000 images. The train batch has 50,000 images and test batch has 10,000 images. The test batch for each class has 1000 images which are randomly selected. Fig.2 shows sample pictures from CIFAR-10 dataset [14].

B. CNN Models

Convolutional neural networks are deep learning algorithms that take input images and convolves it with filters or kernels to extract features. A NxN image is convolved with a fXf filter and this convolution operation learns the same feature on the entire image[18]. The window slides after each operation and the features are learnt by the feature maps. The feature maps capture the local receptive field of th eimage and work with shared

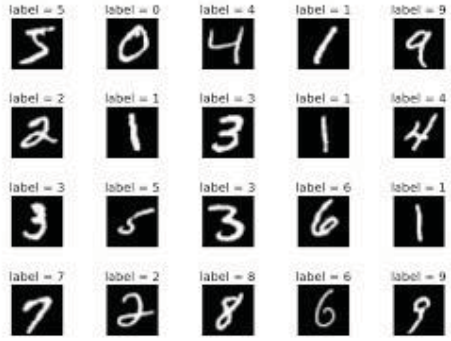

Fig.1 A sample MNIST Dataset

Fig.2 A CIFAR-10 dataset images

weights and biases [15,21]. Equation (1) shows the size of the output matrix with no padding and Equation (2) shows the convolution operation. In order to preserve the size of input image padding is used . In a 'SAME' padding the output image size is same as input image size and a "VALID" padding is no padding. The size of th eoutput matrix with padding is depicted in equation (3).

$$NXN * fXf = N - F + 1 \quad (1)$$

$$O = \sigma(b + \sum_{i=0}^{2} \sum_{j=0}^{2} w_{i,j}\, h_{a+i,b+j}) \quad (2)$$

$$NXN * f * f = (N + 2P - f)/(s + 1) \quad (3)$$

Here, O is the output , P is the padding, s i sthe stride, b is the bias, σ is the sigmoidal activation function, w is a 3x3 weight matrix of shared weights and $h_{x,y}$ is the input activation at position x, y. CNN has application in fields of large scale image recognition [17], Emotion detection through speech [9] [19] facial expression recognition [20], biometric systems, genomics and many others.

C. CNN model for MNIST dataset

The CNN model for MNIST dataset is shown in figure (c). The input image is a vector with 784 pixel values. It is input into the convolutional model where the convolution layers along with filters generate th efeature maps using the local receptive field. The pooling and fully connected layers follow the convolution layers. Dropout is

introduced after each convolutional layer. The pooling layers simplify the output after convolution. There are two types of pooling: Max pooling and L2 pooling. In max pooling the maximum activation output is pooled into a 2 x 2 input region and L2 pooling takes the square root of the sum of squares of the activation 2x2 region. Finally, the fully connected layers connect each layer of max pooling layer to the output neurons. The architecture of the developed model is as follows:

- Batch Size (training): 128, Batch size (test): 256, Number of epochs:10
- Dropout: Yes
- Optimizer: RMS prop, Learning rate=0.001, the parameter β : 0.9
- Keep_prob:0.8

The architecture of the model is :

Convolution_layer 1 → Relu → Max_pool → dropout → Convolution_layer 2 → Relu → Max_pool → dropout → Convolution_layer 3 → Relu → Max_pool → fully_connected → dropout → output_layer → Result

The input is a 28X28 image which passed to filters to generate fetaure maps. . The first filter is of size 5x5x1x32 (32 features to learn in the first hidden layer), 3x3x32x64 for the second convolution layer (64 features to learn from second hidden layer), 3x3x64x128 for the third layer, (128*4*4,625) for the fourth layer and (625,10) for the last layer. The stride is 1 for convolution layer and 2 for max-pooling layers. Stride defines the number of block to move forward after one calculation. Generally, the value of stride for convolution layer is 1 and for pooling layer is 2. The accuracy of this model is 99.6%.

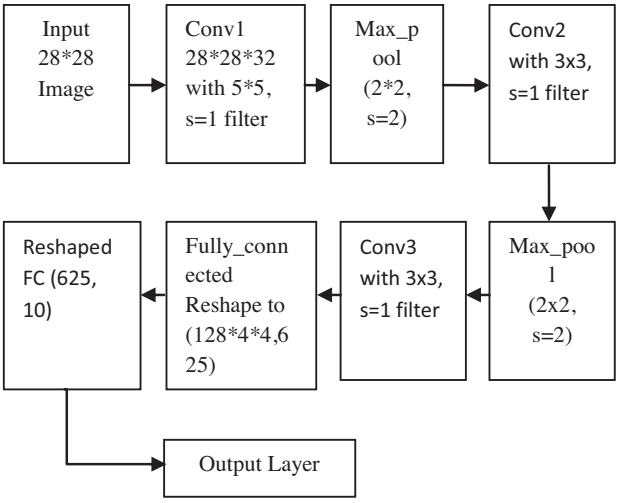

Fig. 3. CNN model for MNIST dataset

D. *CNN model for CIFAR-10 dataset*

The CNN model for CIFAR-10 dataset is as follows:

- Batch size: 32, Number of epochs:50
- Optimizer: RMS Prop, Decay rate= 1e-6
- Data augmentation: True, Rotation ; in range of 0-180, horizontal flip: TRUE, Vertical flip: FALSE
- Dropout : True

The architecture is as follows:

Conv1→Relu→Conv2→Relu→Max_pooling→Dropout →Conv3→Relu→Conv4→Relu→Max_pooling→Dropout→Conv5→Relu→Max_pooling→Dropout→Flatten→ Dense→Relu→Dropout→Dense→Softmax→Result

Fig.4 shows the architecture of the model. The 32X32 input image is given to the model where the first convolutional layer learns 32 features through a 5x5 filter and 'same' padding. After the activation, another convolutional layer is stacked up that learns 64 features through a 5x5 filter. Then Relu activation is added and forwarded to max pooling layer. After the max pooling layer a dropout is implemented with the dropout of 0.25. This entire layer is repeated again with 3x3 filters and the conv 3 layer this time learns 64 features similar to conv 4 features. All the remaining parameters are same. The conv 5 layer learns 64 features with a 3x3 filter followed by Relu, max-pooling and dropout. Then the output is flattened and we have a fully connected layer with 512 outputs. The dropout is again applied as 0.5 and denser to number of output classes i.e. 10 and passed to the softmax layer for the final output. The accuracy of the model is 80.01 % on a CPU unit on test dataset.

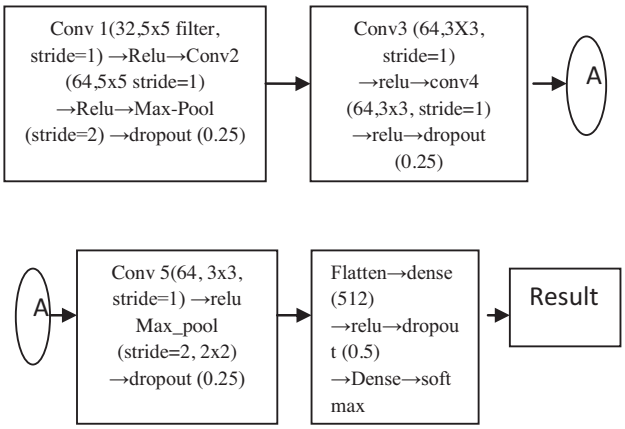

Fig.4 CNN Model for CIFAR-10

E. Relu non-linearity activation function

There is a wide range of activation function available when training neural network models. The mainly used activations are sigmoid, tanh, relu and leaky relu. The relu non linearity is a popular activation function used in deep learning algorithm sand has replaced the use of sigmoidal activation function which is generally used for binary classification techniques. Relu non linearity $f(x) = \max(0, x)$ works several times faster than the tanh non – linearity $(e^z - e^{-z})/(e^z + e^{-z})$, and results the output of the activation as either 0 or a positive number. With relu it is easier to train larger neural networks. Fig.5 shows the graph of relu non-linearity [16].

F. Overfitting and under fitting

Another aspect of training a deep neural network is the issue of high bias (resulting in underfitting) or high variance (resulting in overfitiing). When the data is

underfitting is not generalizing or fitting the data points well. In case of high bias, which is on training batch the network needs to be trained longer and have much bigger network of hidden layers. In case of overfitting, the data has high variance i.e. it is generalizing too well on the test sets. To reduce high variance, regularization techniques and data augmentation techniques can be implemented.

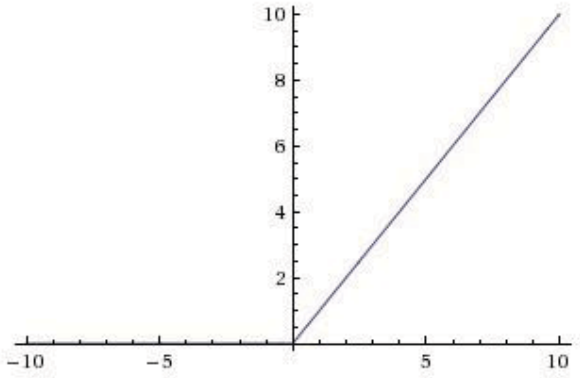

Fig.5. Relu non-linearity

G. Dropout to reduce overfitting

Dropout is a regularization technique which is used to reduce overfitting [7]. In dropout, the network deads some of its nodes randomly based on a parameter. The probability parameter determines whether the node should remain in the network or not. Keep_prob is the probability parameter to keep the hidden nodes in the network. The activation is unaffected during this process as it only determines whether to keep the node in the network or not.

H. Data Augmentation

Another technique to reduce overfitiing is to train the data on large datasets. If dataset is limited the dataset can be artificially created by data augmentation techniques [7]. The data augmentation techniques include distortion and altering of data images for processing to get more data. Some of the techniques are:

- Mirroring – The images are flipped and laterally inverted.
- Random cropping- Cropping some parts of the image and creating subsets from the main image.
- Rotation- This includes rotating the images in any direction at various angles and generating new images.
- Color shifting- Shifting the RGB pixel values of the image to get a new coloured image.

I. RMS prop optimizer and learning rate

RMS prop or root mean square prop is an optimizer which works on the root mean square value of the change in gradients. The change in weights and bias determine the gradient parameters with help of rms value. The learning rate determines the steps the algorithm will take to converge to the global minimum. If the learning rate is too high the algorithm faces exploding gradient problem i.e. it takes larger steps and fails to converge at the local minimum. If the learning rate is too small it faces a vanishing gradient problem i.e. the gradient takes smaller steps and ultimately becomes so small that the changes in weights are insignificant. Thus the learning rate is a hyper parameter that needs to be finely tuned and can be done with the help of learning rate decay. In learning rate decay, the learning rate decays exponentially after every epoch.

IV. RESULT ANALYSIS

The results of the experiments are as shown below:

- CNN Model for MNIST dataset: Accuracy 99.6% shown in figure 6

```
7 0.98828125
8 0.9921875
9 0.99609375

Process finished with exit code 0
```

Fig.6 Accuracy of CNN model on MNIST in 10 epochs

The result shows that as the number of epochs get increased and the best accuracy achieve in recognizing the digits on MNIST data set is 99.6 % with 10 epochs.

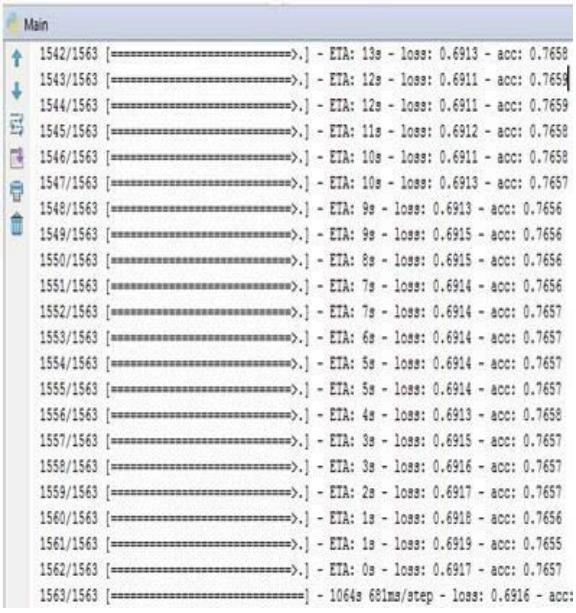

Fig. 7 Accuracy of the CNN model in 50 epochs

CNN model for CIFAR-10 dataset: Accuracy of 80.17% on test set as shown in figure 7.

V. CONCLUSION

The article discusses various aspects of deep learning, CNN in particular and performs image recognition and detection on MNIST and CIFAR -10 datasets using CPU unit only. The accuracy of MNIST is good but the accuracy of CIFAR-10 can be improved by training with larger epochs and on a GPU unit. The calculated accuracy on MNIST is 99.6% and on CIFAR-10 is 80.17%. The

training accuracy on CIFAR-10 is 76.57 percent after 50 epochs. The accuracy on training set may also be improved further by adding more hidden layers. And this system can be implemented as a assistance system for machine vision for detecting nature language sysmbols.

REFERENCES

[1] Yann LeCun, Yoshua Bengio, Geoffery Hinton, "Deep Learning", Nature, Volume 521, pp. 436-444, Macmillan Publishers, May 2015.

[2] Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif, "Handwritten Recognition Using SVM, KNN and Neural Network", www.arxiv.org/ftp/arxiv/papers/1702/1702.00723

[3] Cheng-Lin Liu∗, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques", ELSEVIER, Pattern Recognition 36 (2003) 2271 – 2285).

[4] Ping kuang, Wei-na cao and Qiao wu, "Preview on Structures and Algorithms of Deep Learning", 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP), IEEE, 2014.

[5] Mahmoud M. Abu Ghosh ; Ashraf Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks", IEEE, 2017.

[6] Youssouf Chherawala, Partha Pratim Roy and Mohamed Cheriet, "Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network," IEEE Transactions on Cybernetics, VOL. 46, NO. 12, DECEMBER 2016.

[7] Alex Krizhevsky, "Convolutional Deep belief Networks on CIFAR-10". Available: https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf.

[8] Yehya Abouelnaga , Ola S. Ali , Hager Rady , Mohamed Moustafa, " CIFAR-10: KNN-based ensemble of classifiers", IEEE, March 2017.

[9] Caifeng Shan, Shaogang Gong, Peter W. McOwan, "Facial expression recogniton based on Local binary patterns: A comprehensive study", ELSEVIER, Image and Vision Computing 27, pp. 803-816, 2009.

[10] Li Deng, "A tutorial survey of architectures, algorithms, and applications of Deep Learning", APSIPA Transactions on Signal and Information Processing (SIP), Volume 3, 2014.

[11] Yann LeCun, Corinna Cortes and Christopher J.C. Burges, "The MNIST Database of handwritten digits". Available: http://yann.lecun.com/exdb/mnist/ - MNIST database

[12] The CIFAR-10 dataset. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[13] MNIST dataset introduction, 2017. Available: http://corochann.com/mnist-dataset-introduction-1138.html

[14] Robust Vision Benchmark. Available: https://robust.vision/benchmark/about/

[15] Neural Network and Deep Learning. Available: http://neuralnetworksanddeeplearning.com/chap6.html

[16] Convolutional Neural Networks for Visual Recoginition. Available: http://cs231n.github.io/neural-networks-1/

[17] Krizhevsky, Sutskever and Hinton, "ImageNet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 1106–1114, 2012.

[18] Zeiler, M. D. and Fergus, "Visualizing and understanding convolutional networks". European Conference on Computer Vision, vol 8689. Springer, Cham, pp. 818-833, 2014.

[19] Zhengwei Huang, Min Dong, qirong Mao and Yongzhao Zhan, "Speech Recognition using CNN", IEEE/ACM Transactions on Audio, Speech and Language Processing, pp. 1533-1545, Volume 22, Issue 10, 2014, http://dx.doi.org/10.1145/2647868.2654984.

[20] Shima Alizadeh and Azarr Fazel, "Convolutional Neural networks for Facial Expression recognition", Computer Vision and Pattern Recognition, Cornell University Library, ArXiv:1704.06756v1, 22 April, 2017, arXiv.org.1704.06756.pdf.

[21] Christian szegedy, Wei liu, Yangqing Jia et al., "Going Deeper with Convolutions",Conference on Computer Vision and Pattern Recognition (CPVR) , IEEE explorer, Boston, MA, USA, 2015.