National Conference on Recent Advances in Computer Science and IT (NCRACIT)
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 1 | ISSN : 2456-3307

Recognizing Digits from Natural Images and handwritten Digits using Deep Convolutional Neural Networks

Rohit Thapa[1], Dhrub Kumar[2]

[12]Department of Information Technology, Model Institute of Engineering & Technology, Kot Bhalwal, Jammu, India
rohit.40it14@mietjammu.in[1], dhrub.it@mietjammu.in[2]

ABSTRACT

Recognizing digits from Natural Images and Handwritten digits are one of the famous problems in Computer Vision Applications.Many Machine Learning Techniques have been employed to solve the problem of recognizing Digits from Natural Images and Handwritten Digits.The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). This Paper focuses on Convolutional Neural Networks (CNN), also known as Deep Convolutional Neural Network (DCNN) that operates directly on image pixels, also the three NN approaches are compared and evaluated in terms of many factors such as accuracy and performance. Recognition accuracy rate and performance, however, is not the only criterion in the evaluation process, but there are interesting criteria such as execution time. In addition, DistBelief [15] implementation of deep neural network is employed to train large & distributed neural networks on high quality images. At the end, we find that the performance of this approach increases with the depth of the convolutional network. With the best performance occurring in the deepest architecture with eleven hidden layers. We evaluate this approach on the publically available SVHN dataset and achieve over 96% accuracy in recognizing complete street numbers. This paper also presents blocky artifact as an augmentation technique to increase the accuracy of DCNN for handwritten digit recognition i.e 0-9 and conducts experiments on MNIST dataset only. DCNNs with the proposed augmentation technique give better results than those without such augmentation.

Keywords: DistBelief, SVHN, MNIST, Blocky Artifact, Augmentation

I. INTRODUCTION

The main focus of this paper is to increase the accuracy rate of a deep convolutional neural network (DCNN) for recognizing handwritten digits and recognizing digits from natural images. In computer vision, handwritten digit recognition is a benchmark task. Also, recognizing numbers in photographs is a problem of interest to the optical character recognition community. While OCR on constrained domains like document processing is well studied, arbitrary multi-character text recognition in photographs is still highly challenging.Optical character recognition has many significant usages in our regular life e.g., bank cheque processing, processing numeric fields of forms written by hand, identifying postal codes etc. These are sensitive tasks; tiny mistake can cause severe problem. For this reason, it is important to achieve a good accuracy in handwritten digit recognition.In this paper, for recognizing of digits from natural images we propose a unified approach that integrates three steps via the use of a deep convolutional neural network that operates directly on the image pixels.

The model is configured with multiple hidden layers or you can use deeper architecture to obtain better accuracy, all with feedforward connections. We employ DistBelief to implement these large-scale deep neural networks. The key contributions of this paper are: (a) a unified model to localize, segment, and recognize multidigit numbers from street level photographs (b) a new kind of output layer, providing a conditional probabilistic model of sequences (c) empirical results that show this model performing best with a deep architecture (d) reaching human level performance at specific operating thresholds.By following this above approach on the publicly available Street View House Numbers (SVHN) dataset over 96% accuracy is achieved in recognizing street numbers. Also on a perdigit recognition task, 97.84% accuracy can be achieved. We also evaluated this approach on an even more challenging dataset generated from Street View imagery containing several tens of millions of street number annotations and achieve over 90% accuracy. Our evaluations further indicate that at specific operating thresholds, the performance of the proposed system is comparable to that of human operators. To date, our system has helped us extract close to 100 million street numbers from Street View imagery

worldwide. Also in case of recognizing handwritten digits, different persons write differently, the shapes and sizes of their digits vary. This is why handwritten digit recognition is a difficult task. MNIST is the largest dataset of handwritten English digits[1], which contains 60,000 images.

DCNNs are very popular and efficient for classifying handwritten digits[2]. Accuracy of a DCNN model depends on the size of the training set - better accuracy can be achieved with more training images. As a result, data augmentation, i.e., modification or transformation of the features of the existing training images[3], is one of the best practices to increase the size of the training dataset.Image augmentation can be done in many ways. A common process is to deform original data based on intra-class variation or prior knowledge. Color processing and geometrical variation or transformation such as rotation, resizing, shifting etc. are also used as augmentation techniques in visual recognition problems. This paper focuses on a new augmentation method to improve the accuracy rate of DCNN. It proposes to create a blocky effect on the training

Images; such type of augmentation is unprecedented for handwritten digit recognition, and has not been found in other image classification problems as well. After augmentation of the training set by the blocky artifact, this work applies unsupervised pre-training using autoencoder and DCNN [4].

II. METHODS

A. Method for Transcription of Street Numbers

Our basic approach is to train a probabilistic model of sequences given images. Let S represent the output sequence and X represent the input image. Our goal is then to learn a model of P(S | X) by maximizing log P(S | X) on the training set.

To model S, we define S as a collection of N random variables S1, . . . , SN representing the elements of the sequence and an additional random variable L representing the length of the sequence. We assume that the identities of the separate digits are independent from each other, so that the probability of a specific sequence s = s1, . . . , sn is given by

$$P(S = s|X) = P(L = n \mid X)\Pi n\ i{=}1 P(Si = si \mid X)$$

A simple CNN model can be seen in Fig. 1. The first layer is the input layer; the size of the input image is 28 × 28. The second layer is the convolution layer C2, it can obtain four different feature maps by convolution with the input image. The third layer is the pooling layer P3. It computes the local average or maximum of the input feature maps.

This model can be extended to detect when our assumption that the sequence has length at most N is violated. To allow for detecting this case, we simply add an additional value of L that represents this outcome.

Each of the variables above is discrete, and when applied to the street number transcription problem, each has a small number of possible values: L has only 7 values (0, . . . , 5, and "more than 5"), and each of the digit variables has 10 possible values. This means it is feasible to represent each of them with a softmax classifier that receives as input features extracted from X by a convolutional neural network. We can represent these features as a random variable H whose value is deterministic given X. In this model, P(S | X) = P(S | H).

To train the model, one can maximize log P(S | X) on the training set using a generic method like stochastic gradient descent. Each of the softmax models (the model for L and each Si) can use exactly the same backprop learning rule as when training an isolated softmax layer, except that a digit classifier softmax model backprops nothing on examples for which that digit is not present.

At test time, we predict
s = (l, s1, . . . , sl) = argmaxL,S1,...,SL log P(S | X).

This argmax can be computed in linear time. The argmax for each character can be computed independently. We then incrementally add up the log probabilities for each character. For each length l, the complete log probability is given by this

running sum of character log probabilities, plus log P(l | x).
The total runtime is thus O(N).

We preprocess by subtracting the mean of each image. We do not use any whitening [17], local contrast normalization etc.

B. Augmentation for Handwritten Digit Recognization

Augmentation technique is used to escalate the accuracy. Blocky artifact is applied on the images. Firstly the dimension of the image are reduced to a certain value and then again increase the dimension to it original size. Python's sci-kit-image [5] module is used for resizing the image. It performs interpolation for up-size or downsize images. It down-samples a N-dimensional image by applying the arithmetic mean or sum. It uses linear interpolation for re-sizing the image. Linear interpolation produces blurred but jagged edges. It is mostly used for approximating the value of a function F by using two known values of that function F at other points.

This approximation's error is defined as:

RT = F(x) -G(x)
Here G is the linear interpolation polynomial, which is defined below-

Deep Convolutional Neural Network Model:- The model we will be using contains deep convolutional neural networks(DCNN) and artificial neural network(ANN). DCNN is such an ANN that contains more then one hidden layer [12]. DCNN was used to implement auto encoder. Auto encoder is used for unsupervised pre-training. Auto encoder first encodes the image and then decodes the image. Their aim is to reproduce the input image.
This model first augments the images using rotation range of Encoder of the autoencoder contains three convolutional ges layers. Each convolutional layers has 32 filters and 3x3 kernel size. Each CNN is followed by a 2x2 max pooling layers. Decoder also contains 3 CNN layers followed by 2x2 upsampling layers. Each CNN has 5 layers and 3x3 sized kernel. The encoder was later augmented again by rotation range of 10 and width range shift of 0.1%. In our model decoder has no significant usage. Our target was to use the encoder. When autoencoder encodes the images, it extracts all the important features of that image. So after encoding the encoder consists of all the important features of the image. So we took the encoder and sent it to an fully connected layer which contains 128 layers. Next part contains the output layer with 10 layers, as we have 10 classes. The output layer classifies the image. Figure 1 shows the full architecture of the model while figure 2 shows the Orginal Image, Augumented Image and their Differences.

III. DATASETS

A. MNIST for recognization of handwritten digits
MNIST is database of English ⇥ handwritten numerals. The dimension of the images are 28 28 pixels. This dataset consists total of 60,000 images. Figure 3 shows two image of MNIST dataset. This dataset is divided into two parts. 50,000 images were used for training and other 10,000 images were used for testing.

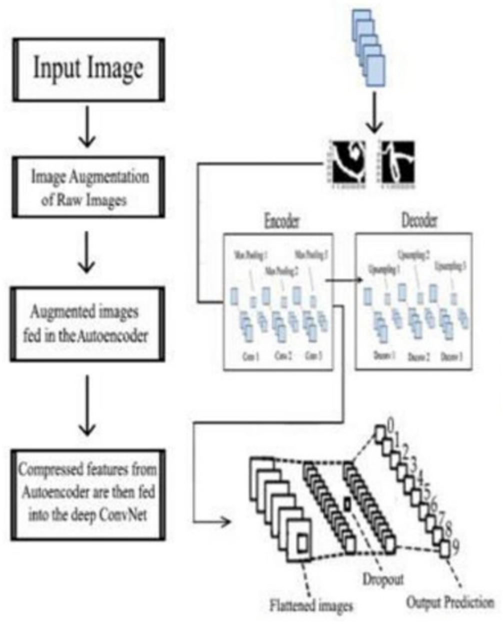
Input Image
Image Augmentation of Raw Images
Augmented images fed in the Autoencoder
Compressed features from Autoencoder are then fed into the deep ConvNet
Encoder
Decoder
Flattened images
Dropout
Output Prediction
0
1
2
3
4
5
6
7
8
9

Figure 1. Model for Augmented Technique for Recognizing Handwritten Digits

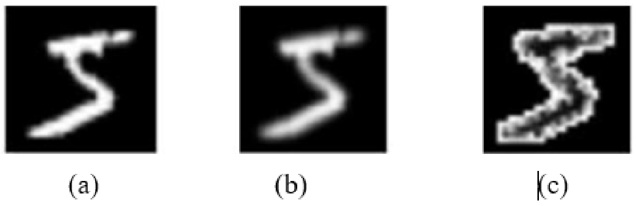

Figure 2. (a) Original Image, (b) Augumented Image and (c) Difference between the Images

Figure 3. MNIST Images

B. Public Street View House Numbers Dataset

The Street View House Numbers (SVHN) dataset [18] is a dataset of about 200k street numbers, along with bounding boxes for individual digits, giving about 600k digits total. The Figure 4 shows internal street number datasets.

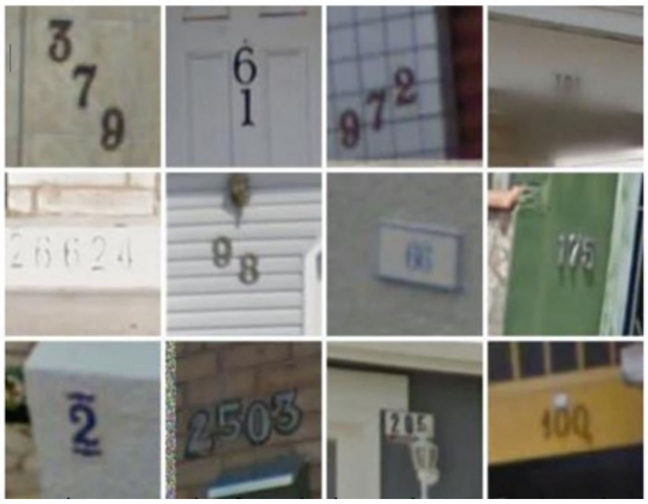

Figure 4. Examples from the Internal Street Numbers Dataset

IV. EXPERIMENTS

A. Recognization of Digits from Natural Images

The dataset is preprocessed in the following way – first we find the small rectangular bounding box that will contain individual character bounding boxes. We then expand this bounding box by 30% in both the x and the y direction, crop the image to that bounding box and resize the crop to 64 × 64 pixels. We then crop a 54 × 54 pixel image from a random location within the 64 × 64 pixel image. This means we generated several randomly shifted versions of each training example, in order to increase the size of the dataset. Without this data augmentation, we lose about half a percentage point of accuracy. Because of the differing number of characters in the image, this introduces considerable scale variability – for a single digit street number, the digit fills the whole box, meanwhile a 5 digit street number will have to be shrunk considerably in order to fit.

Our best model obtained a sequence transcription accuracy of 96.03%. This is not accurate enough to use for adding street numbers to geographic location databases for placement on maps. However, using confidence thresholding we obtain 95.64% coverage at 98% accuracy. Since 98% accuracy is the performance of human operators, these transcriptions are acceptable to include in a map. We encourage researchers who work on this dataset in the future to publish coverage at 98% accuracy as well as the standard accuracy measure. Our system achieves a character-level accuracy of 97.84%. This is slightly better than the previous state of the art for a single network on the individual character task of 97.53% [14].

The best architecture consists of eight convolutional hidden layers, one locally connected hidden layer, and two densely connected hidden layers. All connections are feedforward and go from one layer to the next. The first hidden layer contains maxout units while the others contain rectifier units. The number of units at each spatial location in each layer is [48, 64, 128, and 160] for the first four layers and 192 for all other locally connected layers. The fully connected layers contain 3,072 units each. Each convolutional layer includes max pooling and subtractive normalization. The max pooling window size is 2 × 2. The stride alternates between 2 and 1 at each layer, so that half of the layers don't reduce the spatial size of the representation. All convolutions use zero padding on the input to preserve representation size. The subtractive normalization operates on 3x3 windows and preserves representation size. All convolution kernels were of size 5 × 5. We trained with dropout applied to all hidden layers but not the input.

B. Recognizing Handwritten Digits

A number of experiments were performed with this augmentation method. There are three datasets available, which were augmented in different dimensions. However, we are only mentioning the MNIST dataset. In all the experiments, the training set size was twice of its original size. We added the blocky artifact set with the original data set. ⇥

MNIST: Original size⇥of the MNIST images⇥ are 28 28 pixels. Their dimension was first reduced to 25 25 and then again it was increased to 28 28.

There was two training phase in our used model. The first one was training the auto encoder. We trained auto encoder for 60 epochs with binary cross entropy as its cost function and adedelta as its optimizer. The convolutional layers of auto encoder has rectified linear unit (RELU) as their activation function. Next training phase was for fully connected layer. It was trained for 400 epochs with categorical cross entropy as its cost function and RMS prop as its optimizer. The first fully connected layer has RELU activation function. The last fully connected layer, which is the output layer, has softmax as its activation function.

V. RESULT AND ANALYSIS

A. For Public Street View House Numbers dataset

Now, we will explore the reasons for the unprecedented success of our neural network architecture for a complicated task involving localization and segmentation rather than just recognition. For such a complicated task, depth is crucial to achieve an efficient representation of the task. State of the art recognition networks for images of cropped and centered digits or objects may have between two to four convolutional layers followed by one or two densely connected hidden layers and the classification layers.

In this work, we used several layers that are more convolutional. We find that the depth was crucial to our success of our neural network. This is most likely because the earlier layers can solve the localization and segmentation tasks, and prepare a representation that has already been segmented so that later layers can focus on just recognition.

Moreover, we hypothesize that such deep networks have very high representational capacity, and thus need a large amount of data to train successfully. Prior to our successful demonstration of this system, it would have been reasonable to expect that factors other than just depth would be necessary to achieve good performance on these tasks. For example, it could have been possible that a sufficiently deep network would be too difficult to optimize. In Fig 5 we present the results of an experiment that confirms our hypothesis that depth is necessary for good performance on this task.

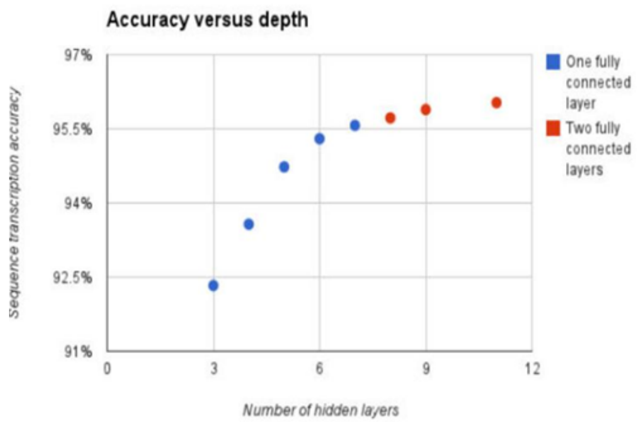

Fig.5 Performance analysis experiments on the public SVHN dataset show that fairly deep architectures are needed to obtain good performance on the sequence transcription task.

B. For Recognizing Handwritten digits from MNIST dataset

Table I show us the previously achieved results on MNIST dataset. The results in the tables are sorted according to the accuracy rate. In 2013, Wan et al. achieved best result so far on MNIST using neural network. The best achieved accuracy on ISI dataset is 98.98%.

Table II, III and IV discuss about the results we achieved using our proposed augmentation method respectively on MNIST, CMATERDB and ISI dataset. We ⇥achieved 99.56% on MNIST dataset. This accuracy was accomplished when we reduced the dimension of our⇥ image to 20 20 pixels. For CMATERDB we achieved 98.67% accuracy. We needed to reduce the image size to 28 28 pixels to achieve this result.

On ISI dataset when we reduced the image dimension to 29 ⇥ 29, we achieved 99.35%. Our results surpasses the state-of-the-art of CMATERDB and ISI dataset.

Table I past Results on MNIST

Work	Accuracy
Liang et al.[6]	99.69%
Lee et al.[7]	99.71%
Chang et al.[8]	99.76%
Sato et al.[9]	99.77%
Ciregan et al.[10]	99.77%
Wan et al.[11]	99.79%

Table II :- Results on MNIST for Different Reductions

Reduction Size	Accuracy
25 X 25	99.51%
24 X 24	99.53%
23 X 23	99.52%
22 X 22	99.51%
21 X 21	99.48%
20 X 20	99.56%

Table III: Results On CMATERDB for Different Reductions

Reduction Size	Accuracy
29 X 29	98.56%
28 X 28	98.67%
27 X 27	98.56%
26 X 26	98.61%
25 X 25	98.56%

Table IV :- Results on ISI for Different Reduction

Reduction Size	Accuracy
29 X 29	99.35%
28 X 28	99.05%
27 X 27	98.97%
26 X 26	99.10%
25 X 25	99.10%

VI. REFERENCES

[1] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[2] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," Neural computation, vol. 22, no. 12, pp. 3207–3220, 2010.

[3] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," arXiv preprint arXiv:1501.02876, vol. 7, no. 8, 2015.

[4] M. Shopon, N. Mohammed, and A. Abedin, "Bangla handwritten digit recognition using auto encoder and deep convolutional neural network," in International Workshop on Computational Intelligence, 2016, accepted and Presented.

[5] "Python scikit-image," https://goo.gl/gBXXDw, accessed: 2016-12-26.

[6] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.

[7] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in International Conference on Artificial Intelligence and Statistics, 2016.

[8] J.-R. Chang and Y.-S. Chen, "Batch-normalized maxout network in network," arXiv preprint arXiv:1511.02583, 2015.

[9] Sato, H. Nishimura, and K. Yokoi, "Apac: Augmented pattern classification with neural networks," arXiv preprint arXiv:1505.03229, 2015.

[10] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 3642– 3649.

[11] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1058–1066.

[12] Buntine, W. (1994). Operations for learning with graphical models. Journal of Artificial Intelligence Research, 2, 159–225.

[13] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36, 193–202.

[14] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In ICML'2013.

[15] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q., Mao, M.,

[16] Ranzato, M., Senior, A., Tucker, P., Yang, K., and Ng, A. Y. (2012). Large scale distributed deep networks. In NIPS'2012.

[17] Sermanet, P., Chintala, S., and LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. In International Conference on Pattern Recognition (ICPR 2012).

[18] Hyvarinen, A., Karhunen, J., and Oja, E. (2001). ¨ Independent Component Analysis. Wiley-Interscience.

[19] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. Deep Learning and Unsupervised Feature Learning Workshop, NIPS.