© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Offline Handwriting Recognition System Using Convolutional Network

Aathira Manoj, Priyanka Borate, Pankaj Jain, Vidya Sanas, Rupali Pashte

Computer Engineering, PVPPCOE, Mumbai, Maharashtra, India

ABSTRACT

In this paper, we have used Convolutional Neural Network (CNN) for offline handwriting recognition. Our Convolutional Network is based on the LeNet-5 network. We have modified it by changing the number of neurons in each layer. We have also added a dropout layer, which has resulted in slower, but accurate learning from the training set. We have used MNIST database for testing.

Keywords : Pre-processing, Segmentation, Feature extraction, CNN, LeNet-5

I. INTRODUCTION

Handwritten English Character Recognition has been a fairly challenging research topic in the field of Image Processing. Up to now, there have been lots of fruitful researches for Handwritten English Character Recognition [1][2][3][4]. However, most of them are carried out with online information, or with online and offline hybrid classifier; researches with pure offline information are rare [1][2][4][5][6]. As handwritten characters are unconstrained and topologically diverse, handwriting recognition with pure offline information has much difficulty. A handwriting recognition system aims at digitizing the handwriting by converting it into machine readable ASCII format with increased accuracy and less time. Several research works have been made to develop techniques which would reduce the processing time and increase the accuracy.

Convolutional Neural Network was first introduced in 1995 by Yann LeCun [7]. It received a lot of attention because of its ability to recognize handwritten digits with increased accuracy. In a CNN recognition system, 2-D image can be directly given as input. Hence, feature extraction is avoided. Many experiments with the CNN have seen moderately good performance. Most researches with CNNs are for handwritten digits [8][9][10][11], handwritten words [12] or printed character recognition [13].

The organization of this document is as follows. In Section 2 (Convolutional Neural Networks), we will give a brief overview of LeNet-5 network. In Section 3 (Proposed System), we give a brief overview of the system that we have implemented. In Section 4 (Result and Discussion), we discuss the results of our system and compare various results.

II. METHODS AND MATERIAL

1. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs. From Hubel and Wiesel's early work on the cat's visual cortex [16], we know the visual cortex contains a complex arrangement of cells. These cells are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. These cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in natural images.

Additionally, two basic cell types have been identified: Simple cells respond maximally to specific edge-like patterns within their receptive field. Complex cells have larger receptive fields and are locally invariant to the exact position of the pattern. The animal visual cortex being the most powerful visual processing system in existence, it seems natural to emulate its behaviour. Hence, many neurally-inspired models can be found in

the literature. To name a few: the NeoCognitron [17], HMAX [18] and LeNet-5 [19].

A. Architecture of LeNet-5

In a LeNet-5 network, the input plane receives the images of characters, which are approximately size normalized and centered. Each unit receives input from a set of units located in a small neighborhood in the previous layer. With locally receptive fields, neuron can extract elementary features such as edges, corners and end points. These features are then combined by the subsequent layers in order to detect higher order features. Units in a layer are organized in planes within which all the units share the same set of weights. The set of outputs of the units in such a plane is called a feature map. Units in a feature map are all constrained to perform the same operation on different parts of the image. A complete convolutional layer is composed of several feature maps with different weight vectors so that multiple features can be extracted at each location. A sequential implementation of a feature map would scan the input image with a single unit that has a local receptive field and store the states of this unit at corresponding locations in the feature map. This operation is equivalent to a convolution followed by an additive bias and squashing function, hence the name convolutional network. Once a feature has been detected, its exact location becomes less important. Only its approximate position relative to other features is relevant. Not only is the precise position of each of those features irrelevant for identifying the pattern, it is potentially harmful because the positions are likely to vary for different instances of the character. A simple way to reduce the precision with which the position of distinctive features are encoded in a feature map is to reduce the spatial resolution of the feature map. This can be achieved with subsampling layers which performs a local averaging and a subsampling, reducing the resolution of the feature map, and reducing the sensitivity of the output to shifts and distortions.

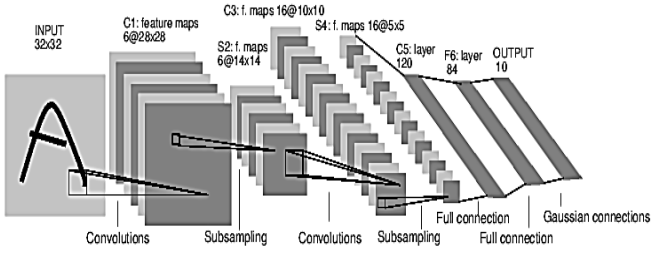

Figure 1. Architecture of LeNet-5

Layer S1 (input layer) is an image of size 32 × 32. Layer C2 is the first convolutional layer with 6 feature maps of size 28×28. Each unit of each feature map is connected to a 5×5 neighborhood of the input in layer S1. Layer S3 is the second subsampling layer with 6 feature maps of size 14×14. Each unit in each feature map is connected to a 2×2 neighborhood in the corresponding feature map in layer C2. Layer C4 is the second convolutional layer with 16 feature maps of size 10×10. The connection way between layer S3 and layer C4 takes much importance for the feature formation [19]. Layer S5 is the third subsampling layer with 16 feature maps of size 5×5. Layer C6 is the third convolutional layer with 120 feature maps of size 1 × 1. Each feature map is connected to all 16 feature maps of layer S5. Layer F7 contains 130 units and is fully connected to layer C6. Finally, layer F8 (the output layer) is composed of Euclidean RBF units and is fully connected to layer F7 [19].

2. Proposed System

Handwriting recognition is done in four phases namely preprocessing, segmentation, feature extraction and classification. Preprocessing deals with making the document ready for the later stages by removing the noise and cleaning the document. In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual characters. Feature extractions deals with extracting various parameters used for describing the character to the machine. Finally, the classifier is responsible for assigning a class to the character and generating the output.

A. Pre-processing

The image provided by the user is a color image. It is first converted into a grayscale image. The pre-processing step included scaling, binarization using adaptive thresholding, skew detection and correction and noise removal.

- **Binarization**
  Binarization is the process of converting grayscale images into binary images. Grayscale images have pixel intensities varying from 0 to 255. Binarization converts it into 0 or 1(bi-level) by using a threshold. All the pixels in the grayscale image having intensity below the threshold are assigned 0 while all the pixels having their intensities above the

threshold is assigned 1 where 0 stands for black and 1 stands for white. There are several methods used for deciding the threshold value. In our project, we have used local or adaptive thresholding because of intensity variation caused due to poor lighting when the image is taken.

In Local or adaptive thresholding, different thresholds are used for each pixel, based on the local area information.

- **Skew detection and correction**
  Skew is inevitably present in the images fed as input to the system. Skew detection and correction algorithms help us in finding the skew angle and removing it. We have used Entropy Based Skew Correction [14]. In this technique, the inherent randomness in the horizontal projection of the image, as the image is rotated, is used to detect the skew. The entropy is minimum when the skew angle is 0. Several techniques have been proposed for this.
- **Noise Reduction Techniques**
  The major objective of noise removal is to remove any unwanted bit-patterns, which do not have any significance in the output. We have used median filtering to remove salt and pepper noise. We have also used morphological operations to remove unwanted elements from the image.

B. Segmentation

Segmentation is the process of decomposing an image into multiple segments. For line segmentation and word segmentation, we have used the pixel counting technique [15]. In this technique, the binarized image is scanned from left to right and top to bottom. In a binarized image, 0 represents black and 1 represents white. Then, the horizontal and vertical projection profiles are created. The image can now be easily segmented into lines and words by cropping it at the minima of the horizontal and vertical profiles respectively. For the segmentation of connected components, we have used over segmentation. To remove incorrect segmentations, we removed those segmentation points which were apart by less than the average width of a character. We determined it to be 7 pixels.

C. Feature Extraction and Classification

It is the process of classifying the output by assigning a class to it. For feature extraction and classification, we have used convolutional neural network based on LeNet-5 [19]. After making some modifications to the LeNet-5, our network is as follows:

- Layer S1 is an image of size of size 28 × 28 pixels.
- Layer C2 is a Convolutional Layer. It contains 20 feature maps of size 24 × 24 where each unit is connected to a 5 × 5 neighborhood of S1.
- Layer S3 is a Sub-Sampling Layer. It contains 20 feature maps of size 12 × 12, where each unit is connected to a 2 × 2 neighborhood.
- Layer C4 is a Convolutional Layer. It contains 50 feature maps of size 8 × 8 where each unit is connected to a 5 × 5 neighborhood.
- Layer S5 is a Sub-Sampling Layer. It contains 50 feature maps of size 4 × 4, where each unit is connected to a 2 × 2 neighborhood.
- Layer C6 is a Convolutional Layer. It contains 500 feature maps of size 1 × 1 where each unit is connected to a 4 × 4 neighborhood.
- We have added a ReLu activation function. It is gives as f(x) = max (0, x). It adds non-linearity to the network.
- We have also added a Dropout Layer with 50% dropout rate. It prevents over-fitting by randomly ignoring half of the weights while predicting. It prevents inter-dependencies between the nodes.
- We have used a softmax layer for error calculation.

III. RESULTS AND DISCUSSION

To test the accuracy of our network, we used MNIST handwritten digit dataset. We used 60,000 images from the training set to train our network and we tested it on 10,000 images from the test set. We trained the network for 100 epochs. We were able to reduce the error to 0.8% as shown in Fig.2.

Overfitting happens when the training set error decreases at a faster rate than the validation set error. Hence, the network might do well on training set but not on the actual data. To solve this problem, we have added a dropout layer, which prevents inter-dependencies between the nodes. As you can see, because of adding the dropout, the training set and validation set error decrease at almost the same rate. Hence, the network does equally well on training and validation/testing set.

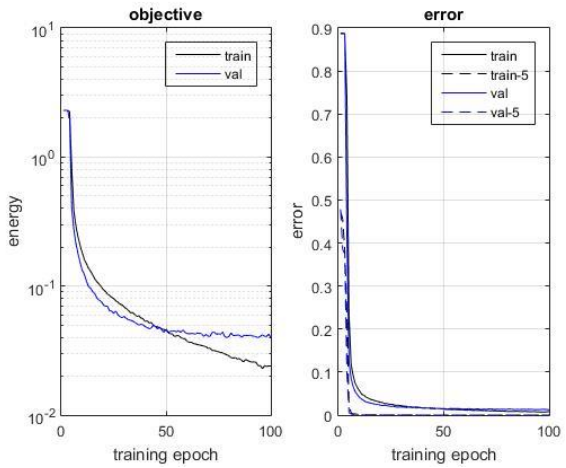

Figure 2. Error rates at training epochs

Table 1 compares different classification methods and their test error rates.

TABLE I. RECOGNITION ERROR RATES (%)

Classifier	Preprocessing	Test Error Rate (%)
K-nearest-neighbors, L3	Deskewing, noise removal, blurring	1.73
SVM, Gaussian Kernel	None	1.4
2-layer NN, 300 HU	Deskewing	1.6
Convolutional net LeNet-4	None	1.1
Convolutional net LeNet-5, [no distortions]	None	0.95
Convolutional net LeNet-5, [no distortions, ReLu, dropout]	None	0.8

IV. CONCLUSION

In this paper, we have used CNN for handwriting recognition. By comparing it with the other state-of-the-art methods, CNNs provide an encouraging solution for handwriting recognition. Nevertheless, further exploration of CNNs is needed. We continue to find more effective ways to solve the handwriting recognition problem using CNNs.

V. REFERENCES

[1]. J.F. H´ebert, M. Parizeau, and N. Ghazzali. Learning to segment cursive words using isolated characters. In Proc. of the Vision Interface Conference, pages 33–40, 1999.

[2]. M. Parizeau, A. Lemieux, and C. Gagn´e. Character recognition experiments using unipen data. In Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on, pages 481–485. IEEE, 2001.

[3]. E.H.Ratzlaff. Methods,reports and survey for the comparison of diverse isolated character recognition results on the unipen database. In Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, pages 623–628. IEEE, 2003.

[4]. L. Vuurpijl and L. Schomaker. Two-stage character classification: A combined approach of clustering and support vector classifiers. In Proc 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, Netherlands, pages 423–432, 2000.

[5].] R. Ghosh and M. Ghosh. An intelligent offline handwriting recognition system using evolutionary neural learning algorithm and rule based over segmented data points. Journal of Research and Practice in Information Technology, 37(1):73–88, 2005.

[6]. V. Nguyen and M. Blumenstein. Techniques for static handwriting trajectory recovery: a survey. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pages 463–470. ACM, 2010.

[7]. Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361, 1995.

[8]. D. Bouchain. Character recognition using convolutional neural networks. Institute for Neural Information Processing, 2007, 2006.

[9]. F. Lauer, C.Y. Suen, and G. Bloch. A trainable feature extractor for handwritten digit recognition. Pattern Recognition, 40(6):1816–1824, 2007.

[10]. Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pages 253–256. IEEE, 2010.

[11]. P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. Document Analysis and Recognition, 2:958, 2003.

[12]. Y. Bengio, Y. LeCun, C. Nohl, and C. Burges. Lerec: A nn/hmm hybrid for on-line handwriting recognition. Neural Computation, 7(6):1289–1303, 1995.

[13]. H. Deng, G. Stathopoulos, and C.Y. Suen. Error correcting output coding for the convolutional neural network for optical character recognition. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, pages 581–585. IEEE, 2009.

[14]. K.R. Arvind, Jayant Kumar, and A.G. Ramakrishnan. Entropy Based Skew Correction of Document Images

[15]. S. Marinai and P. Nesi, "Projection Based Segmentation of Musical Sheets", Document Analysis and Recognition, ICDAR, (1999), pp. 515-518.

[16]. Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. Journal of Physiology (London), 195, 215–243.

[17]. 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.

[18]. Serre, T., Wolf, L., Bileschi, S., and Riesenhuber, M. (2007). Robust object recog- nition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell., 29(3), 411–426. Member-Poggio, Tomaso.

[19]. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.