International Journal of Engineering & Technology, 7 (3) (2018) 1599-1604

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.14053
Research paper

Deformed character recognition using convolutional neural networks

N. Shobha Rani [1] *, N. Chandan [2], A. Sajan Jain [2], H. R. Kiran [2]

[1,2] Department of Computer Science
[1,2] Amrita School of Arts and Sciences, Mysuru,
Amrita Vishwa Vidyapeetham, India
** E-mail: n_shobharani@asas.mysore.amrita.edu*

Abstract

Realization of high accuracies towards south Indian character recognition is one the truly interesting research challenge. In this paper, our investigation is focused on recognition of one of the most widely used south Indian script called Kannada. In particular, the proposed exper-iment is subject towards the recognition of degraded character images which are extracted from the ancient Kannada poetry documents and also on the handwritten character images that are collected from various unconstrained environments. The character images in the degraded documents are slightly blurry as a result of which character image is imposed by a kind of broken and messy appearances, this particular aspect leads to various conflicting behaviors of the recognition algorithm which in turn reduces the accuracy of recognition. The training of degraded patterns of character image samples are carried out by using one of the deep convolution neural networks known as Alex net.The performance evaluation of this experimentation is subject towards the handwritten datasets gathered synthetically from users of age groups between 18-21, 22-25 and 26-30 and also printed datasets which are extracted from ancient document images of Kannada poetry/literature. The datasets are comprised of around 497 classes. 428 classes include consonants, vowels, simple compound characters and complex com-pound characters. Each base character combined with consonant/vowel modifiers in handwritten text with overlapping/touching diacritics are assumed as a separate class in Kannada script for our experimentation. However, for those compound characters that are non-overlapping/touching are still considered as individual classes for which the semantic analysis is carried out during the post processing stage of OCR. It is observed that the performance of the Alex net in classification of printed character samples is reported as 91.3% and with reference to handwritten text, and accuracy of 92% is recorded.

Keywords:*Ancient Documents; Deep Neural Networks; Degraded Character Recognition; Handwritten Text; Kannada Documents; Printed Text; South Indian Script.*

1. Introduction

Robust classification of Kannada characters from scanned images generated from various handwritten and printed samples is a hard classification problem. Integration of automatic transcription capability to machine to recognize Kannada handwritten/degraded printed text from hard copy document to system with a high degree of accuracy is the prime focus of this work. Adoption of pen-based interfaces for recognition of character images to establish human computer interaction through various smart devices is becoming increasingly popular these days [9].Pen based input from paper to a smart device is a unique perspective of offline handwriting recognition particularly emphasizing the south Indian script recognition like Kannada.

Transcription of handwritten Kannada text via optical processor has many advantages, especially these technologies eradicate the barriers of adept typo knowledge to digitize the south Indian scripts, using these devices are more user friendly than any other high end or specialized machines etc.

In the proposed work, an attempt is made to recognize Kannada text both in printed as well as handwritten forms which are present with characteristics like character deformations (handwriting) and back ground degradation (printed) [18]. There exist several contributions in the literature towards the recognition of handwritten and printed text recognition focusing more on the simple characters without any deformations ad noise [16] [17] [20]. Fig. 1 depicts few instances of the input data samples acquired for recognition task in the proposed work.

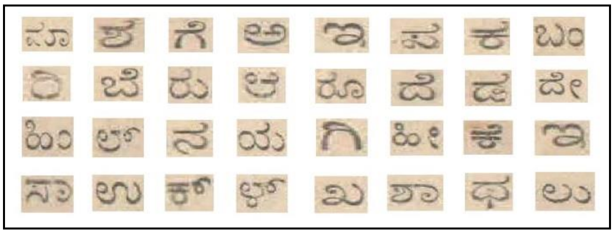


Fig. 1: A) Printed Data Samples.

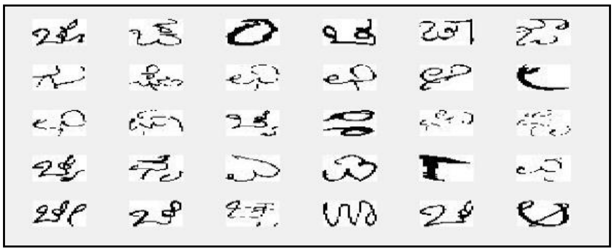


Fig. 1: B) Handwritten Data Samples.

2. Literature review

Some of the significant research contributions reported in this area are detailed subsequently. Arica et al. [4] presented an overview of character recognition for both online and offline handwritten documents. They surveyed various character recognition techniques and came up with the merits, demerits and also the future research scope. The areas identified that have potential research scope are multi-font historical documents which causes difficulties because of lack of noise model over all the stages and also Neural Networks are identified as solution for many pattern recognition problems. Pal et al. [14] presented a survey on Indian script character recognition where they discussed the different properties of Indian scripts and they reviewed the methodologies and the work done on OCR systems for 12 major Indian scripts. The studies found that there has not been intensive work done on the OCR for poor quality documents and also many OCR for different scripts followed the Tree classifiers and SVM classifiers for classification yielding accuracy between 95-97%. Droettboom[8] presented a new technique of correcting the broken characters for the historical printed documents.The proposed technique is based on the graph combinatory which was used to rejoin all the appropriate connected components and called as broken character connection algorithm. First, the algorithm built undirected graphs based on the bounding boxes or morphological dilation which created clusters of graphs where each graph represented a word in the document. Later, all the graphs were evaluated to get the word. Experimentations were carried out on census-like data that was printed in 1799 and it correctly found out 91% of the broken characters. Lavrenko et al. [10] presented holistic word recognition approach for single-author historical documents. The proposed methodology used scalar features like width, height and profile-based features to generate features and hidden markov model to classify them. The experimentations were administered on assortment of George Washington's manuscripts and used Word Error Rate as the performance measure which gave mean word error rate of 0.349 and corresponding accuracy of 65%. Ciresan et al. [6] presented Convolutional Neural Network committees for handwritten character classification. The proposed work suggests that convolutional neural networks are most suitable for the handwritten character recognition especially when the characters are deformed. The training data pre-processing reduces the errors and improve recognition rates. The experimentations were carried on MNIST database with a committee of seven deep CNNs and achieved only 0.27% error rate. Anil et al. [3] presented Convolutional neural networks for the recognition of Malayalam characters. Their proposed methodology used LeNet-5, a CNN trained with gradient based learning and backpropagation algorithm. Experimentations were carried out on Malayalam Character classes and results showed that CNN performance went down once the amount of categories exceeded range of 40. Their algorithm produced 75% accuracy once the characters were ungrouped and improved to 92% when they grouped misclassified characters. Yang et al. [19] presented improved Deep Convolution Neural Network for online handwritten Chinese character recognition using domain specific knowledge. Their planned methodology was twofold. Firstly, the domain specific data like deformations, non-linear normalization, imaginary strokes, path signature, and eight directional features were investigated and integrated with a DCNN to create a composite network to attain higher performance. Then, the ensuing DCNNs were merged using hybrid serial-parallel strategy. The experimentations showed 96.87% of accuracy. Dewan et al. [7] presented a framework for offline character identification by making use of Auto-encoder Networks. They proposed a strategy by creating DNNs by loading the Auto-encoders that are trained in a greedy layer manner in an unsupervised fashion. They likewise made a classifier to expand the rate of grouping. They used a three-hidden layer DNNs and it provided 95.4% accuracy. Loey et al.[11] presented deep learning autoencoder approach for handwritten Arabic Digits recognition. The proposed methodology digit-level stacked auto encoder for digit classification. Their algorithm could handle different types of variations in the human handwriting for Arabic digits. The experimentations were carried out on MADBase Database for Arabic Handwritten digits images and it gave an average accuracy of 98.5%. Pratama et al. [15] introduced recreating Japanese manually written images using auto encoder with residual block. The planned methodology expanded residual block between its input and output using parallel computing. The training was conducted utilizing the Gradient Descent with AdaDelta enhancer that created loss degraded as time. Experimentations were regulated on 5000 Japanese manually written images and the auto encoder with residual blocks produced higher outcomes than classical auto encoders. Ahmed et al [1] presented offline Urdu Nastaleeq Optical Character Recognition based on stacked denoising Autoencoder. The proposed technique utilizes stacked denoising autoencoder for programmed feature extraction straightforwardly from crude pixel estimations of ligature images. Diverse stacked denoising autoencoders were prepared on 178573 ligatures with 3732 classes from un-corrupted (clamor free) UPTI (Urdu Printed Text Image) data collection. The trained networks were approved and tried on debased versions of UPTI data set. The results showcased accuracies in range of 93% to 96% which were superior to the current Urdu OCR frameworks for such substantial data collection of ligatures. Pal et al. [13] introduced Recognition of online manually written Bangla characters utilizing Hierarchical framework with denoising autoencoders. The proposed method at first pre-trains the denoising autoencoder with MLP through backpropagation and afterward they are stacked to frame Deep Denoising Autoencoders. At last, characterization is made from the Deep Classifier. The experimentations were done on wide classes of Bangla like Vowels, Consonants, Special symbols and numerals and it gave 93.12% exactness. Bhowmik et al. [5] presented identification of Bangla manually written characters by making use of an MLP classifier based on stroke features. The proposed approach recognizes the strokes essentially and 10 features are extricated from every one of them. The stroke features are then attached in an fitting method to frame the feature vector of the character image on premise of which the MLP classifier is trained. The experimentations were done on 90 test images and testing was done on 350 images. It gave a precision between 80-82%. It is observed that, most of the works reported aims at improvising the recognition accuracies using the machine learning based techniques and classifiers working based on kernelling those features extracted from images. Also, the degree of experimentations conducted towards recognition of North Indian scripts are high compared to south Indian scripts. Nonetheless, there also exists few significant contributions on the printed Kannada text recognition and even handwritten too confining to the boundaries of simple compound characters where huge complexity of semantic analysis is involved. Thus, in this research deep learning networks are employed for recognition of Kannada text both in printed and handwritten form by considering a large number of classes.

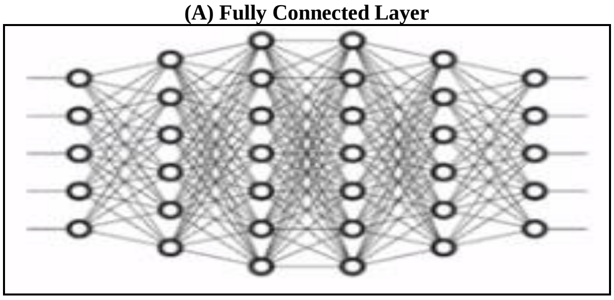
3. Proposed methodology

3.1. Architecture of convolutional neural networks

The architecture of deep convolutional neural networks is an artificial neural network which differentiates from multi-layer perceptron networks and which is widely used for various pattern recognition tasks. According to studies conducted by Nithin et al. [12] and Aloysius et al. [2] in deep learning and convolutional neural networks are more suitable and preferably used for recognition and classification when dealing with variations in data and automatic learning of domain specific features. Local connections, layers and spatial invariances inspire the architecture.

Unlike fully connected networks, neurons in convolutional networks are only connected to a small portion of nodes in its previous layer, through which reduces the load of handling huge number of parameters and reduces number of neurons in subsequent layers as shown in Fig 2. A particular context of classifying handwritten characters that are highly variable geometrical properties compared to printed characters, the task of handwritten recognition is truly complex with usual fully connected networks. As the fact that, each individual possesses more than three to four handwriting styles for writing same character resulting formation of huge dataset and also the deformations along with it. A convolutional network classifies the character patterns in images piece by piece, where each piece is known to be a filter which is considered from the original image.

(A) Fully Connected Layer



(B) Convolutional Networks

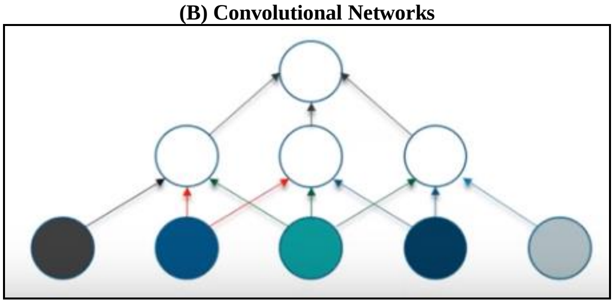


Fig. 2: Types of Neural Networks.

The architecture of the convolutional network employed for handwritten Kannada character classification is as detailed subsequently. Three main types of layers are used to build Convolutional neural network architectures, mainly convolutional layer, pooling layer and fully connected layer. The layers comprising convolutional network in the proposed experimentation is tabulated in table 1 and the architecture of the same is as represented in Fig 3.

Table 1: Layers of Convolutional Neural Network

7x1 Layer array with layers	
1 Image Input	28x28x1 images with 'zerocenter' normalization
2 Convolution	20 5x5 convolutions with stride [1 1] and padding [0 0]
3 ReLU	ReLU
4 Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0]
5 Fully Connected	52 fully connected layer
6 Softmax	softmax
7 Classification output	crossentropyex

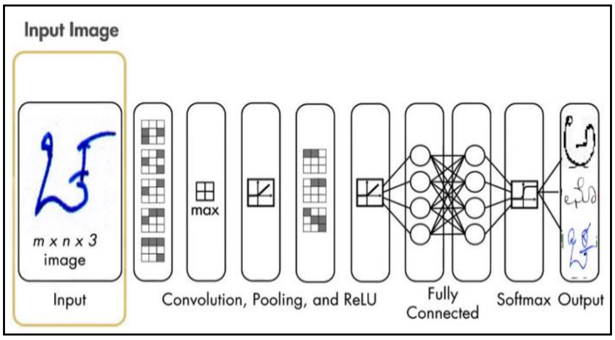


Fig. 1: Architecture of Convolutional Neural Networks.

Let f_i represents a filter of dimension pxq where $i=1,2,3...20$ and $p=q=5$, where each filter f_i is independently subject to convolution process $Conv$ over an image I acquired from input layer and finally ending up with a total of 20 feature maps of size $28x\,28x\,1$. Each filter f_i is slide through an image I with a stride length of [1] to compute the convolved feature maps of image I. For each position of filter f_i over a particular pixel P in an image I, the product of filter weights with corresponding underlying image gray levels is performed, and finally the sum of all the products obtained as the result to derive the feature map which designates the reduced representation of an image I as depicted in Fig. 4.

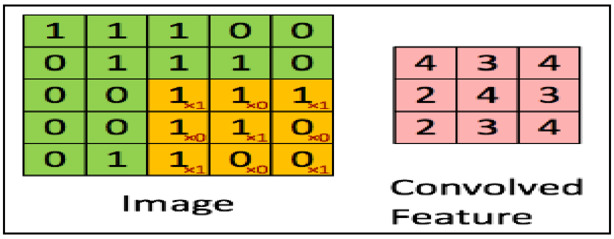


Fig. 2: Convolution of a Filter with an Image.

Further, the convolved feature maps are subject to RELU layer which rectifier that improvises the performance of convolutional networks by reducing the number of computations involved by operating each feature map independently. RELU is a non-linear activation function like sigmoid or tanh that replaces all the negative entities in feature maps to zero. The ReLU function is applied element-wise to the output of matrix-vector product obtained through convolution as given by Eq. (1).

$$f(x)=\max(0,x) \tag{1}$$

Where f is a function and x is the input element which is result of matrix products. The outcome of the RELU is directed to max pooling layer where it progressively reduces the spatial size of the feature map representations, thereby reducing the number of parameters and computations in the network.Max pooling performs the over fitting on the linearized convolved outcomes with the help of max filter and produces more abstract representation of the convolved outcomes. Dimensionality reduction is achieved by transforming a feature map of dimension [m, n] to [m/2, n/2] by retaining the depth parameter as it is through down sampling. Fig. 5 depicts the process of max pooling.

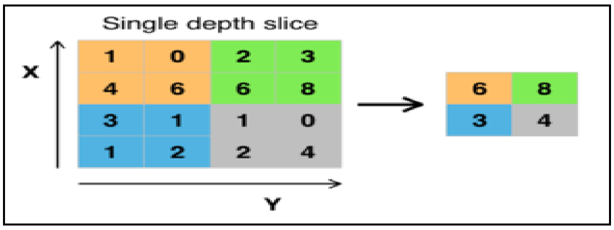


Fig. 3: Feature Map Representation to Max Pooling.

3.2. Equations

The spatial dimensions of a filter f' and stride length S' are the two essential parameters to undergo max pooling. The input volume of size $P_1^{(l-1)}xP_2^{(l-1)}xP_3^{(l-1)}$ is reduced to output volume of size $P_1^{(l)}xP_2^{(l)}xP_3^{(l)}$ and the empirical quantities pertaining to the volume size reduction is given by Eq. (2) through Eq. (4).

$$P_1^{(l)}=P_1^{(l-1)} \tag{2}$$

$$P_2^{(l)} = \frac{(P_2^{(l-1)} - f^{(l)})}{S^l} + 1 \qquad (3)$$

$$P_3^{(l)} = \frac{(P_3^{(l-1)} - f^{(l)})}{S^l} + 1 \qquad (4)$$

The pooling layer operates with a filter $f^{(l)}xf^{(l)}$ over the convolved feature maps by stepping through a stride length of $S^{(l)}$ until the entire feature map is exhausted.

Subsequently, the reduced volumes are forwarded to fully connected layer as input, which converts the max pooled activations into the class probability distributions. Neurons in fully connected layer possess complete set of associations with the neurons in the previous layer. The fully connected layer acts like one of the conventional classifier like SVM. The basic operation performed in this layer is summing up of weights which are obtained as outputs from the previous layer. The higher the probability of particular neuron decides the class to which the test input is categorized in the target classes. The outcome of the fully connected layer is a K-dimensional vector which is collection of arbitrary real values with larger range, hence the soft max function employed to normalize such a wider range in the interval of [0,1]. The function is given by Eq. (5) and Eq. (6).

$$\sigma : R^k \to [0,1]^k \quad \sigma(z_j) = \frac{e^z j}{\sum_{k=1}^{K} e^z k} \qquad (6)$$

For $j = 1,2,3...K$.

Finally, the cross-entropy cost function estimates the error involved in classification of overall samples which are fed as input to the network.

4. Experimental results

The performance of proposed experimentation is carried out with over 5200 broken/distorted handwritten characters which are developed in unconstrained environments and 5200 printed data samples that are extracted from variety of printed ancient documents in Kannada. The overall datasets comprise of 10400 data sample out of which training and testing is conducted with two variant proportions as 60%:40% and 70%:30% respectively. The performance metrics of two different evaluations conducted are as tabulated in table 2 and table 3.

Table 2: Experimental Results with 60% Training Data and 40% Test Data

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	1.93	7.5341	3.91%	1.00e-04
3	50	20.70	3.0766	19.53%	1.00e-04
5	100	39.31	1.6037	60.16%	1.00e-04
7	150	57.53	1.2829	64.84%	1.00e-04
9	200	76.85	0.9812	75.00%	1.00e-04
11	250	95.82	0.5510	85.16%	1.00e-04
13	300	115.24	0.6743	82.03%	1.00e-04
14	350	135.16	0.4954	92.19%	1.00e-04
15	360	138.82	0.4688	89.06%	1.00e-04

Table 3: Experimental Results with 70% Training Data and 30% Test Data

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.45	7.0362	6.25%	1.00e-04
2	50	18.50	3.0596	21.09%	1.00e-04
4	100	36.70	1.6964	53.13%	1.00e-04
6	150	56.27	1.5578	62.50%	1.00e-04
8	200	75.29	1.2707	65.63%	1.00e-04
9	250	93.33	0.8132	78.91%	1.00e-04
11	300	111.71	0.6301	85.16%	1.00e-04
13	350	129.79	0.6031	85.94%	1.00e-04
15	400	147.80	0.4939	90.63%	1.00e-04
15	420	154.99	0.4770	89.06%	1.00e-04

The number of epochs chosen is 15 which have taken over 360 iterations when the training and testing proportions are 60 and 40 whereas 420 iterations when the proportions are considered as 70 and 30. The mini-batch accuracy reported during training corresponds to the accuracy of the mini-batch at the given iteration. An iteration corresponds to the calculation of the network's gradients for each mini-batch. Number of iterations is the number of times the gradient is estimated and the parameters of the neural network are updated using a batch of training instances. A batch size of B is the number of training instances employed in iteration and an epoch corresponds to moving through every available mini-batch during the training and testing process.

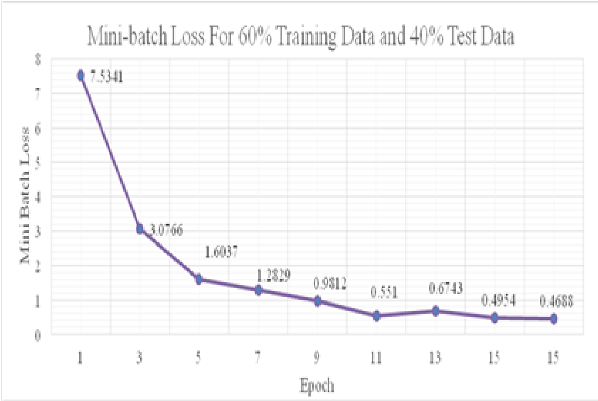


Fig. 4: Performance Metrics- Mini Batch Loss versus Epochs-60:40.

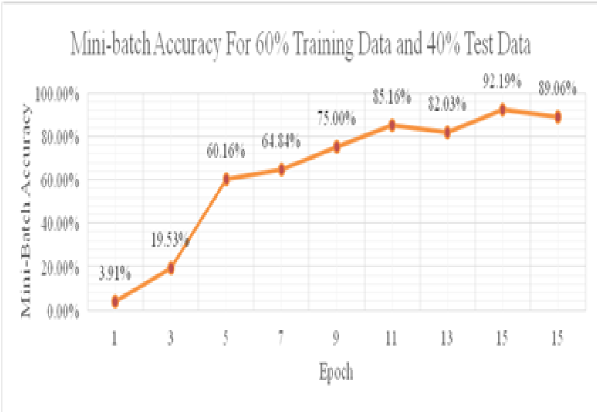


Fig. 5: Performance Metrics- Mini Batch Accuracy versus Epochs-60:40.

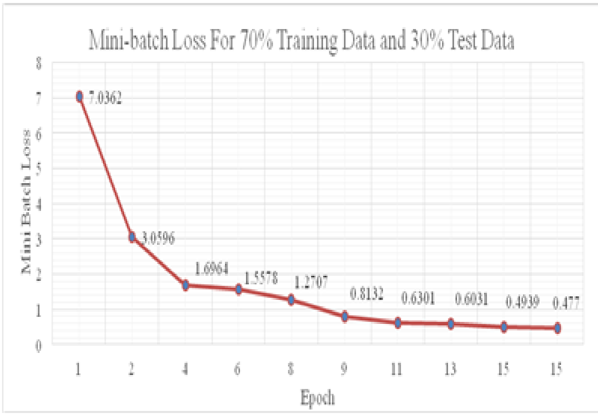


Fig. 6: Performance Metrics- Mini Batch Loss versus Epochs-70:30.

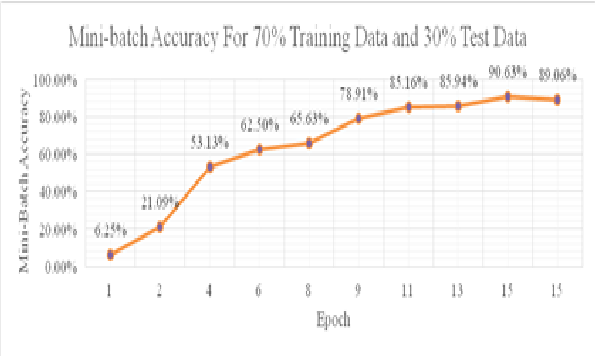


Fig. 7: Performance Metrics- Mini Batch Accuracy Versus Epochs-70:30.

It is observed clearly the mini batch loss had decreased with the increase in the number of epochs which indicates that the reduction in batch size during the training also increases the accuracy along with the reduction in error as depicted Fig. 6 through Fig. 9.

5. Datasets

The datasets employed for training in case of printed data samples are extracted from the ancient Kannada documents whereas the handwritten data samples are collected in varied unconstrained environments as depicted in Fig. 10 through Fig. 11. The printed data samples as depicted in figure 10(a) and (b) are subject to pre-processing for the removal of the artifacts like pen marks, ink marks, show through effects, foxing effect etc. The few instances of data samples as depicted in Fig. 1 are the outcomes achieved through the pre-processing technique. Similarly, the handwritten samples have also underwent required noise removal and morphological enhancements prior to classification using convolutional neural networks.

(A)

(B)

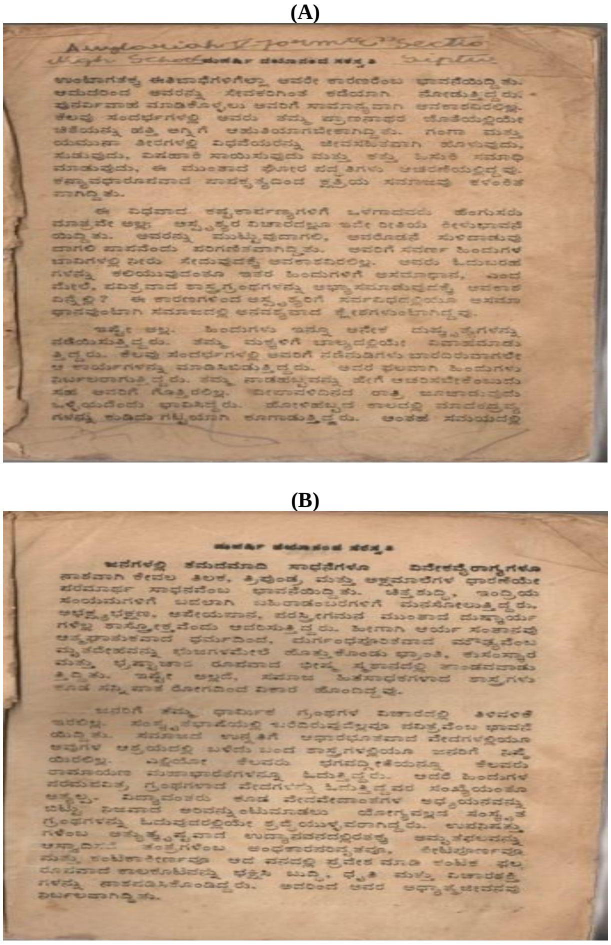


Fig. 8: Instances of an Ancient Printed Document.

(A)

(B)

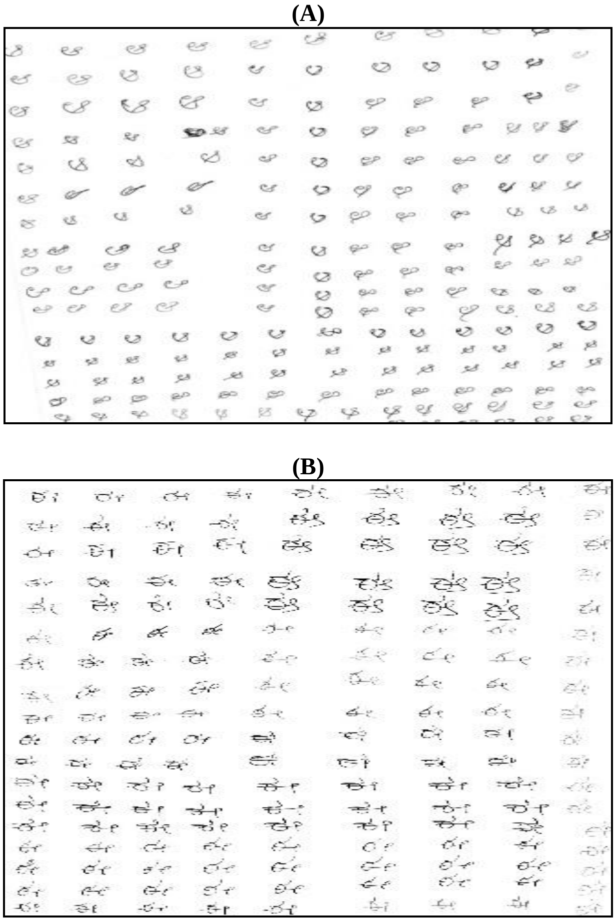


Fig. 9: Instance of a Handwritten Data Collected.

6. Conclusion

In the proposed work, we have conducted the classification of the deformed character set of printed and handwritten Kannada characters using deep convolutional neural networks. In both the cases of printed or handwritten, the datasets considered are more distorted or broken in appearance leading to challenges during recognition process with conventional classifiers like KNN, SVM etc. An overall accuracy of 87 % is achieved during the recognition of printed character set and an accuracy of over 80% is accomplished with respect to handwritten data samples with training and testing proportions of 60 and 40 % in both of the cases for the number of classes of over 52.

Acknowledgement

First, we feel deeply indebted to Her Holiness Most Revered Mata Amritanandamayi Devi (Amma) for her inspiration and guidance both in unseen and unconcealed ways.

Wholeheartedly, we thank our college, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Mysore campus, India, for providing the necessary environment, infrastructure, encouragement and for extending the support possible at each stage of this project.

We express our sincere gratitude and indebtedness to our parents who have bestowed their great guidance at appropriate times by providing encouragement in planning and carrying out the project.

References

[1] Ahmad, I., Wang, X., Li, R., & Rasheed, S. (2017). Offline Urdu Nastaleeq optical character recognition based on stacked denoising autoencoder. *China Communications*, *14*(1), 146-157.https://doi.org/10.1109/CC.2017.7839765.

[2] Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In *Communication and Signal Processing (ICCSP), 2017 International Conference on* (pp. 0588-0592). IEEE.https://doi.org/10.1109/ICCSP.2017.8286426.

[3] Anil, R., Manjusha, K., Kumar, S. S., & Soman, K. P. (2015). Convolutional neural networks for the recognition of Malayalam characters. In *Proceedings of the Third International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014* (pp. 493-500). Springer, Cham.

[4] Arica, N., &Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *31*(2), 216-233.https://doi.org/10.1109/5326.941845.

[5] Bhowmik, T. K., Bhattacharya, U., &Parui, S. K. (2004, November). Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. In *International Conference on Neural Information Processing* (pp. 814-819). Springer, Berlin, Heidelberg.https://doi.org/10.1007/978-3-540-30499-9_125.

[6] Ciresan, D. C., Meier, U., Gambardella, L. M., &Schmidhuber, J. (2011, September). Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (pp. 1135-1139). IEEE.

[7] Dewan, S., &Chakravarthy, S. (2012, November). A system for offline character recognition using auto-encoder networks. In *International Conference on Neural Information Processing* (pp. 91-99). Springer, Berlin, Heidelberg.https://doi.org/10.1007/978-3-642-34478-7_12.

[8] Droettboom, M. (2003, May). Correcting broken characters in the recognition of historical printed documents. In *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on* (pp. 364-366). IEEE.https://doi.org/10.1109/JCDL.2003.1204889.

[9] Graves, A., & Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In Advances in neural information processing systems (pp. 545-552).

[10] Lavrenko, V., Rath, T. M., &Manmatha, R. (2004). Holistic word recognition for handwritten historical documents. In *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on* (pp. 278-287). IEEE.https://doi.org/10.1109/DIAL.2004.1263256.

[11] Loey, M., El-Sawy, A., & EL-Bakry, H. (2017). Deep Learning Autoencoder Approach for Handwritten Arabic Digits Recognition. *arXiv preprint arXiv:1706.06720*.

[12] Nithin, D. K., & Sivakumar, P. B. (2015). Generic feature learning in computer vision. *Procedia Computer Science*, *58*, 202-209.https://doi.org/10.1016/j.procs.2015.08.054.

[13] Pal, A., &Pawar, J. D. (2015, April). Recognition of online handwritten Bangla characters using hierarchical system with Denoising Autoencoders. In*International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015*.IEEE,0047-0051.https://doi.org/10.1109/ICCPEIC.2015.7259440.

[14] Pal, Umapada & Chaudhuri, Bidyut. (2004). Chaudhuri, B.B.: Indian script character recognition-A survey. Pattern Recognition 37, 1887-1899. Pattern Recognition. 37. 1887-1899. https://doi.org/10.1016/j.patcog.2004.02.003.

[15] Pratama, M. O & Kareen,P.(2017,October). Reconstructing Japanese handwritten images using auto-encoder with residual block in parallel computing, 2017 *International Conference on Electrical Engineering and Informatics (ICELTICs)*, Banda Aceh, 231-234.https://doi.org/10.1109/ICELTICS.2017.8253262

[16] Rani, D. A. N. S., Vineeth, P., & Ajith, D. (2016). Detection and removal of graphical components in pre-printed documents. *International Journal of Applied Engineering Research*, *11*(7), 4849-4856.

[17] Rani, N. S., & Vasudev, T. (2018). An Efficient Technique for Detection and Removal of Lines with Text Stroke Crossings in Document Images. In *Proceedings of International Conference on Cognition and Recognition* .Springer, Singapore. 83-97.https://doi.org/10.1007/978-981-10-5146-3_9

[18] Sreelakshmi, U. K., Akash, V. G., & Rani, N. S. (2017, April). Detection of variable regions in complex document images. In *Proceedings ofInternational Conference on Communication and Signal Processing (ICCSP), 2017*.IEEE, 0807-0811.https://doi.org/10.1109/ICCSP.2017.8286476.

[19] Yang, W., Jin, L., Xie, Z., & Feng, Z. (2015). Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge. *arXiv preprint* .https://arxiv.org/abs/1505.07675

[20] Zhong, Z., Jin, L., & Xie, Z. (2015, August). High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps. In *13th International Conference on Document Analysis and Recognition* (ICDAR), 2015. IEEE, 846-850.https://arxiv.org/abs/1505.04925