Generalization through augmentation in deep neural networks for handwritten character recognition

Rajneesh Tiwari, Aritra Sen, Arindam Banerjee

Abstract— Handwritten character recognition is an active and widespread research area in the field of computer vision. However successful implementation of handwritten character recognition systems, especially for cursive handwriting, remains a challenge. Over the years, different shallow and deep neural networks have been proposed for Handwritten character recognition and most of the experiments are done on a few widely used benchmark datasets. Gauging the generalization ability of the deep neural networks for different cursive, handwritten, vernacular texts is quite challenging. Our initial research showed that image augmentation can be used as a regularization tool in neural networks and a combination of augmentation techniques (such as Cutout, Cutmix, and Mixup) can be powerful regularizers in classifying handwritten characters from images. In this paper, we focus on comparing a shallow and a deep neural network architecture for recognizing cursive handwritten texts from images using a novel combination of augmentation methods and loss functions (cross-entropy loss and online hard example mining). Through our experiments, we establish the best combination of advanced image augmentation techniques along with loss functions best suited for the multiclass-multilabel image classification task.

Index Terms—Cutout, Cutmix, Handwritten Character Recognition, Image Augmentations, Mixup, Online Hard Example Mining, Resnet, SeResnext.

1 Introduction

Over the last decade, deep learning research has shown significant success for the conceptual bases of machine learning and artificial intelligence. Deep learning-based classification and recognition of handwritten characters [1], [2] from image data is crucial for the advancement of automation and human-machine interaction in numerous real-life use cases. Over the years several benchmark datasets (such as MNIST [3]) have been developed and studied primarily focusing on classifying language-specific components.

Existing deep neural network architectures have been thoroughly studied on the widely used benchmark datasets [4]. However, the model's generalization ability across different languages with different complexity levels, especially for cursive handwriting, are not thoroughly explored. The higher the number of alphabets and potential diacritics (accents) present in a language, the more challenging the character recognition from cursive handwriting task becomes.

The image datasets inherently suffer from variability in terms of cursive writing. In this paper, we focus on image augmentation techniques to improve the accuracy and generalization ability of different neural network architectures. Our experimental results show that heavy image augmentation techniques with "Online Hard Example Mining (OHEM)" [5], [6], [7], [8] achieve the highest validation macro averaged recall score for deeper models. This suggests planning the augmentation strategies based on the depth and complexity of the deep neural network architecture. Our results also demonstrate that the inclusion of Cutout [7] in low augmentation settings ends up hurting the overall model performance. For all experiments, we leverage the Bengali AI dataset [9] and report the macro averaged recall metric on a fixed validation dataset.

2 Dataset Details

The dataset used in this experimentation process contains images of multiple variations of Bengali hand-written characters. In the Bengali language, along with 18 potential diacritics, there are 11 vowels and 38 number of consonants. Due to the high volume (around 13000) of graphemic variations in Bengali, recognition of Bengali hand-written images has additional complexity. There are mainly three components of Bengali graphemes:

- Grapheme root: 168 Classes in the dataset
- Vowel diacritic: 11 Classes in the dataset
- Consonant diacritic: 7 Classes in the dataset

3 Related Work

One of the early applications of back-propagation networks for recognizing handwritten digits from images was presented by Y. Lecun et al. [1]. A comparative study of the performance of several classifier algorithms on standard datasets (NIST's Special Database 3 and Special Database 1) was shown in [10]. Besides raw accuracy, rejection, training time, recognition time, and memory requirements were also considered there. Different machine learning architectures such as Multilayer Perceptron (MLP) [11], [12], the radial basis function (RBF) network [13], [14], deep learning [15], [16], [17] etc. have been

- *Rajneesh Tiwari is currently working as a Data Scientist at Ericsson, India, PH-+91-9899796858. E-mail: rajneesh.vish1@gmail.com*
- *Aritra Sen is currently working as a Data Scientist at Ericsson, India, PH-+91-9832035202. E-mail: aritra.slg@gmail.com*
- *Arindam Banerjee is currently working as a Data Scientist at Ericsson, India, PH-+91-8335887866. E-mail: arindam.banerjee14@yahoo.com*

thoroughly studied in literature for character recognition. It has also been proven that feature extraction techniques are influential in the performance of character recognition (including cursive handwritten texts) [18], [19], [20], [21], [22], [23], [24], [25] by machine learning algorithms.

Unlike existing works, we plan to show here a combination of image augmentation techniques to achieve improved classification performance. In our previous work [26], we thoroughly evaluated the impact of various augmentation strategies on model performance for shallow (in the context of pretrained Convolution Neural Networks) neural network architectures (Resnet34). In this paper, we evaluate all possible combinations of highly advanced image augmentation pipelines for deep CNN models (seResNext50_32X4d).

4 Technical Details

4.1 Pretrained Models

All the modern state of the art Deep Learning image classification problem is now getting solved using the concept of transfer learning. Instead of training a model from scratch, a pretrained model which has been trained previously on a different learning problem can be used with the model architectures and along with the weights. Generally, these models are trained on large benchmark datasets like Imagenet, CIFAR, and the process of using these pre-trained models on different image recognition tasks is called Transfer Learning. To fit any pre-trained models to new learning problems, we remove the last fully connected layer and a new a fully connected layer to match the current image recognition task. In this paper we use two pre-trained models:

- SeResnext50 – a ResNext50 [27] model with special Squeeze and Excitation Blocks [28]
- Resnet34 [29]

4.2 Loss Functions

In the analysis, we deploy Cross-entropy (CE) loss and OHEM (Online Hard Example Mining) loss functions.

Cross-Entropy loss or Log loss can be used as a Performance metrics in case of a classification model with a probabilistic output. The higher the difference between the predicted probability and the actual class label, the higher the Cross-Entropy loss to penalize the deviation from the actual class.

Hard Example Mining is a way to pick hard examples (training examples with greater loss values) to improve model performance. In this process first, we train the neural network for a few instances, then you identify the examples with greater loss values and run the network along with the previously identified examples. This whole process is sub-optimal and computationally expensive since the network is frozen. Online Hard Example Mining [7] helps to solve the above-mentioned issues. In a batch-wise manner, OHEM performs a hard example of mining. For any given batch, once the forward propagation is done, the loss is calculated for all training examples. Then OHEM finds hard examples in that batch that have higher losses and it only back-propagates the loss computed over these selected training examples.

4.3 Augmentation Techniques

Deep Convolutional Neural Networks (CNN) typically runs by leaning millions of parameters for tasks like Image Classification. However, they also run into the potential problem of overfitting. Many Image Augmentation techniques can help to tackle the problem of overfitting by generating new training examples from existing ones to increase generalization. Some of the techniques used in this paper are discussed below:

Cutout: Cutout augmentation [7] adds noise to the incoming batch dataset by dropping of contiguous regions ensures that entire dropped out regions are propagated throughout the network via the feature maps. Cutout introduces regularization by not allowing the network to rely only on a specific set of visual features. Cutout ensures that the entire context of the image is utilized for the task. An example of cutout augmentation is shown in fig. 1:

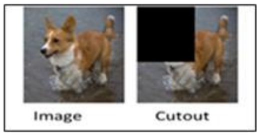

Fig. 1. Cutout

Mixup: Mixup [6] reduces overfitting by creating new images which are the convex combination of image pairs and their labels. Mixup also increases robustness to adversarial examples and much better generalization in case of noisy image labels. An example of mixup augmentation is shown in fig. 2:

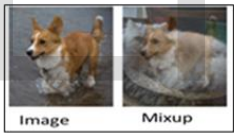

Fig. 2. Mixup

Cutmix: In the case of Cutmix [5], multiple patches are cropped and mixed amongst the images. Ground truth labels are also mixed in the same proportion as the proportion of the cut patches. Cutmix generally provides robustness against corrupted labels. An example of cutmix augmentation is shown in fig. 3:

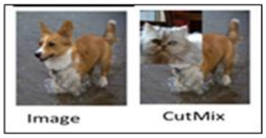

Fig. 3. CutMix

5 Experiments and Discussions

In this section, experimental results on the Bengali Handwritten dataset are discussed. In all total of 14 experiments were conducted with a static-fixed validation dataset comprising 20% data, while the remaining 80% was used for model training. Results suggest that in general deep/large models (seResnext50 variant) demonstrate much better training performance compared to shallow Resnet34 [26] models. These larger models however tend to easily overfit in less aggressive augmentation scenarios and hence we used multiple complex image augmentation techniques such as Cutmix, Mixup, Cutout, and all possible combinations of those to aid model generalization and better validation performance.

Further, we also observe that a vanilla Cross-Entropy loss does not always work best in classification scenarios and a different version of the cross-entropy loss i.e. Online Hard Example Mining loss worked better in almost all the comparable cases.

A. Results

Macro averaged recall is presented for the experiments conducted. The experiment-parameter settings are the same as that of our previous study with Resnet34 models [26]. For completeness, we are listing down the parameters of the pipeline.

- Image Size: 136 X 237
- Augmentation parameters for all experiments:
  - Baseline Augmentation: Gaussian Noise, ShiftScaleRotate (Sigma = -1; shift limit = 0.01, scale limit = 0.1, rotate limit = 30)
  - Cutmix Augmentation: The combination ratio λ between two data points is sampled from the beta distribution Beta (β, β). In our all experiments, we set β to 0.4, that is λ is sampled from the uniform distribution (0, 0.4).
  - Cutout Augmentation: Mask size = 80, Probability = 1.
  - Mixup Augmentation: The combination ratio λ between two data points is sampled from the beta distribution Beta (β, β). In all experiments, we set β to 0.4, which is λ is sampled from the uniform distribution (0, 0.4).

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \quad (1)$$

x_i , x_j are raw input vectors

$$\tilde{y} = \lambda y_i + (1-\lambda)y_j \quad (2)$$

y_i , y_j are one-hot label encodings

  - OHEM Loss: The Rate parameter in OHEM defines the proportion of examples in a batch to be considered as hard examples. In our experiments, we used Rate = 0.5 or 50% of all examples in a batch to be considered as hard examples.
- In any experiments, if more than 1 augmentations (apart from baseline) were used, then equal application probability was assigned to all augmentations [26].

For all experiments, all other augmentations mentioned are in addition to the baseline augmentation. The results from our previous study for ResNet34 models [26] are shown in Table 1:

Table 1. performance of Resnet34 model with a combination of augmentations and loss functions [26]

Aug: 1	Aug: 2	Aug: 3	Loss Function	Train Metric Value	Validation Metric Value
None	None	None	CE Loss	93.94	91.02
None	None	None	OHEM Loss	93.16	91.75
CutMix	None	None	CE Loss	94.30	92.4
CutMix	None	None	OHEM Loss	92.67	91.97
CutMix	Mixup	None	CE Loss	93.39	92.36
CutMix	Mixup	None	OHEM Loss	93.56	92.84
Cutout	None	None	CE Loss	93.21	90.68
Cutout	None	None	OHEM Loss	91.88	90.8
Cutout	CutMix	None	CE Loss	93.53	92.06
Cutout	CutMix	None	OHEM Loss	93.57	92.76
Cutout	Mixup	None	CE Loss	93.57	91.66
Cutout	Mixup	None	OHEM Loss	92.57	91.71
Cutout	Mixup	CutMix	CE Loss	93.46	92.92
Cutout	Mixup	CutMix	OHEM Loss	93.07	92.77

Results from the deep se-Resnext50_32X4D model are appended in the below table:

Table 2. performance of se-Resnext50_32X4d model with a combination of augmentations and loss functions

Aug: 1	Aug: 2	Aug: 3	Loss Function	Train Metric Value	Validation Metric Value
None	None	None	CE Loss	96.64	93.51
None	None	None	OHEM Loss	95.76	94.28
CutMix	None	None	CE Loss	96.87	94.72
CutMix	None	None	OHEM Loss	96.05	94.89
CutMix	Mixup	None	CE Loss	96.23	95.14
CutMix	Mixup	None	OHEM Loss	95.98	95.25

Cutout	None	None	CE Loss	95.87	93.22
Cutout	None	None	OHEM Loss	94.54	93.25
Cutout	CutMix	None	CE Loss	96.76	94.56
Cutout	CutMix	None	OHEM Loss	95.93	94.88
Cutout	Mixup	None	CE Loss	96.93	93.96
Cutout	Mixup	None	OHEM Loss	95.32	94.08
Cutout	Mixup	CutMix	CE Loss	95.94	95.36
Cutout	Mixup	CutMix	OHEM Loss	96.21	95.75

Although no hyperparameter tuning was conducted for these experiments, we believe the observed trends concerning model performance and augmentations, loss functions will hold.

In general, and as expected, the larger/deep the model the better its training and validation score. However, we also observe that these larger models exhibit higher overfitting. To overcome this problem of overfitting, we deploy another loss function called Online Hard Example Mining loss, which, lowers overfitting in deep models.

In the below charts, we present an analysis of the model performance across various criteria such as model category, the strength of augmentations, and loss functions.

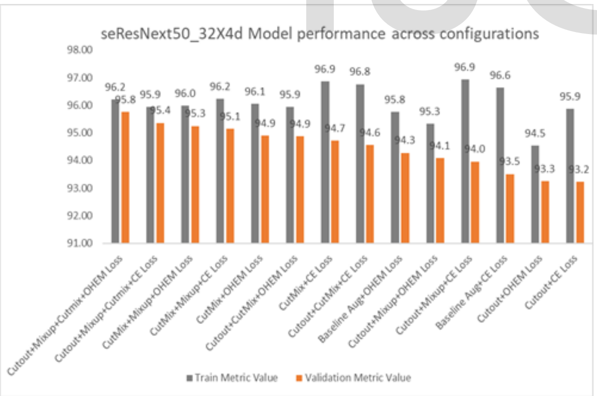

Fig. 4. seResNext50 Model Performance

For deep models, such as seResNext50_32X4d, aggressive augmentations coupled with online hard example-based loss function (OHEM) demonstrates the highest validation performance. This is since the aggressive augmentations help increase generalization power and allows the same model to see newer novel training examples, ultimately helping improve validation performance.

It is further noteworthy to see that for the seResNext50_32X4d model, the OHEM loss always performs better than the respective augmentation configuration's CE loss-based performance (eg: CutMix CE loss vs Cutmix OHEM loss).

B. Discussion

1) *Analysis of augmentations for best macro averaged recall performance for Deep Networks*

To understand which augmentation works best for the deep network (seResNext50_32X4d), we analyze the maximum validation score for each unique augmentation configuration across experiments for the seResNext50_32X4d model.

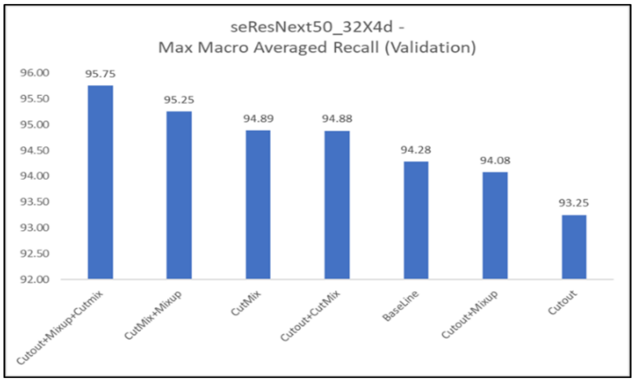

Fig. 5. seResNext50 Maximum Macro Averaged Recall Values

Most aggressive augmentation (i.e. all 3 augmentations in scope) registers the highest macro averaged recall on the validation set, followed by CutMix+Mixup augmentation, while Cutout registers the worst performance overall.

We note that aggressive augmentation helps model performance since the model encounters novel examples while training which ultimately helps in increasing the generalization power of the model.

We further note that for deep models considered here, Cutout generally lowers the validation score, and to use cutout, it is advised to deploy it within a pipeline that includes other aggressive augmentations as well.

2) *Analysis of augmentations for on-average performance in Deep Networks*

We analyze the average performance of each augmentation by averaging unique augmentation's performance across all conducted experiments for the seResNext50_32X4d model.

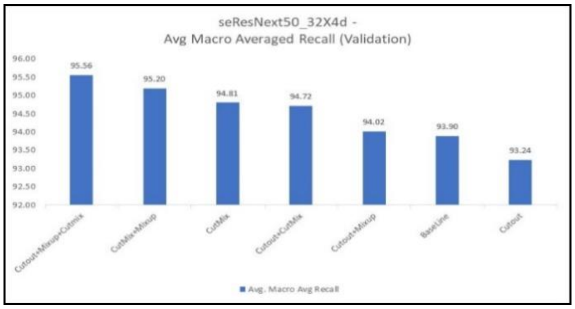

Fig. 6. seResNext50 Average Macro Averaged Recall Values

Most aggressive augmentation registers the highest average score across experiments, followed by CutMix+Mixup. Base-

Line and Cutout form the least performing augmentations. Interestingly, BaseLine performs better than Cutout as cutout leads to loss of information, while the baseline simply adds new/noisy information to the batch pipeline.

3) *Analysis of augmentations for overfitting in Deep Networks*

We analyze the average overfitting (i.e. train vs validation performance) across augmentations for the seResNext50_32X4d model.

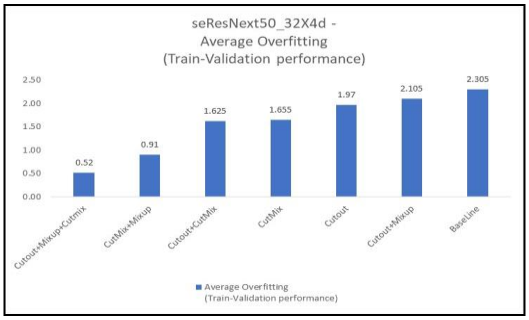

Fig. 7. seResNext50 Average Overfitting Details

As expected, the experiments with the most aggressive augmentation (Cutout+CutMix+Mixup) registers the lowest overfitting followed by CutMix+Mixup. BaseLine augmentation, as expected, registers the highest overfitting.

4) *Analysis of loss functions for best macro averaged recall performance for Deep Networks*

To understand which loss function works best for the deep network (seResNext50_32X4d), we analyze the max validation score for each unique loss function configuration across experiments.

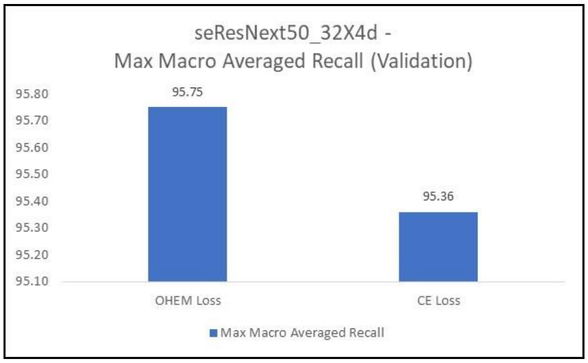

Fig. 8. seResNext50 Max Macro Averaged Recall Details

As expected, OHEM loss registers a higher max score on the validation dataset due to its higher generalization ability.

5) *Analysis of loss functions for on-average performance in Deep Networks*

We analyze the average performance of each loss function by averaging the loss function's performance across all conducted experiments for the seResNext50_32X4d model.

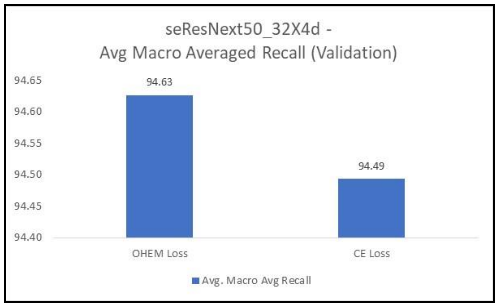

Fig. 9. seResNext50 Average Macro Averaged Recall Details

As expected, OHEM loss registers a higher average score on the validation dataset due to its higher generalization ability.

6) *Analysis of loss functions for overfitting in Deep Networks*

We further analyze the average overfitting (i.e. train – validation performance) across loss functions for the seResNext50_32X4d model.

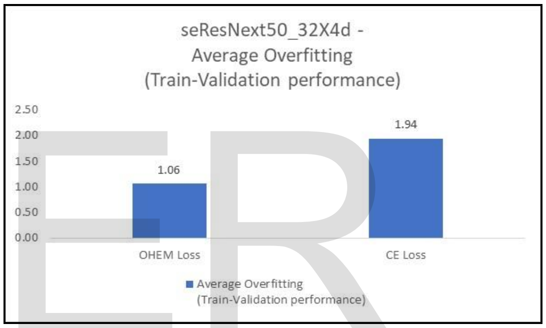

Fig. 10. seResNext50 Average Overfitting Details

OHEM loss performs significantly better than vanilla CE loss in terms of reducing overfitting across experiments. Overall, overfitting is reduced approximately by a factor of half while using OHEM in place of CE loss function for deep networks.

6 Conclusion

In our previous study [26], we thoroughly evaluated the impact of various augmentation strategies on model performance for shallow models (Resnet34). In this paper evaluated all possible combinations of highly advanced image augmentation pipelines for deep CNN models (seResNext50_32X4d) and provide the following recommendations based on our analysis.

In contrast to our previous study [26] on shallow models (Resnet34), current results found from our analysis show that heavy image augmentation techniques with "Online Hard Example Mining" achieve the highest validation macro averaged recall score for deep models.

This suggests separate augmentation strategies depending upon the depth and complexity of the network. Generally, for deep networks, OHEM loss provides better results. In line with results from our previous study on Resnet34 [26], we find that the inclusion of Cutout in low augmentation settings ends up hurting the overall model performance. Thus, it is recommended to use Cutout augmentation only with heavy augmentation pipelines.

REFERENCES

[1] LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Handwritten digit recognition with a back-propagation network." In Advances in neural information processing systems, pp. 396-404. 1990.

[2] Baldominos, Alejandro, Yago Saez, and Pedro Isasi. "A survey of handwritten character recognition with mnist and emnist." Applied Sciences 9, no. 15 (2019): 3169.

[3] LeCun, Yann. The MNIST Dataset Of Handwritten Digits (Images). 1999.

[4] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of Big Data 6, no. 1 (2019): 60.

[5] Yun, Sangdoo, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. "Cutmix: Regularization strategy to train strong classifiers with localizable features." In Proceedings of the IEEE International Conference on Computer Vision, pp. 6023-6032. 2019.

[6] Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. "mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).

[7] DeVries, Terrance, and Graham W. Taylor. "Improved regularization of convolutional neural networks with cutout." arXiv preprint arXiv:1708.04552 (2017).

[8] https://erogol.com/online-hard-example-mining-pytorch/

[9] Bengali.AI Handwritten Grapheme Classification. https://www.kaggle.com/c/bengaliai-cv19

[10] LeCun, Yann, L. D. Jackel, Leon Bottou, A. Brunot, Corinna Cortes, J. Denker, Harris Drucker et al. "Comparison of learning algorithms for handwritten digit recognition." In International conference on artificial neural networks, vol. 60, pp. 53-60. 1995.

[11] Bellili, Abdel, Michel Gilloux, and Patrick Gallinari. "An MLP-SVM combination architecture for offline handwritten digit recognition." Document Analysis and Recognition 5, no. 4 (2003): 244-252.

[12] Javidi, Mohammad Masoud, Reza Ebrahimpour, and Fatemeh Sharifizadeh. "Persian handwritten digits recognition: A divide and conquer approach based on mixture of MLP experts." International Journal of Physical Sciences 6, no. 30 (2011): 7007-7015.

[13] Lee, Yuchun. "Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks." Neural computation 3, no. 3 (1991): 440-449.

[14] Ebrahimpour, Reza, Alireza Esmkhani, and Soheil Faridi. "Farsi handwritten digit recognition based on mixture of RBF experts." IEICE Electronics Express 7, no. 14 (2010): 1014-1019.

[15] Qiao, Junfei, Gongming Wang, Wenjing Li, and Min Chen. "An adaptive deep Q-learning strategy for handwritten digit recognition." Neural Networks 107 (2018): 61-71.

[16] Cireşan, Dan Claudiu, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. "Deep, big, simple neural nets for handwritten digit recognition." Neural computation 22, no. 12 (2010): 3207-3220.

[17] Ashiquzzaman, Akm, and Abdul Kawsar Tushar. "Handwritten Arabic numeral recognition using deep learning neural networks." In 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1-4. IEEE, 2017.

[18] O.D. Trier, A.K. Jain, T. Taxt, Feature extraction methods for character recognition—a survey, Pattern Recognition 29 (4) (1996) 641–662.

[19] M. Yasuda, H. Fujisawa, An improvement of correlation method for character recognition, Trans. IEICE Japan J62-D (3) (1979) 217–224.

[20] Y. Yamashita, K. Higuchi, Y. Yamada, Y. Haga, Classification of handprinted Kanji characters by the structured segment matching method, Pattern Recognition Lett. 1 (1983) 475–479.

[21] F. Kimura, et al., Evaluation and synthesis of feature vectors for handwritten numeral recognition, IEICE Trans. Inform. Systems E79-D (5) (1996) 436–442.

[22] M. Shi, Y. Fujisawa, T. Wakabayashi, F. Kimura, Handwritten numeral recognition using gradient and curvature of gray scale image, Pattern Recognition 35 (10) (2002) 2051–2059.

[23] D.-S. Lee, S.N. Srihari, Handprinted digit recognition: a comparison of algorithms, in: Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition, Buffalo, NY, 1993, pp. 153–164.

[24] C.-L. Liu, Y.-J. Liu, R.-W. Dai, Preprocessing and statistical/structural feature extraction for handwritten numeral recognition, in: A.C. Downton, S. Impedovo (Eds.), Progress of Handwriting Recognition, World Scientific, Singapore, 1997, pp. 161–168.

[25] L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier, C. Olivier, A structural/statistical feature based vector for handwritten character recognition, Pattern Recognition Lett. 19 (7) (1998) 629–641.

[26] Rajneesh Tiwari, Aritra Sen, Arindam Banerjee, Kaushik Dey, “Empirical analysis of generalization through augmentation for classifying images of vernacular handwritten texts”, unpublished.

[27] Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated residual transformations for deep neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492-1500. 2017.

[28] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141. 2018.

[29] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.