International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Japanese Historical Character Recognition using Deep Convolutional Neural Network (DCNN) with DropBlock Regularization

Sujata Saini, Vishal Verma

Abstract: ***The history of a country always helps in the development of making modern society. Japanese classical handwritten/literature dataset Kuzushiji-MNIST and Kuzushiji-49 recently introduced in public domain by Center of Open Data in the Humanities (CODH). The availability of the datasets invite the researchers in different domains. The hidden treasure of information in these datasets resolved and worked by many researchers. The research on these datasets can be helpful to explore and analyze the detail of Japanese era/literature, which ultimately helps in the development of modern society. In this study, Deep Convolutional Neural Network (DCNN) with DropBlock regularization is used to recognize the hiragana characters in Japanese historical script.***

Index Terms: Japanese Handwritten Character Recognition, Historical Document's Classification, Deep Convolutional Neural Network (DCNN), Deep Learning, DropBlock Regularization.

I. INTRODUCTION

Japan has a unique culture and historical pathway. Due to the Japanese isolation from the west, the cursive Kuzushiji language is no-longer be the part of the modern Japanese language system. Most Japanese unable to read 150 years old cursive Kuzushiji-style handwritten documents [3]. Japanese hiragana characters in classical literature are different from modern hentaigana characters. In this study, the purpose of the developed model is to resolve the problem of historical character recognition in Japanese classical literature. The proposed model trained with two highly challenged historical datasets such as Kuzushiji-MNIST, and Kuzushiji-49 are curated by ROIS-DS Center for Open Data in the Humanities (CODH) and created by the National Institute of Japanese Literature (NIJL) [30]. The main objective of this work is trying to achieve a desirable state of the art results with the proposed model using Deep Convolutional Neural Network (DCNN) and DropBlock regularization method to classify Japanese historical dataset. Deep Convolutional Neural Network (DCNN) is a sort of deep learning algorithm that developed to be handle image classification problems. In the case of enlarging data, deep learning gains more popularity over machine learning algorithms for giving a significant state of the art performances [2]. Deep Convolutional Neural Network widely known for accomplish the image classification with different model architectures: LeNet-5 [17], AlexNet [18], GoogleNet [19], VGGNet [20], ResNet [21], and DenseNet [22].

In associated w\ith CNN, dropout was extremely popular for increasing the extensive performance of a neural network [1]. Dropout used to drop the features randomly, it can be effective for fully connected layers but not for feature layers to correlate spatially [13, 14]. The formulation of another dropout method is called *SpatialDropout* associated with Cascade architecture for object localization [15]. The regularization effect of *selective max-drop* and *stochastic dropout* are moderate to the dropout and *SpatialDropout* [14]. On the other hand, DropBlock regularization, a framework of dropout, employ to regularize the convolutional neural network for dropping the contiguous units through the feature map and perform better than dropout and *SpatialDropout* [12]. To ensemble, the batch normalization to the neural network results improved by 4.9% on ImageNet classification [16]. In this study, the efficiency as the primary concern, the proposed convolutional neural network architecture using DropBlock for the purpose of achieving better results comparing other regularization and improving the classification accuracy by enlarging the data to compete for different deep learning models.

II. RELATED WORKS

In this paper, the author' trying to recognize handwritten dataset lies in different categories and complicated in nature. Convolutional Neural Network (CNN) used to recognize the MNIST database for an exact number of predictions [23]. Multi-Column Deep Neural Networks achieved close to human performance on the very competitive MNIST handwriting benchmark [24]. Deep Convolutional Neural Network (DCNN) used to identify Japanese handwritten characters, the dataset consists of three different types of scripts hiragana, katakana, and Kanji and achieve 99.57% recognition rate on overall classification [4]. 4-Nearest Neighbour Baseline Model used to recognize MNIST, Kuzushiji-MNIST and Kuzushiji-49 dataset [3]. A CNN architecture for online detection of handwritten Kanji characters achieved accuracy up to 89% [5]. In the field of Japanese character recognition, previous work yielded an accuracy of accuracy rate of 97% by using fully Convolutional Neural Network (CNN) based filter [6].

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Sujata Saini*, Department of Computer Science and Applications, Chaudhary Ranbir Singh University, Rohtak, India.
Vishal Verma, Department of Computer Science and Applications, Chaudhary Ranbir Singh University, Jind, India.

Recurrent Neural Network (RNN) applied to recognize isolated online handwritten Japanese characters to perform sequence classification task by achieving an accuracy of 97.91% on Nakayosi, and 97.74% on Kuchibue [7].

A geometry-topology based algorithm used to identify the Japanese hiragana characters and performed with an averaged accuracy rate of 94.1% [8]. In the state-of-art recognition, a Deep Convolutional Neural Network achieved an accuracy of 97.20% and 96.87% on CASIA-OLHWDBl.0 and CASIA-OLHWDBl.1 dataset to recognize Chinese handwritten character recognition [9]. To recognizing offline handwritten Japanese character using Deep Convolutional Recurrent Network (DCRN) on TUAT Kondate database and achieve accuracy of 96.35% and 98.05% with and without generating synthetic data [10]. The writer identification for Japanese handwritten documents with the accuracy of best identification 95.0% is and worst identification is 31.5% using CNN [11].

III. PROPOSED ARCHITECTURE

The basic architecture of the proposed model is inspired by the architecture of LeNet-5 [22]. The proposed model employed into this paper based on Deep Convolutional Neural Network (DCNN) and DropBlock Regularization method. The architecture of the developed model consists of feature layers (convolution layers) and classification layers (fully-connected layers). The input images deliver to the network take 64 images (28x28 greyscale) per batch at a time. The overall architecture is depicted in fig. 1.

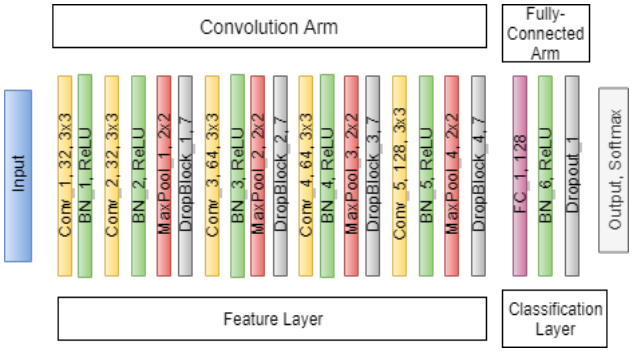


Fig. 1. The Architecture of the proposed model

Adam optimizer used for optimization and it takes the prior values of the parameter for producing effective and smooth learning. The hyperparameters used in Adam optimization are $\beta 1$ and $\beta 2$ are two hyperparameters generally close to 1, α is the learning rate, amsgrad contains boolean variant and gradient of learning parameter and ϵ is a persistent value sum up with the divisor to avoid division by zero. For enforcement, $\beta 1 = 0.9$, $\beta 2 = 0.99$, and learning rate $\alpha = 0.001$ were used for optimization. The parameter and hyperparameter settings for each layer in this architecture are explained in the further subsection.

A. Input layer

The first layer of DCNN architecture is the input layer. The input layer contains zero padding layer so that the input and output will have the same height and width (shown in fig. 2). The input shape of greyscale images will be (height) x (width) x (no. of channels=1). In this study, the input shape is 28 x 28x 1. After applying the zero paddings the input shape will be 30 x 30 x 1. However, DCNN can also accept 1D, 2D, and 3D data volume and process with slight alterations.

B. Convolution layer

The Convolution layer is the second layer takes the input as filter size = 32, stride = (1,1) and, kernel size = 3. In DCNN, several types of activation functions can be used like sigmoid, tanh, ReLU, leaky ReLU function, etc. In this experiment, the popular ReLU activation function is applied. The activation function used for adding non-linearity for obtaining a well-known generalized solution. In the convolution layer, the paddings are same as Zero-padding in the input layer, the output of this layer will pass to the next layer and the bias emerges arbitrarily.

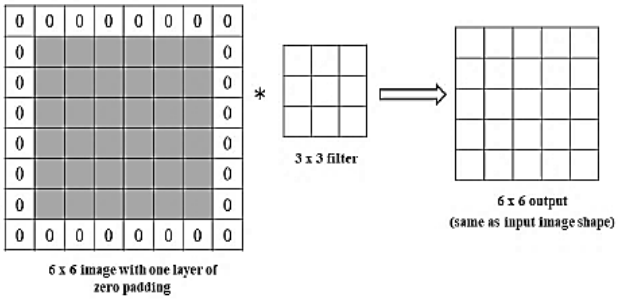


Fig. 2. Zero-padding in input layer before convolution

A. Max-Pooling Layer

After extracting features from the input, Max-Pooling layer reduces the height and width of the input and decreases the chance of overfitting. Pooling layer is basically used after convolution layers. By minimizing the dimension and parameters, this layer increases computing performance effectively. Pooling can be done by using various methods; max pooling, average pooling. After applied Max-Pooling layer with stride=2, the output of the layer pass to the next layer.

C. Batch-Normalization layer

In this layer, the dispersion of every layer's input changes during training. So, the method of using batch-normalization solving the problem of normalizing layer inputs. The model architecture performed better for each training mini-batch in less time with normalization.

D. DropBlock layer

In DropBlock layer, the *block_size* for all feature maps were constant through the training. For implementation, the *block_size*=7 used for improving efficiency with higher *block_size*. In traditional dropout, to hold each activation interpreted through the likelihood of *keep_prob*. In this experiment, *keep_prob*=0.8, having a tendency to work great throughout numerous hyperparameter settings.

$$\gamma = \frac{1 - keep\_prob}{block\_size^2} \frac{feat\_size^2}{(feat\_size - block\_size + 1)^2} \quad (1)$$

The number of features to drop depends upon γ, and it cannot be set explicitly. In equation 1, the dual mask sampled with Bernoulli distribution with the Mean (1-*keep_prob*). Each zero entry in the mask enlarged by $block\_size^2$ and blocks are fully comprised in the feature map. There is γ can be adjusted according to a sampling of an initial binary mask.

The magnitude of a legitimate space $(feat\_size - block\_size + 1)^2$, where *feat_size* indicates the size of the feature map [12].

E. Fully-Connected layer

Transit the input over the featured layers, the output from the last layer is flattened into a one-dimensional array and then feed to the fully connected layer. The fully-connected layer in the proposed architecture consists of 128 nodes and the output from the prior layers is densely connected with these nodes. To reduce overfitting, dropout provides significant advancements with different regularization methods. In this layer, the dropout *rate*=0.5 to reduce overfitting through the connection of the layers.

F. Output layer

The output layer used to get by softmax output unit and the specified output classes (10, 49, etc.) allocated a prospect value that total up to 1. However, the greatest value contained by a class will be the correct class.

For training the model, loss function used to measure the performance by alluring the predicted class and the actual class. For this paper, *sparse_categorical_crossentropy* used as the loss function of the model. Adam optimizer used to set the parameters to make the smallest value of loss function. To validate the model, test data used for generalized the model properly.

IV. EXPERIMENT EVALUATION

Google Colab is used to implement deep learning concept into this study. The experiment is conducted on a Windows 10 operating system laptop containing Intel® Core (TM) i5-7200U CPU @ 2.50GHz 2.70GHz, 8GB RAM, Intel® HD Graphics 620. In this experiment, the proposed model is executed with Keras library functions [31] with Tensorflow as backend. The datasets used in the proposed model, hyperparameter tuning and obtained results are given of following subsections.

A. Dataset Description

In this study, the Kuzushiji dataset contains the characters for the train and test sets are sampled from 35 books, which means the data distributions of each class are consistent between the two sets. The dataset contains total 3,999-character types and 403,242 characters in the form of Kuzushiji-MNIST, and Kuzushiji-49.

Hiragana	Unicode	Samples	Sample Images
は (ha)	U+306F	7000	
ま (ma)	U+307E	7000	
や (ya)	U+3084	7000	
れ (re)	U+308C	7000	
を (wo)	U+3092	7000	

Hiragana	Unicode	Samples	Sample Images
お (o)	U+304A	7000	
き (ki)	U+304D	7000	
す (su)	U+3059	7000	
つ (tsu)	U+3064	7000	
な (na)	U+306A	7000	

Fig. 3. Kuzushiji-MNIST contains 10 classes of hiragana characters dividing into 6000 training and 1000 testing size per class [3].

In this experiment, Kuzushiji-MNIST and Kuzushiji-49 were chosen to perform classification [30]. Kuzushiji-MNIST is balanced across the dataset and contains 10 rows of hiragana characters (shown in fig. 3) which are different from hentaigana (the modern Japanese dataset). Kuzushiji-49 has several rare characters with a small no. of samples (~400) and it consists of 49 classes (266,407 images) [3]. Both Kuzushiji-MNIST and Kuzushiji-49 consists of the 28x28 pixel resolution of greyscale images (shown in fig. 4) The design of Kuzushiji-MNIST dataset is a drop-in replacement of MNIST and also inspired by Fashion-MNIST [3].

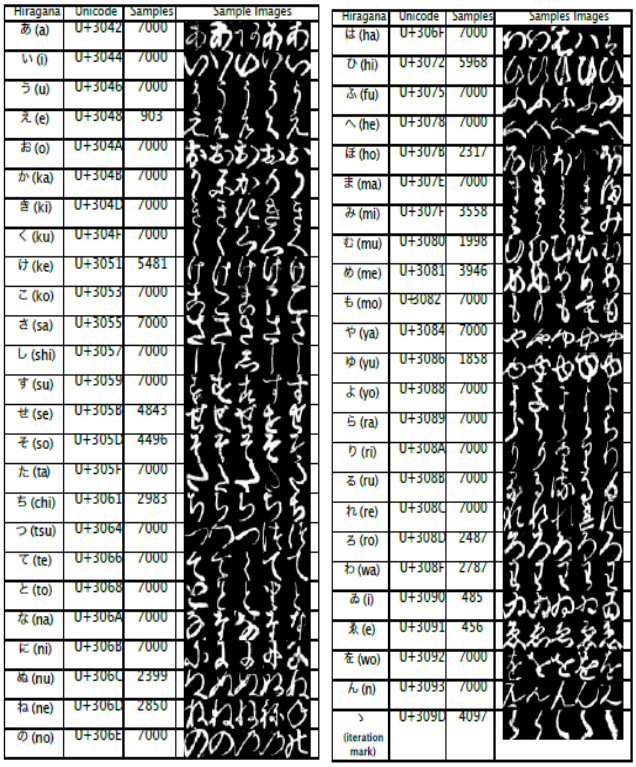


Fig. 4. Kuzushiji-49 contains 49 classes of hiragana characters dividing into 232365 training and 38547 testing samples [3].

In this research, for comparing the performance of Kuzushiji dataset with original MNIST and another useful companion Fashion-MNIST dataset. The MNIST dataset constructed from National Institute of Standards and Technology NIST's database. MNIST refers as Modified Institute of Standards and Technology based on digits database written by 250 writers contains 60,000 training and 10,000 testing images with 28x28 greyscale [28]. On the other hand, Fashion-MNIST dataset is more complicated than the MNIST used in this study. Fashion-MNIST database contains 28x28 greyscales images of fashion items. The dataset contains total 70,000 images (60,000 for training and 10,000 for testing) [29]. The dataset is available on their GitHub repository page.

B. Hyperparameter Tuning

The datasets were trained and tested on Le-Net5 [17] based architecture performed well with Adam optimization and *sparse_categorical_crossentrpoy* loss function. For evaluating the values of the hyperparameters each time, the performance of the model to validate two different datasets. In the case of the number of epochs, the proposed model checked for 10, 30, 50 and, 100 epochs. With little difference of 100 epochs for Kuzushiji-MNIST and 50 epochs for Kuzushiji-49 demonstrated proper to the developed model (shown in fig. 5).

Fig. 5. Performance analysis of several batch sizes.

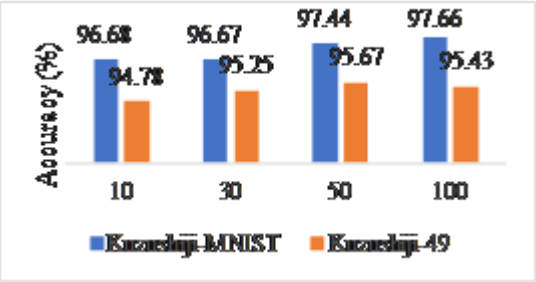
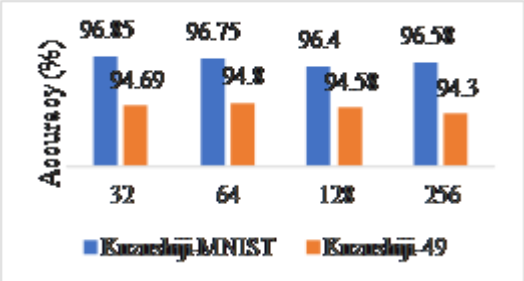


Fig. 6. Performance analysis of a model trained on a various number of epochs.



The developed model trained with four various batch sizes 32, 64, 128 and 256. A batch of 64 represents the finest result (as shown in fig. 6). Due to the large batch size, the required training time per epoch increased substantially.

C. Results and Discussion

In this paper, the developed model trained to identify Japanese Classical handwritten literature datasets such as Kuzushiji-MNIST and Kuzushiji-49. For comparative analysis, MNIST and Fashion-MNIST dataset used to encourage the precise performance of the model.

Table I
Comparative Analysis Of Proposed Model With Different Regularization

Regularization	MNIST (%)	Fashion-MNIST (%)	Kuzushiji-MNIST (%)	Kuzushiji-49 (%)
DropBlock	**99.47**	**93.40**	**97.66**	**95.67**
Dropout	97.99	85.47	86.43	95.34
Spatial Dropout	97.17	84.44	81.08	58.18

According to the Table1, the proposed model train with three regularization methods such as DropBlock, Dropout, and *SpatialDropout*. DropBlock achieves the accuracy rate of 97.66% on Kuzushiji-MNIST, 95.67% on Kuzushiji-49, 99.47% on MNIST and, 93.40% on Fashion-MNIST dataset; it is the highest accuracy rate itself as compare to other regularization methods. The testing accuracy and loss function of Kuzushiji-MNIST dataset depicted in fig. 7. Also, the testing accuracy and loss function of Kuzushiji-49 dataset depicted in fig. 8.

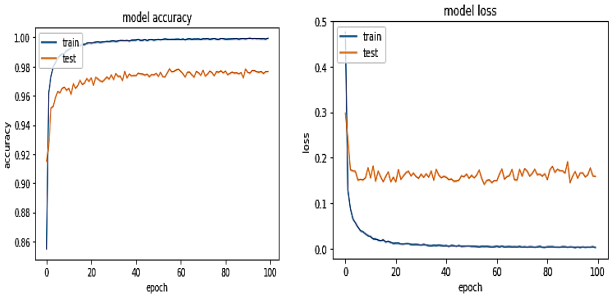


Fig. 7. Accuracy rate and loss of the proposed model with Kuzushiji-MNIST dataset

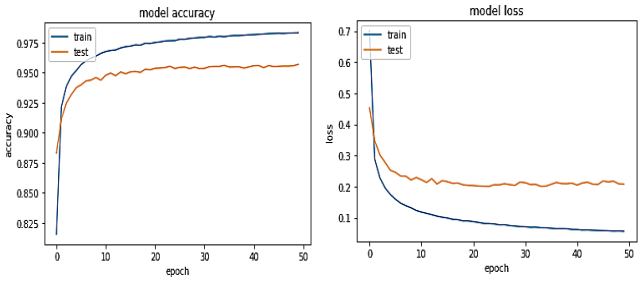


Fig. 8. Accuracy rate and loss of the proposed model on different epochs with Kuzushiji-49 dataset.

For comparative analysis, the developed model compared with other deep learning models such as MobileNet [27], LeNet-5 [17], Transfer Learning with CNN [26], Simple CNN [23], AlexNet [18], Naïve Bayes [25] and 4-Nearest Neighbor Baseline [3] used to measure the performance on four unique datasets.

Table II
Comparative Analysis Of Proposed Model With Other Models

Model	MNIST (%)	Fashion-MNIST (%)	Kuzushiji-MNIST (%)	Kuzushiji-49 (%)
4-Nearest Neighbor Baseline	97.14	85.97	91.59	86.00
Naïve-Bayes	98.06	88.60	92.17	88.44
AlexNet	98.19	87.47	91.82	81.01
Simple CNN	99.08	92.54	95.02	90.42
Transfer Learning with CNN	99.34	97.46	97.06	83.96
LeNet-5	99.13	91.33	94.66	89.64
Mobilenet	99.20	93.04	95.09	91.06
Proposed Model	**99.47**	**93.40**	**97.66**	**95.67**

Table II showed that the developed model is outperforming than other deep learning as well as machine learning models. The best performance of the proposed model chooses on the basis of epochs, batch sizes, and regularization methods. The error measured in this work of various datasets is 0.53% on MNIST, 6.6% on Fashion-MNIST, 2.34% on Kuzushiji-MNIST and 4.33% in Kuzushiji-49. The calculated average error of the proposed model on these four datasets is 3.45%.

Fig. 9. A portion of dataset prediction after training

The dataset prediction after training process represents the orientation of classified characters of Kuzushiji-49 as shown in fig. 9.

V. CONCLUSION

In recent years, modern Japanese character recognition achieves enormous success in digital humanities. At the same time, the heritage of classical Japanese literature undefined and forbidden by the modern society. Due to availability of Kuzushiji-MNIST and Kuzushiji-49 datasets in the public domain, the majority of research purposes gain the opportunity to drive the historical datasets in the field of image classification and character recognition. The proposed model developed with Deep Convolutional Neural Network and DropBlock regularization and the results reflected that the proposed model achieved the highest accuracy rate as compared to dropout and *SpatialDropout* regularization methods. For comparison, the model competes with other various models such as 4-Nearest Neighbor Baseline, Naïve-Bayes, AlexNet, Simple CNN, Transfer Learning with CNN, LeNet-5, and MobileNet. According to the experiment, the testing accuracy of the developed model with desired datasets is 99.47% on MNIST, 93.40% on Fashion-MNIST, 97.66% on Kuzushiji-MNIST and 95.67% on Kuzushiji-49. The proposed model trained on the basis of different regularization methods, epochs, and batch sizes. The calculated average error of the proposed model on these four datasets is 3.45%. After applying Batch-Normalization, the accuracy rate of the developed model increased by 0.55% as comparing without normalization. The future scope of this paper illustrates that the proposed model can be modified to train on more challenging historical datasets such as Kuzushiji-Kanji to increasing accuracy, less time consumption and minimizing the error to handle complex problems.

ACKNOWLEDGMENT

A big thanks to open source community (GitHub, Stack Overflow and Google Colab) for the virtually infinite source of knowledge and platform to operate and share the knowledge; to CODH for their Kuzushiji Dataset for performing the classification task.

REFERENCES

1. Chollet F., Deep Learning with Python, Manning Publications, 2017. Available: https://manning-content.s3.amazonaws.com/download /4/02b02 cf-c1ac-47ae-8ffd25f85daad877/SampleCh01.pdf.
2. Goodfellow, I., Bengio, Y., & Courville, A. (n.d.), Deep Learning, MIT Press, 2016. Available: http://www.deeplearningbook.org/.
3. T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep Learning for Classical Japanese Literature," *32nd Conference on Neural Information Processing Systems)*, no. NeurIPS, 2018, pp. 1–8.
4. Charlie Tsai, "Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks," 2016, pp. 1–7.
5. M. Grębowiec and J. Protasiewicz, "A neural framework for online recognition of handwritten Kanji characters," *Proceedings of the Federated Conference on Computer Science and Information Systems*, no. ACSIS, vol. 15, 2018, pp. 479–483.
6. K. Watanabe *et al.*, "Detection of characters and their boundary from images of modern Japanese official documents using fully CNN-based filter," *Proceedings of NICOGRAPH International*, NICOINT, no. c, 2018, p. 78.
7. H. T. Nguyen, C. T. Nguyen, and M. Nakagawa, "Online Japanese handwriting recognizers using recurrent neural networks," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2018-Augus, 2018, pp. 435–440.
8. S. Das and S. Banerjee, "An Algorithm for Japanese Character Recognition," *I.J. Image, Graphics and Signal Processing*, no. December 2014, 2015, pp. 9–15.
9. W. Yang, Z. Xie, and Z. Feng, "Improved Deep Convolutional Neural Network for Online Handwritten Chinese Characters Recognition using Domain-specific knowledge," *13th International Conference on Document Analysis and Recognition*, ICDAR, 2015, pp. 551–555.
10. N. T. Ly, C. T. Nguyen, and M. Nakagawa, "Training an end-to-end model for offline handwritten Japanese text recognition by generated synthetic patterns," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2018-Augus, 2018, pp. 74–79.
11. R. Nasuno and S. Arai, "Writer Identification for Offline Japanese Handwritten Character using Convolutional Neural Network," *Proceedings of the 5th IIAE International Conference on Intelligent Systems and Image Processing 2017*, no. 16, 2018, pp. 94–97.
12. G. V Ghiasi, Tsung-Yi Lin, Quoc Le, "DropBlock: A regularization method for convolutional networks," *32nd Conference on Neural Information Processing Systems*, no. NeurIPS, 2018, pp. 1–11.
13. G. Hinton, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, 2014, pp. 1929–1958.
14. S. Park and N. K. B, "Analysis on the Dropout Effect in Convolutional," ACCV 2016, Part II, LNCS 10112, 2017, pp. 189–204.
15. J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using Convolutional Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, 2015, pp. 648–656.
16. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.", *ICML'15 Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 448-456.
17. Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "Gradient-Based Learning Applied to Document Recognition," no. November, 1998, pp. 1–46.
18. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*. *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, 2012, pp. 1097-1105.
19. C. Szegedy *et al.*, "Going Deeper with Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1-9.
20. K. Simonyan and A. Zisserman, "*Very deep convolutional networks for large-scale image recognition*," CoRR, vol. abs/1409.1556, 2014, pp. 1-15. Available: https://arxiv.org/abs/14 09.1556.
21. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778.
22. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 2261–2269.
23. H. Wu, "CNN-Based Recognition of Handwritten Digits in MNIST Database," 2018, Available: http://users.cecs.anu.edu.au/~Tom.Ge deon/conf/ABCs2018/paper/ABCs2018_paper_104.pdf.

24. D. Cires and U. Meier, "Multi-column Deep Neural Networks for Image Classification," IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3642-3649.
25. E. Frank and R. R. Bouckaert, "Naive Bayes for Text Classification with Unbalanced Classes," Knowledge Discovery in Databases: PKDD 2006, 2006, pp. 503–510.
26. B. Zoph and J. Shlens, "Learning Transferable Architectures for Scalable Image Recognition," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 8697-8710. Available: https://arxiv.org/abs/1707.07012.
27. A. G. Howard and W. Wang, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," ArXiV abs/1704.04861, 2017, Available: https://arxiv.org/abs/1704.048612012.
28. LeCun., "The MNIST database of handwritten digits," 1998. http://yann.lecun.com/ exdb/mnist/.
29. H. Xiao et al. Fashion-MNIST: A MNIST-like fashion product database, 2017. Available: https: //github.com/zalandoresearch/fashion-mnist.
30. KMNIST Dataset is created by ROIS-DS Center for Open Data in the Humanities (CODH), 2018. Available: https://codh.rois.ac.jp/kmnist/.
31. Chollet F *et al.*, "Keras." https://keras.io, 2015.

AUTHORS PROFILE

Sujata Saini is a research scholar who doing M.Phil. in Computer Science and Applications from Chaudhary Ranbir Singh University. She has done B.Sc. and MSc. in computer science from Maharishi Dayanand University, Rohtak. Her research interest in machine learning, image processing, and natural language processing. She has worked on various projects in deep learning such as image classification, object detection, and text analytics.

Vishal Verma is an Assistant Professor in Department of Computer Science and Applications in Chaudhary Ranbir Singh University, Jind. He completed his B.Tech. Computer Science and Engineering from Kurukshetra University. He has done MSc. And M.CA in Computer Science from Kurukshetra University. He completed M.Phil. Computer Science from Chaudhary Devi Lal. University, Sirsa. He received his Ph.D. Computer Science in 2012 from Kurukshetra University. He has 11 years of diverse experience in teaching and published 12 research papers. His research interest is Database and Aspect Orient Programming.