IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 4, December 2021, pp. 872~878
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i4.pp872-878

Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion-MNIST datasets

Owais Mujtaba Khanday, Samad Dadvandipour, Mohd Aaqib Lone
Institute of Information Science, University of Miskolc, Miskolc, Hungary

Article Info

Article history:

Received Nov 23, 2020
Revised Aug 27, 2021
Accepted Sep 10, 2021

Keywords:

CIFAR10
Convolutional neural network
Deep learning
Fashion-MNIST
Filter size

ABSTRACT

Convolution neural networks (CNN or ConvNet), a deep neural network class inspired by biological processes, are immensely used for image classification or visual imagery. These networks need various parameters or attributes like number of filters, filter size, number of input channels, padding stride and dilation, for doing the required task. In this paper, we focused on the hyperparameter, i.e., filter size. Filter sizes come in various sizes like 3×3, 5×5, and 7×7. We varied the filter sizes and recorded their effects on the models' accuracy. The models' architecture is kept intact and only the filter sizes are varied. This gives a better understanding of the effect of filter sizes on image classification. CIFAR10 and FashionMNIST datasets are used for this study. Experimental results showed the accuracy is inversely proportional to the filter size. The accuracy using 3×3 filters on CIFAR10 and Fashion-MNIST is 73.04% and 93.68%, respectively.

Corresponding Author:

Owais Mujtaba Khanday
Institute of Information Science
University of Miskolc
Egytemvaros, 3525, Miskolc, Hungary
Email: aitowais@uni-miskolc.hu

1. INTRODUCTION

Artificial intelligence (AI) has minimized the gap between human and machine capabilities. The researchers and enthusiasts are doing enormous work on numerous aspects of the field to make extraordinary things happen in many domains. The domain of computer vision is one such area. Various recognition or classification algorithms were developed like support vector machines (SVM), neural networks (NN), multi-level perceptrons (MLP), and many more. A lot of machine learning algorithms have been proposed to achive the task of character recognition [1]-[3]. These advancements helped machines to perceive image and video recognition, natural language processing, and much more. Deep learning is enormously perfected with time, primarily over convolutional neural networks (CNN), and gives much better results than the others [4]-[7]. CNN is now a well-established machine learning tool used for image classification problems, especially in medical sciences and practical life like detecting road signs, recognizing human activity, and facial expression recognition [8], [9].

CNN is a deep learning algorithm that takes an image as input and recognizes it. The architecture of CNN is based on human neurons. These are feed-forward neural networks that can capture the temporal and spatial dependencies by applying relevant filters. The network can extract features without manual handling [10]. CNN isused for a wider range of image recognition task like medical image recognition [11], [12], x rays recognition [13], [14], handwritten character recognition[15]-[17], and offline character recognition [18], [19]. A CNN has the input layer, convolutional layer, a max-pooling layer, and a fully connected layer. In the end, we have the SoftMax applied to classify the image to its respective class [20]. The convolutional layer is a 2D layer, and the filters are used in this layer. CNN input layer has three dimensions. The input image is a three-dimensional image comprising height, width, and depth. The depth is taken as one for the greyscale image and three for the RGB image. The filters are the feature matrixes of various sizes, e.g. 3×3, 5×5, and 7×7. The filters are represented by the vector of weights, which tries to find whether the features are available in the input image. The vector of the weights represents a feature such as a curve, edge, and shape. We have considered three types of filters: 3×3, 5×5, and 7×7, and checked the impact of these filters on the image classification.

In the convolutional layer, the output is calculated as a dot product of the weights and the input and then adds some bias. The filters slide over the input image matrix and produce the convolutional output. If the input is having dimensions of $H \times W \ \times 1$ and the N is the number of filters applied, then the output of the convolutional layer is $H \times W \ \times N$. The stride is applied to determine the number of pixels skipped for the next convolution. If stride is one, then the filter moves one pixel and calculates the convolutional output. If stride is a considerable value, then the convolutional output complexity decreases, but it also affects the accuracy. The general rule suggested is to take the stride's value less than twice the filter size [21]-[23]. A pooling layer is used for the downsampling and is applied along the spatial dimensions. If the stride is 2, then the resulting volume is $H - 2 \times W - 2 \ \times N$. The fully connected layer computes the valid class score. In this layer, every neuron is connected to every other neuron in the previous layer, and the output size will be $1 \times 1 \ \times M$, where M is the number of output classes, e.g., for the handwritten digit classification M is 10 for the cifar10 dataset M is 10 and for cifar100 dataset M is 100.

The hyperparameter called filter sizes is taken into consideration in this study. A filter is a 2d square matrix that is applied to every convolutional layer. These matrixes are of various sizes like 3×3, 5×5, and 7×7. We varied these filter sizes keeping all the other hyperparameters the same to see their effect on the accuracy and to see which ones perform the best among which circumstances.

2. DATASETS

We performed the experiments with the following two datasets i) CIFAR10 [24] and ii) FashionMNIST [25] dataset. These two datasets are small and precise datasets having low computational costs. The results drawn from these datasets can be applied to most of the datasets. So, we use these two datasets to avoid the computation cost of more extensive datasets. CIFAR10 is the dataset of the 60000 images of these categories plane, car, bird, cat, deer, dog, frog, horse, and ship. This dataset has 10 output classes. The second one is the FashionMNIST dataset that contains 10 different categories, so the output classes are 10. All the datasets have 60000 images, out of which we use 40000 for the training and 10000 for the validation, and the rest of 1000 for testing the accuracy of the model. The dataset is split into a validation dataset to train our model efficiently. The validation dataset ensures the model does not cause overfitting or underfitting on training data and performs best on test data.

3. ARCHITECTURE

For both the datasets, CIFAR10 as well as MNIST dataset, we build a CNN model with thirteen layers. The first layer is the input layer, tracked by two Conv2D layers. In each layer, 32 filters are used. Then batch normalization is used for standardizing the inputs followed by the MaxPooling2D layer, a (2, 2) pool is applied. The dropout layer is used for protecting the model from overfitting, and then the Flatten layer is used to flatten the inputs. Two Dropout layers are used in the whole model, and then at the send, one dense layer is used. Two types of activation functions are used. One is ReLU, and at the end, SoftMax is used as an activation function. Adam optimizer is employed for optimizing the model with the learning rate of 0.001, and the categorical cross-entropy function is used to calculate the error. The same model is used with all three different filter sizes, and the effect on the accuracy is measured. The architecture of the models is kept unchanged to track the effect of filter sizes on the accuracy and loss of the model.

4. EXPERIMENTAL RESULTS

For the experiment, we used a simple convolution neural network with a total of 13 layers and just two Conv2D layers. The more layers we use in CNN, the more accuracy we get, but it is not always true the model could run into overfitting. Also, the computational cost increases effectively with each layer. We have to find a balance between the number of layers we use. The architecture is given in

Figure *1*. In the first experiment, both the convolutional layers are trained with a 3×3 filter, and in each layer, 32 filters are used. In the second experiment, we have used the same architecture the number of the filters is kept unchanged, only the filter sizes are changed to 5×5, and in the third experiment, the filter size was altered to 7×7, and the rest of the network is the same.

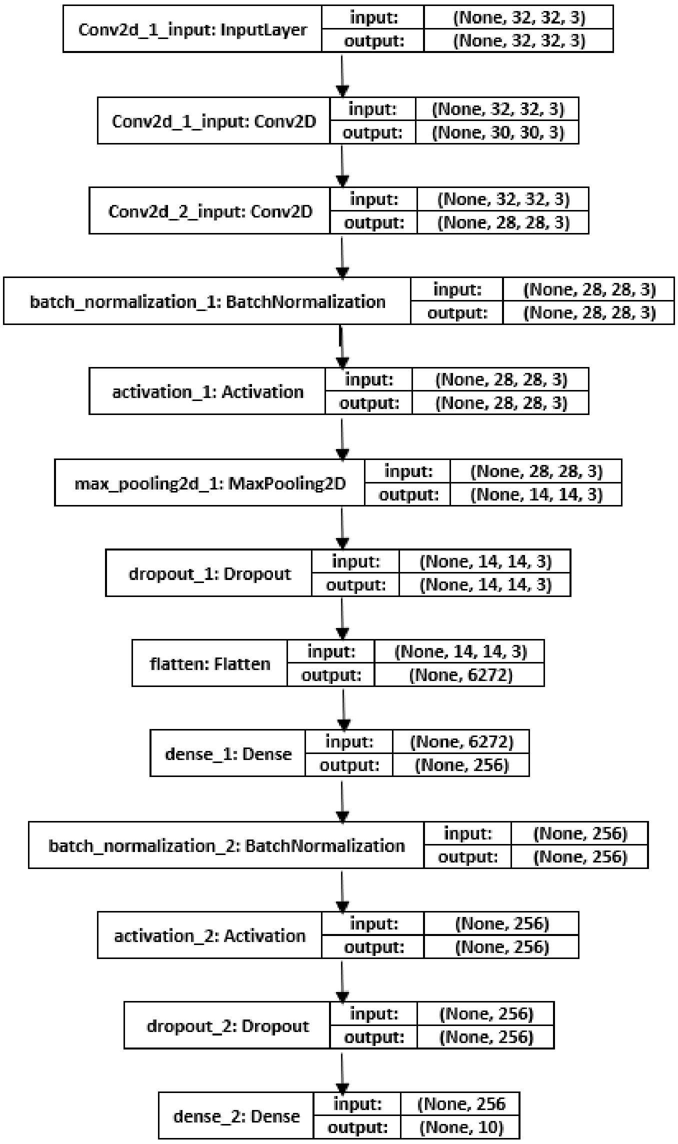
Conv2d_1_input: InputLayer | input: (None, 32, 32, 3) | output: (None, 32, 32, 3)
Conv2d_1_input: Conv2D | input: (None, 32, 32, 3) | output: (None, 30, 30, 3)
Conv2d_2_input: Conv2D | input: (None, 32, 32, 3) | output: (None, 28, 28, 3)
batch_normalization_1: BatchNormalization | input: (None, 28, 28, 3) | output: (None, 28, 28, 3)
activation_1: Activation | input: (None, 28, 28, 3) | output: (None, 28, 28, 3)
max_pooling2d_1: MaxPooling2D | input: (None, 28, 28, 3) | output: (None, 14, 14, 3)
dropout_1: Dropout | input: (None, 14, 14, 3) | output: (None, 14, 14, 3)
flatten: Flatten | input: (None, 14, 14, 3) | output: (None, 6272)
dense_1: Dense | input: (None, 6272) | output: (None, 256)
batch_normalization_2: BatchNormalization | input: (None, 256) | output: (None, 256)
activation_2: Activation | input: (None, 256) | output: (None, 256)
dropout_2: Dropout | input: (None, 256) | output: (None, 256)
dense_2: Dense | input: (None, 256 | output: (None, 10)

Figure 1. Architecture of CNN

Table 1. Accuracy of the CIFAR10 dataset using different filter sizes shows the accuracy of the training, validation, and test dataset for CIFAR10 using different filter sizes, and Table 2 shows the loss on the CIFAR10 dataset using different filter sizes. The accuracy on the CIFAR10 dataset using a filter size of 3 × 3 is highest, i.e., 94.26% on the training dataset, 72.75% validation dataset, and 63.5% on the test dataset, and the lowest accuracy is when 7 × 7 filters are used. All the model configurations are trained on 50 epochs, and the number of filters used is 32. The accuracy and the loss during training and validation are shown in Figures 2 and 3, respectively.

Table 1. Accuracy of the CIFAR10 dataset using different filter sizes

Filter Size	Training Data	Validation Data	Test Data
3 × 3	0.942625	0.7275	0.7304
5 × 5	0.923275	0.7261	0.7297
7 × 7	0.87725	0.7067	0.635

Table 2. Loss of the CIFAR10 dataset different filter sizes

Filter Size	Training Data	Validation Data	Test Data
3 × 3	0.1659	1.02	1.01
5 × 5	0.2209	0.97	0.97
7 × 7	0.3377	1.01	2.97

It is clear from Figure 2 the accuracy is increasing as the epochs increase, but the rate of increase decreases and reaches a point above which the accuracy doesn't change. The accuracy curve of the 3 × 3 filter is above the 7 × 7 and 5 × 5 filter. The validation accuracy looks the same as training accuracy, but there are many spikes, but lesser filter size curves are higher than the bigger filter sizes. Table 1 shows that accuracy is the highest with a 3 × 3 filter on the test data. So, it is concluded that the smaller filter sizes tend to give better accuracy than using larger filter sizes.

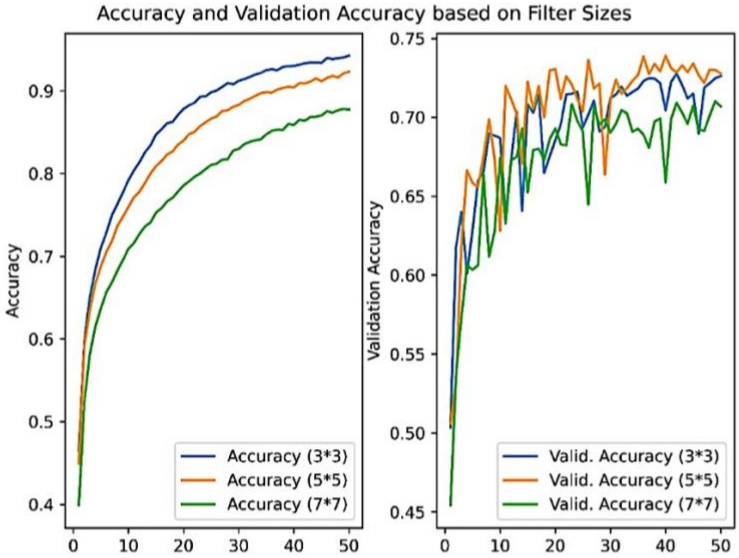

Figure 2. Training accuracy and validation accuracy using different filter sizes on the CIFAR10 dataset

Figure 3 shows that the training loss is given minimum by the 3 × 3 filter and the highest by the 7 × 7 filter. With the validation loss, it looks a bit different, but not too much. There are many spikes, but if we see the generality, the loss decreases as we use the smaller filter size and increases if we use the larger filter size. The second experiment is done on the FashionMNIST dataset with the same network configuration as done on the CIFAR10 dataset. Thirty-two filters with three different filter sizes are used in both the Conv2D layers, and the training, validation, and test accuracy and loss are measured.

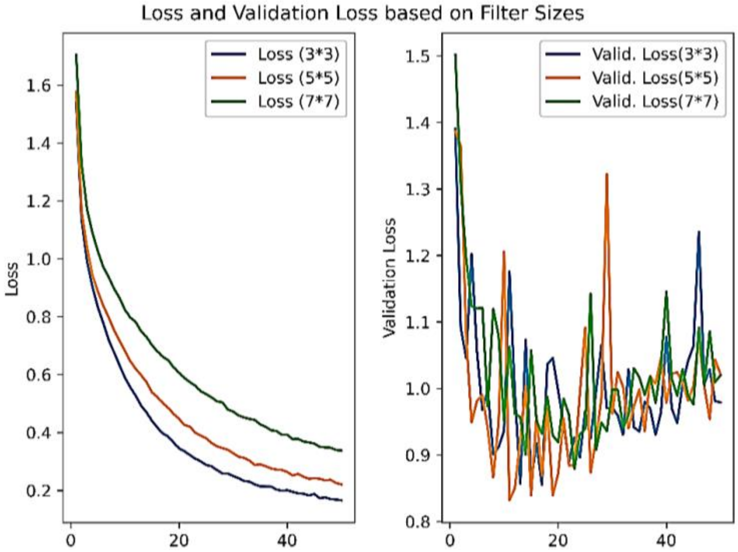

Figure 3. Training loss and validation loss using different filter sizes on the CIFAR10 dataset

Table 3 shows, the accuracy of the training dataset, validation dataset, and test dataset is maximum, i.e., 92.68%, 92.35, and 92.68%, respectively, when 3 × 3 filters are used and lowest, i.e., 91.1% when 7 × 7 filters are used. Table 4 shows the loss of the training dataset, validation dataset, and test dataset is maximum for 3 × 3 filters while in the lowest a 7 × 7 filters are used. Accuracy graphs for FashionMNIST datset is plotted in Figure 4. The 5 × 5 and 3 × 3 curves are almost the same on the training data, and their performances are the same, but the result is seen in the validation and test datasets where the 3 × 3 filter outperforms the other filters.

Table 3. Accuracy of FashionMNIST dataset using different filter sizes

Filter size	Training data	Validation data	Test data
3 × 3	0.929	0.9235	0.9268
5 × 5	0.926	0.9196	0.9264
7 × 7	0.918	0.910	0.911

Table 4. Loss of FashionMNIST dataset using different filter sizes

Filter size	Training data	Validation data	Test data
3 × 3	0.185	0.217	0.200
5 × 5	0.189	0.228	0.202
7 × 7	0.216	0.256	0.243

The loss curve for FashionMNIST dataset using different filter sizes is plotted in Figure 5. In both, the graphs loss is minimum when the 3 × 3 filter is used and maximum when 7 × 7 is used.

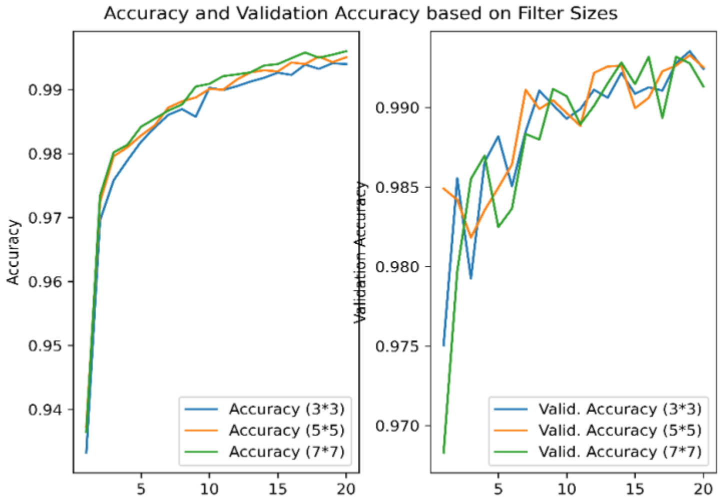

Figure 4. Training accuracy and validation accuracy using different filter sizes on the FashionMNIST dataset

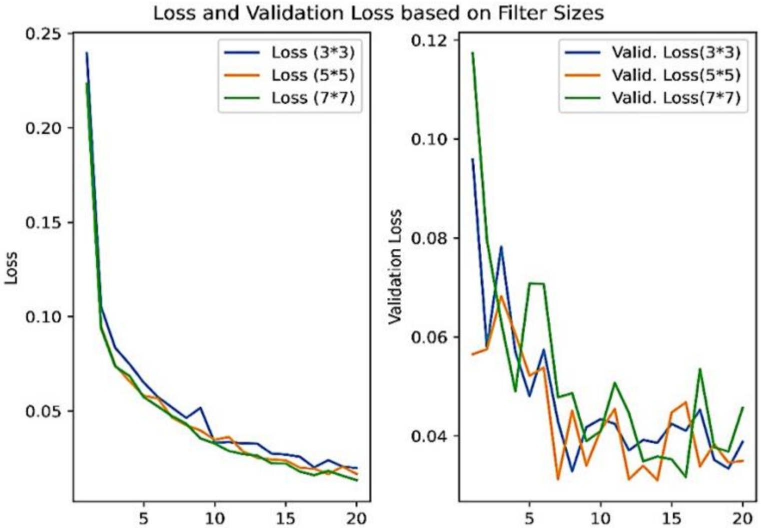

Figure 5. Training loss and validation loss using different filter sizes on the FashionMNIST dataset

5. CONCLUSION

In this paper, different CNN structures are used for image classification. The experiments have been applied on CFIR10 and FashionMNIST datasets, and we used three different filters size, and the number of filters is kept constant 32. From the experiment results, filter size has a major effect on classification accuracy, the accuracy of training, and validation. Accuracy with filter size 3×3 is higher than 5×5 and 7×7, but the training time is high when a smaller filter size is used. Besides, when filter size 3×3 pixels, testing accuracy is better than filter size 5×5 pixels and 7×7 pixels. The accuracy shown by using a 3×3 filter size in both the datasets is 92.68% in FashionMNIST and 73.04% in the CIFAR10 dataset. The only trade of using less filter size is the computational cost associated with the model.

REFERENCES

[1] A. Pal and D. Singh, “Handwritten English character recognition using neural network,” *International Journal of Computer Science and Communication*, vol. 1, no. 2, 2010, pp. 141-144.

[2] L. Heutte, T. Paquet, J. V. Moreau, Y. Lecourtier, and C. Olivier, “A structural/statistical feature based vector for handwritten character recognition,” *Pattern Recognition Letters*, vol. 19, no. 7, pp. 629-641, 1998, doi: 10.1016/S0167-8655(98)00039-7.

[3] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, “A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, pp. 258-262, 1999, doi: 10.1109/34.754617.

[4] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, “PCANet: A simple deep learning baseline for image classification?,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017-5032, 2015, doi: 10.1109/TIP.2015.2475625.

[5] W. S. Ahmed and A. A. A. Karim, “The impact of filter size and number of filters on classification accuracy in CNN,” in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 88-93, doi: 10.1109/CSASE48920.2020.9142089.

[6] O. M. Khandy and S. Dadvandipour, “Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition,” *Indonesian Journal Electrical Engineering Computer Science (IJEECS)*, vol. 21, no. 1, p. 574-581, 2021, doi: 10.11591/ijeecs.v21.i1.pp574-581.

[7] O. M. Khanday and S. Dadvandipour, “Convolutional neural networks and impact of filter sizes on image classification,” *Multidiszciplináris Tudományok*, vol. 10, no. 1, pp. 55-60, 2020, doi: 10.35925/j.multi.2020.1.7

[8] Y. Li, J. Zeng, S. Shan, and X. Chen, “Occlusion aware facial expression recognition using CNN with attention mechanism,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2439-2450, 2018, doi: 10.1109/TIP.2018.2886767

[9] S. Xie and H. Hu, “Facial expression recognition with FRR-CNN,” *Electronics Letters*, vol. 53, no. 4, pp. 235-237, 2017, doi: 10.1049/el.2016.4328.

[10] H.-H. Zhao and H. Liu, “Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition,” *Granular Computing*, vol. 5, pp. 411-418, 2020, doi: 10.1007/s41066-019-00158-6.

[11] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, “Medical image classification with convolutional neural network,” in *2014 13th International Conference on Control Automation Robotics and Vision (ICARCV)*, 2014, pp. 844-848, doi: 10.1109/ICARCV.2014.7064414.

[12] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *Journal of Big Data*, vol. 6, no. 113, 2019, doi: 10.1186/s40537-019-0276-2.

[13] T. Rahmat, A. Ismail, and S. Aliman, “Chest x-ray image classification using faster R- CNN,” *Malaysian Journal of Computing (MJoC)*, vol. 4, no. 1, 2019, doi: 10.24191/mjoc.v4i1.6095.

[14] A. A. Reshi *et al*., “An efficient CNN model for COVID-19 disease detection based on X-ray image classification,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6621607.

[15] D. S. Maitra, U. Bhattacharya, and S. K. Parui, “CNN based common approach to handwritten character recognition of multiple scripts,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1021-1025, doi: 10.1109/ICDAR.2015.7333916.

[16] M. M. Rahman, M. A. H. Akhand, S. Islam, P. Chandra Shill, and M. M. Hafizur Rahman, “Bangla handwritten character recognition using convolutional neural network,” *International Journal Image Graphics Signal Processing*, vol. 7, no. 8, pp. 42-49, 2015, doi: 10.5815/ijigsp.2015.08.05.

[17] L. Chen, S. Wang, W. Fan, J. Sun, and S. Naoi, “Beyond human recognition: A CNN- based framework for handwritten character recognition,” *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR,)*2015, pp. 695-699, doi: 10.1109/ACPR.2015.7486592.

[18] Z. Zhong, L. Jin, and Z. Xie, “High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 846-850, 2015. doi: 10.1109/ICDAR.2015.7333881.

[19] Z. Li, N. Teng, M. Jin, and H. Lu, “Building efficient CNN architecture for offline handwritten Chinese character recognition,” *Computer Science*, 2018.

[20] M. Hussain, J. J. Bird, and D. R. Faria, “A study on CNN transfer learning for image classification,” in *Advances in Intelligent Systems and Computin*, pp. 191-202, 2018.

[21] J. Hannink *et al*., “Mobile stride length estimation with deep convolutional neural networks,” *IEEE Journal of Biomedical and Health Informatics,* vol. 22, no. 2, pp. 354-362, 2018, doi: 10.1109/JBHI.2017.2679486.

[22] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *International Conference on Engineering and Technology (ICET),* 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[23] S. Lai, L. Jin, and W. Yang, “Toward high-performance online HCCR: a CNN approach with DropDistortion, path signature and spatial stochastic max-pooling,” *Pattern Recognition Letters*, vol. 89, pp. 60-66, 2017, doi: 10.1016/j.patrec.2017.02.011.

[24] “CIFAR-10 and CIFAR-100 datasets,” Toronto.edu. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html . (Accessed: Jan, 5, 2021).

[25] Zalando Research, “Fashion MNIST,” [Online]. Available: https://www.kaggle.com/zalando-research/fashionmnist (Accessed: Jan, 5, 2021).

BIOGRAPHIES OF AUTHORS

Owais Mujtaba Khanday received his B.Sc. (I.T) from the University of Kashmir (S.P College Srinagar) and M.Sc. (Computer Science) from the University of Pondicherry. Currently, he is a Ph.D. Student at the University of Miskolc, Hungary, under the Stipendium Hungaricum Scholarship program. aitowais@uni-miskolc.hu +36704202865

Samad Dadvandipour, Associate Professor, Institute of Information Sciences, University of Miskolc, 3515 Hungary dr.samad@uni-miskolc.hu