Improved CNN Based on Batch Normalization and Adam Optimizer

Roseline Oluwaseun Ogundokun[1,2,3(✉)], Rytis Maskeliunas[1], Sanjay Misra[4], and Robertas Damaševičius[5]

[1] Department of Multimedia Engineering, Kaunas University of Technology, 44249 Kaunas, Lithuania
rosogu@ktu.lt
[2] Department of Computer Science, Landmark University, Omu Aran, Nigeria
[3] Landmark University SDG 11 (Sustainable Cities and Communities Research Group), Omu Aran, Nigeria
[4] Department of Computer Science and Communication, Østfold University College, 1757 Halden, Norway
[5] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania

Abstract. After evaluating the difficulty of CNNs in extracting convolution features, this paper suggested an improved convolutional neural network (CNN) method (ICNN-BNDOA), which is based on Batch Normalization (BN), Dropout (DO), and Adaptive Moment Estimation (Adam) optimizer. To circumvent the gradient challenge and quicken convergence, the ICNN-BNDOA uses a sequential CNN structure with the Leaky rectified linear unit (LeakyReLU) as the activation function (AF). The approach employs an Adam optimizer to handle the overfitting problem, which is done by introducing BN and DO layers to the entire connected CNN layers and the output layers, respectively, to decrease cross-entropy. Through a small regularization impact, BN was utilized to substantially speed up the training process of a neural network, as well as to increase the model's performance. The performance of the proposed system with conventional CNN (CCNN) was studied using the CIFAR-10 datasets as the benchmark data, and it was discovered that the suggested method demonstrated high recognition performance with the addition of BN and DO layers. CCNN and ICNN-BNDOA performance were compared. The statistical results showed that the proposed ICNN-BNDOA outperformed the CCNN with a training and testing accuracy of 0.6904 and 0.6861 respectively. It also outperformed with training and testing loss of 0.8910 and 0.9136 respectively.

Keywords: Convolutional Neural Network · Batch normalization · Dropout · Adam optimizer · CIFAR-10

1 Introduction

Classification of images, acquisition of knowledge, and semantic segmentation of images using the convolutional neural network (CNN) have gotten a lot of consideration because

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13381, pp. 593–604, 2022.
https://doi.org/10.1007/978-3-031-10548-7_43

of its success in object recognition [1, 19]. As a result, enhancing its performance is a popular topic in a study by Vieira, Pinaya and Mechelli [1]. The central processing unit (CPU) manages the whole procedure including data preparation of the CNN when solving the image or object detection, while the graphics processing unit (GPU) advances the convolution computation in the neural network (NN) unit and the execution speed of the full-joint layer-integration cell [2].

Even though the neural network's learning speed has been improved recent, the time cost of data preprocessing and preparation for the CPU and the GPU has increased, and a feeble GPU platform is susceptible to execution disruption. The involvement of BN in deep learning (DL) [1] has improved the efficiency of deep CNN and simplified the training procedure. Due to different issues faced during training, training them gets more difficult as the network grows deeper.

The output of the convolutional layers is regularized with extra scaling and ever-changing processes after the convolutional layers (CL) and before the activation layers (AL) [20]. These are introduced after forwarding the output to the AL and then to the next CL [1]. Just by adding this layer, the network improves test data accuracy and trains quicker than a comparable model without BN. BN lowers the internal covariate shift, resulting in improved outcomes. The impact of BN on several of the most lately postulated existing networks, including the Residual Network [4], Dense Network [4], VGG Network [5], and the Inception (v3) Network [6], are investigated. The architecture's inclusion of Batch Normalization layers has not been acknowledged explicitly.

The authors presented that adding BN layers to shallow networks improves performance when compared to a network without BN. They also showed that BN is required to train a deep network in the first place. We go one step more and demonstrated that when BN is employed, Exponential Linear Units [7–9] for Residual Networks perform better than Rectified Linear Units for all other networks.

With operator theory as the AF, reference [10] integrated multi-variable all-out product and interpolation operator theory into the CNN framework, focusing on CNN convergence speed. Their research provides a comprehensive mathematical formula derivation but no findings from the experimental testing. By incorporating companion objective functions (COF) at the entire hidden layers with an inclusive objective function at the output layer, Lee et al. [11] presented deeply-supervised nets. Companion goal functions are frequently used to provide another constraint or regularization to the learning process. As a result, the effectiveness of this strategy is dependent on the creation of a COF, which is not straightforward to achieve. The CNN technique introduces a novel way of extracting image features.

After evaluating the difficulties of CNNs in extracting convolution features, this paper suggested an improved convolutional neural network (ICNN) algorithm (ICNN-BNDA), which is based on batch normalization, dropout layer, and Adaptive Moment Estimation (Adam) optimizer. The ICNN-BNDA uses a seven-layered CNN structure with the LeakyReLU unit as the activation function to circumvent the gradient problem and accelerate convergence. The approach employs an Adam optimizer to handle the overfitting problem, which is done by introducing a BN layer into the overall connected and output layers to reduce cross-entropy.

The core contributions made in this study are as follows:

1. Batch normalization is used to minimize overfitting, increase generalization, and help the model converge rapidly, reducing training time. a greater learning rate was utilized to optimize the model's training duration, and a batch size of 32 was selected since small batch sizes perform better with batch normalization and help provide the needed regularization in the proposed system.
2. Dropout (0.2) was introduced to assist prevent overfitting by randomly setting an input unit with a frequency rate of each step during the training period.
3. The Adam optimization technique is used to make the model more computationally efficient, as well as to optimize the model with big data and parameters.

The rest of this article is prearranged as follows: Sect. 2 discussed a few correlated kinds of literature that have been conducted on CNN. Section 3 presented the rudiments of our proposed model and its details. Section 4 discussed the experimental results, analysis, and interpretations. Section 5 finally concludes the article and a few further study directions were also suggested and presented.

2 Literature Review

Drop-Activation is a regularization approach suggested by Liang et al. [12], which introduces randomization into the activation function. Drop-Activation employs a deterministic network with modified nonlinearities for prediction and drops nonlinear activations in the network at random during training. The proposed technique has two distinct advantages. First, as the numerical tests show, Drop-Activation is a simple yet efficient approach for regularization.

Yamada et al. [13] presented ShakeDrop, an innovative stochastic regularization approach that can be effectively used with ResNet and its enhancements as long as the residual blocks are normalized in batches. Experimentations on the CIFAR-10 and CIFAR-100 datasets confirmed their usefulness. ShakeDrop outperformed the competition in both datasets used for the implementation.

Zhang et al. [14] introduced an innovative Residual network (RN) of Residual network architecture (RoR), which was shown to achieve a new image classification performance on CIFAR-10 and CIFAR-100. Their approach not only improved image classification performance through empirical research, but also has the potential to be a successful supplement to the RN family in the upcoming.

The exponential linear units (ELUs) were suggested by Clevert et al. [9] for quicker and further precise learning in deep NN. The network can drive the mean activations closer to zero by using ELUs with negative values. As a result, ELUs reduce the distance between the normal and unit natural gradients, speeding up learning. In its negative regime, ELUs have a distinct saturation plateau, permitting them to absorb an extra robust and durable system. On various visual datasets, experimental results reveal that ELUs greatly outperform numerous activation functions.

Furthermore, ELU networks outperform ReLU networks that have been trained with BN. Without the necessity for multi-view test, assessment, or model averaging, ELU networks obtained one of the highest ten superlative conveyed outcomes on CIFAR-10 and established an innovative state-of-the-art in CIFAR-100. Furthermore, ELU networks

on the ImageNet delivered competitive results in many fewer epochs than an equivalent ReLU network.

For (static) object recognition, Liang and Hu [15] suggested a recurrent convolutional neural network (RCNN). The primary concept was to include recurrent connections in each of the feedforward CNN's convolutional layers. This structure allowed units in the same layer to modulate the units, improving CNN's capacity to grasp statistical regularities in the context of the object. By weight sharing between layers, the recurrent connections enhanced the depth of the innovative CNN while keeping the number of parameters the same. The advantages of RCNN over CNN for object identification were proved in experiments. RCNN beat the existing models on four benchmark datasets with fewer parameters. Enlarging the number of variables resulted in even improved results.

Although many researchers have sought to enhance the performance of deep learning algorithms [16], the issues encountered when utilizing DL models for picture categorization are as follows.

a. DL normally requires a huge amount of dataset for training and this is lacked
b. Imbalanced dataset
c. Interpretability of dataset
d. Uncertainty scaling
e. Catastrophic forgetting
f. Model compression
g. Overfitting
h. Vanishing gradient problem
i. Exploding gradient problem
j. Underspecification

To overcome these issues, this work developed an enhanced activation function that includes dropout layers and batch normalization layers between the fully connected and output layers of CNN to raise the convergence rate. This overcomes the problem of overfitting caused by increasing the number of repetitions by adding the Adaptive Moment Estimation (Adam) optimizer to the gradient descent technique, which reduces the cumulative error and increases training speed. Internal covariate shift is also addressed by batch normalization. It is possible to employ a faster learning rate, which has a regularizing impact. As a result, an improved CNN algorithm (ICNN-BNDA) is suggested, which is based on dropout, batch normalization, and the Adam optimizer.

3 Materials and Methods

3.1 Dataset

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton got the CIFAR-10. There are 60,000 32×32 color photos in the collection. It is divided into ten classes, each having 6000 photos. The number of training photos is 50,000, whereas the number of testing images is 10,000. The dataset is split into five training batches, each of which contains precisely 1000 photos from each class. The remaining photos are arranged in a random order in the training batches. The test and validation datasets are encompassed in Table 1, and the suggested approach employed in this study is depicted in Fig. 1.

Table 1. Training and test datasets for CIFAR-10

Size of training data	Size of test data	Number of image category
50, 000	10, 000	10

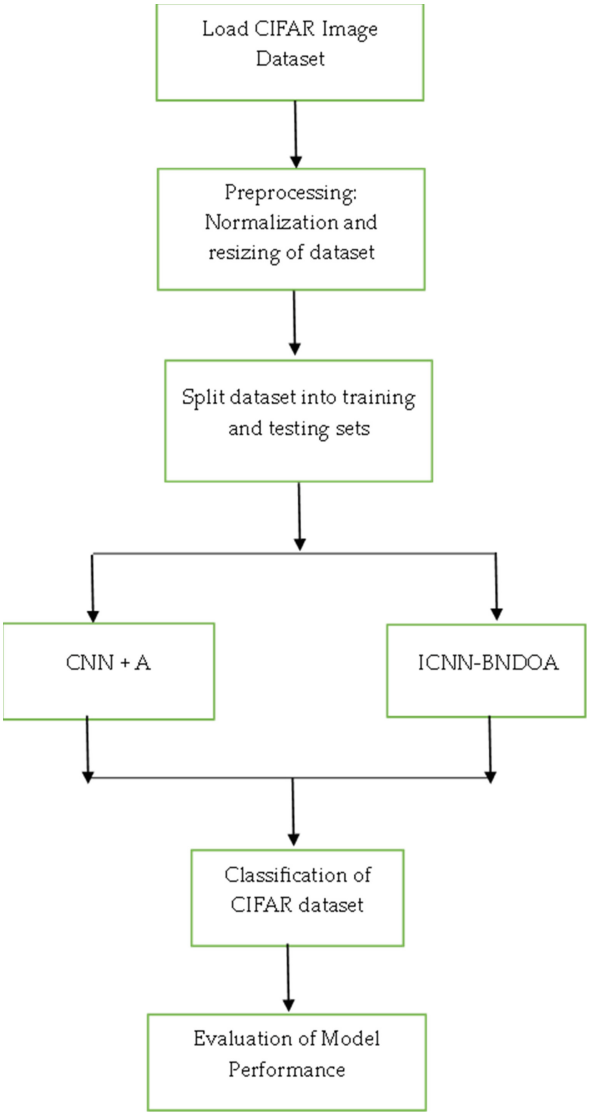

Fig. 1. Schematic illustration for the proposed system

3.2 CNN

Convolutional layers (CL), pooling layers (PL), and fully-connected (FC) layers are the 3 main layers made up by the CNN algorithm. In the algorithm, architecture will be constructed when the layers are layered. The DO layer and the AF are 2 extra key factors in addition to these 3 layers. The architecture is further diverged into 2 components, as shown in Fig. 2.

a. A convolution tool: This is a procedure that is identified as Feature Extraction (FE), which discerns and recognizes the different features of an image for analysis.
b. A fully connected layer: This layer employs the output of the convolution execution to forecast the image's class utilizing the characteristics extracted in earlier stages.

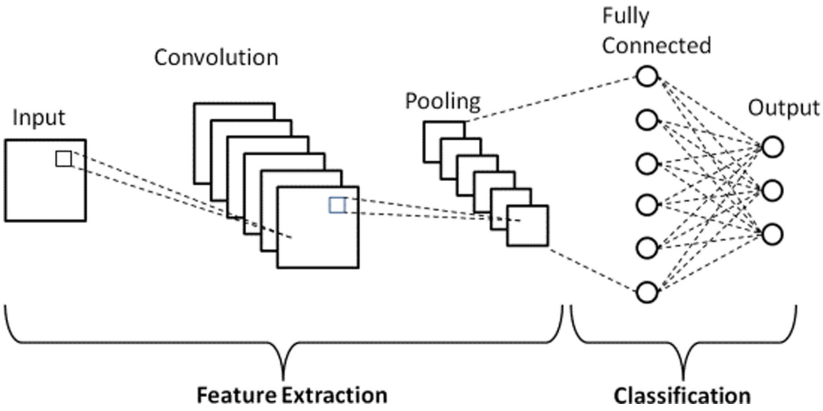

Fig. 2. An example of a CNN Architecture

3.3 Adam

Adaptive Moment Estimation (Adam) is another frequently utilized optimization approach or learning algorithm. Adam [17] exemplifies the most recent deep learning optimization advances. The Hessian matrix, which uses a second-order derivative, is used to express this. Adam is a deep neural network training approach that was created exclusively for Adam. Adam has two advantages: it is more memory efficient and requires fewer processing resources. Adam works by calculating adaptive LR for each parameter in the model. It combines the advantages of both Momentum and RMSprop. It uses squared gradients to scale the learning rate with RMSprop, and it's comparable to momentum because it uses the gradient's moving average. Adam's equation is given in E1. The equation is a mathematical expression of how Adam optimizer is used in image classification task.

$$w_{ij^t} = w_{ij^{t-1}} - \frac{\eta}{\sqrt{E\left[\delta^2\right]^t + \in}} * \widehat{E\left[\delta^2\right]^t} \tag{1}$$

where w is the model weights, eta (look like the letter n) and is the step size which can depend on iteration.

3.4 Activation Function Based on LeakyReLU

The following are the characteristics of the ReLU function f(x) = max (0, x) (x (0, +): (1) Gradient that is unsaturated: Ix > 0 is the formula for the gradient. As a result, the reverse propagation process' gradient dispersion problem is solved, and the parameters in the first layer of the neural network may be updated quickly. (2) Simplicity of computation: The thresholds are determined by the ReLU function: if x 0, then f(x) = 0; if x > 0, then f(x) = x. Regrettably, ReLU units might be brittle and "die" during training [17]. The value generated by a "dead" ReLU is always the same. The "dead" problem of ReLU was solved with the Leaky ReLUs AF [18], which has a modest positive gradient for negative inputs. The Leaky ReLU AF has the formula f = max (0.01x, x), which offers all of the benefits of the ReLU AF but without the "dead" issue. Due to LeakyReLU advantages in solving the gradient saturation problem and improving convergence speed, this AF in this study.

3.5 The Proposed Approach

This research created a seven-layer CNN model that includes an input layer with BN and DO layers, two hidden layers with BN and DO layers composed of convolution and pool layers, a flatten layer with DO layer (0.3), a fully connect-ed layer, two dense layers with BN and DO layers, and an output layer with sigmoid activation. The layers' activation functions are all rectified linear unit (LeakyReLU) functions, with the exception of the output. Conv2D is a function that performs 2D convolution operations. The ICNN founded on Batch normalization, Dropout and the Adam Optimizer (ICNN-BNDOA) is created on the foundation of the CNN architecture, the LeakyReLU AF, and the overfitting avoidance approach that is based on batch normalization and the Adam. A 3x3 matrix is used in the pooling procedure to guarantee that the image input and output after FE are of the same size. In each stage, the LeakyReLU activation function is employed to activate the neuron.

4 Implementation and Evaluation

4.1 Proposed Execution Environment

The seven-layers CNN learning method is developed using the python programming language on a 64-bit Windows 10 system. To evaluate the performance of the established model, Tensorflow is deployed. The performance of the CNN algorithm based on batch normalization and Adam optimizer is tested, assessed, and compared to that of the traditional CNN method at various Adam learning rates. This research compares CNN with and without BN and DO between layers.

4.2 Implementation Evaluation Results

BN and DO layers were supplied to evaluate the performance of the proposed model and determine the effects of the various added layers, according to the implementation setup stated in the preceding section. With the addition of BN and DO to the CIFAR-10 dataset, the model's identification rate rises. The following are the specifics of the two algorithms utilized in the implementation:

1. Conventional CNN is the first algorithm. This approach achieves input signal processing by using a SoftMax function as the activation function and a convolution and pooling layer.
2. Improved CNN algorithm. A convolutional neural network with batch normalization and a dropout of 0 is used in this approach. 2. The sigmoid function was utilized in the last dense layer as the AF. The addition of BN and DO layers, as well as an Adam optimizer with a learning rate of 0.0001 to maximize the cross entropy, enhances model performance.

4.3 Comparative Analysis of the Two Algorithms

The proposed algorithms run using 50 epochs. The two algorithms are compared using metrics like training loss, validation loss, testing loss, training accuracy, testing accuracy, and validation accuracy. Figures 3, 4, 5 and 6 show the model accuracies and losses for the two algorithms.

Figure 3 shows the ICNN-BNDOA training and testing accuracy of 0.6904 and 0.6861 respectively. Figure 4 shows the ICNN-BNDOA training and testing loss of 0.8910 and 0.9136 respectively. Figure 5 show the CNN-A training and testing accuracy of 0.4954 and 0.6739 respectively with their training and testing loss of 1.4170 and 0.9636 respectively as shown in Fig. 6.

Table 2 shows the parameters set and used for the execution (training, validation, and testing) of the proposed model. Table 3 shows the comparative analysis results for the two algorithms implemented and it was deduced that the introduction of BN and DO in the CNN layers produced an improved training and test accuracies with lower value for their losses.

Table 2. Parameter set for the model execution

Parameters	Values
Epochs	50
Batch size	32
Validation split	20
Training split	80
Optimizer learning rate	0.0001
Loss	Categorical cross-entropy

Table 3. Comparative analysis of the two algorithms

Algorithm	Training loss	Training accuracy	Validation loss	Validation accuracy	Test loss	Test accuracy
Proposed ICNN-BNDA	0.8910	0.6904	0.9137	0.6861	0.9136	0.6861
CNN-A	1.4170	0.4954	1.3374	0.5337	0.9636	0.6739

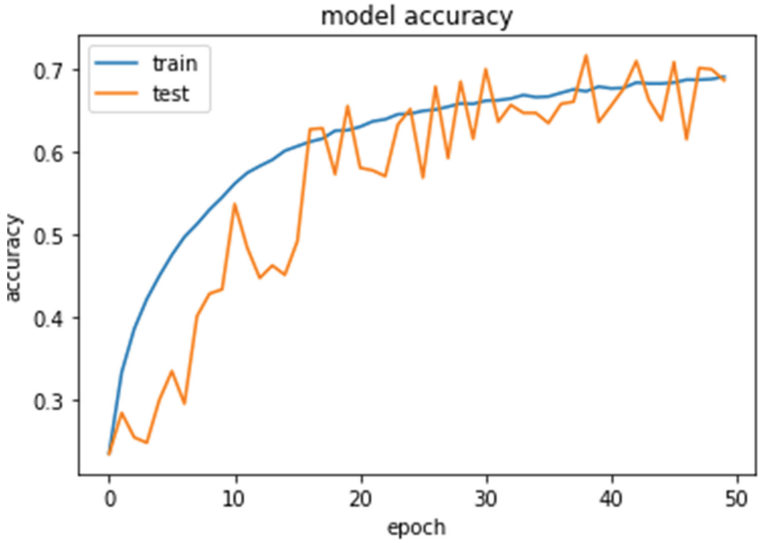

Fig. 3. ICNN-BNDOA training and testing accuracies

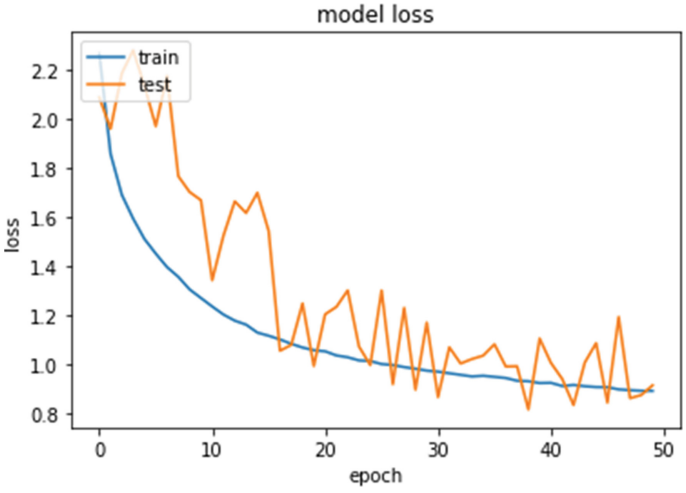

Fig. 4. ICNN-BNDOA training and testing losses

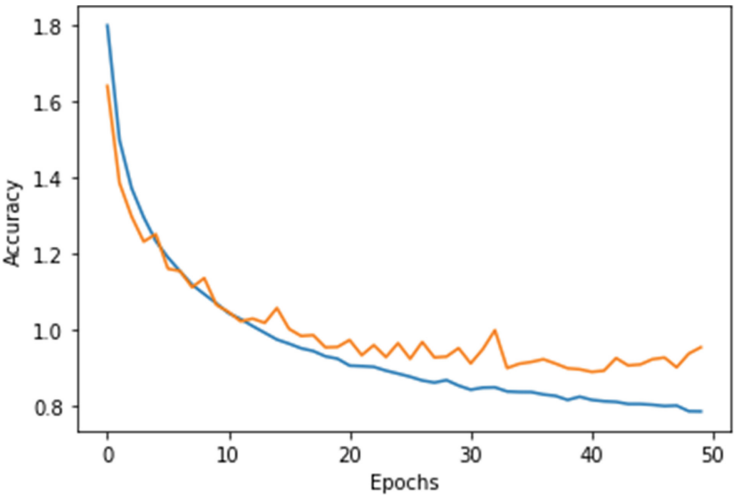

Fig. 5. CNN-A training and testing accuracies

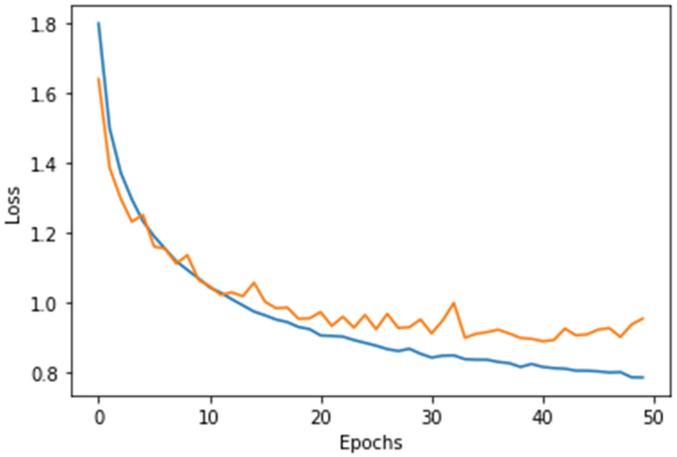

Fig. 6. CNN-A training and testing losses

5 Conclusion

This study looks at the architecture of the CNN, the overfitting issues in DL, and combines the dropout layer, batch normalization, and Adam optimizer with the CNN to increase FE accuracy while minimizing time costs. Tensorflow is used to parallelize an ICNN algorithm based on BN, DO, and Adam optimizers. Due to the fact that the Adam optimizer does not utilize all of the training cases in each combination and may energetically alter the approximation of the first and second-order matrices of the gradient of the individual parameter according to the loss function, the ICNN-BNDOA works quicker.

Additionally, by enhancing the AF, the technique evades the problem of the neuron node output being 0, enhances recognition accuracy, and shortens the processing time.

Since the current test data is small, future research will necessitate employing more test datasets to increase the algorithm's classification accuracy performance. Furthermore, limiting the number of neurons in the CNN can reduce calculations during the training, validation, and testing stages; nevertheless, the dropout layer technique employed during the learning stage upsurges processing time. At the start of the study, we concentrated on enhancing detection accuracy. The amount of time each stage takes necessitates additional examination.

References

1. Vieira, S., Pinaya, W.H., Mechelli, A.: Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: methods and applications. Neurosci. Biobehav. Rev. **74**, 58–75 (2017)
2. Li, S., Dou, Y., Niu, X., Lv, Q., Wang, Q.: A fast and memory saved GPU acceleration algorithm of convolutional neural networks for target detection. Neurocomputing **230**, 48–59 (2017)
3. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR, June 2015
4. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
5. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
8. Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289
9. Costarelli, D., Vinti, G.: Pointwise and uniform approximation by multivariate neural network operators of the max-product type. Neural Netw. **81**, 81–90 (2016)
10. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In Artificial Intelligence and Statistics, pp. 562–570. PMLR, February 2015
11. Liang, S., Khoo, Y., Yang, H.: Drop-activation: implicit parameter reduction and harmonious regularization. Commun. Appl. Math. Comput. **3**(2), 293–311 (2021)
12. Yamada, Y., Iwamura, M., Kise, K.: Shakedrop regularization (2018)
13. Zhang, K., Sun, M., Han, T.X., Yuan, X., Guo, L., Liu, T.: Residual networks of residual networks: multilevel residual networks. IEEE Trans. Circuits Syst. Video Technol. **28**(6), 1303–1314 (2017)
14. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3367–3375 (2015)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

16. Zhang, Z.: Improved Adam optimizer for deep neural networks. In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), pp. 1–2. IEEE, June 2018
17. Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., Yan, S.: Deep learning with s-shaped rectified linear activation units. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, No. 1, February 2016
18. Awotunde, J.B., Ogundokun, R.O., Ayo, F.E., Matiluko, O.E.: Speech segregation in background noise based on deep learning. IEEE Access **8**, 169568–169575 (2020). 3024077
19. Odusami, M., Maskeliunas, R., Damaševičius, R., Misra, S.: Comparable study of pre-trained model on Alzheimer disease classification. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications – ICCSA 2021. LNCS, vol. 12953, pp. 63–74. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86976-2_5