ISSN: 0974-8571 Vol.13 No. 2 December, 2021

International Journal of Computational Intelligence in Control

Convolutional Neural Network with Randomized Pooling to Reduce Computational Complexity

Hafiz Muhammad Imran[1], Dr.Khalid Mahmood Aamir[1], Dr.Muhammad Ilyas[1]

[1]Department of Computer Science and IT, University of Sargodha, Sargodha 40162, Pakistan.

imran.wahab@uos.edu.pk, khalid.aamir@uos.edu.pk, m.ilyas@uos.edu.pk

***Abstract*-Over the past decade, the popularity of convolutional neural networks (CNN), a multilayer network used in machine learning applications, has grown rapidly. Due to considerable research efforts, the accuracy of such networks has been greatly improved. However, the computational, temporal, and spatial complexity of CNNs remains a major challenge to overcome. In this work, the temporal complexity of classical CNN is improved by randomization. Randomization is the use of random elements as logical components in algorithms. The traditional CNN model pooling layer uses averages to reduce the size of feature maps. The average pooling layer (AvgPool) in CNN is very computationally intensive and reduces accuracy. We change the AvgPool layer to a randomization pooling layer (RandPool). The RandPool technique reduces the number of operations while increasing the accuracy. The proposed randomization algorithm improves the accuracy while reducing the computational cost. We compared our proposed CNN with the RandPool model and the traditional CNN model. Simulation results demonstrate 96.95% accuracy and 8.85% reduction in training time for our proposed RandPool model.**

Index Terms - Computational complexity, Convolutional neural network, Pooling, Randomization, Training time.

I.INTRODUCTION

To meet increasing demands, deep learning (DL) researchers have developed a variety of learnable strategies and algorithms. Machine learning emerged from scientists' observation of human learning activities and their transfer of the learning process to machines. The self-learning feature of machine learning, which requires less effort, is its main characteristic. Careful feature extraction is a prerequisite for traditional machine learning [1]. Since DL automatically extracts features from raw data that are critical to solving a particular problem, no expert is required for feature extraction. At different levels of abstraction, the different layers of DL process and extract different aspects of the dataset.

A neural network (NN) with many layers is known as a deep neural network (DNN). Deep neural networks have made substantial advancements in processing images and videos [2], [3]. The time needed for these networks to analyses data has greatly increased. Low latency is crucial for real world applications since network prediction algorithms take a lot of time and resources. A multilayer neural network forms the well-known deep learning model CNN [4]. The CNN layers execute two primary mathematical operations. One is to apply mathematical operations between matrices to obtain the feature maps, which is called convolution. The second method is pooling, which minimizes the dimensions of the data by average, maximum or minimum pooling. Due to the numerous pooling processes, the efficiency of the network is reduced. It is important to keep the pooling processes to a minimum. A vector is softmax to create its probability distribution in the last layer, a fully connected layer [6].

To increase CNN's effectiveness, numerous tests have been run. Denil [7] that there are numerous repetitions in neural network parameterization. Guo [8] and Gupta [9] compress these neural network models without considerably compromising their performance. The networks may only operate with a particular level of precision, according to [10], [11], and this study offers a way to lessen the computational complexity. The majority of academics work to improve CNN accuracy while keeping training and testing times constant. By consuming less time and resources during training and testing, CNN accuracy can still be improved, although it is still challenging.

The pooling layer in the CNN model aims to minimize time and spatial complexity by reducing the dimension of the data. Randomization is used in the pooling layer to cut down on the overall amount of operations, speed up processing, and increase accuracy. The innovative aspect of this study is the reduction in operations, and the output of the RandPool layer contains the dataset's original values without any editing. The RandPool layer of CNN increases accuracy and decreases training time with randomization. As a result of this research, the time required to train and test the CNN model has been reduced while its accuracy has increased. We use specific data points instead of the arithmetic mean to achieve this. Main contributions of this study are stated below.

1. Randomization is used in this study to reduce the number of operations, speed up calculation, and increase CNN accuracy.
2. The literature search revealed that the research community has generally focused on CNN accuracy, holding training and testing times constant. We also considered training and testing time in this study.
3. Image classification problem is targeted in this study with our proposed RandPool for CNN.
4. Modified national institute of standards and technology (MNIST) dataset is used to analyze and verify the accuracy of our proposed RandPool model. The rest of the research is organized as follows: Section II discusses state-of-the-art literature review along with preliminaries for background understanding of the novice. Section III describes the details of the simulation setup and mathematical modelling of our proposed convolutional neural network with a randomized pooling. The results are presented in Section IV, and final considerations along with limitations and future work are discussed in Section V.

II. LITERATURE REVIEW

Before proceeding to the proper discussion of the study, we refer to provide some background information. For this reason, we'll first define some basic terms and concepts in following sub-section that will serve as the cornerstone of the main findings of our study.

A. PRELIMINARIES

1) Convolutional neural networks

The model of DL, used in many applications, is the convolutional neural network (CNN). CNN are well suited for natural language processing (NLP), language translation, signal processing, topic classification, sentiment analysis, and speech analysis [3]. This architecture can categorize input data based on contextual data. Figure 1 represents the classical architecture of convolutional neural network [3].

A convolutional neural network is made up of several layers, including the convolutional layer (CL), the pooling layer (PL), the activation function (AF), and the fully connected layer. The CL creates a feature map by applying the convolution process to the raw image received at its input. The second convolutional layer (CL) performs the convolution and creates feature maps using Stride 2 [5], as shown in Figure 1. The core component of the CNN is the convolutional layer, which has the task of characterizing the input. This layer contains several feature maps that extract regional features from different areas.

The pooling layer reduces the amount of data by averaging the values from the feature map or by selecting the highest or lowest values. As a result, the number of dimensions is decreased, and feature evaluation is enhanced. The pooling layer typically functions between two convolutional layers. Depending on the kernel and step size, the feature map's dimension changes. The convolution layer and the pooling layer capture high-level input information, whereas the pooling layer utilizes average, minimum, or maximum operations.

High weighting values are passed from the activation function to the following layer, which normalizes the values of the feature map. The Rectified Linear Unit (ReLU), the most popular activation function, is one of many activation functions that are employed [3]. It has been determined after extensive research that the rectified linear unit (ReLU) is better than the former when sigmoid activation functions are used in existing machine learning techniques. The sigmoidal linearity with saturation ratio given in Equation 1 is slower than the unsaturated linearity. Equation 2 [12] illustrates a nonlinearity similar to the ReLU.

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \tag{2}$$

The output layer receives a categorized result from a fully linked layer [3]. The number of outputs is based on the number of classes underlying the dataset or problem. Softmax or normalized exponential function is used to transform the vector into a probability distribution using Equation 3 [13].

$$\sigma(\underset{Z}{\rightarrow})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{3}$$

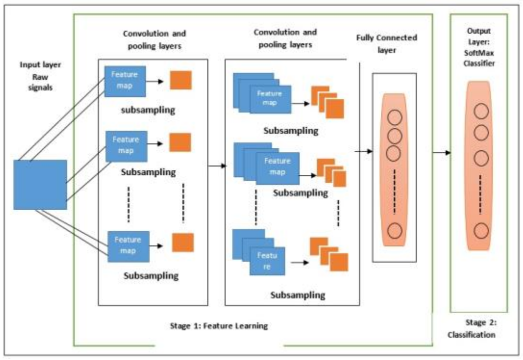

Figure 1: The classical architecture of convolutional neural network

2) Pooling

In a nonlinear process known as pooling, the results from a given location can be combined into a single value. The accuracy and sensitivity of feature translation for smaller input data are increased by computing this single value from the statistics of future outputs [14]. The pooling layer continuously reduces the dimensionality of the input data while minimizing memory usage to preserve variables and improve statistical performance [15]. The pooling process provides a spatially modified representation and reduces the computational cost of the top layers by removing the incomplete connections between the convolutional layers. The most important pooling types are briefly described below to support the background knowledge and understanding of the novice.

Feature samples of the preceding layer are generated to produce a number of feature maps with a constrained resolution. Reducing the number of parameters or weights, which lowers computational cost and prevents overfitting, is one of the pooling layer's main objectives [16]. The optimal pooling procedure should only retrieve pertinent data, and should discard extraneous data. The pooling methodology used is actually crucial to resolving the computer vision problem because the main goal is to turn the aggregated visual characteristics.

- **Max pooling:** The convolutional layer's feature map's most noticeable feature is used by the layer during maxpooling. In a nutshell, it chooses the element with the highest value from the region that the filter in any feature map has selected, as shown in Figure 2. Max-pooling helps to extract the sharpest features from the image, which are the best lower-level representations of the image like edges, points, etc. [17].

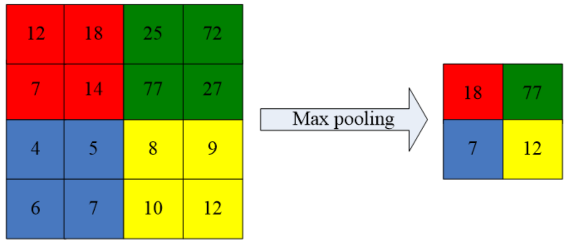

Figure 2: Max pooling process

- **Min pooling:** When using min pooling, the convolutional layer's feature map is used to operate with the feature that is least noticeable. In simpler terms, we can state that it chooses the element with the lowest value from the region that the filter in any feature map has selected, as shown in Figure 3. Min pooling helps to extract the features from the image that have lower sharp values or the edgeless features.
- **Average pooling:** Average pooling, where the values of a group of neighboring pixels are averaged, is a typical method used in convolutional neural networks (CNNs) to downscale feature maps. It can reduce the computational complexity of deep learning models, such as [18]. Average pooling can be more effective in maintaining the spatial structure of the input data, as shown by several studies comparing its effectiveness with other

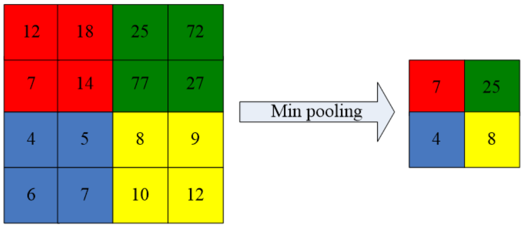

Figure 3: Min pooling process

pooling approaches such as max-pooling and stochastic pooling [19]. Deep learning models have been shown to perform better on a number of tasks, including object detection and picture segmentation when adaptive average pooling is used. This technique adjusts the pooling size as a function of the input size [20]. Average pooling process is shown in Figure 4.

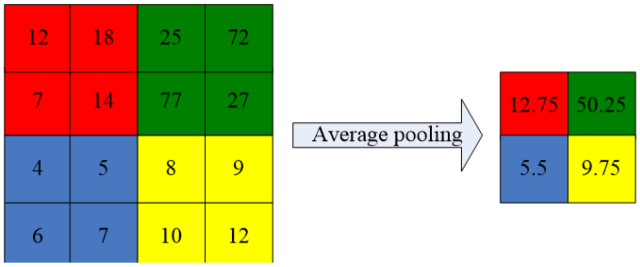

Figure 4: Average pooling process

Randomized pooling: It is a promising method for reducing the computational cost of deep learning models without compromising accuracy. According to studies, Convolutional neural networks (CNNs) can use it to perform better on large-scale image and video processing tasks [21], [22]. The computational complexity of deep learning models can be reduced by randomized pooling while maintaining accuracy. For example, in a recent work [23], authors proved that randomized pooling can increase CNN performance on image classification tasks, achieving equivalent or better performance than traditional pooling methods.

B. LITERATURE REVIEW

The literature review, which presents a thorough examination of the available research on the subject, is a crucial part of this study. We start by conducting a thorough literature analysis of the most recent and pertinent studies in the field in order to contextualize our research. Matthew used pooling to implement the notion of stochastic process. The probability was calculated for each pooling region, and the selected value was then forwarded to be activated. He argues that whereas stochastic pooling also employs non-maximum values, maximum pooling solely selects maximum values. Figure 5 depicts the stochastic pooling process, and Equation 4 [24] describes the stochastic operation.

$$P_i = \frac{a_i}{\sum_{K \in R_i} a_k} \quad (4)$$

The average pooling method is computationally expensive than the proposed randomized pooling, while the stochastic pooling process requires significantly more computational power than the average pooling [24]. The biologically inspired pooling known as "LP pooling" resembles complicated cells [25]. The processes are enumerated in Equation 5, where I and O stand for the input and output feature maps, respectively, and G for the Gaussian kernel. By using LP pooling, the weights of stronger characteristics are reduced while those of prominent features are enhanced. When P=1, the LP pooling behavior changes to average pooling, and when p = 1, the results resemble maximum pooling [26].

$$O = \sum\sum I(i,j)^p \times G(i,j))^{1/p} \quad (5)$$

The visual reordering technique in [27] involves spatial pooling, where nearby feature detectors are merged to obtain task-related information and exclude unnecessary information. Clutter pooling is used to produce denser, more noise resistant, and trans-form invariant representations. The study at [28] provides a comparison of different pooling methods. The efficiency of RunPool, Max, and Average pooling is investigated in this study. When determining the classification performance of the model, many pooling factors are taken into account. RunPool pooling and Average pooling are evaluated by setting Max pooling hyper parameters, tweaking and environmental performance. Because the convolutional layer rather than the pooling layer is utilized in the first experiment, the classification performance is below average. Consequently, a CNN without a pooling layer also yields below-average results. Information is aggregated within a domain using handcrafted pooling techniques [29], but these operations do not reduce training error. This vulnerability prompts the use of a recently discovered pooling technique called LEAP (LEAring Pooling). In this method, a joint linear combination of neuron methods is used to perform the learning process for each feature map. When all weights are equal, this method gives the same results as average pooling.

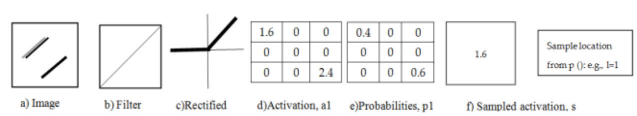

Figure 5: Stochastic pooling process

In this study, a simpler convolution, combined with LEAP, is used instead of the CNN convolutional layer. Proposed model achieves the best classification performance with ImageNet dataset [30], CIFAR100 dataset [31] and CIFAR10 dataset [32].

Convolutional neural networks have an essential layer, the region of interest (RoI) pooling layer, which is primarily used for object segmentation [33] and recognition [34]. Individual bounding box processing has been moved to the back of the network architecture to allow the RoI pooling layer to function. A deep network processes an input image, producing CNN feature maps with fewer spatial dimensions than the original image. The input to the RoI pooling layer is made up of the feature map of the entire image and the coordinates for each RoI. The features connected to a specific object can be roughly located using the RoI coordinates. The features thus obtained have different spatial sizes, since each RoI can have its own dimension.

Another technique of pooling is edge-aware pyramid pooling. An edge-aware pooling module was suggested by Xu et al., [35] to get more information about the edge structure, and the edge-aware feature map is integrated into the pedestrian motion detection task. Finding boundaries between objects and edges in actual photos is the goal of the edge detection task. An essential first step in accomplishing the computer vision tasks of segmentation and target detection is edge detection. Authors of [35] used supplemental data along with edge-related data to help predict pedestrian movement.

DeepPano, a deep representation for categorization and retrieval of 3-D shapes, was presented by the authors of [36]. Representations are learned and extrapolated from 3-D shapes to create panoramic perspectives. DeepPano significantly outperforms previous approaches on classification and retrieval tasks. They have also empirically confirmed rotational invariance of the representation.

Attention weights are generated by additional dense layers in the preceding pooling techniques. This leads to a larger model that needs to be trained. Bhattacharjee et. al., [37] reduce the size and complexity of the model by pooling using Genetic Algorithm (GA). It is a method for solving difficult optimization problems that mimics the biological evolutionary process. GA the main operations of the algorithm are mutation, crossover, and selection. Members of one generation are selected or chosen as parents, and crossover and mutation produce children who can be candidates for the following generation. Over several generations, a population evolves toward an ideal outcome.

Abhinav Kumar et. al., in [38], proposed a randomized weight matrix from uniform distribution and claimed that there is no need to train the net if we use a proposed randomized weight matrix. The reduction in computational complexity is made by generalizing the randomized neural network to the non-Ecludian domain by using a "graph convolution operation" that aggregates neighbourhood structure. Randomized General Convolutional Neural Network is given in figure 6 [39].

The VGG-Inspired stochastic pooling neural network (VISPNN) model was proposed by the authors of [40] and is based on three components: A 20-way data augmentation with salt-and-pepper noise, Gaussian noise, Poisson noise and speckle noise is a better version of the stochastic pooling, average pooling and VGG-inspired neural network. Additionally, the ablation investigations propose two networks (Net-I and Net-II). By switching out stochastic pooling for regular max pooling, Net-I is based on VISPNN. Data augmentation in 20 ways is eliminated by Net-II

In [41] authors have stated that regularization of the network which prevents the structure from overfitting, is a crucial component of training. This study examines a number of regularization techniques that have emerged in recent years, demonstrating notable advancements for several CNN models.

The works are divided into three primary categories: the first is referred to as "data augmentation," and all of the approaches used in this category are aimed at altering the raw data. The second is referred to as "internal changes," and it describes methods for changing feature maps produced by kernels or neural networks. The final one, "label," deals with changing the labels of an input.

Advantages and disadvantages of both conventional and cutting-edge pooling methods for readers and offers a critical comprehension of each [42]. Additionally, a qualitative analysis and evaluation of the effectiveness of pooling approaches on various datasets is conducted in this study. A thorough knowledge of the significance of CNNs and pooling methods in computer vision problems is anticipated as a result of this study.

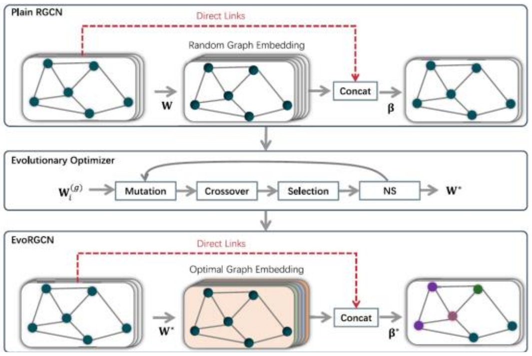

Figure 6: Evolution-driven randomized graph convolutional networks (EvoRGCN) [39]

III. MATERIALS AND METHODS

This study used handwritten grayscale images of size 28×28 from the MINIST dataset [1]. The input of our model is 28×28 images and 256 kernels of size 9×9. The convolution operations are applied with stride 1, i.e. the kernel window moves to the next value without skipping any value. The result of this operation is 256 feature maps that are entered into the next layer of the model. Each feature map is 20×20, and serves as input to the following layer, which has a 9 × 9 kernel and a stride of 2. The second convolutional layer generates 6×6 images. The size of the images (feature map)
M is given in equation 6 [28].

$$M = \frac{i_s - k_s + 1}{n_s} \tag{6}$$

The input image size is denoted by is, the kernel size by ks, and the number of stride by ns. The size of featuremap is reduced through a pooling operation in the poolinglayer In contrasted to the classical operations in pooling layer this study used pooling window and randomized pooling (RandPool). The specifications of machine used in simulation are shown in Table 1. The definition of pooling window is given in next subsection.

Table 1:Machine specification

Machine Type	Processor	RAM
Core i3	1.70 GHz	8 GB

A. MATHEMATICAL MODELLING

In convolutional neural networks (CNNs), the goal of the pooling layer is to reduce the amount of data in order to improve the efficiency of a network. Pooling methods require extensive operations to consolidate the data, which help overcome the time complexity of the model. The output of these methods changes the original data values, which affects the results. In this study, randomized operations were proposed to minimize the complexity. Randomly selecting current values without any computation and passing them to the next layer improves the efficiency of the existing model in two ways: it reduces training time and increases accuracy because the original values are not manipulated.

The mathematical modelling of our proposed convolutional neural network with a randomized pooling model is escribed in the following. The pooling window is the k × k matrix of original values given in equation 7, where we select the random value. These fixed values pass to the next layer.

$$w = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ \vdots & \vdots & \cdots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{bmatrix}$$
(7)

Formation of w matrix to linear indices of Equation 7 are
shown in Equation 8 for m = 1...k.

$p = k(l-1) + m$ for l, m=1…, k
(8)

Therefore, x_{ij} maps to yp and it is obvious that p = 1...k_2.
Formula for calculation of average pool is shown in Equation 9 and expected average pool is given in Equation 10.

$$x_{avg\_pooling} = \frac{1}{k^2} \sum_{p=1}^{k^2} y_p$$
(9)

$$E\left[x_{avg\_pooling}\right] = E\left[\frac{1}{k^2}\sum_{p=1}^{k^2} y_p\right]$$
(10)

The expected average pool with assumption that all y_p for p = 1 . . . k_2 are independent and identically distributed
(IID), is shown in in Equations 11 and 12.

$$E\left[x_{avg\_pooled}\right] = \frac{1}{k^2} \sum_{p=1}^{k^2} E\left[y_p\right] \tag{11}$$

$$E\left[x_{avg\_pooled}\right] = \frac{1}{k^2} \cdot k^2\, E\left[y_p\right] = E\left[y_p\right] \tag{12}$$

We have randomly selected any of the $y'_{ps'}$ from defined randomized pooling in Equations 11 and 12

Assume that y_p is linearly indexed and randomly selected element of w. Consequently, our proposed method (Rand-Pool) and (AvgPool) are the same as shown in Equation 13, due to the assumption of IID.

$$\mathrm{E}[x_{avg\_pooled}] = E[y_p] \tag{13}$$

Figure 7 shows how the data of the original image, the average image and the random image were distributed. This graph shows that the randomly selected values are close to the values of the original image, which increases the accuracy of the model. CNN uses the average values of the dataset that are not from the original dataset, which decreases the accuracy.

IV. RESULTS AND DISCUSSION

The computational complexity of the pooling layer is decreased due to the ran-domization pooling (RandPool), which randomly chooses a value from the feature map. In Equation 14 formula for calculation of typical CNN model is shown.

AvgPool = 2 × (n2 × 256 × no.of images)
(14)

The size of images is n2 and number of filters are 256 of size 9×9.The training per-formed on 8000 images, and 2000 images used to test model. Dimension of each image is 28 × 28.The MNIST dataset [1] is used for testing and training which have hand written grayscale images. The amount of key operations perform in testing is given in Equation 15.

$$2 \times (28)2 \times 256 \times 8 \times 103 = 3.21e^{+9} \tag{15}$$

Comparative analysis of average pool and our proposed RandPool technique are shown in Table 2. Simulations results are recorded for 10 Epochs for both pooling techniques (AvgPool and RandPool). Results depict gradual rise in training time for both pooling techniques with respect to number of epochs whereas, testing time is variable and shown against each. Precision for epoch # 01 is 87.30% (AvgPool) and 87.25% (RandPool) and it is improved till 96.85% (AvgPool) and 96.95% (RandPool) at epoch # 10. Finally, our proposed RandPool help improve the precision up to 96.95%.
The results, given in Table 2 and figure 8, shows that RandPool reduce n2 × m operations as compare to classical CNN model. The classical algorithm of CNN and proposed RandPool both were simulated on same machinimas shown in Table 1 to compare the efficiency. The operation calculation for classic CNN model are shown in Equation 16. Let p × p be the size of pooling window and the size of an image A is n × n such that p ≪ n.

$$\text{Total computations} = \left(\frac{n}{P}\right)^2 . P^2 = n^2 \tag{16}$$

AvgPool take the arithmetic mean for each pooling window, while RandPool pick value from feature map randomly without any calculation. RandPool and AvgPool were applied to the MNIST dataset containing 10,000 handwritten grayscale images of the numbers 0 to 9. Training and testing times and accuracy are shown for each epoch in Table 2. The training and testing times decrease as the number of epochs increases, while the accuracy of RandPool increases. In contrast, the training and testing times of AvgPool become longer and the accuracy decreases as the number of epochs increases. The accuracy is determined using the formula in equation 17.

$$\text{Precision} = \frac{Ta}{Ti} \times 100 \tag{17}$$

Where T_i is input images, and the sum of all correct answers is denoted by Ta. The graphical comparative representation of AvgPool and RandPool is given in Figures 8.

Table 2: Comparison of Average and RandPool (Proposed)

Epoch No.	Method of Pooling	Time of Training	Time of Testing	Precision
1	AvgePool	73.02	7.90	87.30%
	RandPool	65.30	6.25	87.25%
2	AvgePool	230.45	10.97	92.35%
	RandPool	131.31	5.84	92.50%
3	AvgePool	313.87	10.70	94.10%
	RandPool	197.22	6.03	94.85%
4	AvgePool	433.50	11.13	95.05%
	RandPoo	265.71	5.84	95.40%

	l			
5	AvgePool	481.82	11.54	96.00%
	RandPool	330.95	5.81	96.50%
6	AvgePool	488.16	8.40	96.35%
	RandPool	393.93	5.87	96.35%
7	AvgePool	511.00	8.50	96.30%
	RandPool	461.90	5.99	96.40%
8	AvgePool	585.83	7.97	96.30%
	RandPool	532.65	5.88	96.35%
9	AvgePool	673.22	7.97	96.60%
	RandPool	603.49	5.68	96.85%
10	AvgePool	743.55	7.95	96.85%
	RandPool	666.44	5.87	96.95%

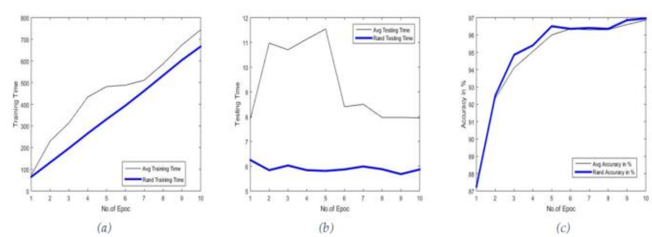

Figure 8: Comparison of Average and RandPool (Proposed) (a) Training time, (b) Testing time and (c) Accuracy

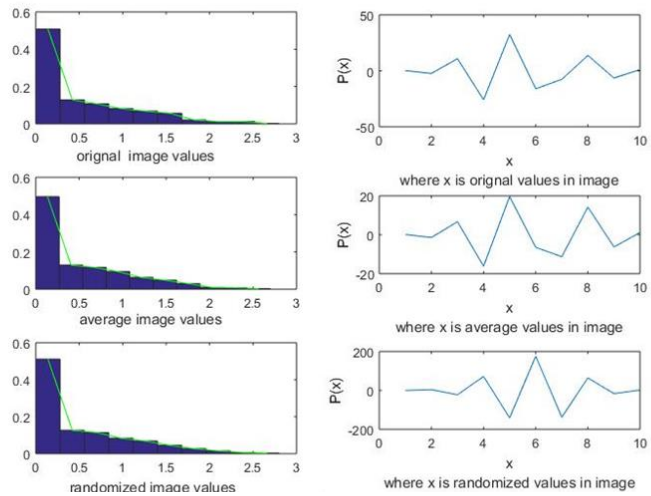

Figure9: Comparative analysis of average pooling (AvgPool) and proposed randomized pooling (RandPool)

V. CONCLUSION AND FUTURE WORK

Complexity in terms of time and space is the fundamental problem of algorithms in computer science. In the current study, we proposed a randomized pooling method (Rand-
Pool) to increase the accuracy of CNN while preserving the original values of the images. Randomized pooling can be used in all CNN variants to improve the accuracy and reduce the computational complexity. This technique in CNN reduces the spatial dimension of the input data. It helps extract the most needed features while reducing computational complexity. However, RandPool can affect the interpretability of the CNN model due to coarse feature representation, translation invariance, and importance assignment. Simulation results proved the superiority of our proposed RandPool technique, and we achieved 96.95% accuracy and 8.85% reduction in training time in 10 # epoch. Since our proposed RandPool technique helps to reduce the test time, it can be effectively used in real-time image processing systems. The robustness of CNN to noise, model accuracy on unseen data, and non-deterministic behavior of the model during training are future research directions.

References

[1] LeCun, Y. & C. CortesYann. (2010) "MNIST handwritten digit database," Microsoft Research, Redmond, Google Labs, New York, [Online]. Avail- able: http://yann.lecun.com/exdb/mnist/. [Accessed on September 18, 2022].

[2] Sairamya, N. J., Susmitha, L., George, S. T., & Subathra, M. S. P. (2019). Hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features. In Intelligent Data Analysis

for Biomedical Applications (pp. 253-273). Academic Press.

[3] Si, L., Xiong, X., Wang, Z., & Tan, C. (2020). A deep convolutional neural network model for intelligent discrimination between coal and rocks in coal mining face. mathematical Problems in Engineering, 2020, 1-12.

[4] Aslam, S., Herodotou, H., Mohsin, S. M., Javaid, N., Ashraf, N., & Aslam, S. (2021). A survey on deep learning methods for power loadand renewable energy forecasting in smart microgrids. Renewable and Sustainable Energy Reviews, 144, 110992.

[5] Riad, R., Teboul, O., Grangier, D., & Zeghidour, N. (2022). Learning strides in convolutional neural networks. arXiv preprint arXiv:2202.01653.

[6] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Un- derstanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

[7] Denil, M., Shakibi, B., Dinh, L., Ranzato, M. A., & De Freitas, N. (2013). Predicting parameters in deep learning. Advances in neural information processing systems, 26.

[8] Guo, Y., Yao, A., & Chen, Y. (2016). Dynamic network surgery for efficient dnns. Advances in neural information processing systems, 29.

[9] Gupta, S., Agrawal, A., Gopalakrishnan, K., & Narayanan, P. (2015, June). Deep learning with limited numerical precision. In International conference on machine learning (pp. 1737-1746). PMLR.

[10] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28.

[11] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In Neural networks for perception (pp. 65-93). Academic Press.

[12] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Interna- tional conference on machine learning (pp. 448-456). pmlr.

[13] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Acti- vation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378.

[14] Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. Communications of the ACM, 64(7), 58-65.

[15] Yildirim, O., Baloglu, U. B., Tan, R. S., Ciaccio, E. J., & Acharya,

U. R. (2019). A new approach for arrhythmia classification using deep coded features and LSTM networks. Computer methods and programs in biomedicine, 176, 121-133.

[16] Wu, H., & Gu, X. (2015). Towards dropout training for convolutional neural networks. Neural Networks, 71, 1-10.

[17] de Souza Brito, A., Vieira, M. B., De Andrade, M. L. S. C., Feitosa,

R. Q., & Giraldi, G. A. (2021). Combining max-pooling and wavelet pooling strategies for semantic image segmentation. Expert Systems with Applications, 183, 115403.

[18] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[19] Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[21] Boureau, Y. L., Bach, F., LeCun, Y., & Ponce, J. (2010, June). Learn- ing mid-level features for recognition. In 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 2559-2566). IEEE.

[22] Yang, M., Li, B., Fan, H., & Jiang, Y. (2015, September). Randomized spatial pooling in deep convolutional networks for scene recognition. In 2015 IEEE International Conference on Image Processing (ICIP) (pp.402-406). IEEE.

[23] Fuan, W., Hongkai, J., Haidong, S., Wenjing, D., & Shuaipeng, W. (2017). An adaptive deep convolutional neural network for rolling bearing fault diagnosis. Measurement Science and Technology, 28(9), 095005.

[24] Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557.

[25] Gholamalinezhad, H., & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485.

[26] Jie, H. J., & Wanda, P. (2020). RunPool: A Dynamic Pooling Layer for Convolution Neural Network. Int. J. Comput. Intell. Syst., 13(1), 66-76.

[27] Deng, L., Abdel-Hamid, O., & Yu, D. (2013, May). A deep convolutional neural network using heterogeneous pooling for trading acoustic invari- ance with phonetic confusion. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6669-6673). IEEE.

[28] Sun, M., Song, Z., Jiang, X., Pan, J., & Pang, Y. (2017). Learning pooling for convolutional neural network. Neurocomputing, 224, 96-104.

[29] Tesema, F. B., Wu, H., Chen, M., Lin, J., Zhu, W., & Huang, K. (2020). Hybrid channel based pedestrian detection. Neurocomputing, 389, 1-8.

[30] ImageNet dataset. https://www.image-net.org/download.php (accessed on March 09, 2023).

[31] CIFAR100 small images classification dataset. https://keras.io/api/datasets/cifar100/ (accessed on March 09, 2023).

[32] CIFAR10 small images classification dataset. https://keras.io/api/datasets/cifar10/ (accessed on March 09, 2023).

[33] He, Z., Yuan, Z., An, P., Zhao, J., & Du, B. (2021). MFB-LANN: A lightweight and updatable myocardial infarction diagnosis system based on convolutional neural networks and active learning. Computer Methods and Programs in Biomedicine, 210, 106379.

[34] Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detec- tion using a robotic vision system. Computer Networks, 168, 107036.

[35] Xu, L., Yan, S., Chen, X., & Wang, P. (2019). Motion recognition algorithm based on deep edge-aware pyramid pooling network in hu- man–computer interaction. IEEE Access, 7, 163806-163813.

[36] Huang, J., Yan, W., Li, T., Liu, S., & Li, G. (2020). Learning the global descriptor for 3-D object recognition based on multiple views decomposition. IEEE Transactions on Multimedia, 24, 188-201.

[37] Bhattacharjee, K., Pant, M., Zhang, Y. D., & Satapathy, S. C.

(2020). Multiple instance learning with genetic pooling for medical data analysis.Pattern Recognition Letters, 133, 247-255.

[38] Kumar, A., Singh, J. P., & Singh, A. K. (2022). Randomized Convolu- tional Neural Network Architecture for Eyewitness Tweet Identification During Disaster. Journal of Grid Computing, 20(3), 20.

[39] Zijia Zhang , Yaoming Cai , Student Member, IEEE, and Wenyin Gong , Member, IEEE. DECEMBER 2022.Evolution-Driven Random- ized Graph Convolutional Networks.IEEE TRANSACTIONS ON SYS- TEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 52, NO. 12,

[40] Wang, S. H., Khan, M. A., & Zhang, Y. D. (2022). VISPNN: VGG- inspired stochastic pooling neural network. Computers, materials & continua, 70(2), 3081.

[41] Santos, C. F. G. D., & Papa, J. P. (2022). Avoiding overfitting: A sur- vey on regularization methods for convolutional neural networks. ACM Computing Surveys (CSUR), 54(10s), 1-25.

[42] Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Kumar Dutta, A., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. Applied Sciences, 12(17), 8643.