

Research Article

Comparative Analysis of Recent Architecture of Convolutional Neural Network

Muhammad Asif Saleem ¹, Norhalina Senan ¹, Fazli Wahid ², Muhammad Aamir ^{1,3},
Ali Samad ^{1,4} and Mukhtaj Khan ²

¹Faculty of Computer Science, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 54000, Malaysia

²Department of Information Technology, The University of Haripur, Haripur 22621, Khyber Pakhtunkhwa, Pakistan

³School of Electronics, Computing, and Mathematics, University of Derby, Derby, UK

⁴Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

Correspondence should be addressed to Fazli Wahid; fazli.wahid@uoh.edu.pk and Mukhtaj Khan; mukhtaj.khan@awkum.edu.pk

Received 27 December 2021; Revised 19 February 2022; Accepted 1 March 2022; Published 31 March 2022

Academic Editor: Vijay Kumar

Copyright © 2022 Muhammad Asif Saleem et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Convolutional neural network (CNN) is one of the best neural networks for classification, segmentation, natural language processing (NLP), and video processing. The CNN consists of multiple layers or structural parameters. The architecture of CNN can be divided into three sections: convolution layers, pooling layers, and fully connected layers. The application of CNN became most demanding due to its ability to learn features from images automatically, involving massive amount of training data and high computational resources like GPUs. Due to the availability of the above-stated resources, multiple CNN architectures have been reported. This study focuses on the working of convolution, pooling, and the fully connected layers of CNN architecture, origin of architectures, limitation, benefits of reported architectures, and comparative analysis of contemporary architecture concerning the number of parameters, architectural depth, and significant contribution.

1. Introduction

Convolutional neural network (CNN) is a neural network that has outperformed computer vision problems [1]. CNNs are considered best for learning data from the image and have performed extraordinarily in image classification, segmentation, and detection. Nowadays, most image processing and computer vision-related problems apply CNN for a better solution [2]. The reason behind the better performance of CNN in the above said task is its ability to work on raw data without having prior knowledge [3]. CNN is a biologically inspired artificial neural network (ANN). In CNN, information travels unidirectionally as a feed-forward network. Its architecture is the same as the human brain's visual cortex, consisting of complex and straightforward cells based on several alternative layers [4]. The figure below shows the complete overview of CNN architecture.

CNN learns by limiting the change in weights according to the target during training using the backpropagation method. The human brain's response-based learning is akin to the optimization of an objective function using a back-propagation method. Deep CNN's multilayered, hierarchical structure allows it to extract low, mid, and high-level information. Lower and mid-level characteristics are combined to form high-level features (more abstract features). CNN's hierarchical feature extraction capabilities mimic the neocortex in the human brain's deep and layered learning process, which dynamically learns features from raw input. CNN's appeal stems mainly from its ability to extract hierarchical features [5]. Several researchers have contributed to performance improvements of CNN. According to the literature, improvements in CNN are made by optimizing the architectural parameters and weights [6]. Improvement in CNN's performance can also be achieved by increasing training data, transforming the training data, and adjusting

the parameters [7]. In this article, several CNN architectures will be discussed, along with their strengths and limitations.

This study will provide understanding of the essential components and theoretical and mathematical design principles of CNN. The rest of the paper is organized as shown in Figure 2. Section 1 develops a basic understanding of CNN, Section 2 provides knowledge about basic CNN components like convolution, pooling, and fully connected layers, Section 3 presents a mathematical representation of CNN, Section 4 shows a study on CNN architectures, including evolution, origin, and comparison of contemporary architectures, Section 5 presents the challenges of several CNN architectures, and finally conclusion and future work are discussed.

2. Review Protocol

The design of this article is based on a systematic research study. It is an appropriate and consistent procedure to record relevant points of interest in the appropriate study range for inspecting and analyzing all current studies identified. A review protocol for searching and selecting relevant research articles from logical research databases is developed in this research. Figure 2 represents an overview of the review protocol. In the proposed review protocol, publishers, selection criteria, and rejection criteria of research articles are considered. The complete review protocol is described in Figure 3.

2.1. Publisher. In this survey study, the article is selected from IEEE, ACM, Springer, and Elsevier. We also picked some articles having significance from Google Scholar and kept them in the category “Others.”

2.2. Selection Criteria. In this research, a standardized paradigm is prepared for the selection and rejection of articles. Several parameters like subject relevancy, year-wise range, and so on are kept in mind to select relevant research articles from digital libraries. A brief description of the selection and rejection criterion is described below.

2.2.1. Subject Relevancy. Article selected for this study must fit in research setting made as per criteria set in review protocol. It must be incorporating relevant responses, and it must be applied as per predefined standards. Remove the nonsignificant findings that are not in agreement with the predefined settings.

2.2.2. Year-Wise Range. For this study, the research articles of the last ten years (2011 to 2021) are considered. The research repository is set as it should not show any research paper more than ten years old.

2.2.3. Result Oriented. The research articles selected for this study must be result oriented. Before finalizing the article, consider a brief overview of the article, especially in the result section, and confirm that the selected paper has a

significant contribution in the relevant field. If the article is not fulfilling the criteria, dismiss it.

2.3. Rejection Criteria. This research is focused on the quality survey. Thus, all irrelevant research articles must be discarded. Below are the settings made for keeping only relevant articles in the research bank.

2.4. Repetition. It is challenging to incorporate all of the research articles collected through a review protocol. Thus, remove the research articles which are not distinguishable as per research settings. Only choose the latest one and remove the remaining.

2.5. Title-Based Rejection. A brief observation of the titles of the research papers will justify the article selected for this research. Although assessing the research article may require some experience, the result will be fruitful. Substantial certainty and experimentation are necessary to support the study proposition and extreme consequences. If the title of the research article does not correspond to the research settings made, dismiss it.

2.6. Abstract Based. It can be challenging to decide when choosing an article by observing only the title of the research. In such a case, a decision may be made by having a brief look at the abstract of the article. From the abstract of article, we can get information about the technique used, and its results can be collected. With these information, the authors can confirm that the paper is suitable or not. If the relevant technique is not applied, then reject the article.

3. CNN Architecture Overview

3.1. Convolution Layers. The convolution layer is an initial part of CNN architecture after the input layer consisting of a combination of convolution kernels (neuron) [8]. Each kernel (neuron) is associated with a small portion. This diminutive portion is called a receptive field. It operates by dividing the input image into smaller pieces of images (receptive fields) and convolving them with a specific set of weights. Operation of convolution layer in CNN can be expressed as follows.

At i th convolution, we can denote as:

Input: \mathbf{a}^{l-1} with size $(\mathbf{n}_H^{l-1}, \mathbf{n}_W^{l-1}, \mathbf{n}_C^{l-1})$, \mathbf{a}^0 being the image input.

- (i) Padding: $\mathbf{p}^{[l]}$, Stride: $\mathbf{S}^{[l]}$
- (ii) Several filters: \mathbf{n}_C^l where each $\mathbf{K}^{(n)}$ has dimensions: $\mathbf{f}^{(l)}, \mathbf{f}^{(l)}, \mathbf{n}_C^{[l-1]}$
- (iii) Bias of the \mathbf{n}^{th} convolution: $\mathbf{b}_n^{[l]}$
- (iv) Activation function: $\vartheta^{[l]}$
- (v) Output: $\mathbf{a}^{[l]}$ with size $(\mathbf{n}_H^l, \mathbf{n}_W^l, \mathbf{n}_C^l)$

We have

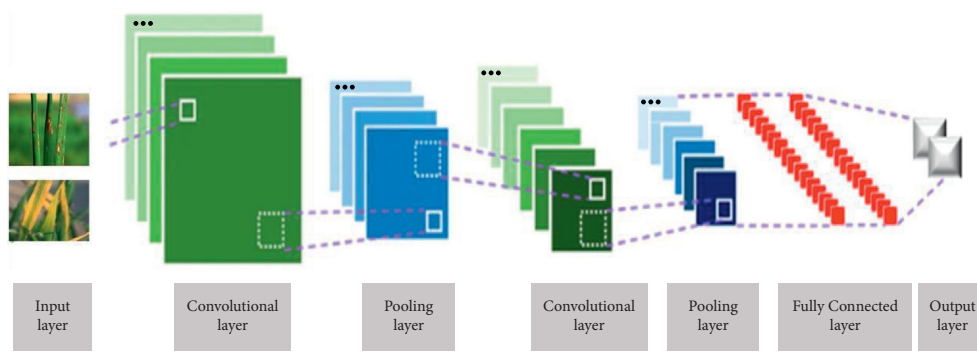


FIGURE 1: CNN structure.

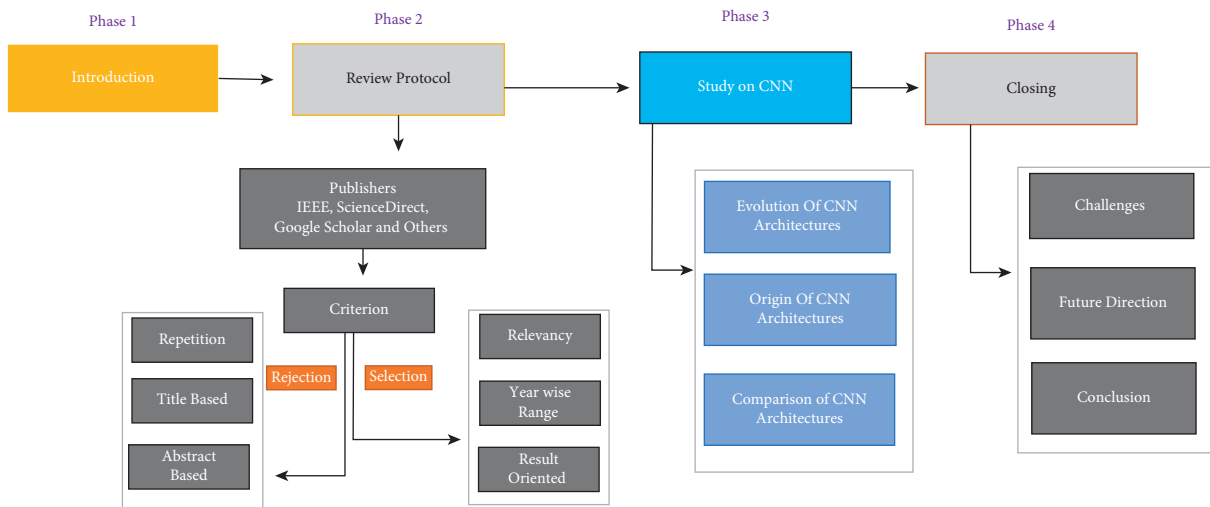


FIGURE 2: Structure of survey.

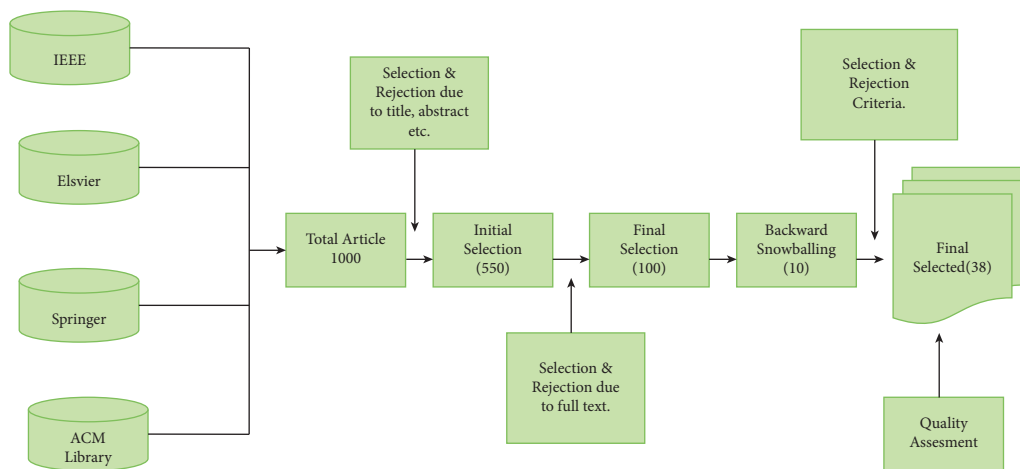


FIGURE 3: Review protocol.

$$\forall_n \in [1, 2, \dots, \mathbf{n}_C^{[l]}], \quad (1)$$

$$\text{Conv}(\mathbf{a}^{l-1}, \mathbf{K}^n)_{x,y} = \vartheta^{[l]} \left(\sum_{i=1}^{\mathbf{n}_H^{l-1}} \sum_{j=1}^{\mathbf{n}_W^{l-1}} \sum_{k=1}^{\mathbf{n}_C^{l-1}} \mathbf{K}_{i,j,k}^{(n)} \mathbf{a}_{x+i-1,k}^{l-1} + \mathbf{b}_n^l \right). \quad (2)$$

$$\mathbf{a}^l = \left[\vartheta^{[l]}(\text{Conv}(\mathbf{a}^{l-1}, \mathbf{K}^{(1)})), \vartheta^{[l]}(\text{conv}(\mathbf{a}^{l-1}, \mathbf{K}^{(2)})), \dots, \vartheta^{[l]}(\text{conv}(\mathbf{a}^{l-1}, \mathbf{K}^{\mathbf{n}_C^{[l]}})) \right]. \quad (3)$$

Dim. $\mathbf{a}^l = (\mathbf{n}_H^l, \mathbf{n}_W^l, \mathbf{n}_C^l)$ with $\mathbf{n}_{H/W}^l = [\mathbf{n}_{H/W}^{l-1} + 2\mathbf{p}^l - \mathbf{f}^{[l]}/s^{[l]} + 1]$; $S > 0$

$$= \mathbf{n}_{H/W}^{l-1} + 2\mathbf{p}^l - \mathbf{f}^{[l]}; s = 0.$$

$\mathbf{n}_C^l =$ number of filters.

The learned parameters at the l^{th} layer will be

- (i) Filters with $(\mathbf{f}^{[l]} * \mathbf{f}^{[l]} * \mathbf{n}_C^{l-1}) * \mathbf{n}_C^l$ (1).
- (ii) Bias with $(1 * 1 * 1) * \mathbf{n}_C^l$ parameters (broadcasting) (2).

The convolution layer can be summed up as a graph given in Figure 4.

3.2. Pooling Layer. After convolution operation, the next layer in CNN architecture is pooling. This layer performs down-sampling [9]. Its task is to downscale the information collected from the convolution layer from each feature and keep the essential information. At the same time, as input to the pooling layer, notations stated below are being considered.

Input: $\mathbf{a}^{[l-1]}$ with size $(\mathbf{n}_H^{[l-1]}, \mathbf{n}_W^{[l-1]}, \mathbf{n}_C^{[l-1]})$, \mathbf{a}^0 being the image input.

Padding: $\mathbf{p}^{[l]}$, Stride: $\mathbf{S}^{[l]}$

Size of pooling filter: $\mathbf{f}^{[l]}$

Pooling function: $\varphi^{[l]}$

Output: $\mathbf{a}^{[l]}$ with size $\mathbf{n}_H^l, \mathbf{n}_W^l, \mathbf{n}_C^l = \mathbf{n}_C^{l-1}$ (3)

The pooling operation can be understood by Figure 5.

3.3. Fully Connected Layer. Fully connected layers are an essential element of convolutional neural networks (CNNs), which proved very effective in image classification [10]. A finite number of neurons are taken as input and classified into relevant classes [11]. Mathematical representations of the fully connected layer are described below.

By considering the j^{th} node of a convolution or pooling layer with the dimensions:

$$\begin{aligned} \mathbf{z}_j^{[i]} &= \sum_{l=1}^{\mathbf{n}^{[i-1]}} \mathbf{W}_j^i \mathbf{I}\mathbf{W}_l^{[i-1]} + \mathbf{b}_j^{[i]} \\ \longrightarrow \mathbf{a}_j^{[i]} &= \varphi^{[i]}(\mathbf{z}_j^{[i]}). \end{aligned} \quad (4)$$

The input \mathbf{a}^{l-1} might be the result of a convolution or pooling operation with dimensions:

$$(\mathbf{n}_H^{l-1}, \mathbf{n}_W^{l-1}, \mathbf{n}_C^{l-1}). \quad (5)$$

$$\text{Dim}(\text{Conv}(\mathbf{a}^{l-1}, \mathbf{K}^{(n)})) = \mathbf{n}_H^l, \mathbf{n}_W^l.$$

Thus,

For plugging into a fully connected layer, we need to flatten the tensor to a 1D vector having the dimension:

$$\mathbf{n}_H^{[i-1]}, \mathbf{n}_W^{[i-1]}, \mathbf{n}_C^{[i-1]}. \quad (6)$$

The learned parameters at these layers are:

Weights: $w_{j,l}$ with $\mathbf{n}_{l-1} * \mathbf{n}_l$ parameters.

Bias with \mathbf{n}_l parameters.

Figure 6 represents the complete working of the fully connected layer.

3.4. Loss Function. In CNN, loss function is considered as one of the most important components. Loss is also known as the error of network and the way by which loss is calculated is called loss function. In CNN, loss functions are being used to calculate the gradients, and gradients are used for updating weights of neural networks. Mean square error, binary cross-entropy, categorical cross-entropy, and sparse categorical cross-entropy are some common loss functions.

3.5. Architectural Evolution. CNNs are considered as one of the best and most widely used biologically inspired techniques [12]. Their origin started with a neurobiological study. They provided platforms for several cognitive models, which all are replaced by CNN. Several researchers made efforts to improve CNN performance [13]. Multiple researchers are focused on the architectural evaluation of CNN. Table.1 shows architectural development of CNN. The main reason behind this focus is to improve the performance of CNN in terms of accuracy, training time, and misrate. However, there is still a gap in automating the architecture development automatically instead of manual.

4. Origin of Convolutional Neural Networks

Application of convolutional neural network is in practice since late 1980s. The first multilayered CNN architecture ConvNet was introduced by LeCuN et al. LeCuN proposed supervised training of ConvNet with backpropagation algorithm making a comparison with unsupervised reinforcement learning by using its predecessor neocognition [14–17]. LeCuN created the basic foundation of modern 2D CNN. ConvNet shows promising results in handwritten digit and zip code recognition [18]. In 1980, ConvNet was improved, and it was known in the neural network family as

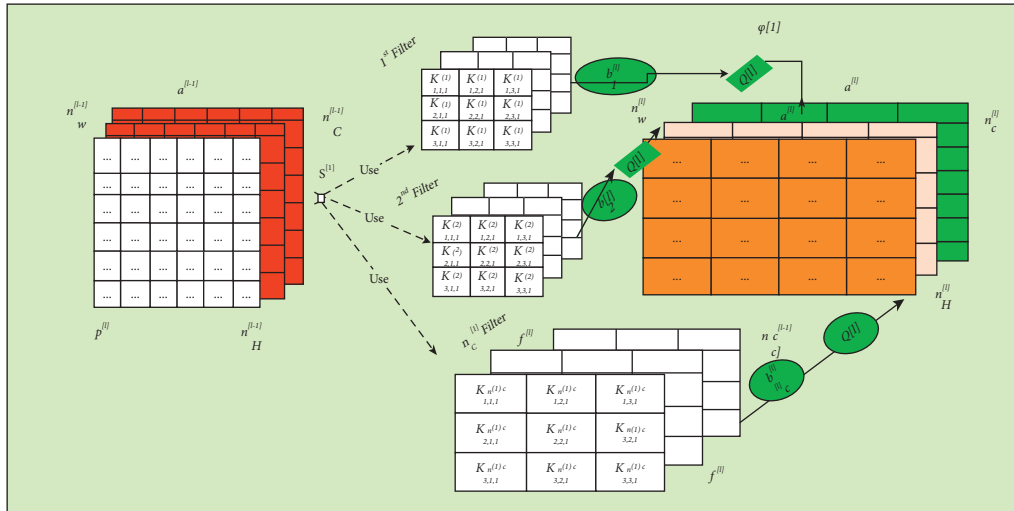


FIGURE 4: Convolution layer.

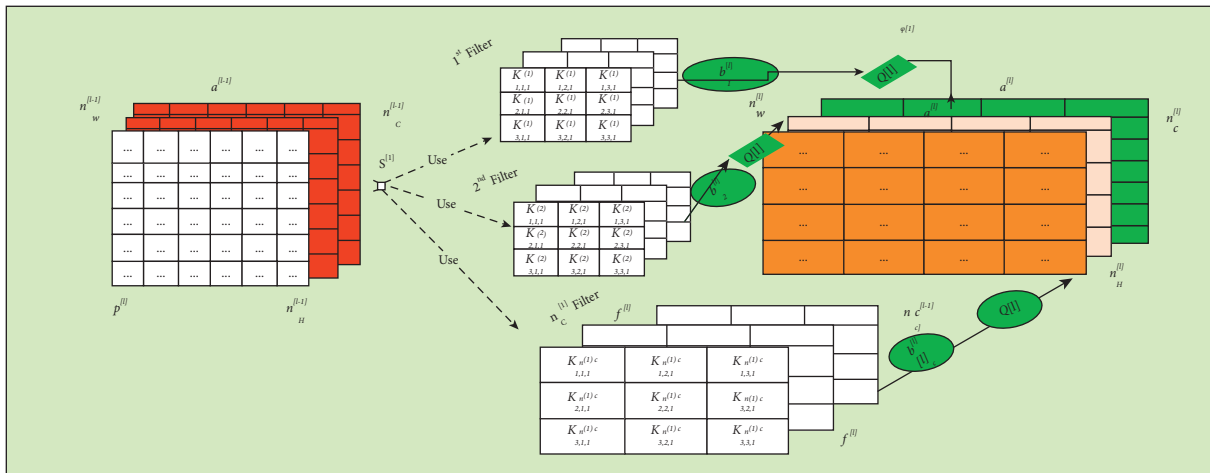


FIGURE 5: Pooling layer.

LeNet-5, and its application started in the classification of characters in document recognition. In early 1990, CNN became the most powerful as per its promising results in fingerprint recognition. Due to its powerful capacity, banks and ATMs started using it for the glory of fingerprints. The major drawback of LeNet-5 is that it does not perform well on image processing problems.

5. Comparative Analysis of CNN Architecture

For the last many years, application of CNN for various tasks like image classification, recognition, and speech recognition has increased [19]. Researchers for specific applications propose several CNN architectures. Table 2 presents brief information about each CNN architecture.

6. Impact of Hyperparameters

CNN has outstanding performance in several tasks, but designing the CNN architecture is still challenging. Its

design is purely based on choosing the best set of hyperparameters like number of convolution layer, type of pooling, type of activation function, of the fully connected layer in architecture, and so on. Recently proposed architectures are very deeper and more complex which need thousands of parameters to be trained for improved performance and need high-performance machines and plenty of time. Usually, tuning hyperparameter of CNN can be performed by the following methods which are manual search, grid search, and random search and are very time consuming and require GPUs for processing. Researchers are now focused on finding optimal ways of tuning hyperparameters. Several researchers applied particle swarm optimization, genetic algorithm, search and rescue algorithm, and so on. But still, there is room for improvement.

7. Challenges of CNN

Convolutional neural networks (CNNs) have performed extraordinarily in image processing and several other

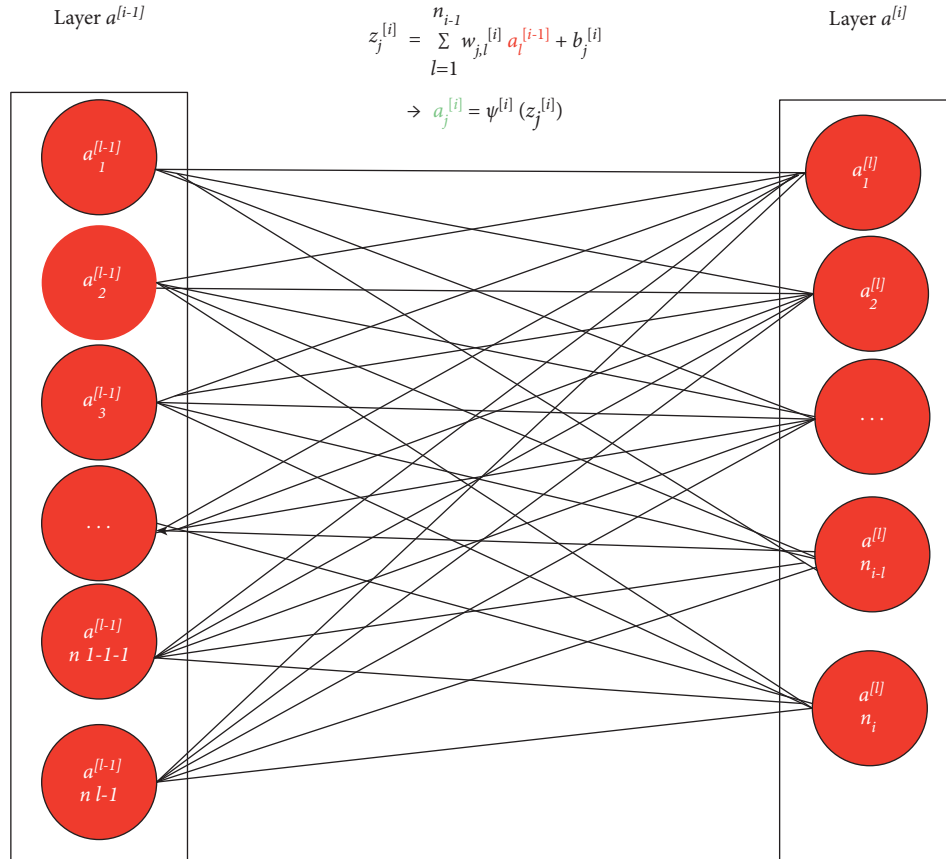


FIGURE 6: Fully connected layer.

TABLE 1: CNN architecture origin.

S.NO	Architecture	Year
1	LeNet	1998
2	AlexNet	2000
3	VGG 16	2003
4	Inception V1	2006
5	Inception V3	2008
6	ResNet 50	2011
7	Xception	2013
8	Inception V4	2015
9	InceptionResNets	2017
10	ResNext 50	2019

vision-related tasks [34]. However, CNN has some issues and limitations which need to be addressed like CNNs are based on a supervised learning mechanism, and therefore, they need a large amount of data for training. Sometimes, it is quite challenging [35]. The selection of hyperparameters has a significant impact on the performance of CNN. The minor change in values of the hyperparameter may affect the overall performance of a CNN [36]. So, careful selection of parameters is a significant design issue that needs to be addressed through some suitable optimization strategy [37]. Powerful hardware like GPU is required for the training of CNN. However, there is still a gap to implement it on smart devices [38]. Architecture-wise limitations and benefits are briefly discussed in Table 3.

8. Future Directions

As discussed in the above sections, creating suitable architecture of convolutional neural network depends upon the combination of convolution layers, number of pooling layers, number of filters, filter size, stride rate, and place of pooling layer, and all these parameters affect a lot on performance of classification in terms of accuracy, misrate, precision, and recall. Suitable parameter selection is purely handcrafted which takes lots of time and high computation powers like GPUs for training and testing combination of parameters again and again. In the future, we are planning to develop an algorithm based on swarm intelligence for the selection of structural parameters automatically.

TABLE 2: Comparison of CNN architecture.

S.NO	Architecture	Year	Major contribution	Parameters	Depth	Reference
1	LeNet	1998	Initial CNN architecture	0.060 M	5	[15]
2	AlexNet	2012	Deeper than LeNet and uses relu and overlapping pooling	60M	8	[20]
3	ZfNet	2014	Visualizing intermediate layer	60 M	18	[21]
4	VGG	2014	Uses small kernel size	4M	19	[22]
5	Google Net	2015	Introduced block concept, split transform, and merge idea	23.6 M	22	[23]
6	Inception V3	2015	Bottleneck issue is sorted and small filter size is added	23.6 M	159	[24]
7	Highway networks	2015	Introduced multipath concept	2.3 M	19	[25]
8	Inception V4	2016	Split transform and merge idea	35M	70	[26]
9	ResNet	2016	Residual learning	25.6	152	[27]
10	Deluge Net	2016	Allows cross layer information	20.2 M	146	[28]
11	Xception	2017	Depth-wise convolution followed by a point-wise convolution	8.6 M	452	[29]
12	ResNeXt	2017	Cardinally homogenous topology grouped convolution	27.5 M	152	[30]
13	Dense Net	2017	Cross-layered information flow	25.6 M	190	[31]
14	Channel boosted CNN	2018	Boosting of the original channel with additional artificial channels	—	—	[32]
15	Convolution block attention module	2018	Feature-map attention	—	—	[33]

TABLE 3: Architectural issues.

S.NO	Architecture	Benefits	Weakness
1	LeNet	Automatic learning of features reduces parameters and computation costs.	Large size filters; improper scaling of diverse classes of images.
2	AlexNet	Every type of feature extraction uses large as well as low-level filter sizes.	In the first and second layers, neurons are inactive.
3	ZfNet	Parameter tuning is introduced by visualizing the output of the intermediate layer.	Additional processing may be required for visualization.
4	VGG	The concept of the receptive field is introduced here. Proposes simple and homogenous topology.	Fully connected layers use high computational power.
5	Google Net	Multiscale filters are introduced. Dimensionality reduction is made by bottleneck layer.	Some information may be lost due to the application of the bottleneck layer.
6	Inception V3	Asymmetric filters and bottleneck are introduced for reducing the computation cost.	Designed architecture is complex.
7	Highway networks	A new training mechanism is introduced.	Complex design.
8	Inception V4	Multilevel feature extraction.	Architecture design which is complex may add up computation costs.
9	ResNet	Error rate is decreased, and residual learning concept is introduced.	Over-adaptation of hyperparameters for a specific task due to stacking of modules.
10	Deluge Net		
11	Xception	Separable convolution is introduced. Application of cardinality for learning good abstraction.	High computational cost.
12	ResNeXt	Application of cardinality in each layer for diverse transformation, and grouped convolution is applied.	High computational power.
13	Dense Net	Ensures the maximum flow of data in layers, and redundant features are avoided in learning.	An increase in feature maps increases parameters.
14	Convolution block attention module	Applies global average and max pooling concept simultaneously and improves flow of information.	The computational load may increase.

9. Conclusion

A convolutional neural network is considered one of the best techniques for vision-related tasks. Researchers have contributed a lot in the last several years. Multiple CNN architectures are proposed as per the need of its application and issue in existing CNN architecture. The improvements in CNN can be classified as activation, loss function, optimization,

regularisation, learning algorithms, and architectural advances. This work examines current advancements in CNN architectures, focusing on processing unit design trends, and proposes a taxonomy for contemporary CNN designs. This article discusses the history of CNNs, their uses, problems, and prospects in addition to categorizing CNNs into several classes.

By utilizing depth and other structural improvements, CNN's learning ability has dramatically enhanced over time.

The greatest gain in CNN performance has been noticed in recent research by substituting the usual layer structure with blocks. The creation of novel and effective block designs is now one of the themes of study in CNN architectures. A block in a network can play the function of an auxiliary learner. To increase performance, these auxiliary learners may use spatial or feature-map information or even boost input channels. By enabling problem-aware learning, these blocks significantly improve CNN performance.

Data Availability

The dataset used to support the findings of this study is available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Yim, J. Ju, H. Jung, and J. Kim, "Image classification using convolutional neural networks with a multi-stage feature," in *Proceedings of the 3rd International Conference on Robot Intelligence Technology and Applications*, pp. 587–594, Springer, Beijing, China, November 2015.
- [2] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [3] N. Senan, M. Aamir, R. Ibrahim, N. Taujuddin, and W. W. Muda, "An efficient convolutional neural network for paddy leaf disease and pest classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020.
- [4] A. Al Maashri, M. DeBole, M. Cotter, N. Chandramoorthy, and Y. Xiao, "Accelerating neuromorphic vision algorithms for recognition," in *Proceedings of the DAC Design Automation Conference*, pp. 579–584, IEEE, San Francisco, CA, USA, June 2012.
- [5] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep roots: improving cnn efficiency with hierarchical filter groups," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1231–1240, Honolulu, HI, USA, July 2017.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [7] A. G. Howard, "Some improvements on deep convolutional neural network-based image classification," 2013, <https://arxiv.org/abs/1312.5402>.
- [8] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cognitive Systems Research*, vol. 53, pp. 31–41, 2019.
- [9] S. Loussaief and A. Abdelkrim, "Convolutional neural network hyper-parameters optimization based on genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 252–266, 2018.
- [10] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *Proceedings of the 2018 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 122–129, IEEE, Kolkata, India, November 2018.
- [11] L. Ali, N. K. Valappil, D. N. A. Kareem, M. J. John, and H. Al Jassmi, "Pavement crack detection and localization using convolutional neural networks (CNNs)," in *Proceedings of the 2019 International Conference on Digitization (ICD)*, pp. 217–221, IEEE, Sharjah, United Arab Emirates, November 2019.
- [12] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, IEEE, Antalya, Turkey, August 2017.
- [13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, <https://arxiv.org/abs/1207.0580>.
- [14] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [15] Y. LeCun, L. D. Jackel, L. Bottou et al., "Learning algorithms for classification: a comparison on handwritten digit recognition," *Neural networks: The statistical mechanics perspective*, vol. 261, p. 276, 1995.
- [16] P. Perera, Y.-C. Tian, C. Fidge, and W. Kelly, "A comparison of supervised machine learning algorithms for classification of communications network traffic," in *Proceedings of the International Conference on Neural Information Processing*, pp. 445–454, Springer, Guangzhou, China, November 2017.
- [17] K. N. S. Nischal, G. N. Sai, C. Mathew, G. C. Gowda, and C. Bm, "A survey on recognition of handwritten zip codes in a postal sorting system," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, 2020.
- [18] X. Zhang and Y. LeCun, "Text understanding from scratch," 2015, <https://arxiv.org/abs/1502.01710>.
- [19] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2012.
- [20] G. Antonellis, A. Gavras, M. Panagiotou, B. Kutter, G. Guerrini, and A. Sander, "Shake table test of large-scale bridge columns supported on rocking shallow foundations," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 141, 2015.
- [21] M. J. Hodan, "Economic aspects of the international wheat Agreement of 1949," *The Quarterly Journal of Economics*, vol. 30, no. 1-2, pp. 225–231, 1954.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [23] C. Szegedy, L. Wei, J. Yangqing et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.
- [25] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, <https://arxiv.org/abs/1505.00387>.
- [26] C. Whitford, N. V. Movchan, H. Studer, A. Elsheikh, and m. i. mechanobiology, "A viscoelastic anisotropic hyperelastic constitutive model of the human cornea," *Biomechanics and Modeling in Mechanobiology*, vol. 17, no. 1, pp. 19–29, 2018.

- [27] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2018.
- [28] J. Kuen, X. Kong, G. Wang, and Y.-P. Tan, "DelugeNets: deep networks with efficient and flexible cross-layer information inflows," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 958–966, Venice, Italy, October 2017.
- [29] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.
- [30] A. Sharma and S. K. Muttoo, "Spatial image steganalysis based on resnext," in *Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 1213–1216, IEEE, Chongqing, China, October 2018.
- [31] G. Huang, Z. Liu, L. J. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, <https://arxiv.org/abs/1608.06993>.
- [32] A. Khan, A. Sohail, and A. J. Ali, "A new channel boosted convolutional neural network using transfer learning," 2018, <https://arxiv.org/abs/1804.08528>.
- [33] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, p. 1807, Munich, Germany, September 2018.
- [34] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Proceedings of the International conference on medical image computing and computer-assisted intervention*, pp. 411–418, Springer, Cambridge, UK, September 2013.
- [35] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2019.
- [36] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, "Speeding up the hyperparameter optimization of deep convolutional neural networks," *International Journal of Computational Intelligence and Applications*, vol. 17, no. 2, p. 1850008, 2018.
- [37] A. Gülcü and Z. KUş, "Hyper-parameter selection in convolutional neural networks using microcanonical optimization algorithm," *IEEE Access*, vol. 8, pp. 52528–52540, 2020.
- [38] S. Potluri, A. Fasih, L. K. Vutukuru, F. Al Machot, and K. Kyamakya, "CNN based high performance computing for real time image processing on GPU," in *Proceedings of the Joint INDS'11 & ISTET'11*, pp. 1–7, IEEE, Klagenfurt, Austria, July 2011.