Understanding of Convolutional Neural Network (CNN): A Review

Article *in* International Journal of Robotics and Control Systems · January 2023

DOI: 10.31763/ijrcs.v2i4.888

CITATIONS
148

READS
17,109

7 authors, including:

Ir. Purwono
Universitas Harapan Bangsa
72 PUBLICATIONS **410** CITATIONS
SEE PROFILE

Alfian Ma'arif
Universitas Ahmad Dahlan
208 PUBLICATIONS **2,195** CITATIONS
SEE PROFILE

Wahyu Rahmaniar
Institute of Science Tokyo
41 PUBLICATIONS **695** CITATIONS
SEE PROFILE

Haris Imam Karim Fathurrahman
Universitas Ahmad Dahlan
14 PUBLICATIONS **355** CITATIONS
SEE PROFILE

International Journal of Robotics and Control Systems
Vol. 2, No. 4, 2022, pp. 739-748
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Understanding of Convolutional Neural Network (CNN): A Review

Purwono [a,1], Alfian Ma'arif [b,2,*], Wahyu Rahmaniar [c,3], Haris Imam Karim Fathurrahman [b,4], Aufaclav Zatu Kusuma Frisky [d,e,5], Qazi Mazhar ul Haq [f,6]

[a] Universitas Harapan Bangsa, Jl. Raden Patah No. 100 Kedunglongsir Ledug Kembaran, Banyumas 53182, Indonesia
[b] Department of Electrical Engineering, Universitas Ahmad Dahlan, Banguntapan, Bantul, Yogyakarta 55191, Indonesia
[c] Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[d] Institute of Visual Computing & Human-Centered Technology, Technische Universität Wien, Vienna 1040, Austria
[e] Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta
[f] Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad, Pakistan
[1] purwono@uhb.ac.id; [2] alfianmaarif@ee.uad.ac.id; [3] wahyu.rahmaniar@gmail.com; [4] haris.fathurrahman@te.uad.ac.id; [5] aufaclav@ugm.ac.id; [6] qazimazhar@mcs.nust.edu.pk
* Corresponding Author

ARTICLE INFO

Article history
Received November 10, 2022
Revised December 15, 2022
Accepted January 15, 2023

Keywords
Deep Learning;
Artificial Intelligence;
Machine Learning;
Convolutional Neural Network;
Computer Vision;
Image Processing

ABSTRACT

The application of deep learning technology has increased rapidly in recent years. Technologies in deep learning increasingly emulate natural human abilities, such as knowledge learning, problem-solving, and decision-making. In general, deep learning can carry out self-training without repetitive programming by humans. Convolutional neural networks (CNNs) are deep learning algorithms commonly used in wide applications. CNN is often used for image classification, segmentation, object detection, video processing, natural language processing, and speech recognition. CNN has four layers: convolution layer, pooling layer, fully connected layer, and non-linear layer. The convolutional layer uses kernel filters to calculate the convolution of the input image by extracting the fundamental features. The pooling layer combines two successive convolutional layers. The third layer is the fully connected layer, commonly called the convolutional output layer. The activation function defines the output of a neural network, such as 'yes' or 'no'. The most common and popular CNN activation functions are Sigmoid, Tanh, ReLU, Leaky ReLU, Noisy ReLU, and Parametric Linear Units. The organization and function of the visual cortex greatly influence CNN architecture because it is designed to resemble the neuronal connections in the human brain. Some of the popular CNN architectures are LeNet, AlexNet and VGGNet.

1. Introduction

In recent years, deep learning technology has been used in various sectors. Deep learning has developed human-like abilities, such as knowledge learning, problem-solving, and decision-making [1]. Big companies have tried to adopt the latest digital technologies, including the Internet of Things (IoT), Big Data, Artificial Intelligence (AI), and Blockchain [2]. Deep learning technology is a development of machine learning and Artificial Intelligence (AI) [3].

In general, machine learning and deep learning can perform self-training without repetitive programming by humans. Deep learning requires initial data collection, called a data set, to predict the outcome of the data. Deep learning will produce output data based on training and testing data [4]. After passing the learning evaluation, deep learning can predict data. Deep learning can be used for pattern recognition or data prediction using big data in several scenarios [5]. Some methods used for the learning system are supervised and unsupervised learning. The supervised algorithm tries to identify the relationship between input and output data, creating a predictive model to predict the output based on the matched input [5]. In contrast, an unsupervised algorithm employs a learning system using non-labeled data. The algorithm can classify training data according to their distinctive characteristics, primarily based on dimension reduction and grouping systems [6].

Deep learning [7] differs from traditional machine learning systems that allow automatic feature extraction of raw data through various representational learning levels, from raw to high and abstract levels. Deep learning can increase learning capacity by amplifying significant patterns and suppressing irrelevant variation in input data along with the exponential advantage of representing complex non-linear functions of large amounts of data that continuously accumulate within hidden deep network layers [8]. Several techniques used in deep learning include convolutional, recurrent, and deep neural networks [9]. Deep learning technology utilizes artificial neural networks, especially the convolutional method.

One of the most widely used deep learning algorithms is the convolutional neural network (CNN). CNN was first introduced in the 1960s [10] and has shown promising performance results in computer vision [11]. CNN has become the most representative neural network in deep learning [12]. CNN has been utilized to solve complicated visual tasks with high computation [11] and is mainly used in image classification [13], [14], segmentation, object detection, video processing, natural language processing, and speech recognition [15]. Some implementations of CNN are video analysis in a study by Shri [16] dan image analysis by Roncancio [17]. The article's contribution is to describe CNN in a brief yet comprehensive explanation. Each constructing element is presented as another point of view in the AI method.

2. Convolutional Neural Network Layer and Architecture

CNN has four layers: convolution layer, pooling layer, fully connected layer, and nonlinearity layer [18]. Illustrations of those four layers are presented in Fig. 1 [19]. Further explanations regarding the description of each layer will be shown in the following subsections.

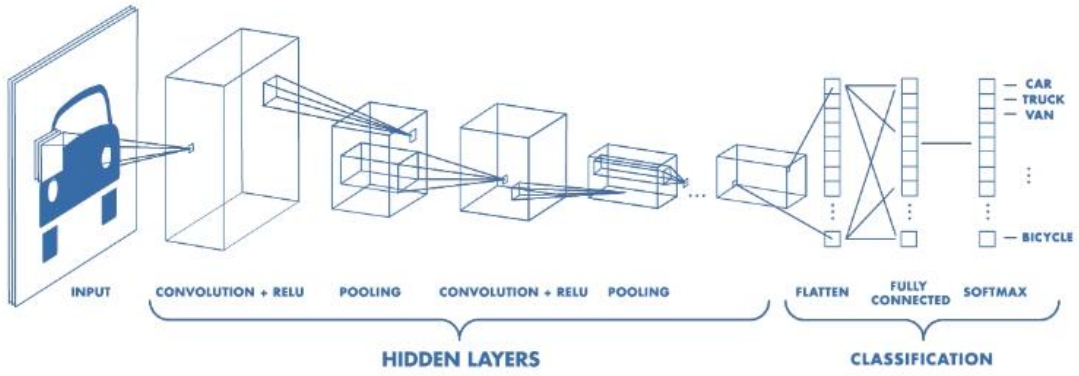

Fig. 1. Architecture of a CNN

2.1. Convolutional Layer

A machine sees an image as a set of numbers, commonly known as matrices. Each number represents the light intensity on a particular point called a pixel. Adam Geitgey illustrates pixels in an image on his website, Medium, as shown in Fig. 2.

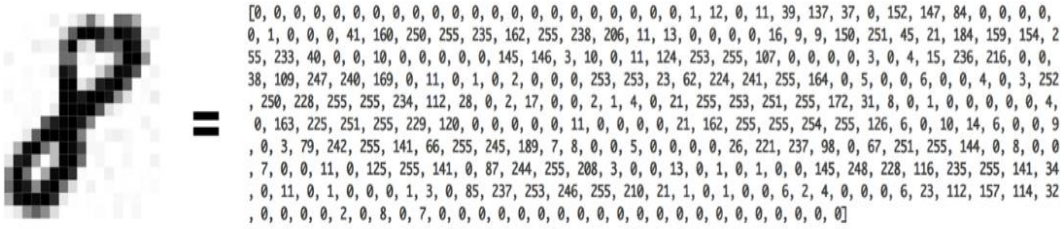

Fig. 2. Pixels of Image

The convolutional layer employs a kernel filter to calculate the convolution of input images, extracting the fundamental features. The filter kernel has the same dimension size but a smaller constant parameter value than the input image [20]. For instance, the acceptable length of a kernel filter for a 2D scalogram with a size of 35×35×35 is $f \times f \times 2$, where f = 3, 5, 7, and so on. However, the filter size has to be smaller than the size of the input image. The filter mask slides across the input image step by step and estimates the product point between the kernel filter weight and the pixel value of the input image. This process results in a 2D activation map. CNN will then learn the visual feature of the image. The general equation of the convolutional layer can be expressed as in the (1). Fig. 3 shows a simple illustration of the computational process in CNN that results in the activation map.

$$Activation\ map = Input * Filter$$
$$= \sum_{y=0}^{columns} \left(\sum_{x=0}^{rows} Input(x-p, y-q) Filter\ (x,y) \right) \quad (1)$$

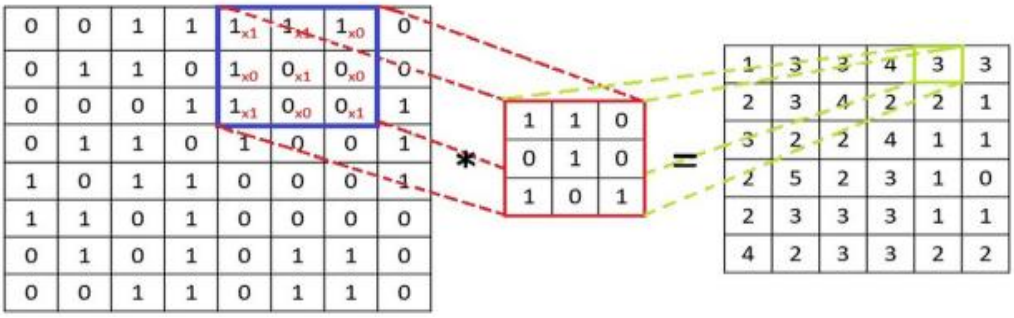

Fig. 3. Convolutional Layer

A convolutional layer is defined by: kernel size, stride length, and padding [21]. Kernel size is the kernel filter's size or the sliding kernel [22]. Stride length is the number of kernels that slide before making product points and creating output pixels [23]. Padding is the size of the 0-th frame set up around the input feature map [24].

2.2. Pooling Layer

The pooling layer will combine two consecutive convolutional layers. It reduces the number of parameters and computation loads by making down-sampling representations. The function in the pooling layer can result in a maximized or averaged value. A maximizing combination is often used for an optimal function [25]. The pooling layer is also helpful in reducing overfitting or computation weights. Fig. 4 represents a simple operation in dimension reduction of an activation map using the max-pooling function [20].

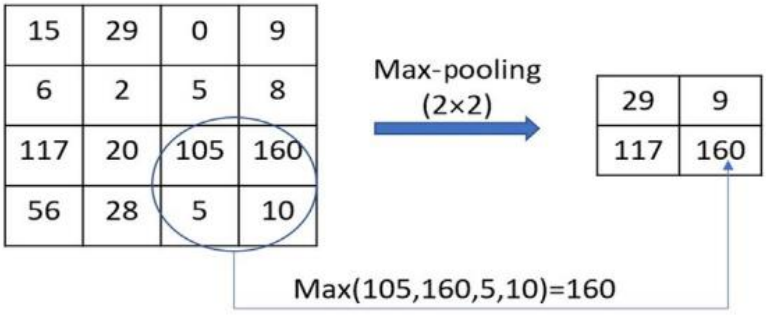

Fig. 4. Pooling Layer

2.3. Fully Connected Layer

The third layer is the fully connected layer, commonly called the convolutional output layer [26]. The fully connected layer is similar to a feedforward neural network, as shown in Fig. 5. The layer is commonly found in the bottom layer of the network. It receives input from the final pooling or the convolutional output layer, flattened before being sent to the subsequent layer. Even distribution of the output means unrolling all the values of the result obtained after the last pooling or convolutional layer into a vector (3D matrix). This method is a simple technique for studying high-level non-linear combinations of a feature represented by the output convolutional layer [26].

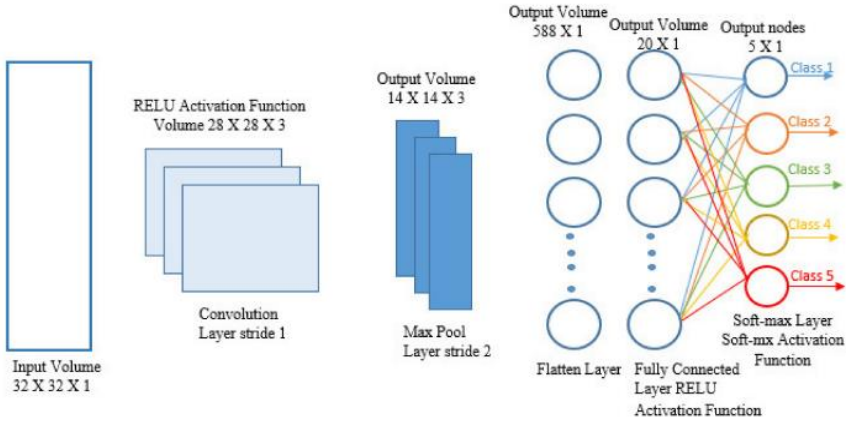

Fig. 5. Fully Connected Layer

2.4. Nonlinearity Layer (Activation Function)

An activation function plays an essential role in CNN layers. The filtered output provides another mathematical function called activation [26]. ReLU, abbreviated from the Rectified Linear Unit [27], is the most common activation function in feature extraction using CNN. The main objective of the activation function is to decide the final output of a neural network, such as 'yes' or 'no'. The activation function maps the output values between -1 and 1, 1 and 0, and so on.

The activation function can be differentiated into two categories, which are [26].

1. Linear Activation Functions. A simplified mathematical expression of linear activation functions can be written as $F(x) = CY$. The input values are multiplied with the constant parameter, c, which is the weight of each neuron. The process results in an output that is proportional to the input. Linear functions can perform more than the step function since they only give a single final answer of yes or no and not multiple choices.
2. Non-linear Activation Functions. Non-linear activation functions are used in modern neural networks. They allow the model to design a complicated mapping between the network's input and output, which is essential for complex learning and modeling systems.

Some of the most common or popular activation functions in CNN and other neural networks are listed as follows [28].

1. Sigmoid: this activation function uses real numbers as inputs and limits the output between 0 and 1. The curve of the sigmoid function is S-shaped and can be mathematically represented as in (2).

$$f(x)sigm = \frac{1}{1+e^{-x}} \tag{2}$$

2. Tanh: Apparently, the tanh function is similar to sigmoid since both use real numbers as their inputs. However, the tanh function limits its output in -1 and 1. The tanh function can be mathematically represented as in (3).

$$f(x)tanh = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}} \tag{3}$$

3. ReLU: ReLU is the most common function used in CNN. All inputs are converted into positive numbers. The computational load of ReLU is relatively lower than other functions. Mathematically, the representation of the ReLU function is presented as in (4).

$$f(x)ReLU = \max(0, x) \tag{4}$$

4. Leaky ReLu: If the ReLU function is responsible for down-scaling the negative inputs, the Leaky ReLU function ensures that inputs are never ignored. This function is used to solve a dying issue in ReLU. A mathematical representation of Leaky ReLU is presented in (5).

$$f(x)LeaklyReLU = \begin{cases} x, & \text{if } x > 0 \\ mx, & \text{if } x \leq 0 \end{cases} \tag{5}$$

5. Noisy ReLU: This function is used to perform Gaussian distribution. A mathematical expression of the Noisy ReLU function is presented in (6).

$$f(x)NoisyReLU = \max(\mathrm{x} + \mathrm{Y}), \mathrm{with Y} \sim \mathrm{N}(0, \sigma(\mathrm{x})) \tag{6}$$

6. Parametric Linear Units: Most of this function adopts the concept of Leaky ReLu. The difference between both functions is shown in the leak factor updated through the training mode. A mathematical representation of Parametric Linear Units can be seen in (7).

$$f(x)ParametricLinear = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \tag{7}$$

3. Popular CNN Architecture

Architecture in CNN is influenced by the organization and function of the visual cortex [26]. The design is made to resemble neuron connections in human brains. After knowing several layers in CNN, we will discuss some popular CNN architectures in this section.

3.1. LeNet

Currently, the development of LeNet has reached the LeNet-5 version. This version is a gradient-based CNN learning structure and was first introduced for digital handwriting character recognition [29]. The structure diagram of LeNet-5 is presented in Fig. 6 [30]. The input of LeNet-5 is grayscale images with a dimension of 32×32×1, which then pass six feature maps of a convolutional layer with a 5×5 filter and a stride. Those six feature maps are pre-processed image channels from the 28×28×6-sized convolutional operation. Stride is used as sliding control of a filter when passing through the dataset. The sliding control uses the tanh activation function. The second pooling layer has a 2×2 filter, six feature maps, and two strides. The tanh function on the second layer results in a 14×14×6 image.

The third step is a second convolutional layer with 16 feature maps, a 5×5 filter, and a stride, resulting in an image with a dimension size of 10×10×16. The fourth layer is a pooling layer with a 2×2 filter, two strides, and 16 feature maps. Four hundred nodes exist in the fourth layer, resulting in an output image with a dimension of 5×5×16. Then, there is a fully connected layer with 120 feature maps using the tanh activation function in the next layer; each has a dimension of 1×1. On this fifth layer, there are 120 nodes connected to 200 nodes on the fourth layer. The sixth layer is fully connected with 84 nodes, resulting in 10164 nodes of trained output parameters. The last layer in LeNet-5 is a fully connected layer with a 5-sized softmax activation function, resulting in a classified output image.

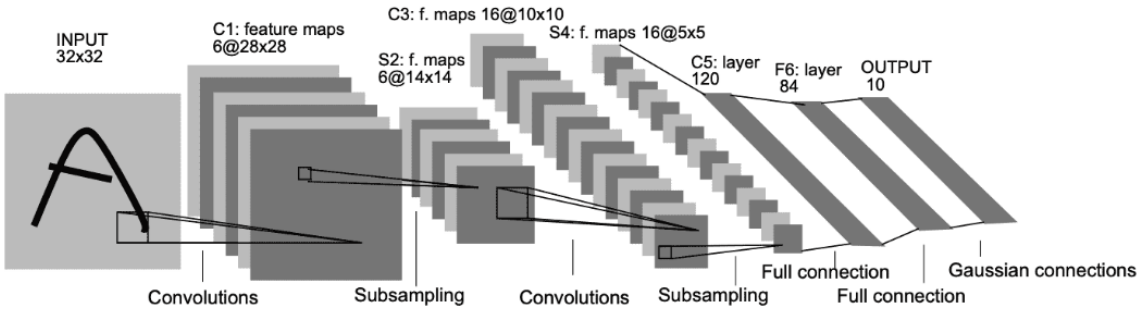

Fig. 6. LeNet-5 Architecture

3.2. AlexNet

Alex Krizhevsky introduced AlexNet in 2012 on a research project called ImageNet LargeScale Visual Recognition Challenge [31]. This architecture is one of CNN architectures with a basic, simple, yet effective layer design. AlexNet has five convolutional layers, followed by a pooling layer on its fourth layer and three layers of a fully connected layer on its fifth. In AlexNet architecture, the convolutional kernels are extracted during the back-propagation optimization procedure by optimizing with the stochastic gradient function [31]. The convolutional layer acts with the sliding convolutional kernel, creating convolved feature maps to gain information within a given neighborhood window. Equation 8 is the function used in AlexNet as a half-wave rectifier, which significantly fastens the training phase and avoids overfitting.

$$f(x) = \max(x, 0) \tag{8}$$

The dropout technique in Alexnet is used as a stochastic regulator in determining the number of input neurons with 0 values to reduce co-adaptation neurons, which is commonly used in the fully connected layer. The architecture of Alexnet can be seen in Fig. 7 [31].

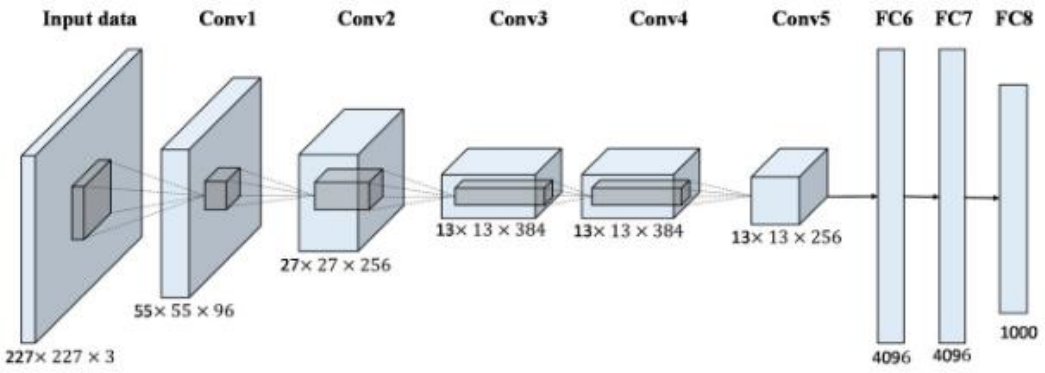

Fig. 7. AlexNet Architecture

3.3. VGGNet

The latest version of VGGNet to the day the article was made is the VGGnet-16. This architecture employs 13 convolutional layers and 3 fully connected layers [32]. The convolutional layer in VGG-16 has a size of 3×3 with a 1-sized stride and padding. Meanwhile, the pooling layer has a size of 2C2 with a 2-sized stride. The resolution of the input image in VGG-16 is 224×224. After each pooling layer is run, the size of the feature map will be reduced by 50%. The last feature map made before the fully connected layer is 7×7 with 512 channels and continues to be expanded to a vector with a size of 7×7×512 channels [33]. The architecture of AVGGNet-16 is represented in Fig. 8.

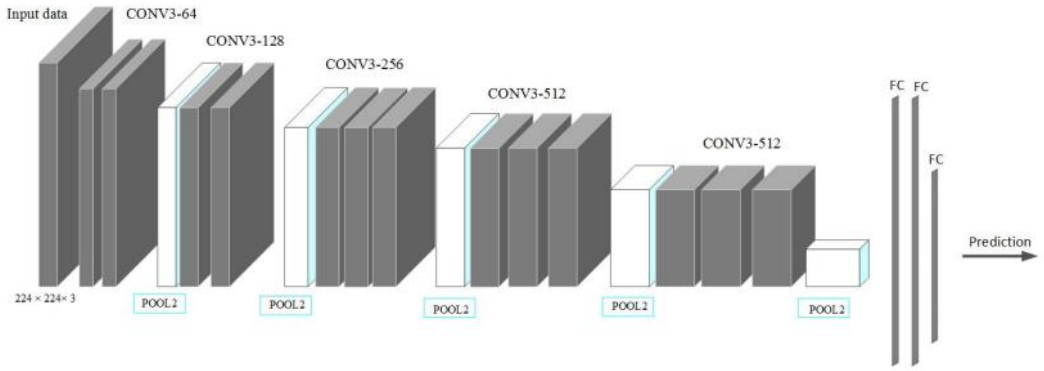

Fig. 8. VGGNet Architecture

4. Discussion

Artificial intelligence combined with a deep network is commonly called deep learning. In this study, deep learning is explained by some popular network architectures, such as LeNet [34], AlexNet [35], dan VGGNet [36]. In general, all network architectures can be differentiated by the depth of the network and the architectural approach method. The resolution of input images used in each architecture differs based on the initial input criteria. LeNet uses a smaller input image (32×32) than AlexNet and VGG Net. The convolutional layers used in the architecture also differ; for instance, VGG Net has 13 layers. Then, LeNet in the study utilizes the MNIST database to measure accuracy, resulting in an accuracy greater than 90% of the prediction truth level.

Meanwhile, AlexNet and VGG Net utilize the ILSVRC database in their error measurement, resulting in 15.3% and 6.8% error rates. In detail, the distinctive characteristics of each CNN architecture are listed in Table 1. Another study finding by Swapna [37], explained error rates in each CNN architecture and is in accordance with the results of this study.

Table 1. CNN Architecture

Methods	LeNet	AlexNet	VGG Net
Image resolution	32×32×1	227×227×3	224×224×3
Number of convolutional layers	3	5	13
Number of pooling layers	2	3	5
Activation function	ReLu, softmax	ReLu, softmax	ReLu, softmax
Accuracy/error rate	>98% (accuracy based on MNIST database)	15.3% (error rate based on ILSVRC)	6.8% (error rate based on ILSVRC)

5. Conclusion

In general, machine learning can perform self-learning without any repetitive programming by humans. Meanwhile, deep learning is an implementation of machine learning that aims to imitate

human brains' work using artificial neural networks. One of the most popular methods in deep learning is the convolutional neural network (CNN). This algorithm has many essential advantages, including image classification, segmentation, object detection, video processing, natural language processing, and speech recognition. CNN has four layers: a convolutional layer, a pooling layer, a fully connected layer, and a nonlinearity layer. The main technique in CNN algorithms is convolution; a filter will slide upon an input, then combine the input and filter values in the feature map. The pooling layer will combine two consecutive convolutional layers. It also minimizes the number of parameters and computational load by performing a down-sampling representation. The function in the pooling layer can result in maximized or averaged results. The fully connected layer connects all activation neurons from the retrospective layer to the next layer. An activation function plays an important role in CNN layers.

The filtered output provides another mathematical function called an activation function. The layer has different functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Noisy ReLU, and Parametric Linear Units. Sigmoid uses real numbers as inputs and limits the output between 0 and 1. Tanh is similar to sigmoid since both use real numbers as inputs, but the tanh function limits its output in -1 and 1. ReLU becomes the most commonly used function in CNN. All inputs are converted into positive numbers. The computational load of ReLU is relatively lower than other functions. If the ReLU function is responsible for down-scaling the negative inputs, the Leaky ReLU function ensures that inputs are never ignored; this function is used to solve a dying issue in ReLU. Noisy ReLU is used to perform Gaussian distribution.

Meanwhile, most of the functions of Parametric Linear Units adopt the Leaky ReLU concept. The difference between both functions is shown in the leakage factor, updated through the training mode. Some of the popular CNN architectures are LeNet, AlexNet, and VGGNet. LeNet has become one of the simplest CNN architectures, which 2 convolutional and 3 fully connected layers. In comparison, AlexNet has 5 convolutional and 3 fully connected layers. VGGNet uses 13 convolutional and 3 fully connected layers. Various advantages of each CNN architecture make it suitable for solving complex visual tasks with high computational loads. CNN is also one of the most representative neural networks in deep learning.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] M. M. Khan, S. Hossain, P. Mozumdar, S. Akter, and R. H. Ashique, "A review on machine learning and deep learning for various antenna design applications," *Heliyon*, vol. 8, no. 4, p. e09317, 2022, https://doi.org/10.1016/j.heliyon.2022.e09317.

[2] M. Ancin, E. Pindado, and M. Sanchez, "New Trends in the Global Digital Transformation Process of the Agri-Food Sector: An Exploratory Study Based on Twitter," *Agricultural Systems*, vol. 203, p. 103520, 2022, https://doi.org/10.2139/ssrn.4093526.

[3] P. W. Tien, S. Wei, J. Darkwa, C. Wood, and J. K. Calautit, "Machine Learning and Deep Learning Methods for Enhancing Building Energy Efficiency and Indoor Environmental Quality – A Review," *Energy AI*, vol. 10, p. 100198, 2022, https://doi.org/10.1016/j.egyai.2022.100198.

[4] M. Parzinger, L. Hanfstaengl, F. Sigg, U. Spindler, U. Wellisch, and M. Wirnsberger, "Comparison of different training data sets from simulation and experimental measurement with artificial users for occupancy detection — Using machine learning methods Random Forest and LASSO," *Build. Environ.*, vol. 223, p. 109313, 2022, https://doi.org/10.1016/j.buildenv.2022.109313.

[5] M. Zhu *et al.*, "A review of the application of machine learning in water quality evaluation," *Eco-Environment Heal.*, vol. 1, no. 2, pp. 107–116, 2022, https://doi.org/10.1016/j.eehl.2022.06.001.

[6] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, 2019, https://doi.org/10.1016/j.ins.2019.02.008.

[7] G. Fu, Y. Jin, S. Sun, Z. Yuan, and D. Butler, "The role of deep learning in urban water management: A critical review," *Water Res.*, vol. 223, p. 118973, 2022, https://doi.org/10.1016/j.watres.2022.118973.

[8] C. Shen, "A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists," *Water Resour. Res.*, vol. 54, no. 11, pp. 8558–8593, 2018, https://doi.org/10.1029/2018WR022643.

[9] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The Understanding of Deep Learning: A Comprehensive Review," *Math. Probl. Eng.*, 2021, https://doi.org/10.1155/2021/5548884.

[10] M. Wu, X. Liu, N. Gui, X. Yang, J. Tu, S. Jiang, and Q. Zhao, "Prediction of remaining time and time interval of pebbles in pebble bed HTGRs aided by CNN via DEM datasets," *Nucl. Eng. Technol.*, 2022, https://doi.org/10.1016/j.net.2022.09.019.

[11] M. M and S. P, "COVID-19 infection prediction from CT scan images of lungs using Iterative Convolution Neural Network model," *Adv. Eng. Softw.*, vol. 173, p. 103214, 2022, https://doi.org/10.1016/j.advengsoft.2022.103214.

[12] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–21, 2021, https://doi.org/10.1109/TNNLS.2021.3084827.

[13] M. K. Bohmrah and H. Kaur, "Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 476–483, 2021, https://doi.org/10.1016/j.gltp.2021.08.003.

[14] W. L. Mao, H. I. K. Fathurrahman, Y. Lee, and T. W. Chang, "EEG dataset classification using CNN method," *Journal of physics: conference series*, vol. 1456, no. 1, p. 012017, 2020, https://doi.org/10.1088/1742-6596/1456/1/012017.

[15] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, https://doi.org/10.1007/s10462-020-09825-6.

[16] S. J. Shri and S. Jothilakshmi, "Crowd Video Event Classification using Convolutional Neural Network," *Comput. Commun.*, vol. 147, pp. 35–39, 2019, https://doi.org/10.1016/j.comcom.2019.07.027.

[17] R. Roncancio, A. El Gamal, and J. P. Gore, "Turbulent flame image classification using Convolutional Neural Networks," *Energy AI*, vol. 10, p. 100193, 2022, https://doi.org/10.1016/j.egyai.2022.100193.

[18] T. Bezdan and N. Bačanin Džakula, "Convolutional Neural Network Layers and Architectures," *International Scientific Conference on Information Technology and Data Related Research*, pp. 445–451, 2019, https://doi.org/10.15308/Sinteza-2019-445-451.

[19] The Mathworks, *Introducing Deep Learning with MATLAB*, 2018, https://www.mathworks.com/campaigns/offers/deep-learning-with-matlab.html.

[20] S. A. Singh, T. G. Meitei, and S. Majumder, "Short PCG classification based on deep learning," *Deep Learning Techniques for Biomedical and Health Informatics*, Elsevier Inc., pp. 141–164, 2020, https://doi.org/10.1016/B978-0-12-819061-6.00006-9.

[21] S. A. Suha and T. F. Sanam, "A deep convolutional neural network-based approach for detecting burn severity from skin burn images," *Mach. Learn. with Appl.*, vol. 9, no. April, p. 100371, 2022, https://doi.org/10.1016/j.mlwa.2022.100371.

[22] C. Ding, Y. Li, Y. Xia, L. Zhang, and Y. Zhang, "Automatic kernel size determination for deep neural networks based hyperspectral image classification," *Remote Sens.*, vol. 10, no. 3, 2018, https://doi.org/10.3390/rs10030415.

[23] R. Riad, O. Teboul, D. Grangier, and N. Zeghidour, "Learning strides in convolutional neural networks," *International Conference on Learning Representations*, pp. 1–17, 2022, https://doi.org/10.31219/osf.io/4yz8f.

[24] A. Nguyen, S. Choi, W. Kim, S. Ahn, J. Kim, and S. Lee, "Distribution Padding in Convolutional Neural Networks," *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 4275–4279, 2019, https://doi.org/10.1109/ICIP.2019.8803537.

[25] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Computer vision for human-machine interaction," *Computer vision for human–machine interaction*, *Computer Vision For Assistive Healthcare*, pp. 127–145, 2018, https://doi.org/10.1016/B978-0-12-813445-0.00005-8.

[26] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electron.*, vol. 10, no. 20, p. 2470, 2021, https://doi.org/10.3390/s19010217.

[27] Z. J. Wang *et al.*, "CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1396–1406, 2021, https://doi.org/10.1162/neco_a_00990.

[28] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, p. 83, 2021, https://doi.org/10.1186/s40537-021-00444-8.

[29] G. Wei, G. Li, J. Zhao, and A. He, "Development of a LeNet-5 gas identification CNN structure for electronic noses," *Sensors*, vol. 19, no. 1, pp. 1–17, 2019, https://doi.org/10.3390/s19010217.

[30] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, pp. 2352–2449, 2017, https://doi.org/10.1162/neco_a_00990.

[31] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sens.*, vol. 9, no. 8, 2017, https://doi.org/10.3390/rs9080848.

[32] U. Muhammad, W. Wang, S. P. Chattha, and S. Ali, "Pre-trained VGGNet Architecture for Remote-Sensing Image Scene Classification," *Proceedings - International Conference on Pattern Recognition*, pp. 1622–1627, 2018, https://doi.org/10.1109/ICPR.2018.8545591.

[33] Q. Guan *et al.*, "Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: A pilot study," *J. Cancer*, vol. 10, no. 20, pp. 4876–4882, 2019, https://doi.org/10.7150/jca.28769.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, https://doi.org/10.1109/5.726791.

[35] 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, https://doi.org/10.1145/3065386.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015, https://doi.org/10.48550/arXiv.1409.1556.

[37] M. Swapna, D. Y. K. Sharma, and D. B. Prasad, "CNN Architectures: Alex Net, Le Net, VGG, Google Net, Res Net," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 953–959, Mar. 2020, https://doi.org/10.35940/ijrte.F9532.038620.