Essential Building Blocks of Convolutional Neural Network for Deep Learning

OLASUNKANMI F. O[1], DANJUMA S.S[2], HARUNA B[3], KOLAWOLE S.F[4]
[1] *NISMES, Kaduna, affiliated to Nuhu Bamalli polytechnic,Zaria, Nigeria*
[2, 3] *Nuhu Bamalli Polytechni, Zaria*
[4] *Nigerian Defence Academy Kaduna, Nigeria.*

Abstract- Computations for Deep Learning (DL) are designed to mimic the functionality of the neurons found in the brains of mammals. To achieve this, DL utilizes different algorithms to learn patterns in number fed as input to the system. One among the algorithms it uses to achieve this is the Convolutional Neural Network (CNN). CNN is best used to manipulate images in order to enable machines learn the patterns in them. In this article, kernel or filter, stride, padding, pooling and flattening shall be considered as fundamental building blocks of CNN for DL.

Indexed Terms- CNN, DL, Neural network, kernel, stride, padding, pooling, flattening

I. INTRODUCTION

Deep Learning (DL) is a type of Machine Learning (ML) that uses numerous layers of processing to extract increasingly higher-level features from input and it is based on neural networks [1]. This entails converting data into numbers and looking for forms among them [2]. DL has exploded in prominence in scientific computing, with its techniques being utilized by a wide range of sectors to solve complicated issues [3]. To perform certain tasks, all DL algorithms employ various forms of neural networks [4]. Machines are trained using DL algorithms that learn from examples [5]. DL is extensively used in trades such as online business, healthcare services, showbiz, and publicizing [6].

II. NEURAL NETWORKS

A neural network is made up of artificial neurons, also known as nodes that are organized like the human brain [7]. Three layers of nodes are placed on top of each other which are the input layer, the hidden layer(s) and the output [8] as shown in Fig 1.1.

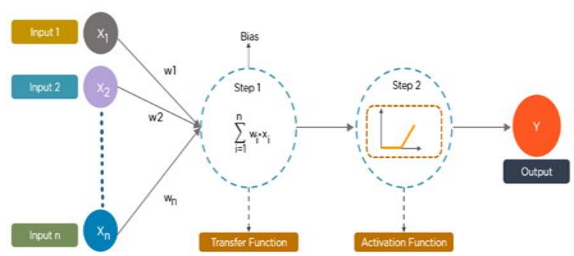


Fig 1.1 Structure of neural network [1]

Data inputs are fed into each node at the input layer. These inputs are multiplied with random weights, this procedure is referred to as transfer function. Finally, a nonlinear function called the activation function is used to generate the output.

III. HOW DEEP LEARNING ALGORITHMS WORK

While DL algorithms use self-taught demonstrations, they count on Artificial Neural Networks (ANNs) that simulate how the brain processes data [9]. Algorithms leverage unknown elements in the input supply to extract features, organize objects, and uncover important data patterns throughout the training phase [10]. This occurs at several levels, the algorithms used to construct the models are similar to how self-learning machines are trained. Several algorithms are used in deep learning models. While no network is flawless, certain algorithms are better suited for specific jobs than others. To select the best, it is necessary to have a thorough understanding of all primary algorithms. This work shall focus on the fundamentals of Convolutional Neural Networks (CNNs).

IV. CONVOLUTIONAL NEURAL NETWORKS (CNNS)

CNNs are deep learning algorithms that take input (picture) and convolve it with filters or kernels to extract features. A fXf filter is used to convolve the same feature across an entire NxN image[11, 12].

V. KERNEL

This is a type of filter that is utilized to obtain features from an input data. It is a matrix that moves over the input data by performing the dot product operation on a region of the data to obtain an out in matrix format [13]. Kernel moves on the input data by the stride value. Equation (1) is used to derive the size of the feature map.

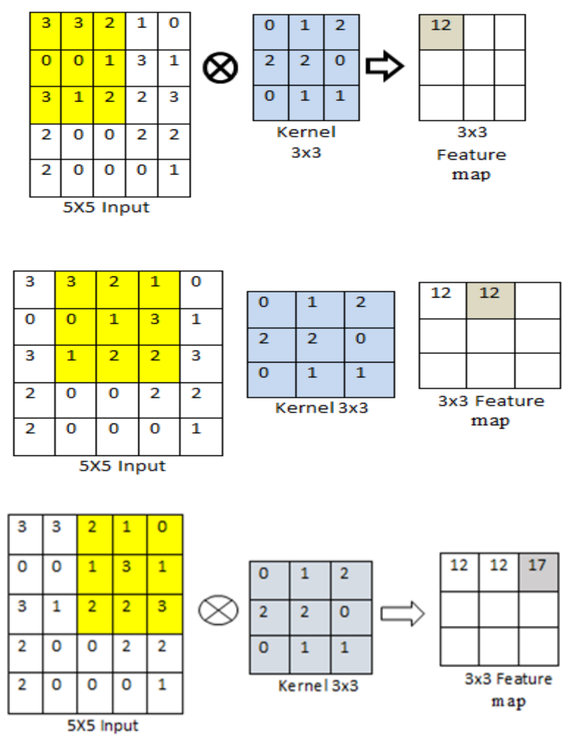
O = [Z – K] +1 (1)

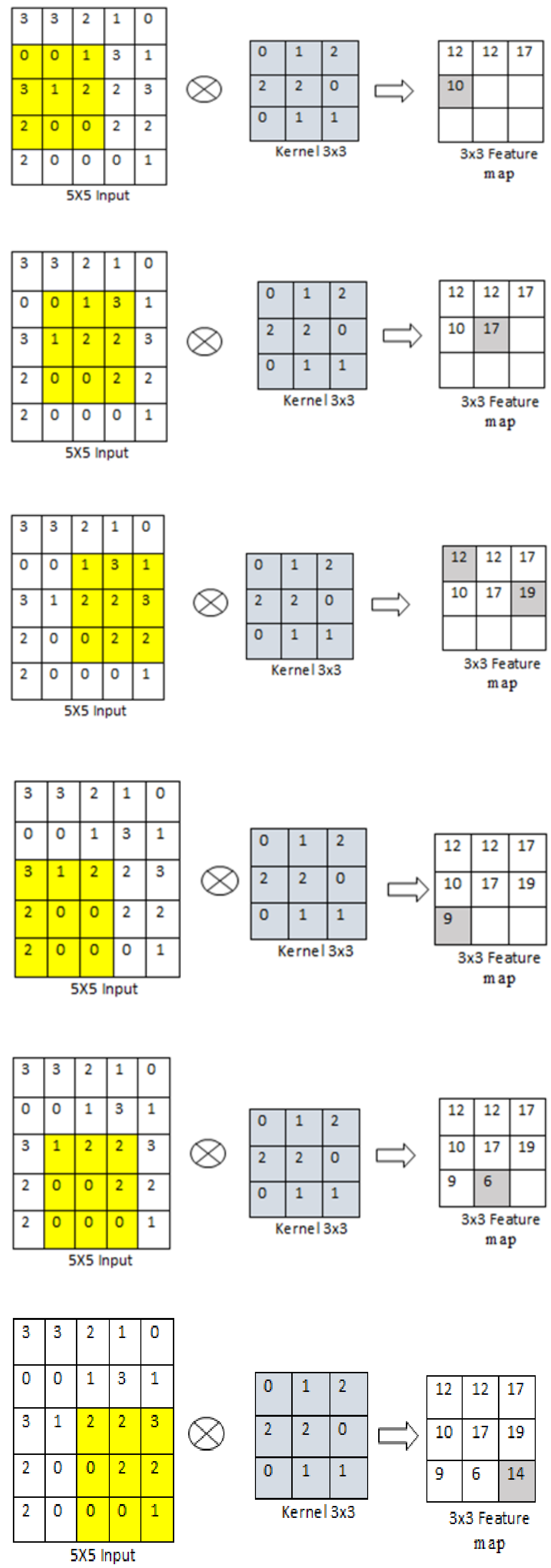
O = Feature map

Z = Input size

K = Kernel size

O = [5 – 3] + 1 = 3



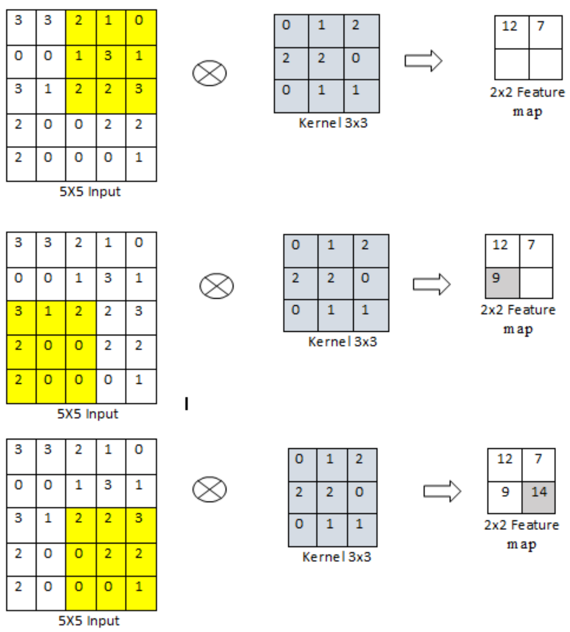


VI. STRIDE

Stride allows the kernel or filter to move one pixel across the input data from top to bottom and from left to right. Stride is the measure of how much the input data moves when the filter is applied. Equation (2) explains how to use stride to get the feature output.

$$O = ([Z - K]/S) + 1 \quad (2)$$

O = Feature map
Z = Input size
K = Kernel size
S = Stride
O = ([5 – 3]/2) + 1 = 2

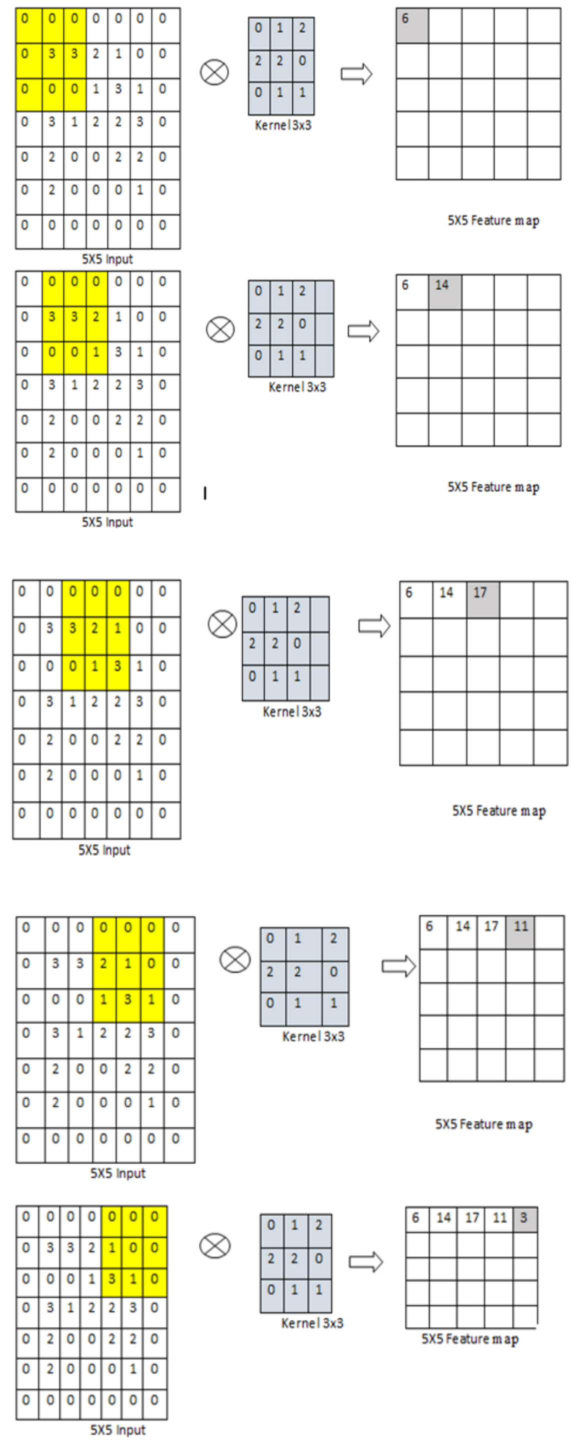


VII. PADDING

This approach is used to fix issues related to the border of the input data; it does this by adding pixels to the edge of the data. The size of the feature output can be determined using equation (3)

$$O = [(Z - k + 2P)/S] + 1 \quad (3)$$

O = Feature map
Z = Input size
K = Kernel size
S = Stride size
P = Padding size
O = ([5 – 3 + 2 * 1]/1) + 1 = 5



6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

Final 5X5 Feature map

VIII. POOLING

This utilizes down-sampling the detected features in the feature maps by summarizing the presence of features in patches [14, 15]. There are various pooling techniques, two of which are average pooling and maximum pooling, respectively, which are used to summarize the average presence of a feature and the most activated existence of a feature. . As long as all of the crucial attributes are preserved, the idea of pooling is to reduce the dimension of the data.

- Max pooling

When a group of data is a 2x2 matrix, this takes the maximum number of the group's chosen elements and discards any characteristics that don't fit.

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

5X5 Feature map

14	

2x2 Max Pooling

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

5X5 Feature map

14	17

2x2 Max Pooling

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

5X5 Feature map

14	17
11	

2x2 Max Pooling

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

5X5 Feature map

14	17
11	19

2x2 Max Pooling

- Average pooling

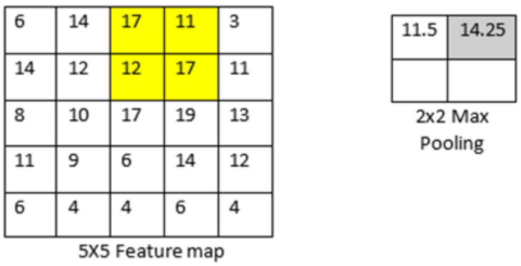
If the group of data is a 2x2 matrix, characteristics that do not fit into the matrix are removed and the average of all the numbers in the specified group of elements in the matrix is taken.

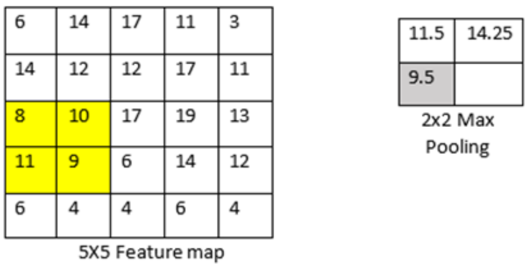
6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

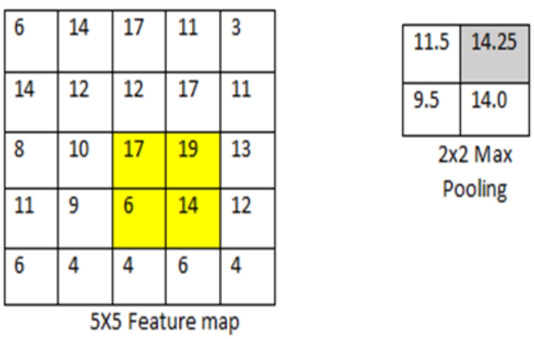
5X5 Feature map

11.5	

2x2 Max Pooling







IX. FLATTENING

In order to feed the neural network for classification, this converts the featured map matrix into a one-dimensional matrix [16].

Acknowledgment
This research is supported by the Department of Electrical and Electronics Engineering Nigerian Defence Academy, Kaduna State, Nigeria.

REFERENCES

[1] B. Avijeet, Top 10 Deep Learning Algorithms you Should Know in 2022, https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm, 2021

[2] R. Yamashita, M. Nishio, R.K.G Do, *et al.* Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9,** 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[3] Y. Tang, “Deep learning using linear support vector machines,” arXiv prprint arXiv:1306.0239, 2013.

[4] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.

[5] A. A. M. Al-Saffar, H. Tao and M. A. Talab, "Review of deep convolution neural network in image classification," 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), 2017, pp. 26-31, doi: 10.1109/ICRAMET.2017.8253139.

[6] M. N. Islam, T. T. Inan, S. Rafi, S. S. Akter, I. H. Sarker and A. K. M. N. Islam, "A Systematic Review on the Use of AI and ML for Fighting the COVID-19 Pandemic," in IEEE Transactions on Artificial Intelligence, vol. 1, no. 3, pp. 258-270, Dec. 2020, doi: 10.1109/TAI.2021.3062771.

[7] M. Mishra and M. Srivastava, "A view of Artificial Neural Network," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), 2014, pp. 1-3, doi: 10.1109/ICAETR.2014.7012785.

[8] C. Zhang and W. Xu, "Neural networks: Efficient implementations and applications," 2017 IEEE 12th International Conference on ASIC (ASICON), 2017, pp. 1029-1032, doi: 10.1109/ASICON.2017.8252654.

[9] R. E. Uhrig, "Introduction to artificial neural networks," Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics, 1995, pp. 33-37 vol.1, doi: 10.1109/IECON.1995.483329.

[10] T. Maimaitiaili, L. Dai. Deep neural network based uyghur large vocabulary continuous speech recognition Journal of Data Acquisition and Processing, 2015, 30 (2): 365-371.

[11] R. Chauhan, K. K. Ghanshala and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First

International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.

[12] M. D. Zeiler and Fergus, “Visualizing and understanding convolutional networks”. European Conference on Computer Vision, vol 8689. Springer, Cham, pp. 818-833, 2014.

[13] Hyeonuk Kim, Jaehyeong Sim, Yeongjae Choi and Lee-Sup Kim, "A kernel decomposition architecture for binary-weight Convolutional Neural Networks," 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1-6, doi: 10.1145/3061639.3062189.

[14] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, “Fundamental Concept of Convolutional Neural Network”,2020, DOI: 10.1007/978-3-030-32644-9_36

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR, abs/1511.00561, 2015.

[16] Binod Suman Academy, “Convolutional Neural Networks | CNN | Kernel | Stride | Padding | Pooling | Flatten | Formula”, https://www.youtube.com/watch?v=Y1qxI-Df4Lk&t=13s