Comparative Analysis of Different Architectures of Deep Convolutional Neural Networks

Mahwish Umer, Mansoor Ahmed Khuhro, Tahir Qadri,Kamlesh Kumar, Asif Ali Laghari

Abstract— Deep Neural Network (DNN) contains many layers of neurons, which provides better performance in terms of accuracy over a large dataset of images based on the DNN's artificial sense of recognition. The Deep Convolutional Neural Network (CNN) architecture is based on multiple convolutional layers of neurons and each neuron in a layer is designed towards feed-forward direction. CNN can learn features of unstructured data such as images, voice and videos during a model's training process. If the number of weighted layers increases in a CNN model, it results in higher object detection accuracy. On the other hand, if more weighted parameters are included in the CNN, it requires a high-performance GPU in the training session. The Xception model is better than VGG-16, ResNet50 and DenseNet121 because it passes the same input to the depth-wise isolated blocks and later merges the output of these blocks as input for the final classification layer The model sizes are directly proportional to the number of parameters involved, affecting the models' performance during the object recognition process. These all models are pre-trained and require transfer learning for fine-tuning in a new dataset. There is a limitation to the models that the dataset must be in the RGB- system. This study compares the CNN architectures of VGG-16, ResNet50, Xception and DenseNet121. This paper's findings show that the Xception model of CNN has performed well while classifying .digital images

***Keywords*— Deep Convolutional Neural Network, VGG-16, ResNet50, Xception, DenseNet121**

INTRODUCTION

The architecture of a CNN can be considered a human brain is connecting with diverse types of neurons. It is possible to classify a raw image with the help of CNN without any image preprocessing. CNN is beneficial in reducing images into a compressed format which can quickly be processed without excluding the original features of an image. CNN's are very good at picking up patterns in an image like gradients, circles, and lines. CNN can classify an unprocessed image without any preprocessing. In CNN Neurons are trained in a forward direction. That's why CNN is also called a feed-forward neural network, which has a special type of layer called a Convolutional layer [1].

Convolutional Neural networks (CNN) are often applied on large number of images and videos in order to recognize and classify them based on their structure or features. CNN is designed by constructing different types of layers such as Convolutional Layer, Pooling Layer, Flatten Layer, Dense Layer and Fully Connected Layer.

This paper consists of different layers of CNN in Section II, different CNN architectures in section III, section IV shows the result and discussion, while section V contains the conclusion.

DIFFERENT LAYERS OF CONVOLUTIONAL NEURAL NETWORK

This section provides the functionality of each layer of CNN over input, and this defines how input gets changed from layer to layer.

Convolutional layer

The convolutional layer (Conv Layer) is a particular layer in Convolutional Neural Network used to detect vertical and horizontal lines from the input images. Conv Layer has weighted parameters which are needed to be trained. Filters applied in the Neural Networks resulted in an activation map, and the size of filters should be smaller than the input data. The output data of Conv Layer is obtained by stacking the activation maps of all filters along the depth dimension, as shown in Figure .1[2].

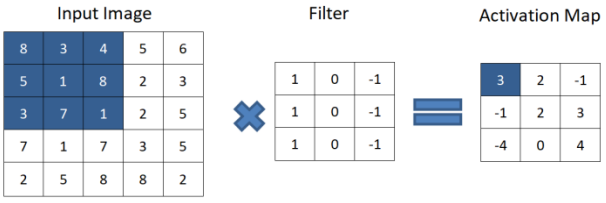

Fig. 1. A graphical example of the convolution process

Pooling layer

A pooling layer is used between two convolutional layers. Pooling Layers are commonly used to minimize the dimensions of the input feature map. It reduces the number of parameters for models to learn further. It also reduces computational work, such as reducing the number of input and output parameters required in a neural network. There are two main types of pooling, but max-pooling is widely used in

Sindh Madressatul Islam University Karachi, Pakistan.

CNN [3]. In Max Pooling layer maximum value is selected from pooling region [4]. In the Average Pooling Layer average value is selected from the pooling region.

Stride

It is the component of the CNN. It has the ability to compress an image and video data. The purpose of stride is to modify the total length of a pixel of an image into small, compressed images without any loss of the features of an image as shown in Figure. 2. For example, if any neural network has a stride equals to 1 as a value, then it means the applied movement of the neural network will direct one pixel forward. It is recommended that we assign the stride value as a whole integer rather than setting it to a decimal value [5].

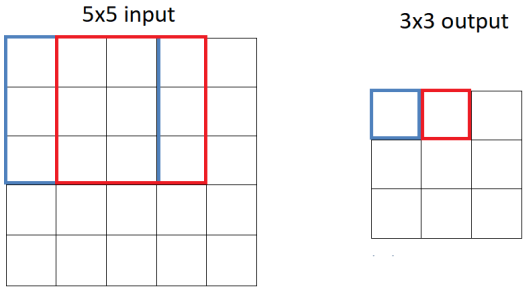

Fig. 2. Stride compression mapping

Padding

Padding is the addition of pixels to a given image when the neural network kernel is processing the given image [6].

Dense Layer

Firstly, the data of the image is read by convolutional layers and then the image is passed through pooling layers. After the data is passed through pooling and convolutional, layers, the output of the image is transferred to dense layer. Output from the convolutional layer is multi-dimensional shape and it is the main reason that we should not pass multi-dimensional image to the dense layer because the dense layer only accepts 1-D shape of image.

Flatten Layer

We usually call Flatten method to convert multi-dimensional data into 1D array. To do this we revoke the Flatten () method between convolutional Layers and the dense layers.

Batch Normalization Layer

Batch layer is also used in deep neural networks as a latest toolkit from many practitioners working on deep learning projects. Batch layer is a significant layer in the deep learning. It is used in architecture as a linear block. It also helps to normalize the network during the training session [7].

Activation Function

In Neural Network (NN), the aggregation of the weights of inputs is transformed to the next node of the network. The capability and performance of the neural networks are affected by the choice of different activation functions. In NN we have basic three layers, input layer, hidden layer, and output layer. We apply the activation function on the hidden or output layer to normalize data. In the hidden layer we use activation functions namely ReLU, Sigmoid and Tanh [8]. And in the output layer we use the activation function Sigmoid for binary classification and Softmax for multi-label classification.
Convolutional neural network architectures

The researchers have designed time to time different models. This section presents the comparison between these models.

VGG-16

The VGG-16 model is also called Visual Geometry Group and is used in deep learning convolutional neural network. Its performance has been tremendous in image recognition in case of a huge dataset. Its performance directly relies on the 16 or maximum 19 convolutional weighted layers. The more convolutional layers are the deeper the network in the architecture of CNN. In order to have a compressed image, we should apply filters of (3×3) in each convolutional layer from left to right and from upper to lower of the input data.

It is also recommended to provide the dimension size of the RGB image as 224× 224. The size of the padding in the VGG-16 is set to level 1. The next layer after the convolutional layer is max pooling layer but it is not the case that after every convolutional layer is max pooling layer. After three stack of Convolutional layers, one max pooling layer comes up. Pool size of max pooling layer is 2×2 dimension. After the stack of convolutional and max pooling layer, three dense layers followed. The initial two dense layers have ReLU activation functions, and the Last dense layer has Softmax activation function. Dense layer is also called as Fully Connected layer as shown in Figure. 3 [9].

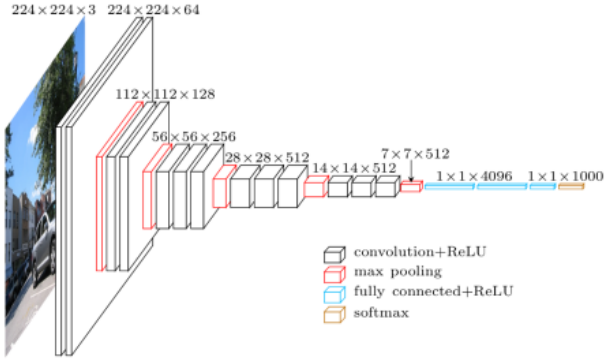

Fig. 3. The architecture of VGG-16[9]

The learning was stopped after 370K iterations (74 epochs) while the deep convolutional network was being trained up.

Deep neural network has large number of parameters that is why more training and epochs are required. VGG-16 has 1138.4M parameters, the size is 528MB, achieve top-1 accuracy 71.3%, and top-5 accuracy is 90.1%. VGG-16 has 16 layers for which 11 are weighted layers. Time inference step in CPU is 69.5ms and in GPU 4.4ms [9-10] [13].

ResNet50

ResNet can be thought as special case of highway network. It is stands for residual network with 50 or more than 50 layers in depth. It is used in object detection and image recognition. There is a common trend in the research community to make the model's architecture go deeper without any effect on its efficiency and performance. However extremely deeper neural network leads to overfitting. Deep neural networks are considered as hard to train. In ResNet Residual block connect with few intermediate layers in feed forward direction called shortcut connections.

ResNet have many stack building blocks called as "Residual Units". "Identity Shortcut Connection" is a method to skip one or more Residual Units, this idea is introduced in ResNet. It is implemented in ResNet to skip double and triple layers. These layers may be Conv Layer, ReLU and Batch Normalization Layer. ResNet use Shortcut connection, but Shortcut Connection is not used by ResNet. Similar idea has been used in Long Short-Term Memory (LSTM).

This model inspires by the Philosophy of VGGNet. The convolutional layers in ResNet50 have 3×3 filters. Two rules are followed; Rule one is, for same size of output feature maps have the same number of filters used. Rule two is, if output feature map size is half, then filter size is doubled. Down sampling performed on the convolutional layer have stride 2. Batch Normalization is used after each convolutional layer and before the activation. This network is used Batch Normalization to ensure forward propagation have non-zero variance. Training done on ResNet from scratch has been required 60 × 1040 iterations. Weight decay used 0.0001 and momentum used as 0.9. Dropout is not used in this network. The network is connected with average pooling layer. The network ends with a fully connected layer with 1000 classes by using the SoftMax function, as shown in Figure. 4.

ResNet50 has 25.6 million parameters. Size is 98Mb, and it achieves top-1 accuracy 74.9%, and top-5 accuracy 92.1%, Time inference step in CPU is 58.2ms and in GPU 4.6ms [11-13] .

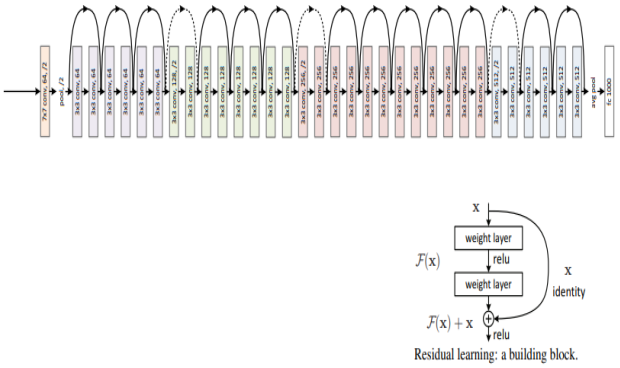

Fig 4: Architecture of ResNet[11]

Xception

Xception stands for "Extreme Inception" and its Architecture. Xception is inspired from Inception architecture. To understand Xception it is needed to understand Inception first. Inception module is developed to make a process more effective and easier. Inception architecture consists of repeating module and connects with linear stack. In the inception module, the same input passes to different transformations and combines all the output.

Xception contains depth wise independent convolutional layers. It is inspired from Inception architecture. There are two major differences in both modules: operation order and non-linearity. Xception has residual connection with linear stack of depth wise separable convolutional layers. Xception has 22.9 M parameters. Size is 88MB, achieve top-1 accuracy 79.0%, and top-5 accuracy 94.5%, Time inference step in CPU is 109.4ms and in GPU 8.1ms[14-15].

Dense Net121

Recent study shows that if a layer contains shorter connection between layers at input and layers at output it increases performance and accuracy. In DenseNet each layer is connected with preceding layers in network in feed-forward fashion. Each layer in dense layer takes additional input from preceding layers, and passes its own features to next layers. Considering n layer has n features of all previous convolutional layers, its output is passed to next layers. Instead of creating very deep architecture, DenseNet increases network potential by its reusable feature for improved efficiency and information flow between the layers of the DenseNet introduces direct connections of any layers with preceding layers as shown in Figure.5 [16].

DenseNet121 has 8.1 M parameters. Size is 33MB, achieve top-1 accuracy 75.0%, and top-5 accuracy 92.3%, Time inference step in CPU is 77.1ms and in GPU 5.4ms.

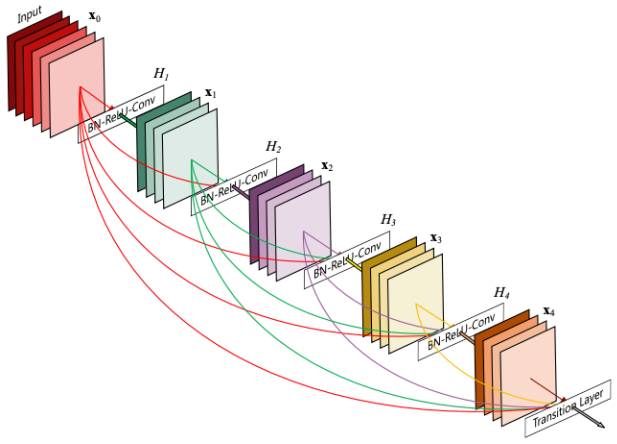

Fig. 5. Architecture of DenseNet[16]

RESULT AND DISCUSSION

This section shows the results of different CNN architecture. In Table I, we compare different CNN models in terms of size, top-1 and top-5 accuracy, parameters, depth, CPU time and GPU time.

Table1. COMPARISON OF DIFFERENT CNN ARCHITECTURES [13]

Models	Size (MB)	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Parameters in (Millions)	Depth (layer)	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4

CONCLUSION

The performance of the CNN is higher while classifying digital images and videos. If a number of weighted layers increase in a CNN model it resulted in the higher level of accuracy. If more weighted parameters are passed to the CNN, then it consumes more time for CPU and GPU during the training session of the models. It can also be concluded that very deep neural network with 1000 or more layers leads to overfitting.

The Xception model shows better result than VGG-16, ResNet50 and DenseNet121 because it passes same input to depth wise isolated blocks and later merges the output of these blocks as input for the final classification layer. VGG-16 uses stack of multiple convolutional layers (weighted layers). Stack of Convolutional Layers increases size of the model for storing learning weights of the deep neural network. The training session time in VGG-16 is higher because of many learning parameters, but it does not affect its performance. Unlike VGGNet ResNet uses Residual Block. Each block is connected in the forward direction with a short connection. Each Residual Block has convolutional layer, Batch Normalization layer and RelU Activation Layer. Residual block repeats in ResNet Architecture stride size vary in each block. In DenseNet output of each previous layer connected with input of next layers. The model sizes are directly proportional to the number of parameters involved which ultimately affects the performance of the models during the object recognition process.

REFERENCES

[1] Wood, Thomas. "Convolutional Neural Network." DeepAI, 14 July 2020, deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network.

[2] Sharma, Anshu, and Poonam Tanwar. "Machine Learning Techniques for Autism Spectrum Disorder (ASD) Detection." International Journal of Forensic Engineering, vol. 5, no. 2, 2021, p. 111. Crossref, https://doi.org/10.1504/ijfe.2021.118912.

[3] "Comparison of Methods Generalizing Max- and Average-Pooling." DeepAI, 2 Mar. 2021, deepai.org/publication/comparison-of-methods-generalizing-max-and-average-pooling.

[4] DeepAI. "Max Pooling." DeepAI, 25 June 2020, deepai.org/machine-learning-glossary-and-terms/max-pooling.

[5] DeepAI. "Stride (Machine Learning)." DeepAI, 25 June 2020, deepai.org/machine-learning-glossary-and-terms/stride.

[6] DeepAI "Padding (Machine Learning)." DeepAI, 25 June 2020, deepai.org/machine-learning-glossary-and-terms/padding.

[7] . "Batch Norm Explained Visually — How It Works, and Why Neural Networks Need It." Medium, 6 Jan. 2022, towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739.

[8] Doshi, Ketan. "Batch Norm Explained Visually — How It Works, and Why Neural Networks Need It." Medium, 6 Jan. 2022, towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739.

[9] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[10] Thakur, Rohit. "Step by Step VGG16 Implementation in Keras for Beginners." Medium, 11 Dec. 2021, towardsdatascience.com/

step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

[11] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[12] Feng, Vincent. "An Overview of ResNet and Its Variants - Towards Data Science." Medium, 21 June 2018, towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

.

[13] Team, Keras. "Keras Documentation: Keras Applications." Keras.Io, keras.io/api/applications. Accessed 3 May 2022.

[14] Akhtar, Zuhaib. "Xception: Deep Learning with Depth-Wise Separable Convolutions." OpenGenus IQ: Computing Expertise & Legacy, 3 Mar. 2021, iq. opengenus.org/xception-model. Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[15] Saleem, Muhammad Asif. "Comparative Analysis of Recent Architecture of Convolutional Neural Network." Www.Hindawi.Com, 31 Mar. 2022, www.hindawi.com/journals/mpe/2022/7313612.

[16] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[17] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014, pages 818–833. Springer, 2014

[18] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015.

[19] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[20] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In Proceedings of The 32nd International Conference on Machine Learning, 2015.

[21] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In Computer Vision–ECCV 2014, pages 184–199. Springer, 2014.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[23] Cimpoi, M., Maji, S., and Vedaldi, A. Deep convolutional filter banks for texture recognition and segmentation. CoRR, abs/1411.6836, 2014

[24] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of The 32nd International Conference on Machine Learning, pages 448–456, 2015.

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015.

[26] Kattenborn, Teja, et al. "Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing." ISPRS Journal of Photogrammetry and Remote Sensing, vol. 173, 2021, pp. 24–49. Crossref, https://doi.org/10.1016/j.isprsjprs.2020.12.010.

[27] ROBINSON, PIERS. "The CNN Effect: Can the News Media Drive Foreign Policy?" Review of International Studies, vol. 25, no. 2, 1999, pp. 301–09. Crossref, https://doi.org/10.1017/s0260210599003010.

[28] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[29] 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.

[30] K. Jin, "Handwritten digit recognition based on classical machine learning methods," 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), 2022, pp. 163-173, doi: 10.1109/IWECAI55315.2022.00040.

[31] Chhajro, M. Ameen. "Handwritten Urdu Character Recognition via Images Using Different Machine Learning and Deep Learning Techniques." Indian Journal of Science and Technology, vol. 13, no. 17, 2020, pp. 1746–54. Crossref, https://doi.org/10.17485/ijst/v13i17.113.