A Review of Convolutional Neural Networks

Arohan Ajit
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India
1706211@kiit.ac.in

Koustav Acharya
School of Electrical Engineering
KIIT Deemed to be University
Bhubaneswar, India
1703207@kiit.ac.in

Abhishek Samanta
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India
1728169@kiit.ac.in

***Abstract*— Before Convolutional Neural Networks gained popularity, computer recognition problems involved extracting features out of the data provided which was not adequately efficient or provided a high degree of accuracy. However in recent times, Convolutional Neural Networks have attempted to provide a higher level of efficiency and accuracy in all the fields in which it has been employed in most popular of which are Object Detection, Digit and Image Recognition. It employs a definitely algorithm of steps to follow including methods like Backpropagation, Convolutional Layers, Feature formation and Pooling. Also this article will also venture into use of various frameworks and tools that involve CNN model.**

***Keywords*— Convolutional Neural Networks, Deep Learning, Digit Recognition**

I. Introduction

Since the beginning, the basic idea behind working of Neural Networks is that it is to mimic the working of human brain to the highest degree possible. Convolutional Neural Network contributes to this by working with the visual sensory organs of the living beings and in process recognizing various types of object be it Digit, Image or a particular action in any object using a string of various techniques followed in a particular order i.e. Convolutional Operation, ReLu Layer, Pooling, Flattening, and Softmax Cross Entropy.

The first step towards developing CNN was taken when a research paper regarding visual cortices of monkeys and birds was published by Hubel and Wiesel. Then in 1980s, convolution process was introduced in field of CNN by Kunihiko Fukushima named neocognitron which was inspired by work of Hubel and Wiesel. However it was Yann Le Cunn who played a major role in bringing CNN to the level it has reached today when he developed a 7 level convolutional network called LeNet-5 using back propagation and adaptive weights for various parameters. All the major architectures present today are different versions LeNet-5.

II. Strcuture Of Each Layer: Explained

A. Convolutional Layer

Convolution Layer is the most basic but at the same time most important layer in the CNN. It basically convolves or multiplies the pixel matrix generated for the given image or object to produce an activation map for the given image.

The main advantage of activation map is that it stores all the distinguishing features of a given image while at the same time reducing the amount of data to be processed. The matrix with which the data is convolved is a feature detector which basically is a set of values with which the machine is compatible. Different versions of image are generated using different values of feature detector. The convoluted model is also trained with backpropagation in order to ascertain minimal error in each layer. According to the lowest error set, depth and padding is set.

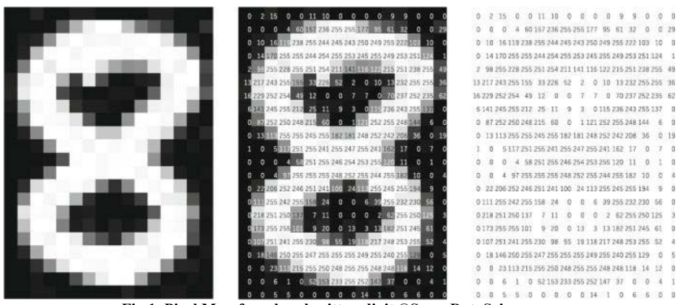


Fig 1. Pixel Map for a handwritten digit ©SuperDataScience

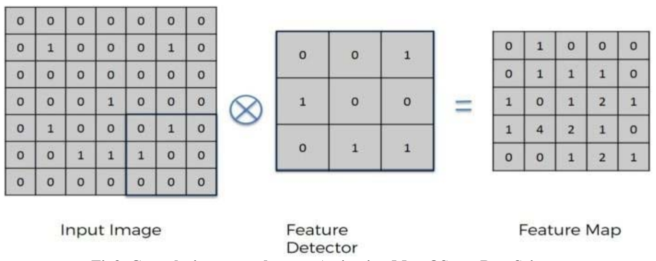


Fig2. Convolution to produce an Activation Map ©SuperDataScience

Fig. 1 above shows how convolution works. This step involves convoluting the matrix containing the image data and then feature detector which gives us an activation map or a feature map. What happens in convolution is that the values on identical positions in the data and feature map i.e. values having value 1 or more than 1 are kept while rest are removed. The matrix from the image data is compared 3x3 at a time. The size of feature detector varies with the type of CNN used. For example there are versions of CNN which use 5x5 or even 7x7 scale filters for convolution. Convolution follows

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

which aims to show how one function modifies the shape of the other. Explaining the above image, the data generated for this image was modified using the filter to generate the activation map. This was we create many feature maps using various filters to generate activation maps which together form a convolution layer.

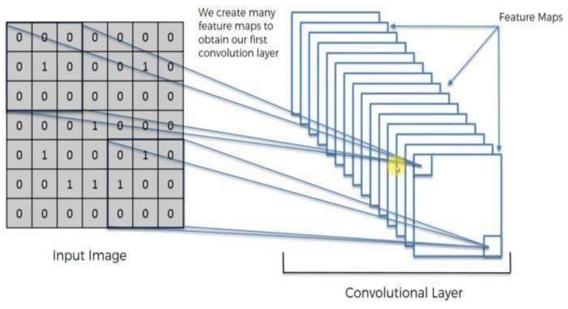

We create many feature maps to obtain our first convolution layer
Feature Maps
Input Image
Convolutional Layer

Fig. 3. Convolution layer ©SuperDataScience

B. *Pooling*

Pooling is an important step to further reduce the dimensions of the activation map, keeping only the important features while also reducing the spacial invariance. This in turn reduces the number of learnable features for the model. This helps to resolve the problem of overfitting. Pooling allows CNN to incorporate all the different dimensions of an image so that it successfully recognises the given object even if its shape is skewed or is present at a different angle. There are various types of pooling like max pooling, average pooling, stochastic pooling, spatial pyramid pooling.Out of them most popular is max pooling.

Max Pooling

Max Pooling takes the highest value from each sub matrice of the activation map and forms a seperate matrice from it. Doing this ensures that the learnable features remain limited in number while also preserving the key features of any image. Max Pooling is usually done using a 2x2 filter.

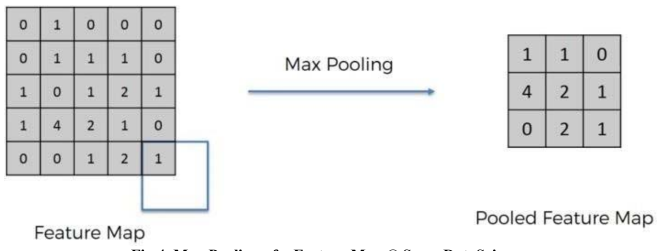


Fig 4. Max Pooling of a Feature Map © SuperDataScience

C. *Fully Connected Layer*

This is the final layer which is feeded to the neural network. Generally matrice is flattened before getting passed on to the neurons. It is hard to follow data after this point due to presence of lot of hidden layer with variable weight for output of each neuron. All the reasoning and computation on data is done here.

III. How CNNs Work

A. *Input (The Training Data)*

Input layer is represented in terms of 3 dimensions i.e. width, length and height. It is commonly denoted as width*height*depth which is pixels for the image shown in form of a matrix. For example. If input is (64x64x3) then width: 64px, height: 64px, depth: 3px. Depth is mainly used to represent color pictures in form of RGB. Commonly Input layer is even and can be divided by 2 multiple times.

B. *Filter*

Also called as kernels or feature detectors. Feature detection uses commonly uses a small matrix. For example, in ConvNet, first layer is of dimension 5x5x3.

C. *Convolved Feature*

Also called Activation Map or Feature Map. A filter of defined is moved through the image taking dot product of each sub matrix producing the Output volume.

D. *Receptive Field*

Region of input matrix similar to size of filter.

E. *Depth*

The number of filters.

F. *Depth Column*

Also called fiber, these are a group of neurons that point towards the same receptive field.

G. *Stride*

Sets the amount of distance stride will shift after each convolution. Having a larger stride gives a smaller output volume. For example a filter having a stride 2 will shift 2 columns after each convolution. It is necessary to set stride in such a way that the output volume is an integer. Smaller strides are used for better results.

H. *Zero – Padding*

Adding zeroes on the border of the input volume so as to maintain the size of input volume and output volume. Not doing so will result in loss of information on border of the image and dimension reduction leading to low performance,

I. *Parameter Sharing*

If a certain feature is common in many images, then it is beneficial to look for that feature in all images which is Done through parameter sharing. However it is an uncommon practice. For example, in a face detection, there is no need to for facial features if the face is positioned at same location in each image.

J. *Dilation*

Dilation is filters which have spaces in its cells. If, we have one dimension filter W of size 3 and an input X:
Dilation of 0: w [0]*x [0] + w [1]*x [1] + w [2]*x [2].
Dilation of 1: w [0]*x [0] + w [1]*x [2] + w [2]*x [4].

K. *ReLU Layer*

ReLU layer is used to apply activation function element-wise which converts all the negative values to zero Therefore converting the threshold to zero. It does not affect volume or hyper parameters.

L. *POOL Layer*

Pool Layer performs a function to reduce the spatial dimensions of the input, and the computational complexity of our model. Pool layer function reduces the dimensions of the input which helps in computational complexity of the model, it helps to control overfitting. It operate independently on every slice of the input. Some commonly used POOL are average pooling, max pooling and 1-2 norm pooling. Below is an example of a Max pooling with 2x2 filter and stride = 2. So, for each of the windows, max pooling takes the max value of the 4 pixels.

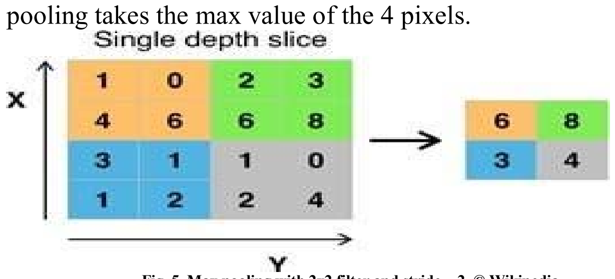


Fig. 5. Max pooling with 2x2 filter and stride = 2. © Wikipedia

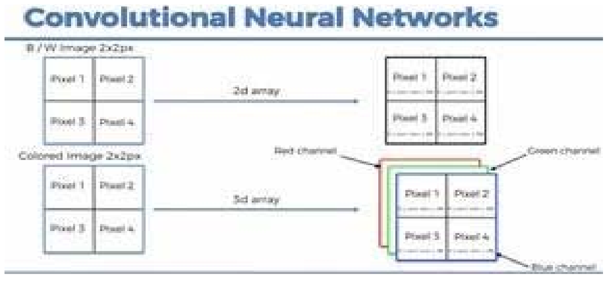


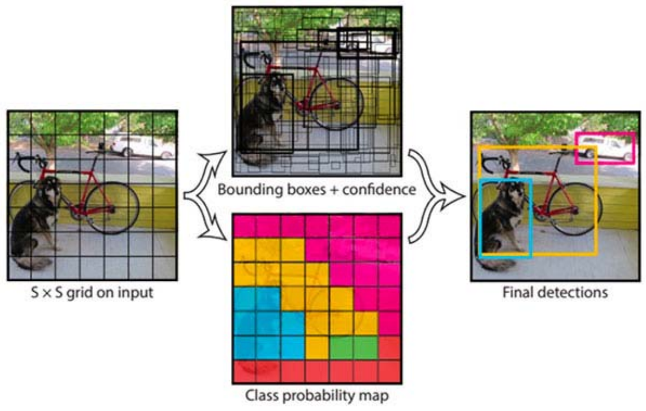
Fig: 6. How a neural network sees a real-world image. ©SuperDataScience

IV. CNN Framework

CNNs helps to modify local features, in order to learn context. Non-static representations are used, therefore, weights for these vectors are learned during the training phase. After training a fully connected layer followed by softmax layer is applied to output final results.

A. *You Only Look Once (YOLO) – Unified Real time Object Detection*

Joseph Redmond has explored the field of object Detection. Before this, approaches to object detection had proposed two stages- one for localizing and other to detect. This caused a disconnection.



Fig, 7. YOLO Object Detection © https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

B. *Single Shot Detector (SSD)*

R-CNN was developed for object detection, localization and classification with its output being a group of bounding boxes. However, the training is slow and consumes multiple phases.

Single shot detector is faster than R-CNN. A single forward pass is needed to localize and classify an object. Multibox loss is used compute the error for SSD. Finally, a detector is used for classification of detected objects. Object detection by SSD consists of:

1. Extracting feature maps, which is done using VGG16
2. Applying features to detect objects.

Architecture:

The SSD model uses a feedforward convolution network producing a network of bounding boxes and the presence of object class instances in those boxes, followed by non- max suppression. An auxiliary structure is built upon base layers to produce detection.

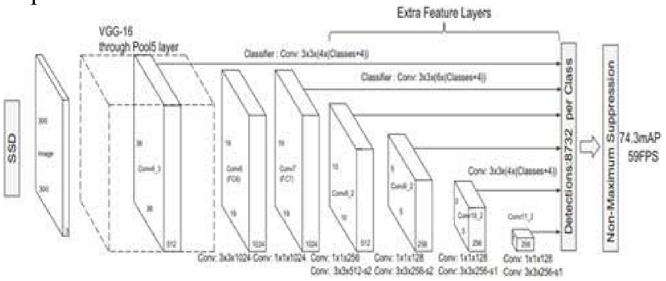


Fig. 8 The SSD Architecture

C. *Artistic Style Transfer using Neural Nets*

Leon A. Gatys has shown an artificial system based on neural network that creates artistic images using convolutional neural networks. This system uses neural representation to separate and recombine different aspects of an image like style and content. Convolutional Neural Networks, when trained on object recognition, develop a recreation of that image that makes information increasingly explicit along the processing.

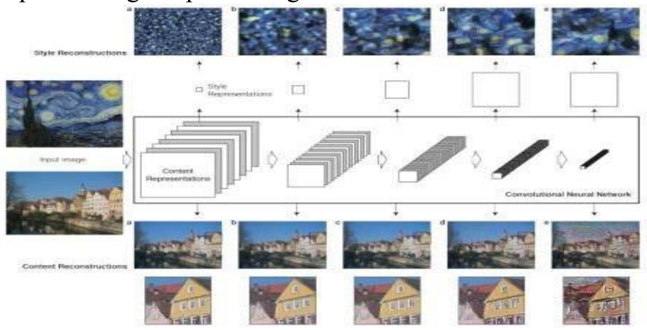


Fig. 9. Content reconstruction using CNN

Style Reconstruction

A new feature is built on the top of the original network that captures the style of the given input image. This representation finds the correlations. Among the different features in the different layers of the network. This helps in creating images which matches the style of a given image on an increasing scale while disregarding information of the general global arrangement of the scene.

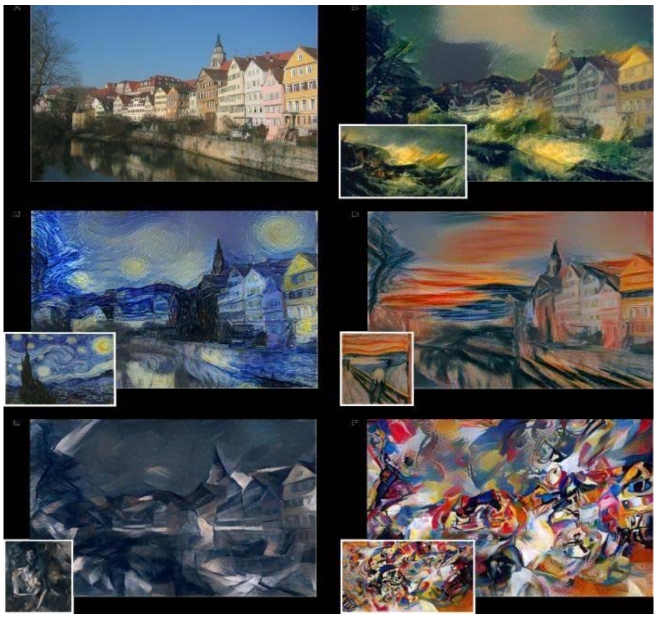


Fig 10. Images that have successfully combined the content of a photograph in the style of well-known artworks. It is shown in

A. (Photo by: Andreas Praefcke).
B. The Shipwreck of the Minotaur by J.M.W. Turner, 1805.
C. The Starry Night by Vincent van Gogh, 1889.
D. Der Schrei by Edvard Munch, 1893.
E. Femme nue assise by Pablo Picasso, 1910.
F. Composition VII by Wassily Kandinsky, 1913.

Fig. 11. Results for the style of the painting composition VII by Wassily Kandinsky.

V. Issues

A. Translational Invariance

Convolution in itself is translation equivariant. However, due to dimensionality reduction by virtue of pooling, the CNN architecture does achieve translation invariance.

B. Spatial Relationships

Orientations and relative position of objects in an image are not considered well in a CNN architecture, which does not take into account the rotational and translational relationship. Max pooling has been able to solve that problem, nevertheless, at the cost of loss of valuable information

VI. Architectures

A. LeNet

LeNet is one of earliest CNN which was mainly used for digit recognition. This is still considered one of the most important work in field of digit recognition with 18261 citations. This architecture published in 1998 was the ones that kickstarted development of CNN and subsequently Deep CNN for digit recognition based on MNIST database. General architecture for this network includes applying convolution(5x5) on input and subsequent average pooling(2x2) with stride of 2 repeated twice and is finally ended with 2 fully connected layers. The final input that is provided to the FCN is of dimension 120x1x1. No. of parameters taken into account are approximately 60,000.

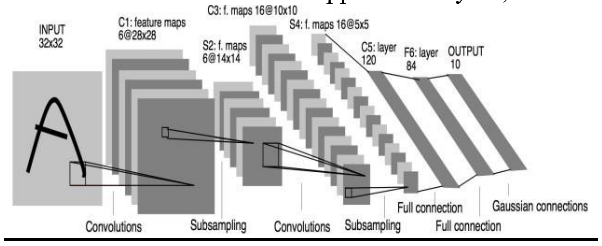


Fig. 12. LeNet ©https://medium.com/@pechyonkin/key-deep-learning-architectures-lenet-5-6fcc59e6f4

B. AlexNet

This architecture was published in 2012 and was among first CNN to win 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge). Input provided was about 15 million RGB image data of dimensions 3x224x224 image (which was later corrected to 227x227) from about 22,000 categories. This Deep CNN consists of 7 layers and 60 million parameters. The architecture of this network in very similar to that of LeNet except it is deeper, The operations consists of 11x11,5x5,3x3 convolutions, 3x3 max pooling ending with 2 fully connected layers of 4096 layers each. In terms of optimization, Layer Normalisation and ReLU activation was used instead of tanh activation as it was found to increase speed by 6 times. To avoid overfitting dropout layers was used, although it increased the training time. The model is divided into 2 parts as the architecture was trained in 2 GTX 580 GPUs for 6 days.

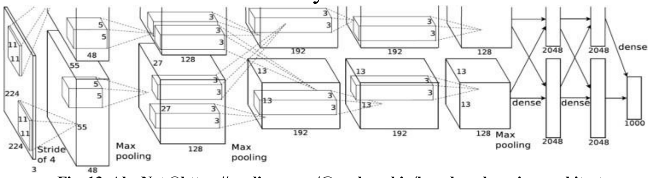


Fig. 13. AlexNet ©https://medium.com/@pechyonkin/key-deep-learning-architectures-lenet-5-6fcc59e6f4

C. VGG(16) Net

VGG stands for Visual Geometric Group. This particular network was introduced in ILSVRC 2014 challenge where although it was a runner up, was widely accepted and appreciated. The design was similar to that of AlexNet which also meant that VGG Net also had a huge number of features. This network contains about 138 million parameters. There are total 16 convolutional layers in VGG 16, distributed in 3 blocks containing 2 layers of 3x3 convolutions followed by 2x2 max pooling and 2 blocks containing 3 layers of 3x3 convolutions followed by 2x2 max pooling. The architecture is finally finished by 2 fully connected layers of 4096 hidden layers each. Some prime advantages of this network is that it provides greater discrimination due to non linearity within the blocks, also computational power required is also less as it processes larger receptive fields in succession of 3x3 convolutions as opposed to processing them directly as it directly reduces the no. of parameters to be computed. One more advantage of having blocks was the ReLU activation could be performed twice in each block after each convolution.

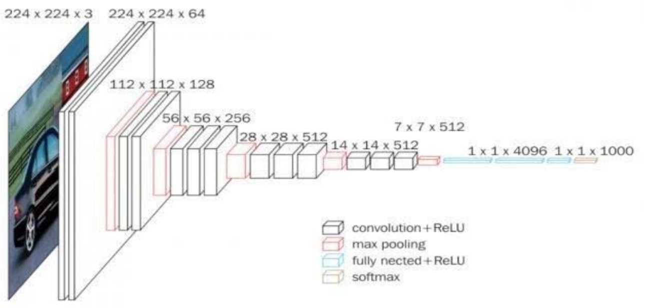


Fig. 14. VGG(16) Net © https://neurohive.io/en/popular-networks/vgg16/

D. GoogleNet

The GoogLeNet also followed the block pattern as followed by VGG 16 and each block is called Inception Module. GoogLeNet uses total of 9 inception module. This architecture won the ILSVRC 2014. This network 22 layers deep but compensated by relatively much lesser no. of parameters of about 5 million compared to runner up VGG's 138 million parameters. Each layer has multiple filter sizes (1X1, 3X3, 5X5 convolutions) and which filter is to be updated was decided by backprop. Each module also contained a 3x3 max pooling layer. Each convolution layer was preceded by a bottleneck layer of 1x1 convolution in order to reduce the depth of feature map which in turn reduces no. of parameters to be computed significantly. In Order to optimise results, 1x1 convolutions were followed by ReLU activation. Instead of fully connected layer, Global Average Pooling was used in this architecture at the end basically taking average of each individual feature map.

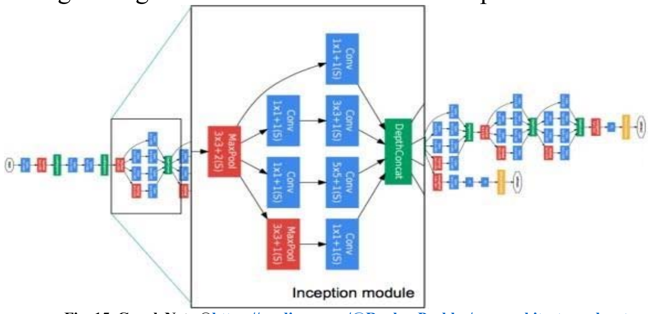


Fig. 15. GoogleNet –©https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet- and-resnet-7c81c017b848

E. Microsoft ResNet

ResNet or Deep Residual Network was the winner for ImageNet challenge for 2015 surpassing the human accuracy error for the first time with an error rate of about 3.6%. The network is very very deep and the one presented at challenge was 152 layers deep. It was observed that as network depth increased accuracy got saturated and then further on it degraded rapidly, not only that with deeper network team also had to solve the Vanishing Gradient Problem. In Order to counter this, ResNet introduced a feature of skip connection. The concept behind renet was that even if the network was depp, the training of the network was similar to that of shallow network by skipping after every 2 layer. In order to compute, the input and output both were copied to the next layer basically learning the residual of previous computation. No. of parameters computed were about 65 million. Some layers also have bottleneck starting and ending by 1x1 convolution. Batch Normalization was used after each convolution

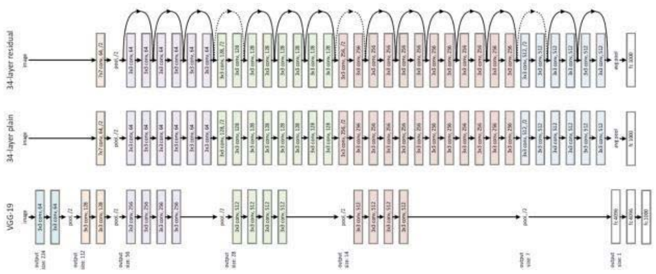


Fig 15. ResNet © https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification- localization-detection-e39402bfa5d8

VII. References

[1] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, Kaori Togashi: Convolutional Neural Networks : an overview and application in radiology(2018), Insights into Imaging(2018) 9:611-629. Available at: https://doi.org/10.1007/s13244-018-0639-9

[2] Neena Aloysius, Geetha M: A Review on Deep Convolutional Neural Networks, International Conference on Communication and Signal Processing(2017)

[3] Domink Scherer, Andreas Muller, Sven Behnke: Evaluation of Pooling Operations in Convolutional Architecture for Object Recognition(2010). Available at: http://www.ais.uni-bonn.de

[4] Wei Liu1(B) , Dragomir Anguelov2, Dumitru Erhan3, Christian Szegedy3, Scott Reed4, Cheng-Yang Fu1, and Alexander C. Berg: SSD: Single Shot MultiBox Detector

[5] "A Neural Algorithm of Artistic Style" by Leon A. Gatys, Alexander S. Ecker, Matthias Bethge.

[6] Key Deep Learning Architectures: LeNet-5 - https://medium.com/@pechyonkin/key-deep-learning-architectures-lenet-5-6fc3c59e6f4

[7] https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html

[8] A Walk-through of AlexNet – Hao Gao - https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637

[9] CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more - https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

[10] GooglNet - Artificial Intelligence - https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/googlenet.html

[11] Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 (Image Classification) - https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7

[12] Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image… - https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

[13] Review: ResNet—Winner of ILSVRC 2015 (Image Classification, Localization, Detection) - https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8

[14] Microsoft Presents: Deep Residual Networks - https://medium.com/@bakiiii/microsoft-presents-deep-residual-networks-d0ebd3fe5887