ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse196952020.pdf

https://doi.org/10.30534/ijatcse/2020/196952020

# A Comprehensive review of Transfer Learning on Deep Convolutional Neural Network Models

**Aswathy Ravikumar[1], Harini S[2]**

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
aswathy.ravikumar2019@vitstudent.ac.in

[2] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
harini.s@vit.ac.in

## ABSTRACT

In deep learning the most significant break though in the field of image recognition, language processing was done by Convolutional Neural Network (CNN). The CNN is a powerful algorithm which is capable of using features extraction at multiple levels and this is done based on the distribution of data automatically. The rapid growth in Big Data and the high-performance architectures have made it possible for the growth of CNN. Transfer Learning is the method for performance improvement in the specific problem with the transferring of knowledge learned from other domains. Transfer learning have a wide range of applications and it is widely used in deep learning algorithms There is rapid growth in the use of transfer learning in the pre trained models in CNN. The transfer learning helps to improve the performance of the models and it helps in saving time by acting as an optimization technique. This survey attempts to provide a comprehensive review in understanding the mechanisms of transfer learning and the pretrained models in CNN.

**Key words:** Neural Network, Machine Learning, Transfer Learning, Deep Convolutional Neural Networks.

## 1. INTRODUCTION

The deep learning is mainly used in the image and pattern recognition applications like the autonomous driving car projects, driver assistance, medical diagnosis [32] and the detection of the target [1]. The main principle behind the deep learning algorithms are the neural network architecture [2]. The neural network can be CNN (Convolutional Neural Networks), Pre trained network models, Recurrent Neural Networks, Unsupervised pretrained networks like Generative Adversarial Networks, Deep belief networks, Auto Encoders etc. Complex problems can be solved effectively using deep architectures which works on the principle of combination of multiple processing units to perform different layer of feature abstraction. Studies have proven that with enough amount of training data the CNN is able to create effective representation of the data and have good performance in solving the problems. Lately the transfer learning help in the transferring of the features learned from one domain to another. CNN have undergone many modifications in the recent years in terms of architecture, methods, and complexity. CNN is the widely used deep learning architecture for image recognitions and it works based on the features extraction at different levels using convolutional filters. RNN is used for the sequential or time series data prediction, audio and video analysis with the feed forward neural networks with the capability to deal with the hierarchical sequences in the data. CNN is the algorithm that is widely used for image detection, classification and retrieval applications. Due to the high possibilities of the algorithm many companies like Microsoft, Google, Face Book are developing the new models of CNN. The main computer vision-based image processing applications are now using CNN based models which make use of both the temporal and spatial relations in the data. CNN consist of many layers consisting of the convolutional, pooling, fully connected layers. Convolution is the process of feature extraction from the data automatically and these features are used to learn the data distribution and this eliminates the need of a feature extractor step. The main advantages of CNN are the multitasking capability, automatic feature extraction using convolutional filters and the weight sharing. In CNN the weight training is done using back propagation algorithm. The CNN is capable of extracting the low, mid-level, high level (combination of low and mid-level) features and this is the deep learning similar to the human brain capable of processing the data from the features automatically. The CNN became widely accepted when AlexNet showed high performance ImageNet challenge in 2012.

Transfer Learning is the transferring of knowledge from one domain to another for solving similar problems. Transfer learning is based on the generalization theory which was used in psychology. This theory requires a connection between the events. It depends on the new event whether the transferred knowledge is having a positive impact or not. The negative transfer [3,4] is the case in which the knowledge transfer has a negative impact on the new domain. The negative transfer depends on the target and source domains and domain access of the knowledge [5]. Based on the domain discrepancy there are the homogeneous (domain with the same feature vectors and slight difference in the distribution) and heterogeneous transfer learning (domains are of different feature vectors with distributions that need feature adaptions) [6]. The models in deep learning can be shared with the other applications in terms of weights and parameters. The few famous pretrained models in CNN are VGG-16/9, Inception V3, ResNet-50 etc.

## 2. DEEP CONVOLUTIONL NETWORK

Convolutional Neural Networks are the feed forward neural networks used mainly for the image recognition tasks, optical character recognition and the natural language processing of handwritten documents and can be used for content analysis using the convolutional networks for diagrams. The advancement of CNN has led to the use in many applications like Robotics, drones, self-driving cars, security system and medical field. The CNN perform picture analysis as tensors with numbers and additional dimensions. The feature extraction in CNN is done automatically at different levels [10,11,12] using the concept of visual perception [13]. The kernels in the CNN are used for the different features and the function stimulation is done using the activation functional at each level. The CNN is superior to ANN in terms of the local connections which helps to reduce the number of parameters and helps to make the model convergence faster, efficient weight sharing which further helps to reduce the number of parameters, dimensionality reduction using down sampling and all these features make CNN the most appealing algorithm in deep learning. CNN have mainly four steps – Convolution (for feature extraction), Padding (Image adjusting), Striding (to avoid overfitting [14]), Pooling [15] (max and average pooling) and Fully connected layer.

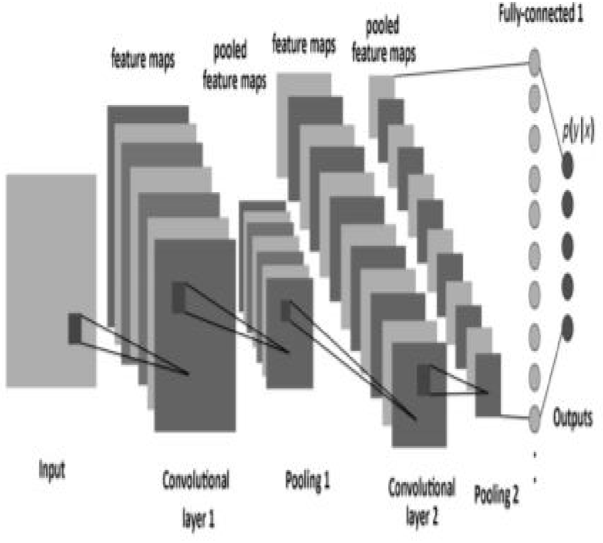

**Figure 1:** CNN Architecture

The Convolutional layer is a combination of multiple convolutional kernals which divides the image into multiple small parts to make the feature extraction process faster. Convolution is expressed as

$$f_l^k(p,q) = \sum_c \sum_{x,y} i_c(x,y).e_l^k(u,v)$$

$i_c(x,y)$ is the input image tensor $I_c$ elemnts multipled with the convolutional kernal $e_l^k(u,v)$ to obatin the feature map. The different feature maps are obatined using the different kernals and based on the number and size of the convolutional filters and direction in which it I applied the convoution operation varies.

Pooling layers have the feature vectores obatined from the convolutional layers which perform the down sampling by summing up the similar data in the neighborhood of the required field and the most prominent response is given as output .

$$\mathbf{Z}_l^k = g_p(\mathbf{F}_l^k)$$

$Z_l^k$ is the feature map after pooling of the $l^{th}$ layer and $k^{th}$ input feature and g reprentw the pooling function . It is used for dimensinality reduction while the most relevant features are not lost. There are different types of pooling like maxmium ,sum and average pooling . The decision making function is the activation function which is useful in understanding the hidden patterns and the appropriate activation function makes the model faster and improves the performance. ReLu is the most commonly used activation function in the CNN which is a non linear function that is used in mapping the output to a postive value or zero and the negative values are mapped to zero this ensures the non linearity in the CNN which is more relistic. The covarience shift that occurs in the feature maps due to the hidden parmeters change in distribution are solved using the batch normalization.Generalization can be improved by dropout which skips random units based on a fixed probabilty to avoid overfitting in the neural network . The final fully connected network is used to perform the classification and the activation function used is Softmax for classification to high order features.

## 3. CNN MODELS

The CNN have undergone evolution over the time for better performance and faster training. The rapid growth in big data and the high-performance computing facility paid way to the research and development of CNN architectures. The 2012-ILSVRC marked the beginning of the CNN architectures growth with the AlexNet which reduced the error rate to a great extent. The parameter optimization helps to reduce the computational complexity and improve the performance. The new architectures were formed by varying the number of layers, filters and complexity of the model. The main difficulty in designing the new CNN models are the degerming the filter dimension, padding and each layer hyperparameters. The uniform modular design is preferred over customized layer due to the ability to use it for other applications easily

### 3.1 LeNet5

LeNet5[19] is the CNN put forward in 1998 which has seven layers (5 convolutional and pooling layers and two FCN layer) used for grouping digits in the grey scale images 32x32 pixel size for object detection. It uses the logic of the neighboring pixels being connected to each other and the features are distributed. It helped to reduce the parameter count and automatic feature extraction from pixels It plays with different convolutions using maximum pooling with different tasks and the final layer and associated layers are interfaced by the final convolutional layer in the. Padding may or may not be done but the size of each frame is reduced by 4 after every convolutional filter and pooling is done. This architecture was initially used for feature extraction using the spatial patterns which were in common from Brain Computer interface [18] but it has a drawback the accuracy was low than the original CNN network. Multiple locations with similar features can be easily extracted using the convolutions done using the learnable parameters. The sharable parameter learning helped to change the individual pixel view without correlation.

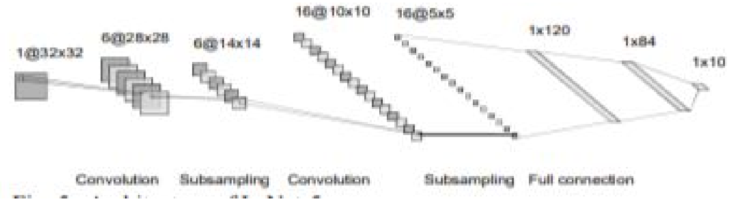

**Figure 2**: LeNet

### 3.2 AlexNet

The main drawback of the LeNet is it fails for other problems other than handwritten digit recognition. The first model to perform well for image recognition task in multiple domains. In AlexNet the deeper optimization is done using multiple layers. The main limitation faced by the networks are the lack of hardware computational power. The NVIDIA GTX 580 GPU was used to train the AlexNet. It is 8-layer CNN for better generalization of the pictures with different resolution. AlexNet faced the challenge of overfitting due to the depth and it was later addressed using the dropout of transformational units in the training phase to ensure the all features are learned and normalization of the local responses. It faced the vanishing gradient issue which was rectified using the ReLU activation function

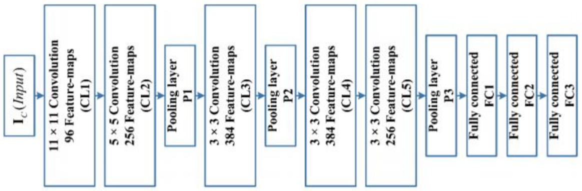
I_C (Input)
11 × 11 Convolution 96 Feature-maps (CL1)
5 × 5 Convolution 256 Feature-maps (CL2)
Pooling layer P1
3 × 3 Convolution 384 Feature-maps (CL3)
Pooling layer P2
3 × 3 Convolution 384 Feature-maps (CL4)
3 × 3 Convolution 256 Feature-maps (CL5)
Pooling layer P3
Fully connected FC1
Fully connected FC2
Fully connected FC3

**Figure 3:** AlexNet

### 3.3 ZFNet

Multiple layer Deconvolutional Neural Network which was introduced by Zeiler and Fergus in 2013. The main motivation for designing this network was for the visualization of network performance by the analysis of the activation of the neurons in the CNN. Earlier in 2009 Deep Belief Network was optimized using the same idea of hidden layer visualizing. In 2011 the Autoencoder was evaluated using the classes in the last layer of the network. In Deconvolutional Neural Network the convolutional and pooling operations are reversed whereas all other steps remain the same as a forward pass CNN. This idea was implemented in AlexNet and it was found only few neurons were active and the others were inactive in the initial two layers and showed aliasing artifacts. To overcome this the parameter optimization was done in CNN and the stride and filter size was reduced to increase the learning rate. The visualization of features helped to improve the performance of CNN and the identification of the drawbacks in the design and helps to correct it.

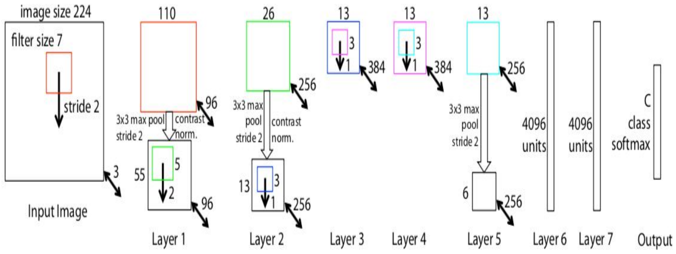

**Figure 4:** ZFNet

### 3.4VGGNets

Visual Geometry Group proposed a 16 convolutional layer CNN algorithm VGGNet [22] which had many variants like VGG-11, VGG-13, VGG16, VGG-19 etc. The ImageNet challenge was won by VGGNet in 2014 and it was shown as depth of the network increases the performance is increased till an extent. VGGNet removed the LRN layer and it used mainly the multiple 3x3 convolutional filters for the feature extraction. VGG is widely used in many practical applications like Blind-Image Quality Prediction [23], Biometric Authentication, Crack Detection etc.

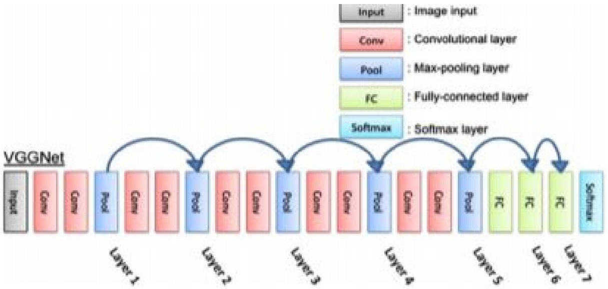

**Figure 5:** VGGNet

### 3.5 GoogLeNet /Inception Net

GoogLeNet [27] won the ILSVRC in 2014 and it was based on the principle of large-scale stacking of Inception units. The main versions of the Inception network are the Version1 [27], Version2 [28], Version 3 [29], and Version4 [30]. The patterns were recognized using the Inception module. In Inception v1 the images with higher data concentration needs bigger convolutional kernel but the training of larger kernels is difficult since the number of parameters is large. Inception v1 uses 1 × 1(for reducing computational cost), 3 × 3, 5 × 5 convolution kernels for the generation of the wide network structure. In Inception v2 batch normalization is used to overcome the covariate shift. The high learning rate training of the model is increased using the normalization if the normal distribution of each layer. Inception v2 showed the use of the medium sized wider feature maps makes the model better for high dimensional feature representations. Inception v3 included factorizing of factorizing 5 × 5 and 3 × 3 kernels into two single dimensional filters for improving the training time of the network. RMSProp is used for optimization in Inception v3. Inception v4 is the best among the all versions in terms of performance

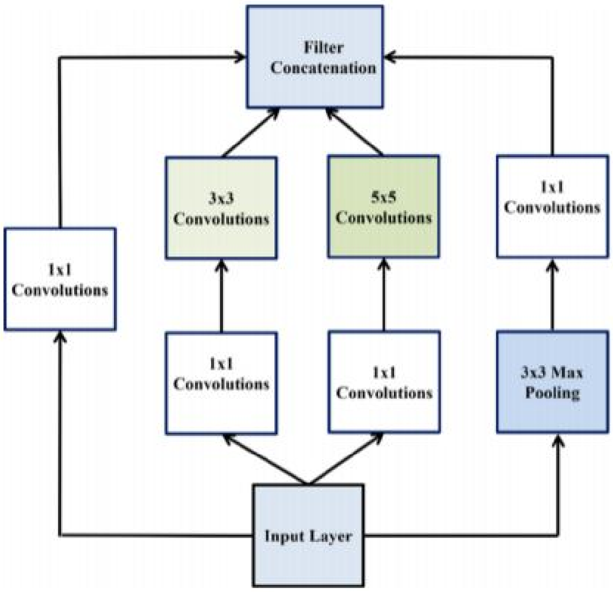

**Figure 6:** GoogleNet

### 3.5ResNet

Batch Normalization and "skipconnections" also known as gated units made the main base of Residual Neural Network [24] developed in 2015. The skip layer connection helps the network have better performance and faster training and it removes the reflections. Starting from the initial 34 layer the extended versions were having 50-layer, 101 layer and 152-layer ResNet with three-layer residual blocks. The vanishing gradient problem was mitigated to an extent in ResNet without the degeneration of the network. Advanced versions of ResNet like stochastic depth ResNets, pre-activation ResNet , ResNet in ResNet and wide ResNet were developed.

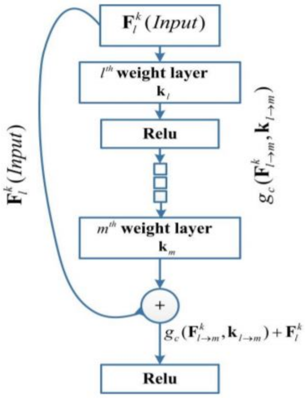
$\mathbf{F}_l^k\,(Input)$
$l^{th}$ weight layer $\mathbf{k}_l$
Relu
$m^{th}$ weight layer $\mathbf{k}_m$
+
$g_c(\mathbf{F}_{l\to m}^k, \mathbf{k}_{l\to m}) + \mathbf{F}_l^k$
Relu
$\mathbf{F}_l^k\,(Input)$
$g_c(\mathbf{F}_{l\to m}^k, \mathbf{k}_{l\to m})$

**Figure 7:** ResNet

### 3.6MobileNet

The light weight networks embedded in mobile phones developed by Google using the depth wise convolutions. The MobileNet has mainly three versions MobileNet v1, MobileNet v2, and MobileNet v3. MobileNet v1 uses width multiplier for reducing the resolution of the images inputted. MobileNet v2 have the inverted residual blocks (opposite of the normal residual block) and the linear bottleneck modules. And in this model, it is applied to each input channel separately. Both depth wise and point wise convolution are applied to the training dataset and the optimum parameter are considered and the light weight neural net is developed. MobileNet v3 has the NetAdapt (for layer wise search) and excitation based lightweight attention model, activation function h-swish and platform specific neural architecture-based search NAS which implement block wise search. It is widely used in many practical applications like smartphones-based solutions for medical diagnosis [31], medical analysis.

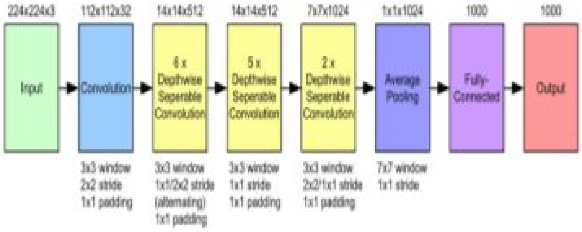
224x224x3
112x112x32
14x14x512
14x14x512
7x7x1024
1x1x1024
1000
1000
Input
Convolution
6 x Depthwise Seperable Convolution
5 x Depthwise Seperable Convolution
2 x Depthwise Seperable Convolution
Average Pooling
Fully-Connected
Output
3x3 window 2x2 stride 1x1 padding
3x3 window 1x1/2x2 stride (alternating) 1x1 padding
3x3 window 1x1 stride 1x1 padding
3x3 window 2x2/1x1 stride 1x1 padding
7x7 window 1x1 stride

**Figure 8:** MoileNet

### 3.7 ShuffleNet

The ShuffleNets[26] were developed to solve the lack of computing power in mobiles using various steps like shuffling channels, convolutions. There are mainly two version ShuffleNetv1 and ShuffleNetv2.These networks highly reduced computational complexity by using pointwise group convolutions of single shuffle. In ShuffleNet v1the different color pointwise group convolutions were used to represent the different groups and the channel shuffle operation is done after the initial feature is obtained .TO reduce the parameters used in the network the ReLU activation function is neglected and the concatenation operations are preferred over addition in the pooling stages. ShuffleNet v2 brought in the idea of FLOPs dominating the network and Memory Access Cost to be considered for measuring the speed of the network.

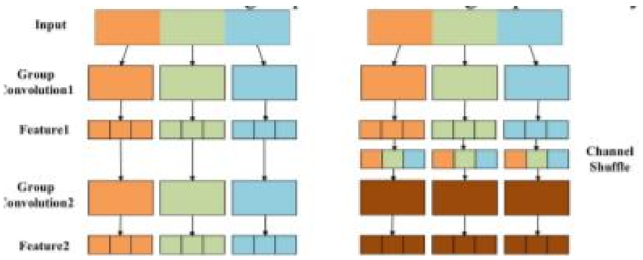

**Figure 9:** ShuffleNet

### 3.8 GhostNet

The GhostNet [25] was successful in reducing the computational cost by removing the redundancy in the features extracted in the convolutional layers. The similar feature maps extracted by the CNN are known as Ghosts. The initial convolutional layers were used for getting the feature representations and they work like normal convolutions. These features are later processed as linear transformations to obtain the multiple feature maps.

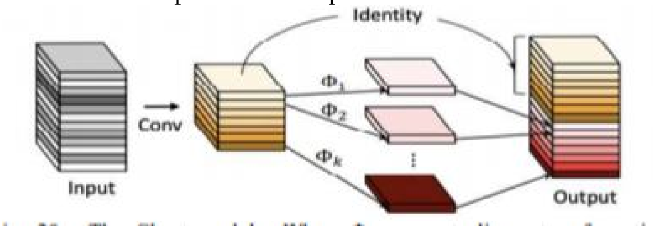

**Figure 10:** GhostNet

### 3.9 DenseNet

DenseNet connects the layers in a feed forward manner. DenseNet have direct connections of layers with same feature size and used for high level scaling with no optimization difficulties. It was success to remove the vanishing gradient problem. The features obtained the previous layer acts as input of the current layer. DenseNet strengthened the feature propagation, it helps in feature reuse and helps to reduce the number of parameters in the network.

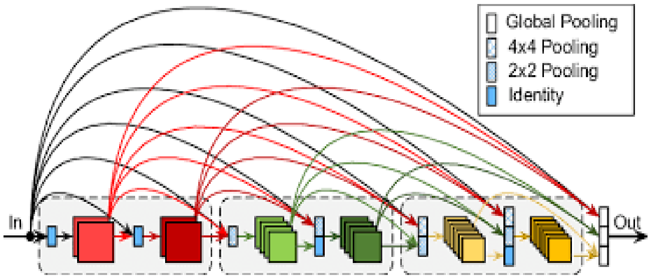

**Figure 11:** DenseNet

## 4. TRANSFER LEARNING

Knowledge transfer from one domain to another known as Transfer Learning used for the many machine learning tasks and feature extraction. Deep learning has been widely benefited by the transfer learning to increase the overall performance of neural networks. For the efficient working of a neural network model it needs a huge amount of data and transfer learning gives the ability to adapt to different data sources when the training data available is small. Deep learning models learn intermediate features better when transfer learning from different domains are applied since the generalization is more effective then. It was found from studies that the features acquired from the bottom layers are general and are more specific when acquired from top layer. The transfer learning faces challenges like the task specific top layers and the network splitting makes optimization strategies difficult. But despite these problems the transfer learning showed better performance than building a model from scratch. CNN models trained using large dataset can be used for other domains using the transfer learning. The CNN models are trained using the ImageNet and they can be later fine-tuned for the new domain specific dataset which is helpful in making the model more efficient and easier to develop. The low-level features are learned in the initial layer and the higher-level features in the high-level layers. In the transfer learning in CNN the final layers mainly the fully connected layer and the output layer used for classification is modified as per the domain. Transfer learning can be used for deep feature extraction where fine tuning is avoided and the feature vectors are extracted using the activation function in CNN [20]. The initial layer activation functions provide the low-level feature representations and the deeper layers provide more higher-level feature representation that can be further used for the classification.

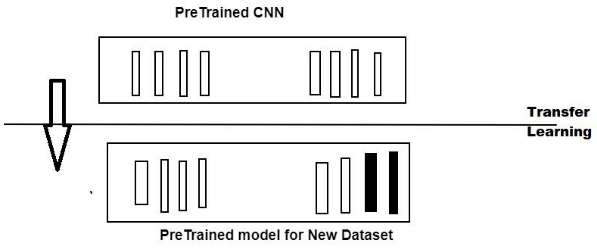

**Figure 12:** Transfer Learning

## 5. CONCLUSION AND FUTURE SCOPE

CNN due to weight sharing capability, dimensionality reduction, local connections made it very useful in solving many practical real-world applications. In this paper a detailed review of the most commonly CNN models based on the activation functions, learning algorithm, optimization and regularization strategies and network structure. The CNN structure was discussed in detail and how the transfer learning can be used for make use of the pretrained models for the new application domains for better accuracy and performance. The models can be further refined and expanded in terms of model size, depth of the network and hyper parameters tunning. Lack of generalization ability, equivariance are some of the problems faced by CNN. The comparative study was conducted for identifying the powerful CNN models and it was found that the ResNet outperformed VGGNet in Image net Challenge. The transfer learning in CNN can be further improved by doing distributed processing of the models in parallel in the future.

For future research in image processing field the new CNN architecture designs are developed. Ensemble learning in CNN with combining different diverse architectures can be implemented in the future for better performance and acceleration of the models. In the future the better method for normalization and optimization can be incorporated into the models. The generative learning capacity of CNN need to be explored further for better feature extraction. Spatial relevance of the images together with the features need to be explored in future since it is highly related to the attention mechanism of our visual system. For better learning capacity of the network need to be enhanced both in terms of size and hardware computing facilities like FPGA for reducing the power and training time. More work needs to be done in the hyper parameter tunning which is mainly intuition driven method. Pipeline Parallelism can be used for scaling the CNN without hyper parameter tunning. Distributed transfer learning can be done for CNN in the future to reduce the training time.

## REFERENCES

1. Srinivas S. S. Kruthiventi , Kumar Ayush , R. Venkatesh Babu. **DeepFix: A Fully Convolutional Neural Network for Predicting Human Eye Fixations**, *IEEE Transactions on Image Processing*, Vol. 26(9), pp. 4446 - 4456, Sept. 2017
2. Zhang Bo, Zhang Ling, **The analysis and improvement of artificial neural network models**, in *IEEE International Conference on Intelligent Processing Systems* (Cat. No.97TH8335), Oct. 1997.
3. D.N. Perkins and G. Salomon, **Transfer of Learning**. Oxford, England: Pergamon, 1992.
4. S.J. Pan and Q. Yang, **A survey on transfer learning**, *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
5. Z. Wang, Z. Dai, B. Poczos, and J. Carbonell, **Characterizing and ´ avoiding negative transfer,** in Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 11293– 11302.
6. K. Weiss, T.M. Khoshgoftaar, and D. Wang, **A survey of transfer learning**, *J. Big Data,* vol. 3, no. 1, Dec. 2016.
7. J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, and B. Sch¨olkopf, **Correcting sample selection bias by unlabeled data,** in Proc. *20th Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2006, pp. 601–608.
8. M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. Bnau, and M. Kawanabe, **Direct importance estimation for covariate shift adaptation,** *Ann. Inst. Stat. Math.*, vol. 60, no. 4, pp. 699–746, Dec. 2008.
9. O. Day and T.M. Khoshgoftaar, **A survey on heterogeneous transfer learning**, *J. Big Data,* vol. 4, no. 1, Dec. 2017.
10. Yusen Zhan and Matthew E Taylor. **Online transfer learning in reinforcement learning domains.** arXiv preprint arXiv:1507.00436, 2015
11. Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. **A simple neural attentive meta-learner**. In NIPS 2017 Workshop on Meta-Learning, 2017.
12. T. Ahonen, A. Hadid, and M. Pietikainen, **Face description with local binary patterns: Application to face recognition,** *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037-2041, 2006.
13. W. T. N. Hubel D H, **Receptive fields, binocular interaction and functional architecture in the cat\"s visual cortex**, *The Journal of Physiology*, vol. 160, no. 1, pp. 106-154, 1962.
14. D. M. Hawkins, **The problem of overfitting**, *Journal of chemical information and computer science*s, vol. 44, no. 1, pp. 1-12, 2004.
15. K. Fukushima, **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**, *Biological Cybernetics*, vol. 36, no. 4, pp. 193-202.
16. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
17. https://medium.com/@sidereal/cnns-architectureslenet-alexnet-vgg-googlenet-resnet-and-more666091488df5
18. Siavash Sakhavi, Cuntai Guan, Shuicheng Yan, **Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks**, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29(11), pp. 5619 - 5629, March 2018.
19. B. L. Lecun Y, Bengio Y, et al., **Gradient-based learning applied to document recognition**,” *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.
20. Orenstein EC, Beijbom O. **Transfer learning and deep feature extraction for planktonic image data sets**. In: 2017 *IEEE Winter conference on applications of computer vision (WACV).* IEEE. 2017

21. A. Krizhevsky, I. Sutskever, and G. Hinton, **ImageNet Classification with Deep Convolutional Neural Networks**, *Advances in neural information processing systems*, vol. 25, no. 2, 2012.
22. K. Simonyan, and A. Zisserman, **Very Deep Convolutional Networks for Large-Scale Image Recognition**, *Computer Science*, 2014
23. Jongyoo Kim, Anh-Duc Nguyen, Sanghoon Lee, Deep **CNN-Based Blind Image Quality Predictor**, *IEEE Transactions on Neural Networks and Learning Systems,* vol. 30(1), pp. 11 - 24, June 2019.
24. K. He, X. Zhang, S. Ren, and J. Sun, **Deep residual learning for image recognition.** pp. 770-778.
25. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, **GhostNet: More Features from Cheap Operations**, arXiv preprint arXiv:1911.11907, 2019.
26. X. Zhang, X. Zhou, M. Lin, and J. Sun, "**ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices.**"
27. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "**Going deeper with convolutions**." pp. 1-9.
28. S. Ioffe, and C. Szegedy, **Batch normalization: Accelerating deep network training by reducing internal covariate shift,** arXiv preprint arXiv:1502.03167, 2015.
29. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, **Rethinking the inception architecture for computer vision.** pp. 2818- 2826.
30. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, **Inception-v4, inception-resnet and the impact of residual connections on learning**
31. J. Velasco, C. Pascion, J. W. Alberio, J. Apuang, J. S. Cruz, M. A. Gomez, B. J. Molina, L. Tuala, A. Thio-ac, and R. J. Jorda, **"A Smartphone-Based Skin Disease Classification Using MobileNet CNN,"** *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, 2019. https://doi.org/10.30534/ijatcse/2019/116852019
32. D.K. Kirange, J.P. Chaudhari, K.P. Rane, K.S. Bhagat, Nandinichaudhri, **Diabetic Retinopathy Detection and Grading Using Machine Learning,** *International Journal of Advanced Trends in Computer Science and Engineering,* Vol 8, No.6, pp. 3570-3576, 2019. https://doi.org/10.30534/ijatcse/2019/139862019