Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549546

PERFORMANCE OF TRANSFER LEARNING MODELS FOR UNIVARIANT HAND WRITTEN CHARACTER RECOGNITION

B.Meena
Research Scholar, Department of CS &SE, AUCOE(A), Associate Professor, CSE Department, Raghu Engineering College, Dakamarri, Visakhapatnam,
meenabhagavathula100@gmail.com

K.Venkata Rao
Professor, Department of CS & SE, AUCOE(A), Andhra University, Visakhapatnam.

Suresh Chittineni
Professor, CSE Department, GITAM University, Visakhapatnam

Abstract—In the field of Deep Learning and Computer Vision, there have been several advancements. These models helped to produce state-of-the-art results on tasks like Image recognition and Image Classification, especially with the advent of extremely deep Convolutional Neural Networks. As a result, Deep Learning Architectures have evolved through time (adding more layers) to tackle increasingly complicated problems, which has aided in boosting Classification and Recognition task performance as well as making them more resilient. Deep Learning models are used to train the model from scratch or also use the existing pre-trained models using Transfer Learning method. Our paper proposes the comparative analysis of pre-trained models using Transfer Learning method for GoogLeNet, ResNet50, ResNet101 and VGGNet models. The models are trained on user defined dataset to recognize hand written digits in Telugu Language.

Keywords— Computer Vision ; Convolution Neural Network ; Hand Written Character; Image Recognition ; Transfer Learning

I. INTRODUCTION

Transfer learning is a Deep Learning technique that uses a model that has been trained for one job as a starting point for a model, that performs a related task. Transfer learning makes updating and retraining a network considerably faster and easier than training a network from the beginning . Object Identification, Image Recognition, and Speech Recognition are just a few of the applications that employ this method. Transfer Learning is popular because of various reasons such as : Use of popular models that have previously been trained on huge datasets to train models with less labelled data . Potential to cut training time and computer resources. The weights are not learnt from scratch with transfer learning since the pretrained model has previously learned them based on earlier learnings. Model Designs established by the deep learning research community, including prominent architectures like GoogLeNet , VGGNet and ResNet, are available. Figure 1 Depicts the basic flow of Transfer Learning.

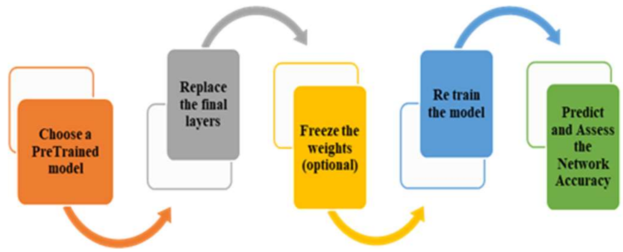

Fig 1 . General Work flow of Transfer Learning

There are several pre-trained models available, each with its own set of benefits and drawbacks. Any pre-trained model is selected depending on the model size and its usage in various environments, network size , accuracy , prediction speed . Prediction speed is affected by other factors such as hardware , batch size and also the architecture of the chosen model and its size . The Two most prevalent deep learning methodologies to train the model are Training a model from scratch and Transfer learning . For very specialized jobs where prior models cannot be used, developing and training a model from scratch works better. The disadvantage of this strategy is that proper results usually need a big amount of data. For example, for Text Analysis, pretrained models cannot be accessed but do have a big number of data samples, building a model from scratch is probably the best option here. Transfer learning is beneficial for problems like object identification, where there are a number of popular pretrained models. For example: To perform Image Classification with fewer datasets , An AlexNet , GoogLeNet , VGGNet or ResNet or any other pretrained network can be used the model can be re trained for the user defined datasets by adjusting the output layers and few parameters to reach the required efficiency . Transfer learning can help to obtain improved model accuracy in less time in certain situations. In Figure 2 , the graph clearly depicts the network performance and training with and without transfer learning.

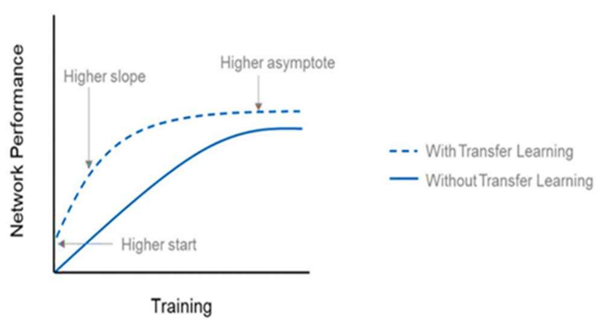

Fig 2: Network Performance with and without Transfer Learning

II. PRE-TRAINED MODELS

A. ResNet : ResNet is a well-known deep learning model that was first introduced in a paper by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang. ResNet is one of the most widely used and effective deep learning models to date . learning models to date. The introduction of these Residual blocks alleviated the challenge of training very deep networks, and the ResNet model is built up of these blocks. ResNet allows the model to train hundreds or even thousands of layers while still achieving excellent results.

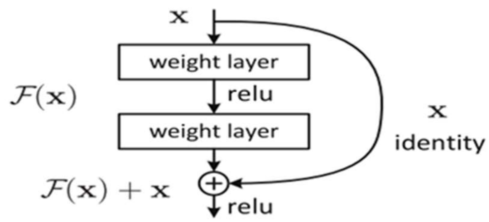

Fig 3 : Residual Block

Many computer vision applications other than classification of images have improved as a result of its great representational capacity, such as object identification and face recognition. ResNet's main concept is to provide an "identity shortcut link" that bypasses one or more levels, as indicated in the diagram below. Many computer vision applications other than picture classification have improved as a result of its great representational capacity, such as object identification and face recognition[1]. The network takes as input image size 224 x 224 by 3 . The proposed model consists of a Deep Convolution Neural Network model (Pretrained ResNet-50) was fine-tuned in this study, and transfer learning was used to identify the images. Additional layers include drop out, fully linked, and a SoftMax activation layer with a size of 2 output layer for the two classes of data for classification. The existing architecture after fourth stage is modified by adding new layers a drop out, fully connected layer, softmax layer and a classification layer. The details of Datasets and Results are show in section 4 and 5 respectively. The initial layer is an input layer that takes images of size 224 * 224 * 3 size , this images are applied 7 X 7 Convolution with striding 2 and a max pool layer is applied . In various stages the convolution operations are applied.

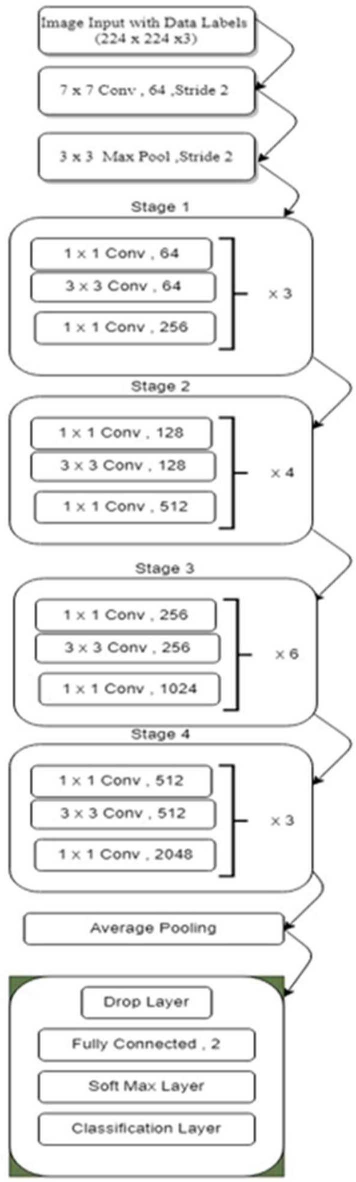

Fig 4. ResNet Layers

B. GoogLeNet :

The initial version of Inception Models, GoogLeNet, was first suggested and won the 2014 ILSVRC (ImageNet Large Scale Visual Recognition Competition). It was first published as a research paper in 2015 at the IEEE Conference on Computer Vision and Pattern Recognition[2] (CVPR), and it has been referenced 34,784 times. GoogLeNet is a 22-layer deep convolutional neural network built by Google researchers as a variation of the Inception Network, a Deep Convolutional Neural Network. 11 convolutions in the center of the design and global average pooling are used in this architecture in comparison to preceding state-of-the-art models like AlexNet and ZF-Net, GoogLeNet is drastically different. It uses a range of methods, including the necessary deep architecture techniques like 1x1 convolution and global average pooling, to produce deeper architecture.

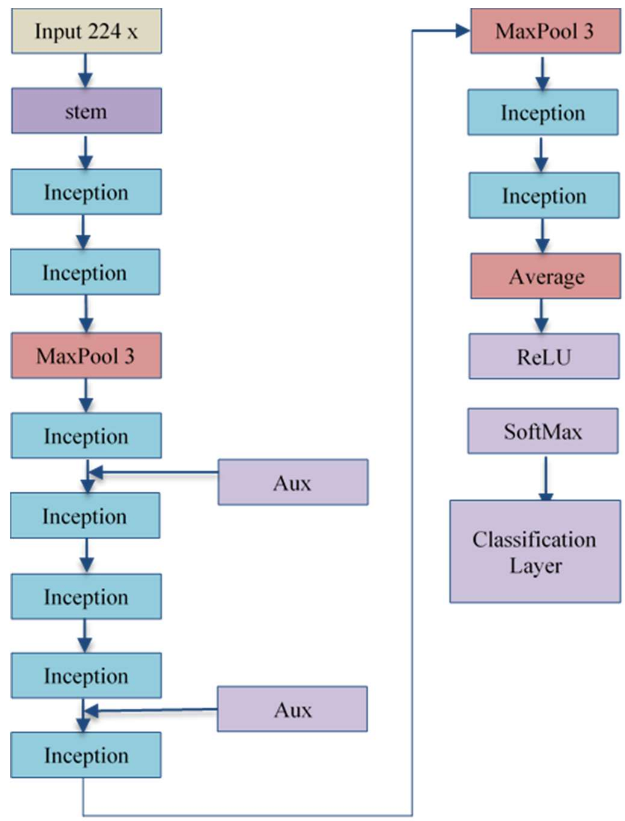

Fig 5. GoogLeNet Layers

Average Pooling: In earlier systems like AlexNet, the entirely connected layers are used at the network's end. Many designs incorporate the majority of their parameters in these fully linked layers, which raises the cost of calculations. A method called as global average pooling is used to wrap up the GoogLeNet architecture. A 7x7 feature map is averaged down to a 1x1 feature map in this layer. Additionally, this results in the elimination of all trainable factors and an increase in top-1 accuracy of 0.6 percent. In contrast to preceding systems like AlexNet and ZF-Net, the inception module is unique. In this architecture, the convolution size is fixed for each layer. The 1x1, 3x3, 5x5, and 3x3 max pooling are carried out in parallel at the input of the Inception module, and the outputs are stacked together to obtain the final result. Theoretically, convolution filters of different sizes will be better able to handle objects of different scales.

C. VGGNet :

In 2014, Karen Simonyan and Andrew Zisserman from the University of Oxford devised the convolutional neural network architecture known as VGGNet. A 224*224 size image of RGB is used as the input for VGG-based ConvNet. In the preprocessing layer, mean image values that are computed for the whole ImageNet training set are subtracted from the RGB image with pixel values in the range of 0-255.

These weight layers are applied after preprocessing the input photos. A stack of convolution layers is applied to the training pictures. The VGG16 architecture consists of 3 fully linked layers and a total of 13 convolutional layers. Instead of using huge filters, VGG uses smaller (3*3) filters with more depth. It ultimately had the same effective receptive field as if there had simply been one 7 × 7 convolutional layer. The first two layers are convolution layer using 3*3 filters, and the first two levels have 64 filters, producing a volume of 224*224*64 since the identical convolutions are used in both layers. Filters are indeed 3*3 with a 1 stride. The volume's height and breadth were then reduced from 224*224*64 to 112*112*64 by using a pooling layer with a max-pool of 2*2 size and stride 2. The next two convolution layers have 128 filters each. The new dimension as a consequence is 112*112*128. Volume is decreased to 56*56*128 once the pooling layer is employed. Following the addition of two further convolution layers with 256 filters each, the size is decreased to 28*28*256 using down sampling. A max-pool layer separates two more stacks, each of which has three convolution layers. 7*7*512 volume is flattened into a Fully Connected (FC) layer with 4096 channels and a softmax output of 8 classes after the last pooling layer.

At each level or stack of the convolution layer, the number of filters we may employ doubles. This fundamental idea guided the development of the VGG16 network's architecture. The fact that the VGG16 network is a large network and requires more time to train its parameters is one of its major drawbacks. The depth and quantity of completely linked layers in the VGG16 model make it larger than 533MB. This lengthens the process of putting a VGG network into practice. Many deep learning image classification issues employ the VGG16 model, however smaller network topologies like GoogLeNet and SqueezeNet are frequently used. In any event, because it is so simple to implement, the VGGNet is a fantastic building block for learning purpose. The VGGNet accepts 224224-pixel images as input. To maintain a constant input size for the ImageNet competition, the model's developers chopped out the central 224 x 224 patch in each image. The convolutional layers of the VGG use the smallest feasible receptive field, or 33, which nevertheless catches up, down, and left to right movement. A linear transformation of the input is also performed via 11 convolution filters. A ReLU unit, a significant advancement from AlexNet that shortens training time, is then presented. ReLU, which stands for rectified linear unit activation function, is a piecewise linear function that, in the event that the input is positive, produces the input value; otherwise, it produces zero. To ensure that the spatial resolution is maintained after convolution, the convolution stride is kept at 1 pixel (stride is the number of pixel shifts over the input matrix).

ReLU is utilized by each hidden layer [7] in the VGG network. Local Response Normalization (LRN) is often not used with VGG as it increases memory use and training time. Furthermore, it doesn't increase overall accuracy.

Three completely interconnected layers make up the VGGNet. The first two levels each have 4096 channels, while the third layer has 1000 channels with one channel for each class.

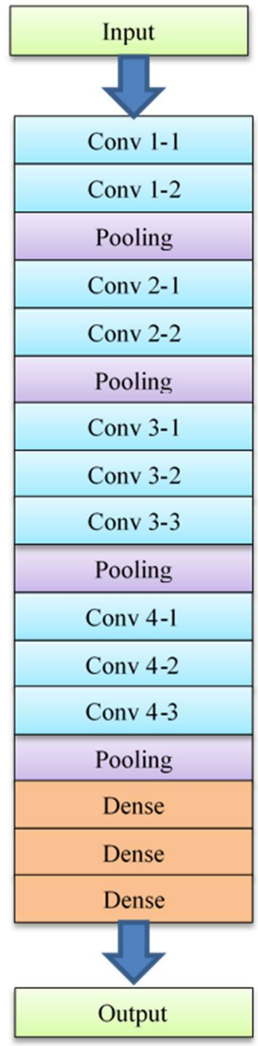

Fig 5. VGGNet Layers

III. Data Collection :

The model is trained with user created datasets initially with 8 different classes having the image input size 224 x 224 x 3 . The images are hand written images collected from different age groups. Each class has 400 images. The model is trained with 3200 images. . The sample datasets are show in the Fig 5.

Fig.6.Telugu Vowels and Consonants
(Hand Written Samples used in Dataset)

IV. Results :

The below table and the graph show the test accuracy of ResNet50 , ResNet101 , GoogLeNet and VGG19 .It is observed that ResNet 101 has shown a better result .

Model	Accuracy (%)	Elapsed Time	Hardware source	Learning rate	No.Of epochs	Iterations per epoch	Max iterations	Frequency
ResNet 101	91.25	18 sec	Single GPU	0.01	5	2	10	30 iterations
GoogLeNet	87.5	14 sec	Single GPU	0.01	5	2	10	30 iterations
ResNet 50	86.25	12 sec	Single GPU	0.01	5	2	10	30 iterations
VGG19	85	14 sec	Single GPU	0.01	5	2	10	30 iterations

Fig 7 : Test Accuracy Table

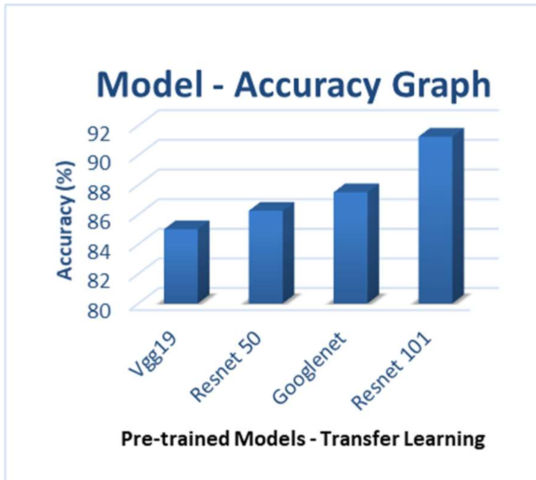

Fig 8. VGGNet Layers

V. Conclusion:

Transfer learning is a well-known concept that can be used to both classic machine learning and deep learning applications. Transfer learning has several advantages, including cost savings and increased efficiency when training new models. These models can be used to improve the deployment efficiency. Models can be trained in simulations rather than in real-world settings. For our datasets ResNet101 with few modifications using Transfer Learning has shown better results than the other models VGGNet, GoogLeNet. Further this Process can be extended to identify all the telugu characters of various vowels and consonants. Further this methodology and fine tuning of Transfer Learning is recommended for various Image Processing, Image Recognition and Classification techniques.

VI. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun , Deep Residual Learning for Image Recognition - Dec 2015

[2] Karen Simonyan Andrew Zisserman , Very Deep Convolutional Networks For Large-Scale Image Recognition (ICLR 2015)

[3] Prasad, S. D., & Kanduri, Y. (2016, September). Telugu handwritten character recognition using adaptive and static zoning methods , IEEE - 2016

[4] Jagdessh Babu V,Prashant L,Raghunath Sharma R,Prabhakar Rao G.V, Bharath A - HMM-based Online Handwriting Recognition System for Telugu Symbols – 2007

[5] Lakshmi, C. V., Jain, R., & Patvardhan, C. OCR of printed Telugu text with high recognition accuracies. In Computer Vision, Graphics and Image Processing (pp. 786-795). Springer, Berlin, Heidelberg – 2006

[6] Koppula, V. K., & Negi, A. (2011, September). Fringe map based text line segmentation of printed Telugu document images. In Document Analysis and Recognition (ICDAR) - 2011

[7] Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers, Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique

[8] T.V. Ashwin and P.S. Sast ry. Font and size independent OCR for printed Kannad,a documents using SVM classifier. In Proc.of Open Research Forum, Fifth ICDAR, pages 14-18. 1999.

[9] R.C. Gonzalez and R.E. Woods. Digital Image Processing.Addison-Wesley, 1993.

[10] R. Kasturi and S. N. Srihari (Eds.). Design and implementat ionof a Teluguscript recognit ion system. 2000 Proc. Fifth ICDAR.IEEE Computer Society Press, Los Alamitos,CA, 1999.

[11] G. Nagy, S. Seth, and M. Vishwanathan. A prototype documentimage analysis system for technical journals. Computer, 25(7),1992.

[12] Atul Negi, B. Phani Kumar, and B.K. Trinadh.Offline printedKannada script recognition system.In Int. Workshop onPerformance Evaluation Issues in Multiligual OCR.1999.

[13] A. V. S. R. L. Prasanna. Layout analysis of Telugu documentsusing RLSA method. Technical report, Dept . of Computer andInformat ion Sciences, University of Hyderabad, Hyderabad,India, 2000.

[14] M.B. Sukhswami, P. Seetharamulu , and Arun K. Pujari.Recognition of Telugu characters using neural netwroks. Int. J.of Neural Systems, 6(3):317-357,1995.

[15] O.D. Trier, A.K. Jain, and R.Taxt. Feature extraction methodsfor character recognit lon-a survey. Pattern Recognition,29(4):64 1-662, 1996.

[16] K. Wong, R. Casey, and F. Wahl. Document analysis system.
IBM J. Research and Development, 26(6), 1982.

[17] G. E. Hinton, R. R. Salakhut dinov, " Reducing thedimensionality of dat a with neural networks," Science, vol.313,pp. 504-507, 2006.

[18] K. Yu, L. Jia, Y. Q. Chen, W. Xu, "Deep learning: yest erday,today, and t omorrow," Journal of Computer Research andDevelopment, vol. 50, pp. 1799-1804, 2013.
ht tps://ieeexplore.ieee.org/document/9028977

[19] G. E. Hinton, "A Practical Guide to Training Rest rict edBolt zmann Machines," Lecture Notes in Computer Science,pp.599-619, 2012.

[20] B. David, " Charact er recognit ion using convolutional neuralnetworks," Seminar Stat ist ical Learning Theory Universit y ofUlm, Germany Inst itute for Neural Informat ion Processing, pp.2-5, 2006.
G. E. Hinton, S. Osindero, Y. W.Teh, "A fast learning algorithm for deep belief net s," Neural Computation, vol. 18, pp. 1527- 1554, 2006.

[21] C. X. Zhang, N. N. Ji, G. W. Wang, " Introduction of restricted

boltzmann machines," Sciencepaper Online,Beijing,[22] Casey, R.G. and Lecolinet , E., "A Survey of Methods and Strategies in Character Segment at ion". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, Vol.18, No.8,pp.690-706.

[23] Liang, S., Shridhar, M. and Ahmadi, M., " Segmentation ofTouching Characters in Printed Document Recognition". PatternRecognition,1994, Vol.27, No.6, pp.825-840.

[24] Rajiv K. Sharma and Amardeep S. Dhiman, "Challenges inSegmentation of Text in Handwritten Gurmukhi Script ".Proceedings of BAIP 2010, CCIS 70, Springer-Verlag Berlin Heidelberg, pp. 388–392.[25] Rajiv K. Sharma and Amardeep Singh, " Segmentation of Handwritten Text in Gurmukhi Script ". International Journal ofComputer Science and Security, 2008, Vol.2. No.3, pp 12-17.

[26] Rajiv K. Sharma and Amardeep Singh, " Detection andSegmentat ion of Handwritten Text in Gurmukhi Script usingFlexible Windowing", IJCTE, 2010, Vol 2, No. 3, pp. 329 – 332.

[27] A.Kaur, Rajiv K. Sharma, and Amardeep Singh, " A HybridApproach to Classify Gurmukhi Script Characters".Internat ional Journal of Recent Trends in Engineering, 2010, Vol 3, No. 3, pp 103- 105.

[28] Sukumar Burra1, Amit Patel2, Chakravarthy Bhagvati1 andAtul Negi1, "Improved Symbol Segmentat ion for TELUGU Optical Character Recognition", ISDA 2017.

[29] Li Chen, Song Wang, Wei Fan, Jun Sun, Satoshi Naoi, "Cascading Training for Relaxation CNN on Handwritten Character Recognition", 2016 15th Internat ional Conference on Front iers in Handwriting Recognition.

[30] Raj, Jennifer S., and J. Vijitha Ananthi. "Recurrent neural networks and nonlinear prediction in support vector machines."