# TExNet – Tuning Existential Network for High Resolution Edge Detection

Sreyas S, Student, Department of Computer Science and Engineering, BS Abdur Rahman Crescent Institute of Science and Technology

Zaid Baig A, Student, Department of Computer Science and Engineering, BS Abdur Rahman Crescent Institute of Science and Technology

## Abstract

In video-based object detection for either in self-driving or in video surveillance, the issue we practically face in this area is to deal with very deep convolutional neural networks (CNN). Very deep CNNs does a great job in detecting feature sets at different layers of abstractions, right from the edges (close to the shallow layer) to very complex features of the object (nearing the output layer). So deeper the neural network results in the risk of an issue known as vanishing gradients problem, to tackle this issue the concept of inclusion of residual blocks to the neural network architecture was proposed. This helped give rise to the concept of skip connections which help in preventing the gradients to go to zero during the training process. Another issue with deep CNNs is computational cost, which normally occurs due to the number of hyperparameters which the neural network has to train, with inception network we would be able to reduce the number of hyperparameters by 10 folds if we were to work with a 1x1 convolutional feature set. In this proposed work, TExNet is aimed to incorporate the advantages of both, the Inception model and the ResNet model, and come up with a nested Resnet-Inception model which will be able to produce results with much higher accuracy and can work with much deeper CNN model. It makes use of the concept of concatenating multiple CNN feature set result and adding a residual block on top of it. The architecture of the model will comprise of a nested Residual block which will be applied both, inside the inception cell and outside running parallelly between 2 inception cells. The proposed work is aimed to preserve the most minute features, which may be lost in traditional architecture during the training process and help in increasing the number of class labels it would be able to detect. The proposed model will be tested mainly under GPU computing environment and tested using TensorFlow 1.1x.

## Introduction

There has been a significant rise to the number of neural network architectures which the deep learning community had offered us in the field of computer vision. The most common feature which is present in most of today’s well known neural network architecture is the inclusion of the concept of using convolutional in their respective architecture, for the concept of convolution was first introduced in 1998 in the LeNet architecture where the paper proposed the concept of reducing the image size with the help of filters and extracting primary features like edges, lines and other predominant features of the object to be detected. The LeNet architecture was mainly developed for scanning of hand written documents. The convolutional layer mainly comprised of these feature matrixes that help in extracting the predominant features from the image and leaving behind the lesser ones. This concept of the LeNet architecture inspired the creation of all other neural network architecture which have been made today. The same concept is applied on the AlexNet architecture where it consists of a deeper neural network and helps in recognizing more objects. The Google Net architecture used the same concept of convolutions but instead of computing in a sequential order, the convolutional and reduction phase were done parallelly and the result of those were concatenated and sent to the next step. The overall neural network size of the Google Net architecture was significantly larger than the AlexNet or the VGG network, as VGG-16 just represents a neural network architecture with 16 layers of convolution layers. Now as the introduction of much deeper neural networks are being developed, there is a rising problem of what is known as vanishing gradients. To further explain of what vanishing gradient, as we train a neural network model there are two major steps involved, i.e, the forward propagation step and the backward propagation step. The forward propagation step will initially compute the result of the given input and then this initial result is compared with the original result using a loss function, this function then tells the neural network of whether the parameter values should increase or decrease in accordance to the learning rate model and when the parameters of the neural network. Usually the parameter values of the neural network are mostly between 0 and 1, hence the vanishing gradients issue arises when these parameters go ever so close to 0 that it becomes meaningless when being computed during the forward propagation step and when it hits zero the future training process of gradient descent will become meaningless as there will be zero change in the parameters that have hit zero, as the learning rate to directly dependent on the parameter being non-zero.

## Literature Survey

**Perceiving Motion from Dynamic Memory for Vehicle Detection in Surveillance Videos** - Wei Liu, Shengcai Liao\*, Senior Member, IEEE, and Weidong Hu (2018)

This paper proposed in the improvement of how memory modules can affect the process of object detection in videos.

**Single Image Super-Resolution Using Deep CNN with Dense Skip Connections and Inception-ResNet** - Chao Chen and Feng Qi (2018)

**Deep Residual Learning for Image Recognition** - Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2017)

This paper introduced a new technique or a new tool kit which developers and researchers can use to build an effective neural network architecture. It solves the traditional problems of vanishing gradients by making sure the output from each layer is either greater than or lower than the previous result of the previous activation layer or equal to the result of the previous activation layer.

**Going Deeper with Convolutions** - Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich (2014)

The origin of this paper was inspired by the movie named inception. It proposes a simple twist to the already existing convolutional neural network architecture by combining and running multiple filters parallelly at the same step and concatenating the result and repeating the process. This method had a positive impact on the accuracy of the model and also gave the needed stability to support a very deep neural network architecture. But this result came at the cost of increase in the number of hyperparameters and ultimately increasing the computation cost overall. To bring down the hyperparameters the concept of one by one convolutional filters is used, which brought the number of hyperparameter to be tuned by about ten folds.

**Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning**- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi (2017).

The following paper was aimed at combining the perks of the inception network and the Resnet network and come up with a neural network architecture which is supposedly the best architecture for running very deep convolutional neural network.

**Network in Network** - Min Lin, Qiang Chen, Shuicheng Yan (2014)

This paper proposed an efficient way of computing with gradients and help in the reduction of the number of hyperparameters being used. The main idea was basically to make an unorthodox approach to the traditional method of calculating the result of a convolutional filter. By making use of a one by one convolutional filter the model will not drop performance, but will in turn have a massive improvement in computational cost. A slight note to be taken in this is that, the filter will not help in reducing the size of input layer but for each filter step should be followed by a pooling step which will help in the reduction of the input size.

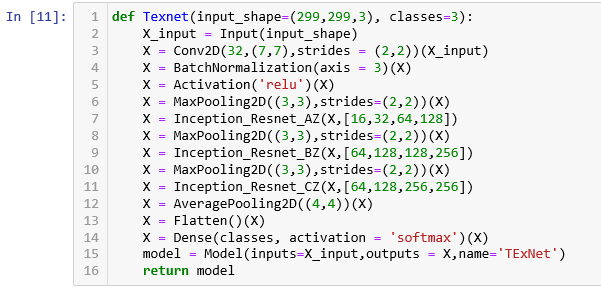
**Gradient Based Learning Applied to Document Recognition** - Yann LeCun Leon Bottou Yoshua Bengio and Patrick Haffner (1998)

This paper is regarded as the beginning of the field of computer vision. It is the first paper that introduced the concept of feature detection using convolutional neural network, where the input is convolved and passed through a learning step involving a forward step and a backward step. The forward step tries to compute and get the result for the given filter value and input, then depending on the result a loss is computed to indicate the direction of change that has to be made on the filter values by a constant learning rate. The change in the gradients is performed in the backward step. This process is repeated several times till the model gives out satisfying result.

**Residual Inception** - Xingpeng Zhang Sheng Huang Xiaohong Zhang Wei Wang Qiuli Wang Dan Yang (2018)

## Algorithm

The proposed TExNet neural network model is built on a non-conventional custom inheritance method, that iteratively builds on the original algorithm. The following is a functional structure of the proposed TExNet model.

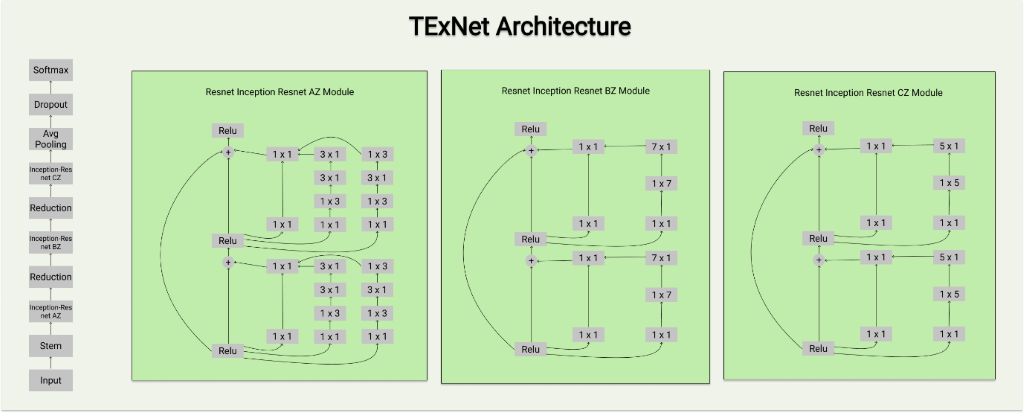


Algorithm explanation:

1. TExNet model arguments
   1. A 3 channel 299x299 pixel image data
   2. An integer dimension of the output space
2. Algorithm structure
   1. Initial convolution layer
   2. Residual block between inception cells
   3. Residual block within inception cells
   4. Final classification step

## Model Overview

Our proposed neural network model, TExNet, uses the existing ResNet and Inception model network, whereby intelligently combining the advantages of ResNet and Inception model. This section will explain the motivation behind the currently proposed model and what exactly is the structure of the proposed model along with its functioning, considering it a virtually independent model.



Model motivation:

1: ResNet has been a rising model in the field of deep neural networks considering its approach towards combing layers of the neural network (NN). The ResNet model uses a concept called residual block which bypasses or in other words fast-forwards the output of previous layer by skip connection method and passes it to deeper layers of the neural network. This approach of ResNet model has been inherited for the proposed TExNet model, which uses the concept of residual blocks for inclusion of deeper neural layers without the loss of information that usually degrades over layers in conventional large deep neural network models, where the backpropagation of the gradient degrades to infinitesimally small values that the model fails to capture resulting in a saturated or overfitted model.

2: Secondly, the success of the Inception model has been in the introduction of 1x1 convolution preceding the conventional NxN convolution, which reduces the number of input channels required to tune the model altogether. This reduction in dimension hugely succeeded in reducing the parameter requirement for the model as a whole by 10-20 folds. This idea of Inception model has also been inherited into the proposed TExNet model which acts as a catalyst for the aim of achieving higher accuracy models with finer performance.

Model structure and function:

The proposed TExNet model comprises of 2 primary components, an independent residual block that connects multiple inception cells together and a transformed inception cell that consists of residual block(s) within it. The purpose of having a residual block and inception cell have already been explained in earlier parts. This part will explain the purpose of this specific order of arrangement. As residual blocks bypass the neural layers that are connected around it, wrapping the inception cells around a residual block will help in faster propagation of information with lesser loss factor. Similarly, for better time-based performance, individual inception cells are embedded with residual blocks theoretically increasing the performance in terms of training time and accuracy for a given number of epochs. The proposed TExNet model is expected to have a decreased gradient factor for larger deep neural networks for better accuracy with particularly a trade-off against run-time caused by relatively deeper networks.

Comparison

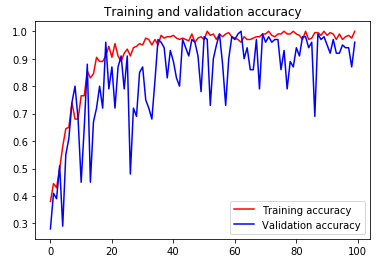
This section will particularly discuss about the already existing model independently with its advantages and disadvantages along with the proposed model and how it takes advantages of the existing models and tries to eliminate their disadvantage

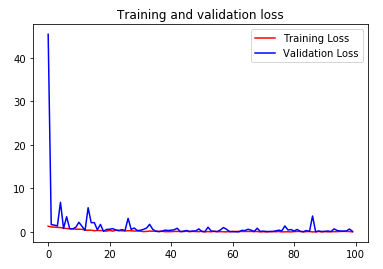
ResNet model:

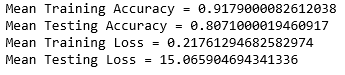
Universal approximation theorem states that given enough capacity, a feed-forward neural network consisting of a single layer is just sufficient enough to represent any function. However, the system can be very large which can end up overfitting the data but simply stacking up the layers on top of each other wouldn’t work. Hence, when the neural network goes deeper and deeper, the back propagation of gradient becomes insignificant. The ResNet model uses the identity connection method to skip layers as discussed briefly in the overview section, which allows deeper network by allowing the gradient travel large number of layers. This also results in having the error rates for deeper layers just about the same compared to their shallower counterparts. A fitting hypothesis is that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlaying mapping. Here, although deeper neural networks can be trained, it still loses its accuracy by a small percentage.

Inception model:

The inception model as discussed in the overview section uses the 1x1 convolution before passing through each of the other NxN convolution and also uses it in the post-max-pooling stage. This helps in reducing the number of parameters to be tuned by many folds helping in better accuracy but increases the training-time for the model by a small proportion.



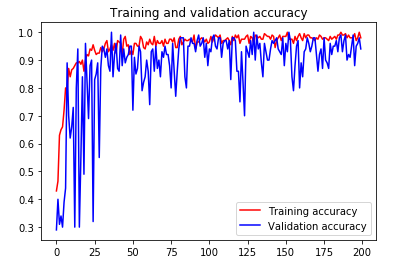


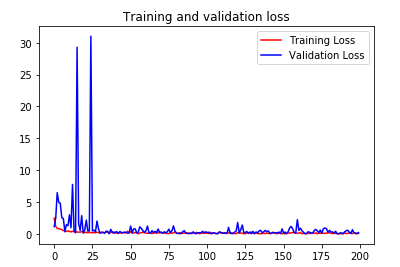


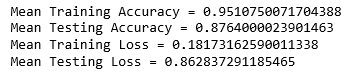
TExNet model [proposed]:

|  |
| --- |
| Total params: 20,313,315 |
| Trainable params: 20,277,731 |
| Non-trainable params: 35,584 |

The proposed model incorporates the advantages of both the model, bypassing neural layers for faster computation in deeper networks and introduction of 1x1 convolution for relaxing the model complexity and increasing performance. The first stage optimization consists of residual blocks between two inception cells, this lets the model pass more accurate and detailed data to deeper layers without more loss of generality, helping stacking up deeper neural layers for finer detection. Secondly, the usage of residual blocks within the inception cell themselves allows the model to cost cut on processing time with lesser performance trade-off.







Comparing the Inception Resnet V4 model with the proposed TExNet model shows that the proposed model doesn’t just theoretically improvise over the original but practically gives performance improvement which can be seen from the above given graphs.

Firstly, the proposed TExNet model provides a significant training accuracy of 95.1%, that is nearly 4% greater than the 91.7% training accuracy of the original Inception Resnet V4 model.

Secondly, the proposed TExNet model provides a significant testing accuracy of 87.6%, that is significantly 7% more accurate than the 80.7% testing accuracy of the original Inception Resnet V4 model.

## Training Methodology

Our networks were trained using very limited computing hardware, comprises of the market standard NVIDIA GTX 1050Ti (Portable). The full extend of the network’s capability is limited as the training takes at least 70-80 seconds per epoch, for the steps per epoch set at 300. Compared to the Inception V4, the training per epoch is twice of it. The hyper-parameter tuning we have opted is to hard code and hard test the model to find the most optimal parameters for which the network can be best trained on for any given dataset. The whole network is trained within Jupyter notebook, using TensorFlow 1.15 – GPU and CUDNN. All test comparison is made initially with the Rock-Paper-Scissors dataset which are stored in terms of their respective directory. The dataset consists of 300 images of each category of dataset, where 50 of the images are used for validation and rest of the images are used for training. The number of epochs set for training the network for the given RPS dataset is set at 200.

The learning rate plays one of the most imported roles for the training of the network. Even the slightest change in the learning rate either improves the training by making it training it faster or very slowly. The aim of this network is to be able to train faster for images that have many features within it. The method applied to find the most optimal parameters for the model is not the best, but by the method of trail and error the model is subjected to give out more information of where the model lacks and what are the parameters that need improving. The training results for each models apart from TExNet take atleast 4-5 hrs to get a stabilized result, but for TExNet takes atleast 6 hours. The results for the RPS dataset indicate that the TExNet model is 1% better than the inception v4 model.

The model will next be trained with the most standard dataset most used for testing out any new network.

Digits\_MNIST\_Dataset: comprises of 60,000 images for training and 10,000 images for training. Each individual image consists of single digit from the range 0-9. (The result of the networks trained with this dataset will be updated, and any additional changes to the model will also be noted and updated in the document.

Conclusion

Thus, the proposed model is a theoretical improvement or an incremental model that leverages the ResNet and Inception models, specifically their advantages. The proposed model has also gone through custom improvements in parameter setting for making the performance and complexity relaxation more significant.

On a conclusive remark, the proposed TExNet model is 1.2% faster than the original Inception ResNet V4 model on the basis of overall training-testing performance measure. Also, the original Inception ResNet V4 model and the proposed TExNet model has been tested on naive datasets and the published results are for the same, further testing and training on large datasets will be performed for a more conclusive and elaborate training-testing performance measure.

The future work may include:

1. Incorporating/Replacing better performing models in the future which can result in increased performance.
2. Usage of custom-built hardware systems for proposed model specific optimization

## Reference

[1] Srivastava R K, Greff K, Schmidhuber J. Training very deep networks[C]//Advances in neural information processing systems. 2015: 2377-2385.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[3] B. Li and Y. He, “An Improved ResNet Based on the Adjustable Shortcut Connections,” IEEE Access, vol. 6, pp. 18967–18974, 2018.

[4] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception resnet and the impact of residual connections on learning[C]//AAAI. 2017, 4: 12.

[5] Dong C, Loy C C, He K, et al. Learning a deep convolutional network for image super-resolution[C]//European Conference on Computer Vision. Springer, Cham, 2014: 184-199.

[6] X. Zhang, S. Huang, X. Zhang, W. Wang, Q. Wang, and D. Yang, “Residual Inception: A New Module Combining Modified Residual with Inception to Improve Network Performance,” 2018 25th IEEE International Conference on Image Processing (ICIP), 2018.