Journal of Computer Science

Original Research Paper

Analyzing Effect on Residual Learning by Gradual Narrowing Fully-Connected Layer Width and Implementing Inception Block in Convolution Layer

Saurabh Sharma

Department of Computer Science, Sir Padampat Singhania School of Engineering, India

Article history
Received: 12-12-2021
Revised: 06-05-2022
Accepted: 10-05-2022

Email: isaurabhsh@gmail.com

Abstract: Research conducted on the advancement of CNN architecture for computer vision problems focuses on strategically choosing and modifying convolution hyperparameters (kernel, pooling, etc.). However, these research works don't exploit the advantage of employing multi fully-connected layers post the core schema to avail further performance improvements, which have been identified as the first research gap. Studies were also conducted to address the challenges of vanishing gradients in deep networks by employing residual learning via skip connections and lowering model training computational costs using parallel convolution rather than sequential convolution operations by employing inception blocks. These studies also don't discuss in detail the impact of sparing features on feature learning, which has been identified as the second research gap. Diagnosis of infectious patterns in chest X-rays using residual learning is chosen as the problem statement for this study. Results show that ResNet50 architecture achieved improved accuracy by 0.6218% and declined error rate by 2.6326% if gradually narrowing FC layers are employed between core residual learning schema and output layer. Also, independent implementation of inception blocks (google net v2) before skip-connections in ResNet50 architecture boosts accuracy by 0.961% and lowers the error rate by 4.2438%. These performance improvements were achieved without regularization and thus, encourage future work in this direction.

Keywords: Fully-Connected Layer, Neuron Layer Width, ResNet50, Residual Network, Skip-Connections, Inception Blocks

Introduction

Convolution Neural Networks (CNN) are neural networks that are applied for problems related to visual imagery. The mathematical operation of convolution, upon which the network is based, solves the high dimensionality challenge incurred due to the multidimensional feature matrix of an image. The research work of Fukushima (2004) and Mozer (1986) showed that the approximate position of features learned by one convolution layer allows subsequent layers to detect more complex patterns or features. Though deeper CNN helps in learning complex features of images producing an efficient classification model, error-rate decrement over epochs starts plateauing which is caused by the vanishing gradient. This limitation of deeper CNN architecture was explained in an article by He *et al.* (2016), which also discussed the residual learning approach based on skip connection or shortcut path. In this study, a fifty-layered architecture was implemented from the aforementioned paper on the multi-class classification problem of chest X-ray images.

Pathology relies on the examination of tissues, organs, and body fluids to identify the causes and effects of a disease on the organ of the body. Using a scan of internal organs to perform a diagnosis of disease requires reading effects and/or patterns of the pathogen on that organ. In the case of pneumonia, the goal of bio-medical diagnosis entails establishing a preliminary finding that entails the presence or absence of infiltrates (white spots) in the lungs from the patient's Chest X-Ray (CXR) which indicates the presence of an abnormal substance in the lung parenchyma. The lung parenchyma Suki *et al.* (2011) is a portion of the lung responsible for gaseous exchange (collectively comprising alveoli, alveolar ducts and respiratory bronchioles). The presence of pulmonary infiltrate could be evident in patients Ellison and

Science Publications

Donowitz (2015); Weinberger *et al*. (2017) suffering from diseases such as pneumonia, tuberculosis and immuno-compromised patients suffering from HIV infection as well as in patients suffering from rejection after organ transplant. The possible disease or infection responsible for pulmonary infiltrates can be identified by studying the patient's medical history, blood or sputum tests as well as by analyzing the distribution patterns of these infiltrates themselves. X-rays are primarily absorbed by the bone structure (ribs, sternum, trachea, clavicle, scapula, humerus, etc.) making them highly illuminated on the CXR image. However, soft tissues that make up the lung parenchyma absorb less amount of X-ray, leaving them less illuminated on CXR images. As soft tissues are less illuminated on CXR images, it becomes challenging for medical practitioners to identify abnormalities and patterns of infection in chest X-ray scans. Despite these challenges, experienced radiologists and surgeons can perform the anatomy of soft tissues (lung parenchyma) ignoring surrounding bones in the CXR images by relying on their experience, perceptual skills and judgment. Through this study, we try to understand whether lung pathology diagnostics can be achieved by a machine learning model that can distinguish and learn soft tissues infiltrate patterns like an experienced radiologist.

By studying architectures of successful and innovative models developed over the past decade, we outlined experimentation to identify the scope of enhancement in fully-connected and convolution layers of ResNet50 architecture for better learning of lung infiltrates patterns for diagnosis of lung infections with high probability. We analyze the resultant performance of original architecture and its architectural variations, against the categorical classification problem of chest X-ray images into normal, viral pneumonia, COVID-19, and lung opacity classes.

Though earlier attempts have been made towards modeling a solution for chest x-ray classification, analysis work achieved here will contribute to the development of more advanced architectures for biomedical diagnosis.

Related Work

As part of a related research work-study, we focused on a few architectures that participated in or won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) Russakovsky *et al*. (2015) in the past few years. ImageNet dataset consists of more than 15 million high-resolution images labeled under thousands of categories, each category consisting of more than a thousand images. The sampling is then distributed into training, testing, and validation with image counts of 1.2 million, 150,000 and 50,000.

Krizhevsky *et al*. (2012) published a Neural Information Processing Systems (NeurIPS) paper that described the architecture of AlexNet, which won the ILSVRC 2012 by substantially reducing the error rate. AlexNet architecture consists of eight layers, comprising five convolution layers and three fully connected Dense layers. Two groups of kernels are used for convolution operation in the first four layers (1,2,3 and 4). The kernel dimensions decrease slowly from 11 to 3 (11 x 11, 5 x 5, 3 x 3 and 3 x 3), while the number of kernels increases from 48 to 192 (48, 128, 192 and 192). In the first two layers, convolution operations are followed by overlapping max pooling and local response normalization. In the fifth layer, 256 kernels of dimension 3 x 3 are used for convolution operation, followed by 3 x 3 overlapping max pooling with 2 strides. The sixth and seventh layers are dense layers, comprising 4096 neurons each. Finally, the last layer is a dense layer of 1000 neurons, with softmax activation to classify the input image into one of 1000 classes of the ImageNet dataset. The demonstrated experimentation in the paper uses input images of size 227 x 227 or 224 x 224 with padding. The paper introduces Rectified Linear Unit (ReLU) activation function which performs six times faster than the Tanh activation function. Overlapping pooling differs from non-overlapping pooling concerning stride size smaller than their kernel size. AlexNet also utilizes a dropout Srivastava *et al.* (2014); Hinton *et al.* (2012) of 0.5 probability in the first two fully-connected layers as regularization. A learning rate of 0.01 was used which was reduced (three times) by a factor of 10 in case the validation error rate did not improve. Convolution operation was conducted using two groups of convolution operations in the first four layers, resulting in two parallel paths. Thus, two GPUS were used for implementing the architecture. The top-1 and Top-5 validation error rates of AlexNet were 39 and 18.2% respectively. In the result section, the author mentions that adding one more convolution layer to AlexNet helped reduce the validation error rate from 18.2 to 16.6%. CaffeNet Jia *et al*. (2014) is a 1-GPU implementation of AlexNet, in which instead of 2 path architecture of AlexNet are combined into one path.

Clarify further fine-tuned AlexNet for better image classification tasks by designing network architecture based on visualizing technique published by Zeiler and Fergus (2014). ZFNet (named after the authors of the paper) proposes two changes to AlexNet architecture's first convolution layer, namely, reduction of the first layer filter size from 11 x 11 to 7 x 7 and changing the stride from 4 to 2. The ablation study showed a top-5 validation error of 16.5%, while AlexNet was at 18.1%. Clarify won the ILSVRC 2013 using ZFNet.

VGGNet developed by VGG (Visual Geometry Group) team by Simonyan *et al*. (2015) from Oxford university was 1st runner-up of ILSVRC 2014. Though the architecture did not win the aforementioned competition, it demonstrated significant improvement in image classification tasks from ZFNet and thus over AlexNet. The paper proposes using two layers of 3 x 3 filters instead of 5 x 5 filters and three layers of 3 x 3 filters instead of 7 x 7 filters in the ZFNet architecture. Replacing large filters with smaller ones helps in reducing the number of trainable parameters. With reduced trainable parameters, the risk of vanishing gradient associated with a deeper network diminishes, net-work convergences faster and the problem of overfitting is reduced. The ablation study in the paper demonstrates VGG-11, VGG-13, VGG-16 and VGG-19 models. The author keeps adding layers to the network and verifies the error rate to check for improvements. VGG-11 obtained an error rate of 10.4%, similar to ZFNet. VGG-13 produced an error rate of 9.9% on the addition of the conv layer to the existing architecture. In VGG16(Conv-1), additional three convolution layers were added to the architecture which further lowered the error rate to 9.4%. VGG-16 with additional filters lowered the error rate to 8.8%. However, VGG-19 raised the error rate to 9.0% which proved that adding further layers to existing architecture was no longer improving accuracy.

Inception architecture was first introduced as GoogLeNet (a tribute to Lenet architecture) by Szegedy *et al.* (2015). The paper introduced the inception module approach, wherein rather than the implementation of convolution layers and pooling sequentially, the inception module implements 1 x 1, 3 x 3 and 5 x 5 convolution operation and max-pooling on an input parallelly and then concatenates the output of these operations to pass onto next layer. This approach saved the decision-making step of choosing the right size of filter for convolution operations in a layer. The GoogLeNet, also known as Inception-v1architecture suffered from saturation problems and consequently gradient descent problems. To resolve the above-stated issue, Ioffe and Szegedy (2015), published a paper on BN-Inception, also known as Inception-v2 was developed. This new version uses Batch Normalization and ReLU activation function also replaces 5 x 5 convolution with two 3 x 3 convolution for parameter reduction. The resultant architecture produced more irregular outputs, thus higher learning rate was advised. Szegedy *et al*. (2016) introduced factorization for the convolution layer to reduce dimensionality and in turn, reduce the overfitting problem. The author proposed that rather than using a square-shaped filter of dimension f x f for convolution operation, two filters of dimension f x 1 and 1 x f will yield similar results while reducing the number of learning parameters. The paper also included an efficient grid size-reduction module which is an equally efficient but computationally cheaper network.

Residual Network a.k.a. ResNet He *et al*. (2016) was the winner of ILSVRC 2015 for image classification, detection and localization. ResN*et al*so won MS COCO 2015 competition for detection and segmentation. Deep learning networks such as AlexNet, ZFNet and VGGNet are based on an architecture that entails convolution layers, followed by Fully Connected (FC) layers for image classification tasks. Such networks are referenced by the authors as "plain" networks.

Such networks when designed for deeper representation, i.e., when the number of layers is increased in their architectures, pose the problem of either vanishing or exploding gradients. This is in contrast to the expectation from neural networks to produce better accuracy predictions with deeper layers. This problem was demonstrated by the authors upon comparing training and testing error representation of 20 layers and 56 layer plain networks using CIFAR-10 dataset. The authors proposed the addition of a skip/shortcut connection after a few weighted layers. If during the weighted layer learning, gradients of features start to vanish, the input layer carried by skip connection, will transfer them back to earlier layers. These blocks formed by employing skip connections are referred to as residual blocks. In the case of deeper networks, the paper adopts the suggested technique of GoogLeNet (Inception-v1) and Network in Network Lin *et al*. (2014) of adding 1 x 1 conv layer after the start and before the end of each residual block. These 1 x 1 conv reduced the number of connections, without much degrading network model performance. In the ablation study, results showed how 34 layered ResNet architecture was better than a 34 layer plain network by comparing their error rate using 10 crop testing, though, not much significant improvement was evident in the comparison between 18 layer plan and 18 layer ResNet architecture on the same test.

Szegedy *et al*. (2017) published two Inception-v4 architectures based on ResNet model's skip connection approach named Inception-ResNet-v1 and Inception-ResNet-v2. In this version, Inception blocks employ the improvements developed in Inception-v2 and Inception-v3, primarily Batch Normalization and Factorization. The paper presents three architectures of inception blocks (Inception-A, Inception-B, and Inception-C) and one architecture of stem. The stem architecture comprises six convolution layers, with one max-pool layer of 3 x 3 dimension after the first three layers. The number of filters/kernels increases from 32 to 256 gradually as 32, 32, 64, 80, 192 and 256 for each layer respectively. The inception block architectures proposed in the paper have skip/shortcut connections, to cater to vanishing gradient problems as was proposed in ResNet architecture. The main architecture comprises the above-mentioned stem, sequentially followed by the proposed Inception-A and its reduction layer, Inception-B and its reduction layer, Inception-C and its reduction layer, followed by Average Pooling and Dropout of 0.8 probability. The computational cost of the Inception-ResNet-v1 model is similar but trained faster in comparison with Inception-v3. However, the accuracy of the model was slightly worse than Inception-v3. Since a large number of filters were used in the overall architecture, the author mentioned that employing more than 1000 filters caused instability with the residual variants, causing the network to die in early epochs of training. Another architecture, Inception-ResNet-v2 mentioned in the paper employs the same overall network schema with variations in Inception blocks (A, B and C) by increasing the number of filters in these blocks to a higher number. The resultant architecture had a similar computational cost to that of Inception-ResNet-v1, but with faster training and better final accuracy.

Wu *et al.* (2019) performed an in-depth analysis of the trade-off between the width and depth of ResNet architecture. The proposed model outperformed the original ResNet architecture in image classification and had good performance for semantic segmentation. The paper discusses that additional depths in ResNet may not bring much improvement and could yield worse results since effective depths in ResNet are not completely trained. Since deeper networks demand high GPU memory, shallower, but wider networks can have more trainable parameters with lesser GPU memory cost. The proposed model employs Batch Normalization and ReLU activation with each convolution layer. With only a depth of 38 layers, the model had 19.2 and 4.7% top-1 and top-5 errors respectively, outperforming ResNet, Inception-4 and Inception-ResNet-v2 models.

Methodology

ImageNet dataset consists of large and uneven dimension images. During the ILSVRC competition, prior to model training and evaluation, resizing input images to 256 x 256 or 224 x 224 dimensions was part of the data pre-processing activity. This squared the input image dimensions and also reduced processing cost. It is worth noting that, the studied models have not been evaluated against images with higher dimensions such as 500x500 or more. However, the dimensions of medical images such as chest X-rays and CT scans are large, to present infectious patterns with higher clarity to assist human vision that suffers from limited capabilities. This assumption inspired a part of the data pre-processing approach employed for this study. Thus, for model training and performance evaluation, input images were re-sized to 500 x 500 pixel dimensions to better enact live scenarios in the field of medical diagnosis. A convolution neural network primarily consists of a convolution layer and Fully-Connected (FC) layer. A study of literature on various CNN architectures for computer vision problems or challenges shows that advancements or performance improvements are achieved by altering either all or some components of the convolution layer, such as kernel dimension, filter count, pooling and normalization method, activation function, etc. However, no specific analysis or attempt towards improving the neural schema of FC layer has been conducted from studied architectures or literature. This observation constitutes the first half of our problem statement. A study of CNN architecture evolution from deeper to wider network from the work of Gilboa and Gur-Ari (2019) showed that the cost of network training is reduced due to the transferability of learned features between parallelly implemented convolution tasks in a wider network. The research also showed that compared to a deeper neural network, a wider network holds and learns from more input features. Szegedy *et al.* (2015) implemented the concept of the wider network in GoogLenet architecture in form of inception blocks. Shallower and performance enhanced architecture was later conceived when Ioffe and Szegedy (2015) and Szegedy *et al.* (2016) introduced kernel factorization for convolution operation and Batch Normalization to the network schema. Though the performance of a wider network holds for novel classification tasks, learning more complex features may require a deeper neural network. However, theoretical understanding of pre- diction performance is proportional to the depth of a network ceases to be true in the case of very deep neural. Simonyan *et al.* (2015) and He *et al.* (2016) attribute the cause of this unexpected neural network behavior to a vanishing gradient. This was evident from the ablation study of VGG architecture which showed that the error rate of VGG16 was less than that of VGG19. ResNet architecture mitigated this issue by employing skip connections by adding parameter weights and from the previous layer onto subsequent layers which were separated by 2 or 3 convolution layers. In this study, we selected ResNet50 architecture as a benchmark to study the advantageous effects on the performance of classification tasks by (i) manipulating the width and depth of FC layer and (ii) adding wider convolution layers in an already deeper architecture. Though different state-of-the-art CNN-based architectures choose different widths and depths of FC layer before the classification (output) layer, in the original paper of He *et al.* (2016), ResNet architecture employs a single classification(output) FC layer for generating prediction. For the first half of this research work we studied the effect on the performance of ResNet50 by (i) insertion of FC layer with gradually decreasing width which is less than the flattened output of the final convolution layer, (ii) insertion of a comparatively narrower dense neural layer before the final classification FC layer, (iii) increasing the depth, but decreasing the width of the FC layer, (iv) increasing the width and depth of the FC layer (Fig. 1). In the second part of this research work, we insert an inception block from GoogLeNet v2 architecture before every convolution block in the ResNet50 (Fig. 2) to investigate if adding width to an already deeper architecture can bring positive performance enhancement for the classification task. For practical implementation and study of the aforementioned experimentation, we chose the problem statement of multi-class (normal, viral-pneumonia, COVID-19 and lung-opacity) classification of bio-medical Chest X-Ray (CXR) images. Before the listed experimentations, a classification model based on ResNet50 schema was trained and evaluated and its performance was recorded. The performance of original ResNet50 against chosen metrics (section 4) was registered as a benchmark, against which all experimentally tweaked ResNet50 models were compared.

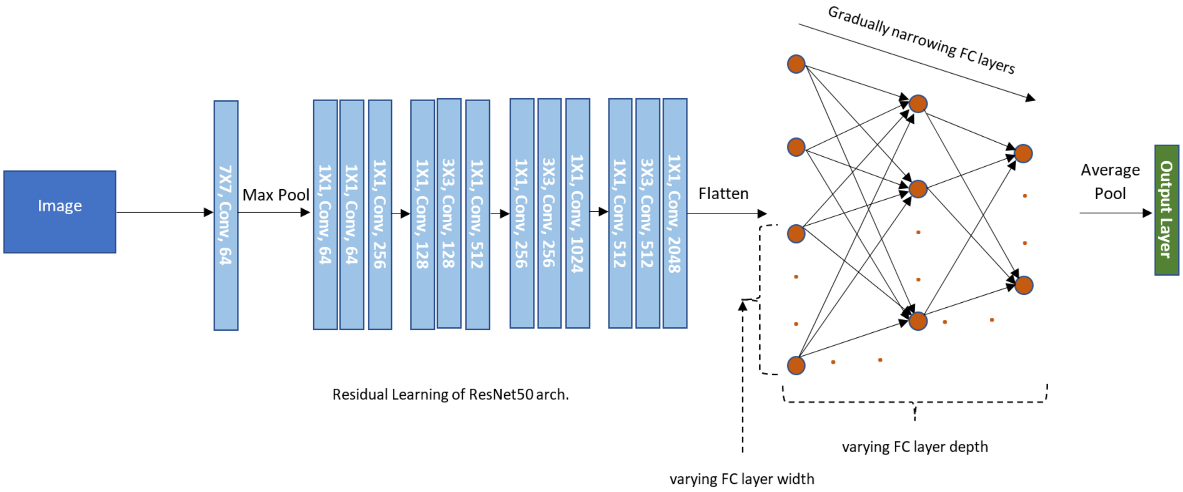

Fig. 1: Insertion of gradually narrowing FC layers after residual learning in ResNet50 arch

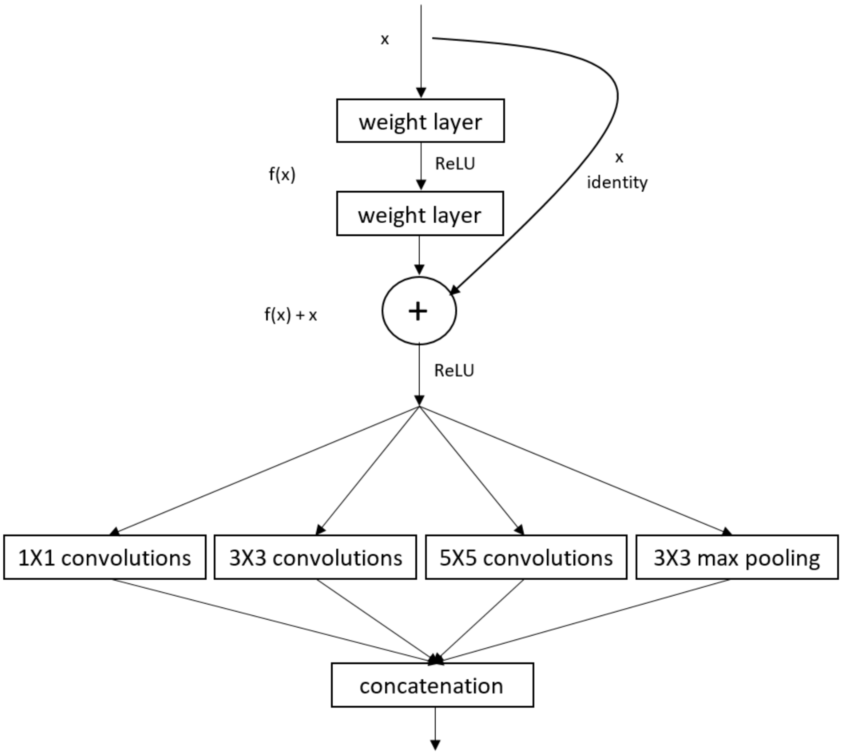

Fig. 2: Insertion of inception block post every skip-connection in ResNet50 arch to exploit spared feature learning

Experiments and Results

Two experimentations Experiment (i) and Experiment(ii) were set up to study the effect on model performance for classification tasks by; (i) manipulating the width and depth of FC layer and (ii) adding wider convolution layers in an already deeper architecture.

Dataset

The chosen dataset was accumulated from varied sources Rahman (2021), consisting of CXR images of resolution 299x299, which constitutes over 21,000 images. Due to limited computational resources (number of GPU devices and time-sharing constraints) and huge class imbalance, a randomly chosen, but fixed subset of the image dataset was utilized in all experimentation setups. The image dataset comprised 2500 samples, each of COVID-19, lung opacity, and normal cases, while 1345 images of viral pneumonia. Of the total 8845 images, 1769 images were separated for testing and 1415 images were kept aside for validation, leaving 5661 images for training.

Data Augmentation

Images were resized to 500x500 pixels to evaluate the performance of the models against high-resolution images, which is normally the case in bio-medical scans. Images were rescaled to the range of [0,1]. Other image augmentation approaches used were rotation, standard normalization, image shift (width and height), shearing, zoom, channel shift, and image flip (vertical and horizontal).

System Configuration

Tesla P100-PCIE-16GB

Implementation

Each model was trained for 200 epochs overtraining and validation batches of size 16. The learning rate was initially set at 0.001 with the Adam optimizer and was scheduled to decrease by a factor of 10 after 80, 120, 160, and 180 epochs. Model training was monitored using categorical-cross-entropy (for loss) and accuracy (for a measure of classification correctness).

Evaluation Metrics

Precision and recall rely on a confusion matrix, which in turn relies on closely balanced classes. However, in a real-life scenario, the number of scanned images of each categorical class may not always be even. Thus, F1-score was selected as the primary evaluation metric to compare the model performance of the benchmark model (original ResNet50) and its experimental variants on the chosen categorical classification task. However, we included precision and recall metrics for detailed performance comparison.

Prerequisite (Benchmark) Evaluation

Benchmark model, based on the original ResNet50 architecture schema was implemented, evaluated and its performance recorded against all considered metrics for later comparison with the performance of experimental models in detailed experiments.

Experiment (i)

A single convolution layer of a simple CNN architecture implementation consists of a convolution operation followed by a pooling layer. The number of kernels at each subsequent layer number is gradually increased to bring the number of trainable features to a manageable size before flattening the output of the final convolution layer and passing it either to a single FC layer that serves as a classification neural layer or to a series of subsequent FC layer densely connected to the classification neural layer for making classification prediction.

Though different state-of-the-art CNN based architectures choose different widths and depths of FC layer before the classification (output) layer, in the original paper of He *et al*. (2016), ResNet architecture employs a single classification(output) FC layer whose width is equal to the number of classes in the problem.

Experiment (i), which is a set of three experiments, studies the effect of changes in neural width and depth of the above-mentioned FC layer on model performance. Each experiment was based on independent approach-based curiosity, which upon result analysis formulated into the base foundation for further experiments in this set of experiments. It needs to be noted that the number of parameters in the flattened layer of the unmodified ResNet50(original) model is 131072.

Experiment (i)-A

This experiment is based on an initial hypothesis that a gradual decrease in several parameters while moving from one neural layer to another can help increase the model's performance. The inspiration for this hypothesis is derived from the resultant decrease in several parameters from every subsequent convolution layer.

To test this hypothesis, a set of gradually narrowing(width) FC layers is inserted before the final classification(output) FC layer.

Three modified versions of ResNet50 models were implemented by manipulating the width of FC layer; (i) Insertion of one FC layer of the width of one thousand neurons, (ii) Insertion of two FC layers of the width of one thousand neurons, and (iii) Insertion of one FC layers of width one thousand neurons followed by one FC layer of one-hundred-twenty-eight neurons.

Experiment (i)–B

Findings of Experiment (i)-A shown in Table 1 shows that the modified versions of ResNet50 model with multiple FC layers, ResNet50-with-FC(2 x 1000) and ResNet50-with- FC(1 x 1000, 1 x 128) lowered the test error-rate by 2.7785 and 2.3744% compared to unmodified ResNet50 benchmark model. Though both variant models performed better than the benchmark model, only ResNet50-with-FC(1 x 1000, 1 x 128) model suffered the least decline in test accuracy i.e., 0.1131%.

Based on the understanding of observations from Experiment (i)-A, Experiment (i)-B was outlined. Since in the earlier experiment, the model with FC layer of one-hundred-twenty-eight neurons before the classification layer lowered the error rate by 2.3744%, at the expense of 0.1131% of test accuracy, experiments in this set were implemented with one-hundred-twenty-eight neurons in the FC layer just before classification layer. However, the wider FC layer was replaced by narrower FC layers of width 246 neurons while increasing the depth of FC layer in subsequent experiments. Implemented FC layers in variant ResNet50 architectures of this Experiment(i)-B set entails, (i) Insertion of two FC layers of width 256 neurons, (ii) Insertion of four FC layers of width 256 neurons, and (iii) Insertion of six FC layers of width 256 neurons; each followed by one FC layer of 128 neurons.

Experiment (i)–C

Performance observation of model variants belonging to Experiment(i)-B from Table 2, upon comparison with the benchmark model based on the original ResNet50 architecture shows distinctive improvement in test accuracy and test loss by a factor of 0.6218% and 2.6326% respectively. Based on the results of Experiment (i)-B, variants of ResNet50 in Experiment(i)-C were outlined by in- section of FC layer of depth of four layers and neuron width of; (i) five-hundred, (ii) one-thousand and (iii) two-thousand-forty-eight; followed by FC layer of one-hundred-twenty-eight neurons. Table 3 shows no satisfactory improvement in test accuracy and loss was observed in this set of experiments, thus ceasing modifications in FC layers of ResNet50 architecture.

Experiment (ii)

To understand the impact of inception block (wider network) on performance of existing ResNet50 architecture, an inception block was added before every convolution block (Conv->BN->ReLU->Conv->BN->ReLU->Conv->BN->Conv->BN->Skip Conn->ReLU) of the residual network (ResNet50). The intuition of adding multiple inception blocks was two folds; to allow the network to learn from a wider set of features at the same time and to allow the model to learn using multiple dimensions of kernels (1 x 1, 3 x 3, and 5 x 5).

The performance comparison between the original ResNet50 model and the Hybrid (Inception stacked within Residual Network) model listed in Table 4, showed a marginal improvement in inaccuracy, but a significant decrease in an overall loss.

Table 1: F1-score, accuracy, loss, and parameter comparison between benchmark and experiment (i)-A

Model variations	Benchmark ResNet50 (Original)	ResNet50 with FC 1 × 1000	ResNet50 with FC 2 × 1000	Experiment (i) - A ResNet50 with FC 1 × 1000, 1 × 128
F1 Score (Class Level) [0,1]				
Classes				
COVID_19	0.9260	0.9085	0.9289	0.9337
Lung Opacity	0.8552	0.8502	0.8586	0.8621
Normal	0.8669	0.8582	0.8561	0.8574
V. Pneumonia	0.9562	0.9544	0.9398	0.9385
F1 Score (Average) [0,1]				
Macro avg.	0.9011	0.8928	0.8958	0.8979
Weighted avg.	0.8939	0.8848	0.8901	0.8926
Accuracy (%)				
Test Accuracy	89.2595	88.3550	88.8638	89.1464
Test Loss	32.9535	34.4329	30.1750	30.5791

Table 2: F1-score, accuracy, loss, and parameter comparison between benchmark and experiment (i)-B

Model variations	Benchmark ResNet50 (Original)	Experiment (i) - B ResNet50 with FC 2 × 256, 1 × 128	ResNet50 with FC4 256, 1×128	ResNet50 with FC6 × 256, 1 × 128
F1 Score (Class Level) [0,1]				
Classes				
COVID_19	0.9260	0.9164	0.9262	0.8629
Lung Opacity	0.8552	0.8410	0.8632	0.8047
Normal	0.8669	0.8601	0.8806	0.8127
V. Pneumonia	0.9562	0.9313	0.9536	0.9396
F1 Score (Average) [0,1]				
Macro avg.	0.9011	0.8872	0.9059	0.8550
Weighted avg.	0.8939	0.8815	0.8996	0.8439
Accuracy (%)				
Test accuracy	89.2595	88.0724	89.8813	84.2849
Test loss	32.9535	32.1273	30.3209	43.9054

Table 3: F1-score, accuracy, loss and parameter comparison between benchmark and experiment(i)-C

Benchmark	ResNet50 with FC	Experiment (i) – C	ResNet50 with	
Model variations	ResNet50	FC 4 × 1000,	FC 4 × 2048,	
(Original)	4 × 500, 1 × 128	1 × 128	1 × 128	ResNet50 with
F1 Score (Class Level) [0,1]				
Classes				
COVID_19	0.9260	0.9099	0.9252	0.8936
Lung Opacity	0.8552	0.8187	0.8455	0.8504
Normal	0.8669	0.8385	0.8526	0.8507
V. pneumonia	0.9562	0.9624	0.9667	0.9389
F1 Score (Average) [0,1]				
Macro avg.	0.9011	0.8824	0.8975	0.8834
Weighted avg.	0.8939	0.8719	0.8885	0.8761
Accuracy (%)				
Test accuracy	89.2595	87.0548	88.7507	87.5071
Test loss	32.9535	35.2765	32.0964	35.5780

Table 4: F1-score, accuracy, loss, and parameter comparison between benchmark and experiment(ii)

Model variations	Benchmark ResNet50 (original)	Experiment (ii) hybrid model
F1 Score (Class Level) [0,1]		
Classes		
COVID_19	0.9260	0.9344
Lung opacity	0.8552	0.8703
Normal	0.8669	0.8757
V. Pneumonia	0.9562	0.9565
F1 Score (Average) [0,1]		
Macro avg.	0.9011	0.9093
Weighted avg.	0.8939	0.9031
Accuracy (%)		
Test accuracy	89.2595	90.2205
Test loss	32.9535	28.7097

Table 5: F1-score, comparison between selected models from experiment sets (i) and (ii) against benchmark

	Benchmark Experiment		Experiment (i)-B	Experiment (i)-C	Experiment (ii)
MODEL	(i)-A	ResNet50 with	ResNet50 with	ResNet50 with	Hybrid Model
VARIA-	ResNet50	FC 1 × 1000,	FC 4 256, 1×	FC 4 ×1000,	(Inception
TIONS	(original)	1 × 128	128	1 × 128	stacked in ResNet)
F1 Score (Class Level) [0,1]					
Classes					
COVID_19	0.9260	0.9337	0.9262	0.9252	0.9344
Lung opacity	0.8552	0.8621	0.8632	0.8455	0.8703
Normal	0.8669	0.8574	0.8806	0.8526	0.8757
V. pneumonia	0.9562	0.9385	0.9536	0.9667	0.9565
F1 score (average) [0,1]					
Macro avg.	0.9011	0.8979	0.9059	0.8975	0.9093
Weighted avg.	0.8939	0.8927	0.8996	0.8885	0.9031

Table 6: Precision comparison between selected models from Experiment sets (i) and (ii) against Benchmark

		Experiment(i)-A	Experiment (i)-B	Experiment (i)-C	Experiment (ii)
	Benchmark	ResNet50 with	ResNet50 with	ResNet50 with	Hybrid Model (original)
	ResNet50	FC 1 × 1000,	F C4× 256,	FC 4×1000,	(Inception stacked
Model variations	(original)	1 × 128	1 × 128	1 × 128	in ResNet)
Precision (Average) [0,1]					
Classes					
COVID_19	0.9821	0.9678	0.9799	0.9777	0.9740
Lung Opacity	0.8427	0.8745	0.8327	0.8595	0.8552
Normal	0.8261	0.8074	0.8694	0.8035	0.8509
V. Pneumonia	0.9804	0.9721	0.9519	0.9631	0.9731
Precision (Average) [0,1]					
Macro avg.	0.9078	0.9055	0.9085	0.9010	0.9133
Weighted avg.	0.8983	0.8968	0.9028	0.8929	0.9055

Table 7: Recall comparison between selected models from Experiment sets (i) and (ii) against Benchmark

Model variations	Benchmark ResNet50 (original)	Experiment (i)-A ResNet50 with FC 1 × 1000, 1 × 128	Experiment (i)-B ResNet50 with FC 4 × 256, 1×128	Experiment (i)-C ResNet50 with FC 4 ×1000, 1 × 128	Experiment (ii) Hybrid Model (Inception stacked in ResNet)
Classes	Recall (Class Level) [0,1]				
COVID_19	0.8760	0.9020	0.8780	0.8780	0.8980
Lung Opacity	0.8680	0.8500	0.8960	0.8320	0.8860
Normal	0.9120	0.9140	0.8920	0.9080	0.9020
V. Pneumonia	0.9331	0.9071	0.9554	0.9703	0.9405
Recall (average) [0,1]					
Macro avg.	0.8973	0.8933	0.9053	0.8971	0.9066
Weighted avg.	0.8926	0.8915	0.8988	0.8875	0.9022

Table 8: Parameter comparison between selected models from Experiment sets (i) and (ii) against Benchmark

Model variations	Benchmark ResNet50 (original)	Experiment (i)-A ResNet50 with FC 1 × 1000, 1 ×128	Experiment (i)-B ResNet50 with FC 4 × 256, 1 × 128	Experiment (i)-C ResNet50 with FC 4 × 1000, 1 ×128	Experiment (ii) Hybrid (Inception st ResNet) acked in
Total Param	24,105,732	154,783,084	57,366,916	157,786,084	24,547,052
Trainable	24,052,612	154,729,964	57,313,796	157,732,964	24,493,932
Non-	53,120	53,120	53,120	53,120	53,120

Table 9: Accuracy & Loss comparison between selected models from Experiment sets (i) and (ii) against Benchmark

Model variations	Benchmark ResNet50 (original)	Experiment (i)-A ResNet50 with FC 1×1000, 1 × 128	Experiment (i)-B ResNet50 with FC 4×256, 1 × 128	Experiment (i)-C ResNet50 with FC 4×1000, 1 × 128	Experiment (ii) Hybrid (Inception stacked in ResNet)
Test Accu- racy	89.2595	89.1464	89.8813	88.7507	90.2205
Test loss	32.9535	30.5791	30.3209	32.0964	28.7097

Discussion

Tables 5, 6, 7, 8 and 9 compare F1-score, precision, recall, parameter size and accuracy, and loss of the models that performed well in experiment setups – Experiment(i)-A, Experiment(i)-B, Experiment(i)-C and Experiment(ii). Based on a study of the F1-score from Table 5, it can be derived that compared to the Benchmark (original) ResNet50 model, two model variants of ResNet50, i.e., ResNet50 with FC (4 x 256, 1 x 128) and hybrid ResNet50(inception block implementation) produced better average and class-wise F1-score than the benchmark ResNet50 model. The noted elevation in macro-average and weighted-average F1-score was of 0.0048 and 0.0057 by ResNet50 with FC (4 x 256, 1 x 128) and, 0.0082 and 0.0092 by hybrid ResNet50(inception block implementation).

The neural changes of ResNet50 with FC(4x256, 1x128) elevated the macro-average and weighted-average precision by 0.0007 and 0.0045 and macro-average and weighted-average re-call by 0.008 and 0.0062, while hybrid ResNet50(inception block implementation) elevated the macro-average and weighted-average precision by 0.0055 and 0.0072 and macro-average and weighted-average recall by 0.0093 and 0.0096, compared to benchmark model, as can be observed from Table 6 and 7.

The neural schema changes in the two experimental models, i.e., ResNet50 with FC (4x256, 1x128) and hybrid ResNet50(inception block implementation), as shown in Table 9, elevated the classification accuracy by 0.6218 and 0.961% respectively and reduced error-rate by 2.6326 and 4.2438% respectively.

Though the applied neural changes in the two models produce better results, the hybrid ResNet50(inception block implementation) employed only 1.83% more parameters than the original ResNet50 model, which is to be expected given the addition of new inception blocks in convolution layers and can be considered as a reasonable trade-off in exchange for reducing error-rate by ~4%.

However, in problems, such as bio-medical classification, where human life is in question, reducing the probability of misdiagnosis by 4.2438% (decrease in error rate) can make a lot of difference for medical practitioners.

Conclusion

As part of this research, we were able to prove our hypothesis that a gradual decrease in neural layer width proportionally, across multiple FC layers before classification (output) FC layer, can produce better results compared to the current implementation of ResNet50 where the reduced features generated by last convolution layer are flattened and passed directly to the classification (output) FC layer. We believe that mathematically deriving the correct width and depth of FC layer by further research and experimentation on more diverse problem sets, can form the foundation of future work to understand the importance of Fully-Connected (FC) layers and inception blocks in computer vision problems. With demonstrated result sets, we were able to achieve a model variation that produced a cross-class performance of more than 90% against F1-score, precision, and recall metrics and reduced error rate by 4.2438% compared to the unmodified ResNet50 benchmark model. It is worth noting, that no regularization was implemented to prevent further overfitting problems and further enhance testing accuracy (except for Batch Normalization which is part of the ResNet50 architecture to reduce internal covariance shift). Thus, we believe that this model can be further improved to achieve even higher test results in image classification problems.

Acknowledgment

I thank my parents and beloved for their blessings, support and encouragement during my study at research program. They continuously nagged and pushed me to give it my best. I also thank my mentor Dr. Mukesh Kalla (HoD, Computer Science) and Dr. Arun Kumar (Dean, Computer Science) at Sir Padampat Singhania University, Udaipur, Rajasthan, India.

Ethics

The author declares that there exist no known competing financial interests that could have influenced the work reported in this study. The author approves the publication of manuscript.

References

Ellison, R. T., III, & Donowitz, G. R. (2015). Acute Pneumonia. Mandell, Douglas, and Bennett's Principles and Practice of Infectious Diseases, 823–846.e5. doi.org/10.1016/B978-1-4557-4801-3.00069-2

Fukushima, K. (2004). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36, 193-202. doi.org/10.1007/BF00344251

Gilboa, D., & Gur-Ari, G. (2019). Wider Networks Learn Better Features. arXiv preprint arXiv:1909.11572. doi.org/10.48550/arXiv.1909.11572

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580. doi.org/10.48550/arXiv.1207.0580

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR. http://proceedings.mlr.press/v37/ioffe15.html

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678). doi.org/10.1145/2647868.2654889

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25. https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Mozer, M. C. (1986). RAMBOT: A Connectionist Expert System That Learns by Example. https://eric.ed.gov/?id=ED276423

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S. B. A., ... & Chowdhury, M. E. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Computers in biology and medicine, 132, 104319. doi.org/10.1016/j.compbiomed.2021.104319

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252. https://link.springer.com/article/10.1007/s11263-015-0816-y

Simonyan, K., Zisserman, A., *et al*., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. doi.org/10.48550/arXiv.1409.1556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

Suki, B., Stamenovic, D., & Hubmayr, R. (2011). Lung parenchymal mechanics. Comprehensive Physiology, 1(3), 1317. doi.org/10.1002/cphy.c100033

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. At the Thirty-first AAAI conference on artificial intelligence.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Weinberger, S. E., Cockrill, B. A., & Mandel, J. (2017). Principles of Pulmonary Medicine E-Book. Elsevier Health Sciences.

Wu, Z., Shen, C., & Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition, 90, 119-133. doi.org/10.1016/j.patcog.2019.01.006

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham. doi.org/10.1007/978-3-319-10590-1_53