Exploration of VGG Models to Benchmark Advanced Deep Convolutional Architecture

Amit Dutta, Antorip Sahu, Armen Kasparian *

Abstract

We investigate the performance of VGG16 and VGG19 image classification architectures, and conduct a comparative performance analysis with competitive architectures. We study how well these models generalize on different data-sets. We also suggest a modified VGG architecture with improved performance in terms of training time with fewer epochs and with the introduction of sparsity for comparable accuracy.

1 Problem Statement and Research Goal

Our main objective for this project is to study existing models for image classification on certain datasets and use those results to mold a new network for the same task. For the study into the existing models used for image classification, an algorithmic approach is taken to get quantitative results that are then used for find the models strengths and weaknesses. This understanding of the models then allows for the creation of a new model based on the existing models that has the goal of maintaining accuracy while maximizing efficiency.

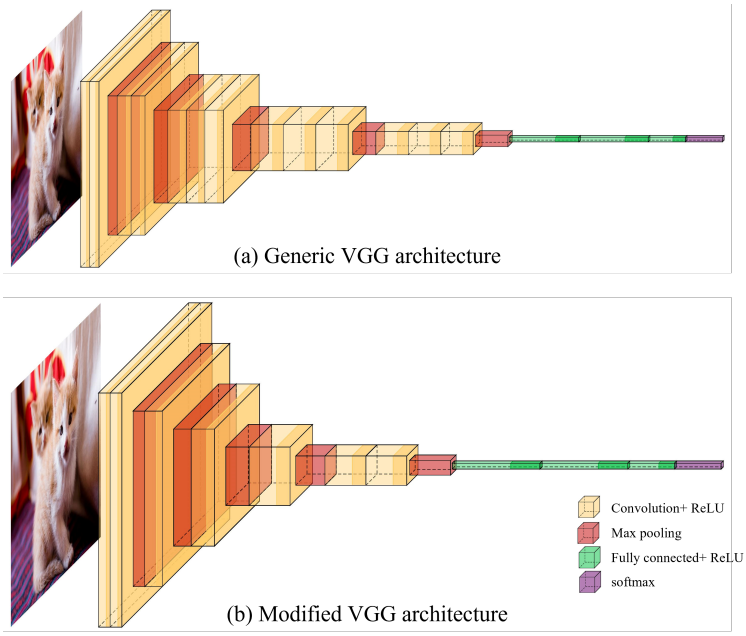


Figure 1: VGG 16 architecture (a) and the proposed modified VGG architecture (b)

*The authors are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA
E-mail: amitdutta@vt.edu, antoripsahu@vt.edu, armen@vt.edu

2 The VGG Architecture

The **VGG** (Visual Geometry Group) architecture was developed to study the affect of depth of convolutions networks on the accuracy for face recognition and image classification problems [1]. This is a well studied popular architecture with satisfactory accuracy guarantees and hence chosen in this project for analysis and development of newer models.

2.1 Architecture Overview

VGG16 has a total of 16 layers that have trainable parameters. To break it down, there are 5 convolutional blocks comprising 13 convolutional layers and 1 fully connected block comprising of 3 fully connected layers. Fig.1(a) describes the architecture. Each set of convolution blocks are associated by max-pooling layers. From the feature maps whose dimensions are given by the height and width the convolutional layers try to pull out as much information as possible by adding more channels after each blocks. Finally after all the convolutional operations we have fully connected layers which produces the final output.

2.2 Input/Output Relation

The input is RGB image with $(224, 224, 3)$ dimension and the output is vector of dimension $(1000, 1)$. The main goal of this architecture was originally to predict the top 5 classes out of 1000 classes from the given training data from the ILSVRC data-set [2] containing 100,000 images. Thus the output is generated after passing through a soft-max function which assigns probability to each class.
From the output layer let the top 5 class be represented by a vector $\mathbf{c} \in \mathcal{R}^5$. Now according to the given training example image let us maintain another vector $\mathbf{C} \in \mathcal{R}^n$ where $1 \leq n \leq 5$ that characterizes the classes in the input image. To test the accuracy of prediction we introduce the following parameter

$$d(c_i, C_k) = \begin{cases} 0 & c_i = C_k \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Next the error can be calculated as $e = \frac{1}{n} \sum_k \min_i d(c_i, C_k)$

3 Simulations and Comparison Results

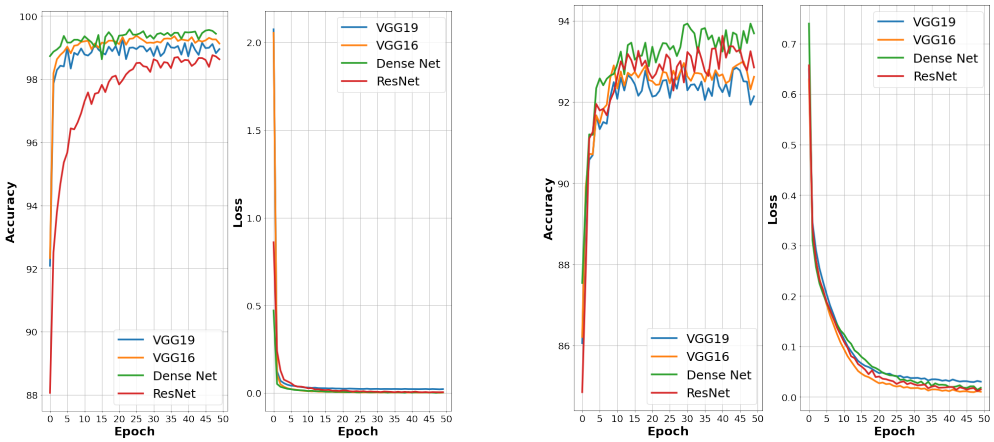
For simulating the system, the main goal of the tests is to see the differences between VGG16 and VGG19 classification models for the CIFAR10/MNIST/FMNIST dataset. The models were both written in Python using Pytorch. Training was completed using CUDA and GPU acceleration while the CPU was utilized testing and validation. The parameters for both networks can be seen below. These datasets were choosen as they are small and allow us to train the models in the alloted time given.

Parameter	Value
Learning Rate	1e-4
Batch Size	64
Training Size	40,000 Images
Validation Size	10,000 Images
Testing Size	10,000 Images

Table 1: Model Parameters for VGG16, VGG19, DenseNet, and ResNet

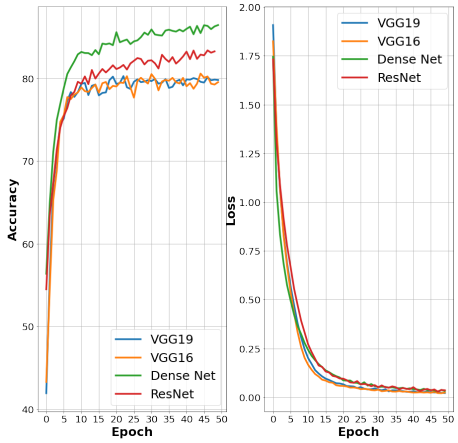
3.1 Training Simulation: Challenges and Results

In simulating with different data sets over different network architectures comes challenges to ensure all the systems are training on the same input data. All the networks by default take in a three channel 224x224 image. This becomes a problem when the dataset does not conform to that structure. The FashionMNIST and MNIST dataset are both 28x28 pixel single channel (grayscale) image databases.



((a)) Training Accuracy on MNIST Dataset

((b)) Training Accuracy on FashionMNIST Dataset



((c)) Training Accuracy on CIFAR10 Dataset

Figure 2: Training Accuracy and Loss results of all datasets on all models

To allow the networks selected to work with the system the images are all enlarged to 224x244 pixels. For testing purposes, the MNIST and FashionMNIST datasets are handled differently when training on the networks. The MNIST dataset is unchanged with the input to the networks all being changed to take in a single channel instead of the default three. The FashionMNIST dataset is modified using a transform to become a stack of the three of the same grayscale image. This in turn allows it to be input into the unmodified networks that accept the three channel input.

Due to the size and nature of the architectures of the networks used, an issue with testing and simulating is the training time required. Overall, over 45 hours of GPU compute were dedicated to just the training phase on a Nvidia RTX 3070. These are also relatively smaller datasets in comparison with ImageNet (∼10x larger) meaning that the training time will only increase with the larger dataset collection.

Model	Accuracy
VGG16	99.38%
VGG19	98.65%
ResNet	92.24%
DenseNet	93.21%

((a)) Fashion MNIST Dataset

Model	Accuracy
VGG16	99.30%
VGG19	99.20%
ResNet	99.01%
DenseNet	99.50%

((b)) MNIST Dataset

Model	Accuracy
VGG16	99.38%
VGG19	98.65%
ResNet	92.24%
DenseNet	93.21%

((c)) CIFAR 10 Dataset

Table 2: Testing Accuracy of all the Datasets on all the Models

3.2 Results with CFAR10 dataset

Firstly we look at the CIFAR10 dataset [3], the results speak for themselves. The extra 3 convolutional layers of the VGG19 network show no increase in performance in accuracy or loss over the 50 epochs of training. Moving to the training dataset, the validation scores are within 0.1 percent of each other. This proves the point that both networks perform identically with the CIFAR10 dataset and the sparsity of the VGG16 network can be exploited. The training results also show that DenseNet stands out with respect to the other models which shows that all image information were successfully pulled through all the dense layers during training with the help of subsequent feed-backs from the previous layers. The testing results also confirms this fact for CIFAR10 dataset.

3.3 Results with MNIST dataset

Next we perform our analysis using the MNIST dataset [4]. Here we observe that the ResNet has the least performance which shows that with increased layers and just a feedback from the feedback from previous layer is not enough for classification using MNIST dataset, however when feedback is introduced in all the subsequent layers as in DenseNet, it helps in performance as shown in the training result in Fig. 2(b). The models however show very similar performance during training. Thus for MNIST dataset a single model like VGG16 can be used to get satisfactory classification results.

3.4 Results with Fashion MNIST dataset

Finally, we look at the Fashion MNIST dataset [5]. Here we observe that the training performance increase with the increase model complexity by increasing more layers and feedback, hence DenseNet has the highest performance followed by ResNet and the VGG architectures as seen in Fig. 2(c). However, in testing we observe that VGG16 has the best performance. This shows that although a more complicated architecture can capture more detailed features in the Fashion MNIST dataset, a simpler model like VGG16 can still classify images better than the others.

4 Improved Existing Architecture: Modified VGG

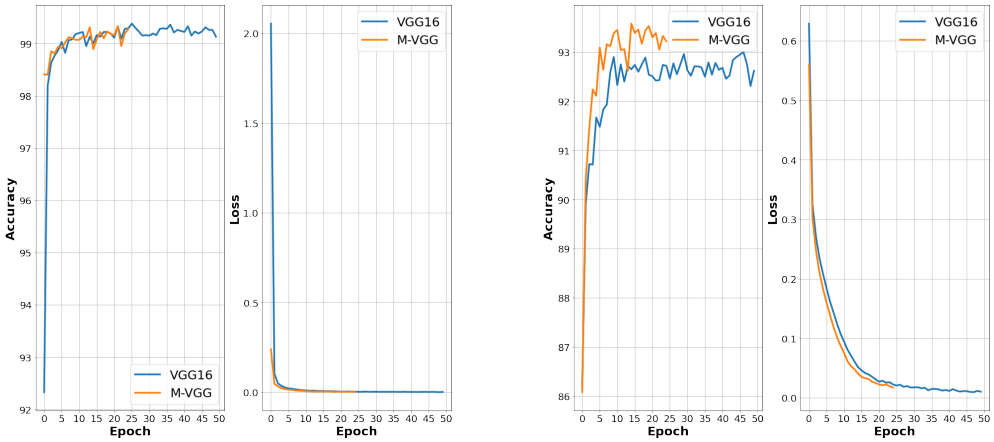
4.1 Introducing Sparsity

As talked about in section 3.1, a large time is spent on training these complex models on relatively small datasets. This leads to the idea that researching sparsity could lead to some increases in performance without decreasing accuracy. The results from these simulations show that removing those six extra convolution layers from the VGG19 network did not sacrifice accuracy which leads to the idea of exploring how many more layers we could possibly remove while maintaining the results with negligible differences.

The next step is determining which layers of the system to remove to introduce the sparsity. The VGG16 network contains six sizes of images that section off the layers in the model. The image starts at the base 224x244 then goes through a series of convolutions to extract all the important features into the final size of 7x7. To introduce sparsity into the VGG 16 network, the idea was not to change the sizes of the architecture meaning the system would still go from a 224x224 pixel to a 7x7 pixel image. To achieve this, six layers are removed from multiple different size layers as seen in figure 1.

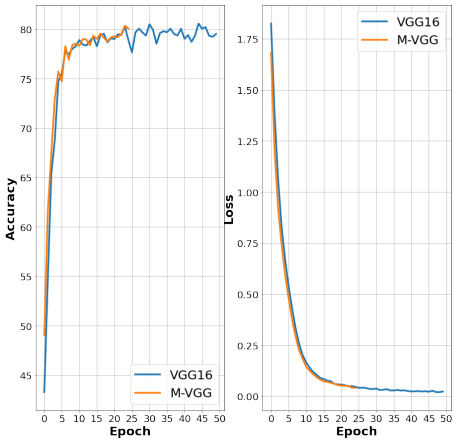
4.2 Results of Modified VGG16

The results of the modified VGG16 network show that the introduced sparsity is able to keep the same accuracy in all the datasets tested while being more efficient. The modified VGG16 model is trained with the same parameters as before and shows promising results when compared in all three datasets. The modified VGG16 network was only trained on 25 epochs as from the preliminary runs it is shown that convergence occurs there. The modified VGG16 network took 2 hours to train in comparison to 4 hours (scaled for 25 epochs). This provided the modified VGG16 network with a training speedup of 4x without sacrificing any accuracy.



((a)) Training Accuracy on MNIST Dataset

((b)) Training Accuracy on FashionMNIST Dataset



((c)) Training Accuracy on CIFAR10 Dataset

Figure 3: Training Accuracy and Loss results comparing VGG16 and M-VGG

Dataset	Model	Accuracy
CIFAR10	VGG16	79.55%
CIFAR10	M-VGG	79.98%
MNIST	VGG16	99.30%
MNIST	M-VGG	99.43%
FashionMNIST	VGG16	99.38%
FashionMNIST	M-VGG	99.38%

Table 3: Testing Accuracy for VGG16/M-VGG on CIFAR10, MNIST and Fashion MNIST Datasets

5 Conclusion and future work

Our project was able to benchmark the VGG16 and VGG19 architectures along with DenseNet and ResNet for the MNIST, Fashion MNIST, and CIFAR datasets. We were able able to show that introducing sparsity is a good way to gain efficiency in smaller datasets. The modified VGG network was able to perform equally with the VGG16 network while using fewer resources and lesser time to train. The most interesting path of future work exploration involves implementing a transfer learning mapper. This mapper would take the larger trained model and create a smaller untrained modified model. This untrained model would be instantiated with the corresponding mapped weights from the larger trained model. Ideally, the modified model would then be able to inherit the learned features from the dataset and perform well without training in itself.

6 Contributions

1. Problem Setup: Antorip Sahu, Amit Dutta
2. Research: Amit Dutta, Antorip Sahu, Armen Kasparian
3. Model Creation: Armen Kasparian
4. Debugging: Armen Kasparian, Amit Dutta, Antorip Sahu
5. Testing: Armen Kasparian, Amit Dutta, Antorip Sahu
6. Observations: Amit Dutta, Antorip Sahu, Armen Kasparian
7. Reporting: Antorip Sahu, Amit Dutta, Armen Kasparian

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[4] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[5] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.