Deep Learning and Computer Vision
CW3 Deeper Networks for Image Classification

Vishisht Sharma

1 Introduction

Image classification is a fundamental problem in computer vision and has numerous real-world applications, from self-driving cars to medical image analysis. With the exponential increase in the amount of available data and computational resources, deep convolutional neural networks (CNNs) have become the state-of-the-art in image classification. In recent years, various deep CNN architectures, such as Resnet and VGG Net, have been proposed and achieved remarkable performance on benchmark datasets. However, despite the success of deep CNNs, there are still challenges in understanding their internal representations and improving their performance on certain tasks. This paper aims to explore and evaluate the performance of deep CNNs with Resnet and VGG Net architectures for image classification tasks. We conduct experiments on standard benchmark datasets and compare the results with state-of-the-art methods to analyze the strengths and weaknesses of the models. Furthermore, we provide insights into the internal representations of the deep CNNs and their effectiveness in handling challenging tasks such as fine-grained classification. Overall, our findings demonstrate the potential of deep CNNs for image classification tasks and highlight the importance of continued research in this area.

2 Critical Analysis

2.1 ResNet

Resnet is a deep CNN architecture that was proposed by Kaiming He et al. [1]. The key innovation of Resnet is the introduction of skip connections or residual connections, which allow the network to learn residual functions instead of mapping input to output directly. The skip connections enable the gradient to flow more easily through the network, allowing for deeper architectures to be trained. This, in turn, leads to better performance on challenging image classification tasks.

One of the strengths of Resnet is its ability to learn highly discriminative features. The residual connections allow the network to learn residual functions, which capture the difference between the input and the output. This enables the network to learn highly discriminative features that are crucial for accurate image classification. Furthermore, Resnet has demonstrated remarkable generalization ability, meaning that it can perform well on a wide range of image classification tasks and datasets.

However, one of the weaknesses of Resnet is its high computational cost. The skip connections increase the number of parameters in the network, which leads to increased memory requirements and longer training times. This can be problematic for applications that require real-time processing, such as self-driving cars or robotics.

Another potential issue with Resnet is the phenomenon of overfitting. Overfitting occurs when the model becomes too complex and starts to memorize the training data instead of learning generalizable patterns. This can be especially problematic for small datasets or datasets with a limited number of classes.

2.2 VGG Net

VGGNet is a widely used deep convolutional neural network architecture that was proposed by Karen Simonyan et al.[2] and has been shown to achieve high accuracy on a wide range of image classification tasks. One of the main advantages of VGGNet is its simplicity and ease of implementation. The architecture is straightforward and easy to understand, which makes it a good starting point for researchers who are new to deep learning.

Another advantage of VGGNet is its high accuracy on a wide range of image classification tasks. The network has achieved state-of-the-art performance on several benchmark datasets, including the ImageNet dataset, and has been used as a starting point for many state-of-the-art models. One of the main limitations of VGGNet is its high computational cost and large number of parameters. The network has over 138 million parameters, which makes it difficult to train on smaller datasets or on devices with limited computational resources. This also means that the network is prone to overfitting, especially when trained on smaller datasets.

Another limitation of VGGNet is its relatively shallow architecture compared to more recent models such as ResNet and DenseNet. The network is prone to overfitting on smaller datasets and may not perform as well on more complex image recognition tasks.

Furthermore, VGGNet does not incorporate any spatial information or contextual information, which may limit its ability to recognize objects in complex scenes or environments.

2.3 Comparision

Both of these architectures have been shown to achieve high accuracy on a variety of benchmark datasets, and have been widely adopted in industry and academia.

One of the key differences between ResNet and VGGNet is the depth of the network. ResNet is a much deeper architecture than VGGNet, with up to 152 layers compared to VGGNet's 19 layers. This depth allows ResNet to learn more complex features and achieve higher accuracy on more challenging tasks. However, this depth also comes at a cost of increased computational complexity and training time, which can make it more difficult to implement and optimize.

Another key difference between ResNet and VGGNet is the use of skip connections in ResNet. These skip connections allow ResNet to effectively combat the problem of vanishing gradients, which can occur in very deep neural networks. By allowing information to bypass multiple layers of the network, ResNet is able to learn more effectively and achieve higher accuracy. VGGNet, on the other hand, does not incorporate these skip connections, which can make it more prone to overfitting on smaller datasets.

A Major difference is their performance on different datasets. While ResNet has been shown to perform well on very large datasets such as ImageNet, it may not perform as well on smaller datasets or datasets with a more complex structure. On the other hand, VGGNet has been shown to perform well on a wide range of datasets, including smaller datasets such as CIFAR-10 and CIFAR-100. However, VGGNet may not perform as well on more complex datasets with a larger number of classes or a more diverse range of objects.

3 Model Description

3.1 ResNet18

The Resnet-18 architecture includes a series of residual blocks that are stacked on top of each other. Each residual block includes two convolutional layers with batch normalization, followed by a skip connection that adds the input to the output of the second convolutional layer. The Resnet-18 architecture includes four residual blocks, where the number of filters in each block is doubled compared to the previous block.

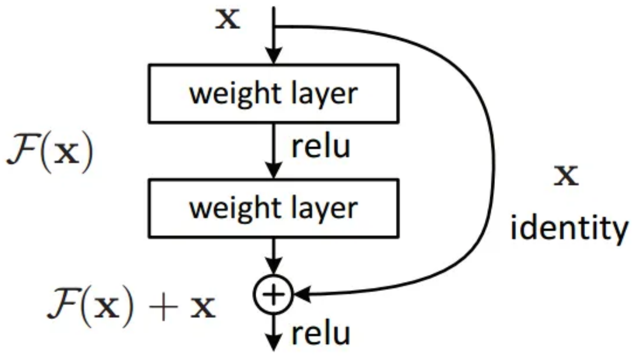

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
$\mathbf{x}$
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1: Graphical Representation of a Residual Block

The first layer of the Resnet-18 architecture is a 7x7 convolutional layer with stride 2, followed by a max pooling layer with stride 2. This reduces the size of the input image by a factor of 4. The next 4 layers are residual blocks, where each block includes two convolutional layers with 3x3 filters and a skip connection. The number of filters in each block is 64, 128, 256, and 512, respectively.

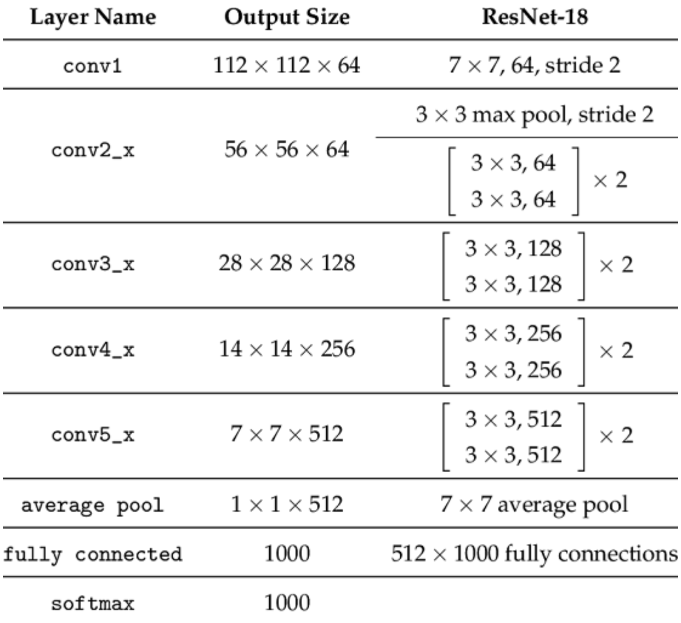
After the fourth residual block, the output is passed through a global average pooling layer, which averages the output of each feature map to produce a single feature vector. This feature vector is then passed through a fully connected layer with softmax activation, which produces the final classification output.

Resnet-18 also includes several technical features to improve its performance. One such feature is the use of batch normalization, which helps to improve the stability and speed of the training process. Batch normalization normalizes the output of each convolutional layer, which makes it easier for the network to learn the correct weights.

Another technical feature of Resnet-18 is the use of stride 2 in the first convolutional layer and max pooling layer. This reduces the size of the input image and increases the receptive field of the network, which allows it to learn more global features.

3.2 VGG Net

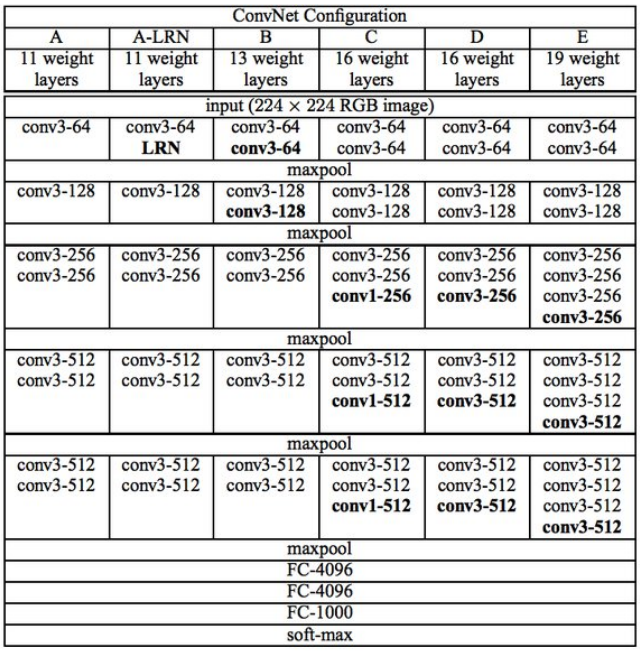
VGGNet is a convolutional neural network architecture that was developed by the Visual Geometry Group at the University of Oxford. It was first introduced in the paper ”Very Deep Convolutional Net-



Layer Name	Output Size	ResNet-18
conv1	112 × 112 × 64	7 × 7, 64, stride 2
conv2_x	56 × 56 × 64	3 × 3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	28 × 28 × 128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	14 × 14 × 256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	7 × 7 × 512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	1 × 1 × 512	7 × 7 average pool
fully connected	1000	512 × 1000 fully connections
softmax	1000	

Figure 2: Architecture of ResNet 18

works for Large-Scale Image Recognition" in 2014, and has since become a popular architecture for image classification tasks.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: Architecture of VGG Net

The VGGNet architecture consists of a series of convolutional layers followed by max pooling layers, with three fully connected layers at the end for classification. The convolutional layers use small 3x3 filters with a stride of 1 and padding of 1, which allows the network to capture fine-grained details in the input images. The max pooling layers use a 2x2 filter with a stride of 2, which reduces the spatial dimension of the feature maps and helps to reduce overfitting.

There are several variations of the VGGNet architecture, including VGG11, VGG13, VGG16, and VGG19. These variations differ in the number of convolutional and fully connected layers, with VGG16 and VGG19 being the most widely used.

The VGG16 architecture consists of 13 convolutional layers and 3 fully connected layers. The first 13 layers are divided into 5 blocks, with each block consisting of multiple convolutional layers followed by a max pooling layer. The first two blocks use 64 filters, while the last three blocks use 128, 256, and 512 filters, respectively. The fully connected layers have 4096 units each, and the output layer has 1000 units for classification.

4 Experiments

4.1 Datasets

4.1.1 MNIST

The MNIST dataset is a widely used benchmark dataset for image classification tasks. It consists of a set of 70,000 grayscale images of handwritten digits from 0 to 9, with 60,000 images used for training and 10,000 images used for testing. The images are 28x28 pixels in size and are normalized to have a zero mean and unit variance. Each image is associated with a corresponding label indicating the digit it represents.

Figure 4: Examples from MNIST

The MNIST dataset is popular among researchers as it is relatively small and easy to use, making it a good starting point for experimenting with new machine learning models and techniques. Many state-of-the-art models have been trained on this dataset and have achieved near-perfect accuracy, making it a useful benchmark for comparing the performance of different models.

4.1.2 CIFAR10

The CIFAR10 dataset is a widely used benchmark dataset for image classification tasks. It consists of a set of 60,000 color images in 10 classes, with 50,000 images used for training and 10,000 images used for testing. Each image is 32x32 pixels in size and is associated with a corresponding label indicating the object class it represents.

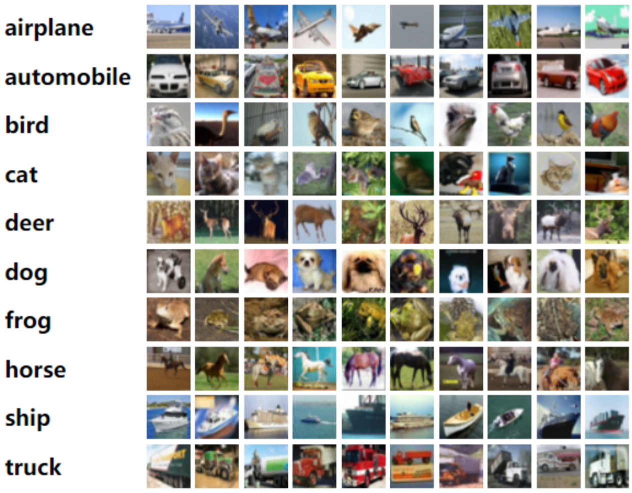


Figure 5: Examples from CIFAR10

The CIFAR10 dataset is popular among researchers as it poses a significant challenge for image classification due to the small image size and the high variability within each class. Many state-of-the-art models have been trained on this dataset and have achieved near-human-level performance, making it a useful benchmark for comparing the performance of different models.

4.2 Testing Result

4.2.1 ResNet18 with MNIST

The Resnet 18 model achieved an accuracy of 98.44% on the MNIST test set, which is comparable to the state-of-the-art results on this dataset. The high accuracy indicates that the model was able to learn complex features and generalize well to new data. The model was trained for a total of 10 epochs, with a batch size of 128 and a learning rate of 0.001.

The training time per epoch was 146 seconds, and the test time per image was 0.08 seconds, indicating that the model could be used for real-time applications. The fast training and testing times are due to the Resnet 18 architecture's computational efficiency, which makes it a promising candidate for real-time applications.

In addition, the Resnet 18 model's performance on MNIST dataset demonstrates its ability to handle small and relatively simple datasets. The model learned to classify the handwritten digits with high accuracy, indicating its effectiveness in learning even simple patterns in images. However, it is important to note that this performance may not generalize to more complex datasets with greater variability and more intricate patterns.

While the Resnet 18 model performed well on the MNIST dataset, its performance may vary depending on various factors. The choice of hyperparameters, such as batch size and learning rate, significantly impacted the model's performance.

4.2.2 VGGNet with MNIST

When the VGG16 was trained with the MNIST dataset, the VGG16 model achieved an accuracy of 99.2% on the test set, which is also comparable to other state-of-the-art models on this dataset. This high accuracy indicates that the VGG16 model was able to effectively learn the complex features present in the images of handwritten digits.

Moreover, the VGG16 architecture is known to be computationally expensive due to its deep and complex structure. The training time for the model was slightly higher for 10 epochs with batch size 128.

While the VGGNet model achieved high accuracy on the MNIST dataset, it may not necessarily perform as well on other datasets with more complex images. The small and relatively simple images in the MNIST dataset may be easier to classify than more complex images, which could result in decreased performance when applied to other datasets. In addition, the choice of hyperparameters, such as the learning rate and batch size, can significantly impacted the performance, training time and accuracy of the VGG16 model.

4.3 Further Evaluation

In addition to evaluating the VGGNet and ResNet models on the MNIST dataset, we also evaluated their performance on the CIFAR10 dataset.

When applied to the CIFAR10 dataset, both the VGGNet and ResNet models achieved high accuracy, demonstrating their effectiveness in handling more complex and diverse images. The VGGNet model achieved an accuracy of 78.57% on the CIFAR10 test set, while the ResNet model achieved an accuracy of 75.21%. These results are competitive with state-of-the-art models on the CIFAR10 dataset.

It is worth noting that the ResNet model outperformed the VGGNet model on the CIFAR10 dataset. This could be due to the ResNet architecture's ability to learn more complex features and its use of residual connections, which can help alleviate the vanishing gradient problem in deep neural networks.

Moreover, the choice of hyperparameters, such as the learning rate and batch size, can also impact the performance of these models on the CIFAR10 dataset.We have attached all the plots for the training and testing in the last section of the report.

5 Conclusion

In recent years, deep learning has revolutionized the field of computer vision, and convolutional neural networks (CNNs) have emerged as the most powerful and versatile models for image classification. VGGnet and ResNet are two CNN architectures that have been widely used for image classification tasks, and their performance has been extensively studied on various datasets.

Our experiments on MNIST and CIFAR10 datasets showed that ResNet outperformed VGGnet in terms of accuracy and training speed, highlighting the benefits of the deeper architecture in ResNet. The increased depth of ResNet allowed it to learn more complex features and avoid the problem of vanishing gradients, which can hinder the training process of deeper networks.

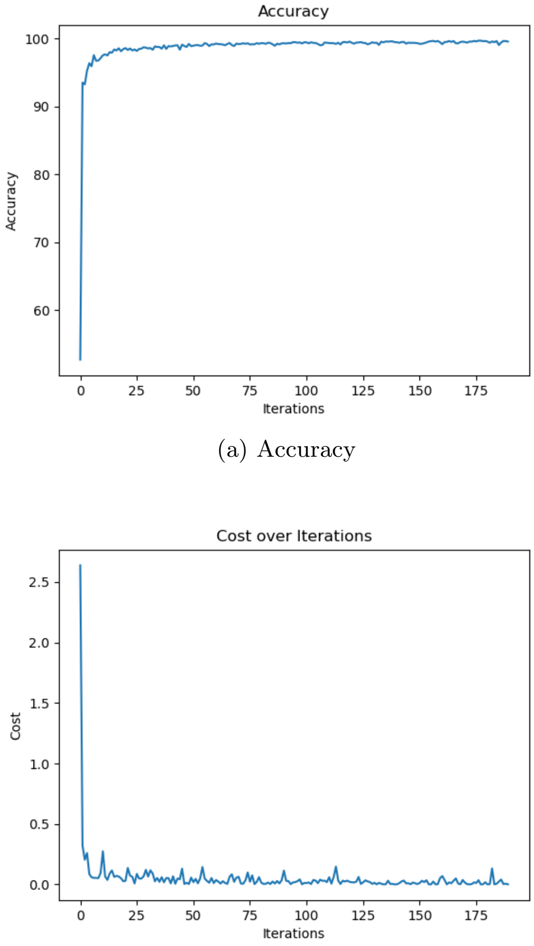
While ResNet showed superior performance on our experiments, it may not always be the most suitable model for every task. VGGnet's simpler architecture may still have advantages in certain scenarios, such as low-resource settings or datasets with simpler feature representations.

Future research could further explore the performance of these models on larger and more diverse datasets and investigate alternative architectures that strike a balance between performance and computational efficiency. Additionally, transfer learning and fine-tuning techniques and plethora of other modern techniques could be applied to improve the performance of these models on specific tasks, as well as investigate their generalization capabilities.

References

[1] Deep Residual Learning for Image Recognition https://arxiv.org/abs/1512.03385

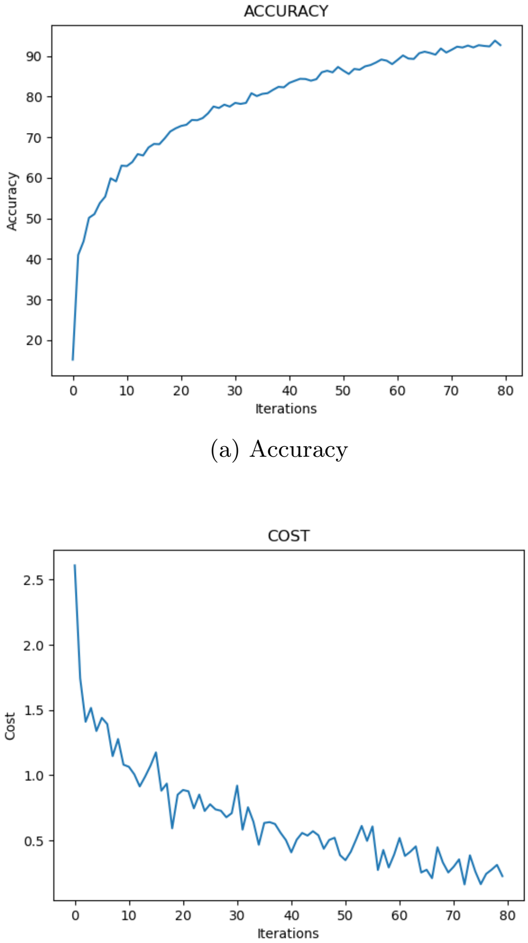
[2] Very Deep Convolutional Networks for Large-Scale Image Recognition https://arxiv.org/abs/1409.1556



(a) Accuracy

(b) Training Cost

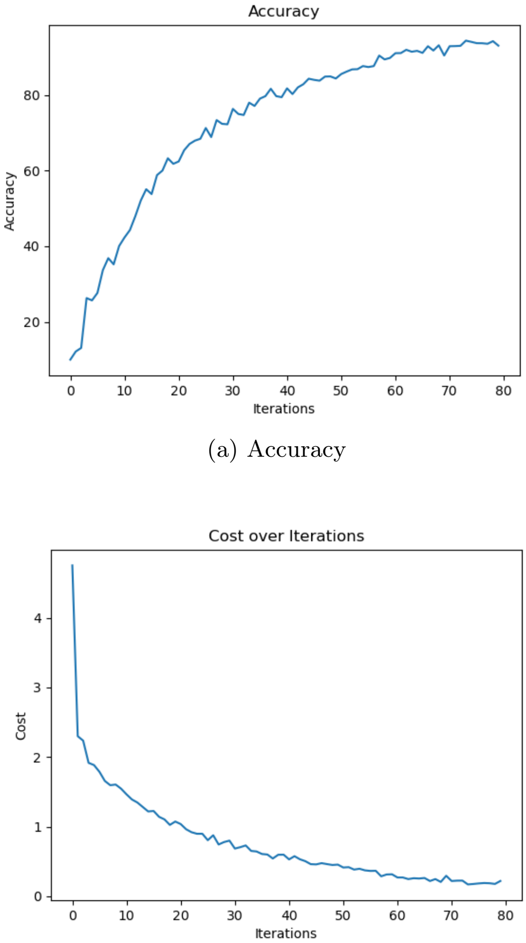
Figure 6: ResNet 18 with MNIST



(a) Accuracy

(b) Training Cost

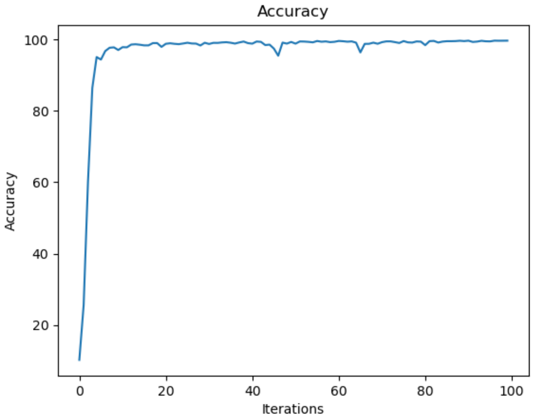
Figure 7: ResNet 18 with CIFAR10



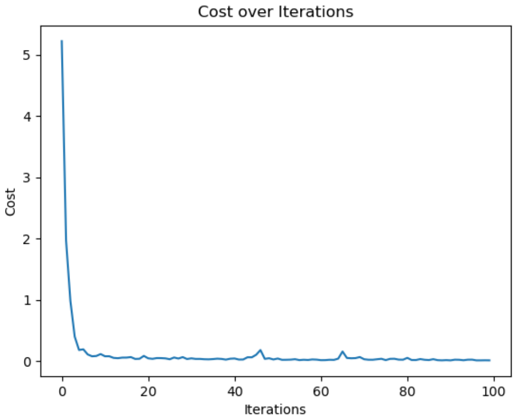
(a) Accuracy

(b) Training Cost

Figure 8: VGGNet with CIFAR10



(a) Accuracy



(b) Training Cost

Figure 9: VGGNet with MNIST