Image Classification Using AlexNet and MobileNet: A Comparative Study

Hanbei Xiong
University of California, San Diego; hax001@ucsd.edu

Abstract

This research paper focuses on evaluating the performance of three different convolutional neural network (CNN) architectures, namely 3-layer CNN, AlexNet, and MobileNet, for image classification on the CIFAR-10 dataset. In addition to comparing their classification accuracy, we also conduct some experiments in image enhancement to see if the performance of the models can be further improved. The results show that AlexNet and 3-layer CNN outperform the MobileNet in terms of classification accuracy, with AlexNet being the most accurate. Furthermore, we observe that image enhancement techniques can improve the performance of all three models, with the best results obtained when using a combination of image enhancement techniques. These findings provide insights into the potential benefits of using different CNN architectures and image enhancement techniques for image classification tasks.

***Keywords*—** Convolutional Neural Network, Image Classification, Image Enhancement

Introduction

Image classification is a crucial task in computer vision, with applications in fields such as autonomous driving, object recognition, and medical diagnosis. Neural networks have become the preferred method for image classification due to their ability to learn complex features from large datasets. The advent of deep learning has led to the development of increasingly complex neural network architectures that can achieve state-of-the-art performance on challenging image classification tasks.

The convolutional neural network (CNN) is a deep learning architecture that has revolutionized image classification. CNNs consist of multiple layers of convolutional and pooling operations, followed by one or more fully connected layers that perform classification. In recent years, several popular CNN architectures have been proposed, including alexnet and mobilenet, which have been shown to outperform traditional 3 layer CNNs on various image classification tasks.

In this research paper, we aim to compare the performance of traditional 3-layer CNN, AlexNet, and MobileNet on the cifar-10 dataset, which is a widely used benchmark for image classification. Our goal is to evaluate the trade-off between model complexity and accuracy, as well as the efficiency of each architecture in terms of training and inference time. By comparing the performance of these three architectures, we hope to provide insights into the strengths and weaknesses of each model and help guide the selection of appropriate models for different image classification tasks.

Related Work

Convolutional Neural Networks (CNNs) have been widely used for image classification tasks due to their ability to capture spatial dependencies in images. One of the earliest and most popular CNN models is AlexNet, which achieved significant performance improvements on the ImageNet dataset in 2012 [3]. Since then, various CNN models with deeper architectures have been proposed to further improve performance. However, with the increasing popularity of mobile devices, efficient CNN architectures that can run on mobile devices have become a research focus. One such architecture is MobileNet, proposed by Howard et al. in 2017, which uses depthwise separable convolutions to reduce computation while maintaining accuracy [1]. In this study, we compared the performance of three different CNN models, including a 3-layer CNN, AlexNet, and MobileNet, to evaluate their effectiveness in image classification tasks. Other recent CNN models with improved performance, such as VGG and ResNet, can also be considered for future studies.

Method

Data Description

The CIFAR-10 dataset is a commonly used benchmark dataset in computer vision research[2]. It consists of 60,000 32x32 color images in 10 categories, with 6,000 images per category. The categories include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is split into 50,000 training images and 10,000 testing images. Each image in the CIFAR-10 dataset is a RGB image, with 3 channels (red, green, and blue) and 8-bit values per channel (0-255). This results in a total of 32 x 32 x 3 = 3072 pixels per image. The small size of the images and the variability in object position and scale make the CIFAR-10 dataset challenging for image classification tasks. To prepare the dataset for our experiments, we applied standard preprocessing techniques, including normalization and data augmentation.

We used the CIFAR-10 dataset to train and test our machine learning models for image classification tasks. The dataset provided a diverse set of images with a variety of objects and backgrounds, allowing us to evaluate the performance of our models on a challenging and realistic dataset.

3-layer CNN

A traditional 3-layer convolutional neural network (CNN) is a simple and widely used model for image classification tasks. The

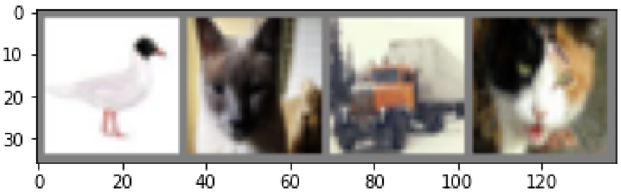


Figure 1: Cifar-10 Input Image Examples

model consists of three layers: a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer applies a set of filters to the input image to extract relevant features. The pooling layer reduces the spatial dimensionality of the output of the convolutional layer. The fully connected layer takes the output of the pooling layer and maps it to the class probabilities using a softmax activation function. The model is trained using the categorical cross-entropy loss function and optimized using stochastic gradient descent.

AlexNet

AlexNet is a deeper and more complex CNN architecture than the traditional 3-layer CNN. It consists of eight layers, including five convolutional layers, two fully connected layers, and a softmax output layer. The model uses larger filter sizes and stride values than the traditional 3-layer CNN, allowing it to learn more complex features from the input images. AlexNet also introduces the use of ReLU activation functions and dropout regularization to improve model performance and prevent overfitting. The model is trained using the categorical cross-entropy loss function and optimized using stochastic gradient descent with momentum.

MobileNet

MobileNet is a lightweight and efficient CNN architecture designed for mobile and embedded devices. The model uses depthwise separable convolutions to reduce the number of parameters and computational complexity while maintaining high accuracy. Depthwise separable convolutions apply a depthwise convolution to each channel of the input separately and then apply a pointwise convolution to combine the outputs. This reduces the computational cost of the convolutional layer without sacrificing accuracy. The model also uses linear bottlenecks and a shortcut connection to further improve performance. The model is trained using the categorical cross-entropy loss function and optimized using stochastic gradient descent.

Experiment

In this section, the paper performs data prepossessing, model training and hyper parameter tuning.

Data Prepossessing

This study transforms the input by resizing the images to a size of 227x227 pixels, converting the images to PyTorch tensors and normalizing them with a mean of 0.5 and standard deviation of 0.5 for each color channel. This normalization step helps to ensure that the input features have similar scales, which makes the optimization process easier and faster.

Training and Hyper Parameter Tuning

Three models were trained using a similar approach, which involves looping over the dataset multiple times and updating the model's parameters using stochastic gradient descent(SGD) optimization. Within each epoch, the training process involves iterating over the training data in mini-batches and computing the forward and backward steps of the neural network using the inputs and labels of the mini-batch. The forward step computes the model's predictions for the given inputs, and the backward step computes the gradients of the loss function with respect to the model's parameters. After computing the gradients, the optimizer updates the model's parameters to minimize the loss function.

The study also tried other optimization method like Adaptive Moments Estimate (Adam). In comparison, although SGD was more challenging to achieve learning, it eventually performed better outcome than Adam.

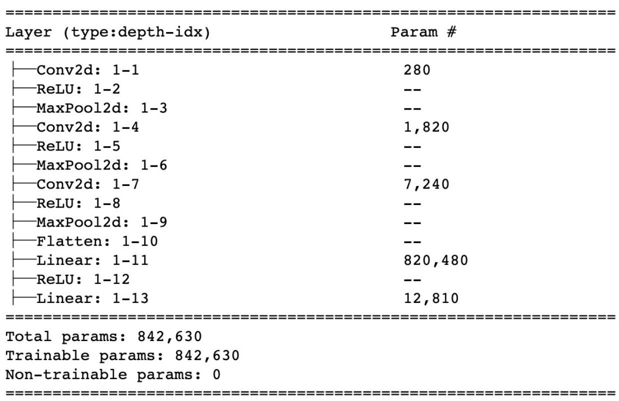
Besides, for all models, the learning rate was modified within the range of 0.001 to 0.0005, but no significant improvements in accuracy were observed. As such, a learning rate of 0.001 was selected for all models for efficiency.

In addition, the activation function was compared between the popular ReLU and sigmoid functions for the 3-layer CNN and AlexNet models. While the AlexNet model showed minimal difference in performance, the 3-layer CNN model exhibited lower accuracy with sigmoid activation. For the MobileNet model, however, due to the high computational requirements for training, only ReLU activation was used to simplify the experimental setup. Furthermore, the number of layers in the MobileNet model was reduced compared to Howard's original study to accommodate for hardware limitations.

3-layer CNN Structure

The model consists of three convolutional blocks, each followed by a ReLU activation function and a max pooling layer. The first convolutional block takes as input images with 3 color channels and produces 10 output feature maps. The second block takes 10 feature maps as input and produces 20 output feature maps. The third block takes 20 feature maps as input and produces 40 output feature maps.

After the convolutional blocks, the output is flattened and passed through two fully connected layers with ReLU activation functions. The first fully connected layer has 1280 neurons and takes as input the flattened output of the third convolutional block. The second fully connected layer has 10 neurons, which corresponds to the 10 possible classes in the CIFAR-10 dataset.



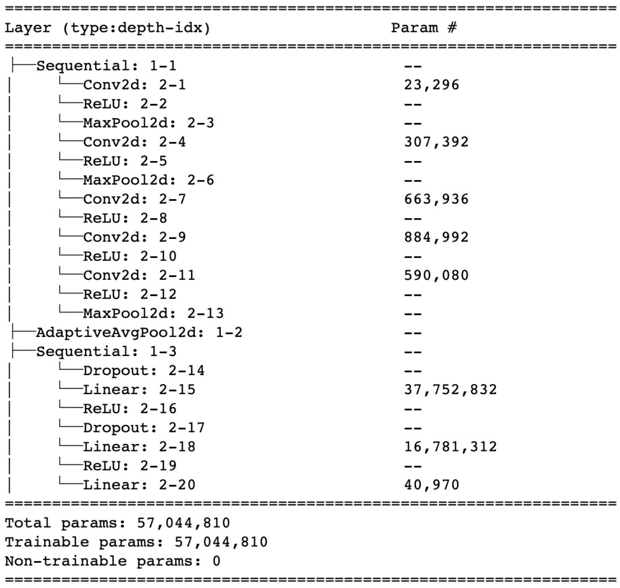
```
=================================================================
Layer (type:depth-idx)                   Param #
=================================================================
├─Conv2d: 1-1                            280
├─ReLU: 1-2                              --
├─MaxPool2d: 1-3                         --
├─Conv2d: 1-4                            1,820
├─ReLU: 1-5                              --
├─MaxPool2d: 1-6                         --
├─Conv2d: 1-7                            7,240
├─ReLU: 1-8                              --
├─MaxPool2d: 1-9                         --
├─Flatten: 1-10                          --
├─Linear: 1-11                           820,480
├─ReLU: 1-12                             --
├─Linear: 1-13                           12,810
=================================================================
Total params: 842,630
Trainable params: 842,630
Non-trainable params: 0
=================================================================
```

Figure 2: 3-layer CNN Structure

AlexNet Structure

The architecture consists of two main parts: feature extraction and classification. The feature extraction part has several convolutional layers with ReLU activation functions and max pooling layers to reduce the spatial dimensions of the feature maps. The output of the last convolutional layer is fed to a fully connected layer with ReLU activation functions, followed by two dropout layers and another fully connected layer with softmax activation to produce the final classification scores.

The architecture is based on the popular AlexNet architecture, with some modifications such as the use of adaptive average pooling instead of fixed-size max pooling, and fewer filters in each convolutional layer.



```
================================================================
Layer (type:depth-idx)                   Param #
================================================================
├─Sequential: 1-1                         --
│    └─Conv2d: 2-1                        23,296
│    └─ReLU: 2-2                          --
│    └─MaxPool2d: 2-3                     --
│    └─Conv2d: 2-4                        307,392
│    └─ReLU: 2-5                          --
│    └─MaxPool2d: 2-6                     --
│    └─Conv2d: 2-7                        663,936
│    └─ReLU: 2-8                          --
│    └─Conv2d: 2-9                        884,992
│    └─ReLU: 2-10                         --
│    └─Conv2d: 2-11                       590,080
│    └─ReLU: 2-12                         --
│    └─MaxPool2d: 2-13                    --
├─AdaptiveAvgPool2d: 1-2                  --
├─Sequential: 1-3                         --
│    └─Dropout: 2-14                      --
│    └─Linear: 2-15                       37,752,832
│    └─ReLU: 2-16                         --
│    └─Dropout: 2-17                      --
│    └─Linear: 2-18                       16,781,312
│    └─ReLU: 2-19                         --
│    └─Linear: 2-20                       40,970
================================================================
Total params: 57,044,810
Trainable params: 57,044,810
Non-trainable params: 0
================================================================
```

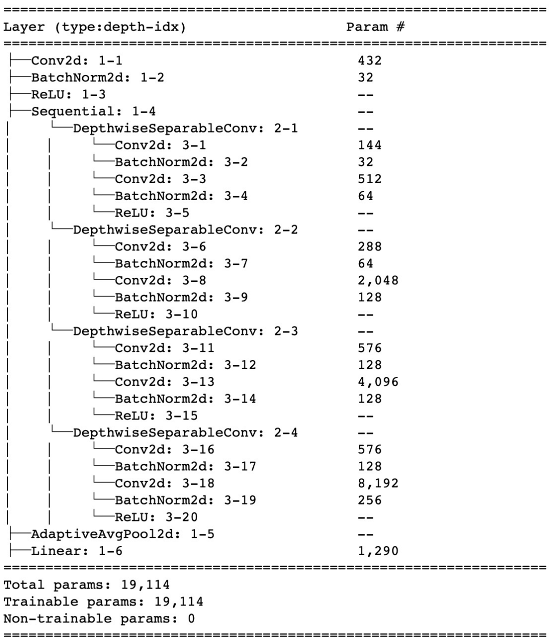
Figure 3: AlexNet Structure

MobileNet Structure

The network consists of several convolutional layers, each followed by batch normalization and ReLU activation functions. The intermediate layers use a specialized type of convolution called ”depthwise separable convolution”, which splits the convolution process into two steps: a depthwise convolution that applies a separate filter to each input channel, followed by a pointwise convolution that combines the results of the depthwise convolution. Finally, the network has a global average pooling layer that averages the values in each feature map, followed by a fully connected layer to output the final classification.

Results

In this section, three models were evaluated for their performance in this task. The results indicate that the 3-layer CNN model had a decent level of performance. However, the AlexNet model outperformed the CNN model, suggesting that it may be a more effective approach for the given task. On the other hand, the MobileNet model had relatively low accuracy when compared to the other two models, indicating that it may not be as suitable for this particular task. Regarding to time efficiency in model training, 3-layer CNN and AlexNet model spent much less time than MobileNet.



```
================================================================
Layer (type:depth-idx)                   Param #
================================================================
├─Conv2d: 1-1                             432
├─BatchNorm2d: 1-2                        32
├─ReLU: 1-3                               --
├─Sequential: 1-4                         --
│    └─DepthwiseSeparableConv: 2-1        --
│    │    └─Conv2d: 3-1                   144
│    │    └─BatchNorm2d: 3-2              32
│    │    └─Conv2d: 3-3                   512
│    │    └─BatchNorm2d: 3-4              64
│    │    └─ReLU: 3-5                     --
│    └─DepthwiseSeparableConv: 2-2        --
│    │    └─Conv2d: 3-6                   288
│    │    └─BatchNorm2d: 3-7              64
│    │    └─Conv2d: 3-8                   2,048
│    │    └─BatchNorm2d: 3-9              128
│    │    └─ReLU: 3-10                    --
│    └─DepthwiseSeparableConv: 2-3        --
│    │    └─Conv2d: 3-11                  576
│    │    └─BatchNorm2d: 3-12             128
│    │    └─Conv2d: 3-13                  4,096
│    │    └─BatchNorm2d: 3-14             128
│    │    └─ReLU: 3-15                    --
│    └─DepthwiseSeparableConv: 2-4        --
│    │    └─Conv2d: 3-16                  576
│    │    └─BatchNorm2d: 3-17             128
│    │    └─Conv2d: 3-18                  8,192
│    │    └─BatchNorm2d: 3-19             256
│    │    └─ReLU: 3-20                    --
├─AdaptiveAvgPool2d: 1-5                  --
├─Linear: 1-6                             1,290
================================================================
Total params: 19,114
Trainable params: 19,114
Non-trainable params: 0
================================================================
```

Figure 4: MobileNet Structure

	3-layer CNN	AlexNet	MobileNet
Overall Accuracy	0.70	0.78	0.57
plane	0.78	0.77	0.64
Car	0.85	0.87	0.77
bird	0.57	0.69	0.38
Cat	0.59	0.60	0.26
deer	0.56	0.76	0.49
Dog	0.64	0.72	0.48
frog	0.72	0.89	0.63
horse	0.75	0.76	0.61
ship	0.82	0.90	0.73
truck	0.76	0.88	0.68
Training Time(second)	423s	480s	2248s

Table 1: Accuracy on the 10000 test images, Accuracy for each class, and Time Spent of the network in training

Furthermore, this study aims to improve the accuracy by data augmentation. Those Images which were incorrectly classified by trained AlexNet model were selected and performed image enhancement by adjusting the brightness, contrast and sharpness. The updated images were passed into the trained model to make classification. Twenty-five percent of those images were classified correctly after enhancement.

Conclusion

This study compares the performance of traditional 3-layer CNN, AlexNet and MobileNet in image classification task using Cifar-10 dataset. It proved that AlexNet has achieved the best performance and relatively high efficiency. This study also examines the effectiveness of image enhancement. For future study, image enhancement before experiment is worth trying to improve the model performance. The MobileNet also have potential to be more effective in this task by adding more layers and nodes when there is no limitation in hardware.

References

[1] Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, abs/1704.04861.

[2] Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical Report*.

[3] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.