BBM 418 - Computer Vision Laboratory
2022 Spring
Assignment 3

Murat Çelik
21827263
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
b21827263@cs.hacettepe.edu.tr

1 Introduction

This assignment includes analysis on Convolutional Neural Networks. A model is created from scratch to understand the CNN architecture. A CNN-based transfer learning is done. The results are analyzed. In this whole process, the Pytorch library was used and the improvements were made on Colab.

2 Data set

There are 15 categories; bean, bitter_gourd, bottle_gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, radish and tomato. The training set includes 3000 images, the validation set includes 750 images, and the test set includes 750 images. There are a total of 4500 images, 300 from each class. In all distributions, in accordance with the stratify rule, there are equal numbers of each class in each section.

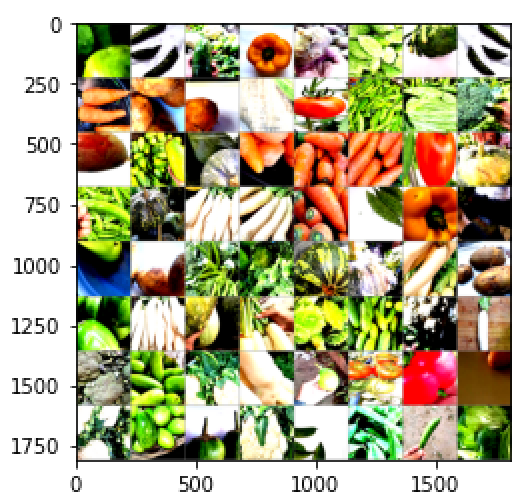

Figure 1: Example of Data Set

3 Part 1 Modeling and Training a CNN classifier from Scratch

3.1 Convolutional Neural Networks Architecture

3.1.1 CNN Information

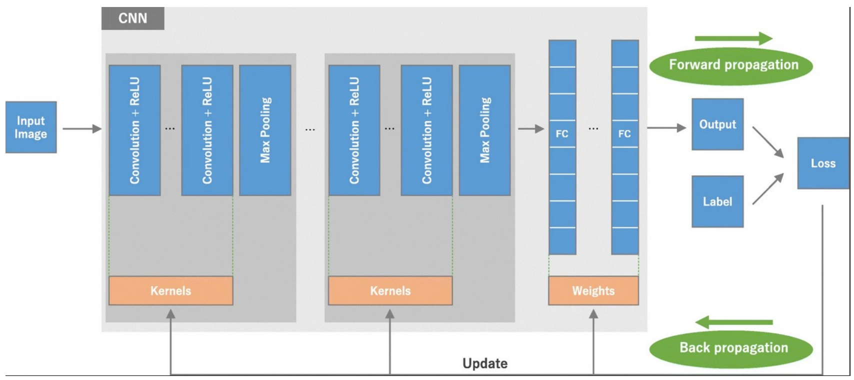

Figure 2: CNN Architecture

The operations in which we select the maximum value in every 4 pixels and thus reduce the input size by half are called (max) pooling.

The process of applying a filter that results in an activation function to the entire image is called convolution.

The purpose of CrossEntropyLoss is to take the output probabilities (P) and measure the distance from the truth values.

The Adam optimization algorithm is an extension of stochastic gradient descent popularized in deep learning applications.

3.2 Residual Connections

Residual Connections are structures formed by adding some data to the neural network via an alternative route. In the model created below, a residual connection is created between the 3rd and 4th and 4th and 5th type layers. Looking at the results, residual networks provided an additional 5% success. Comparing the above results, an accuracy value of more than 5% was obtained by adding the residual network. Although these values do not pose a problem for the algorithm, it has been a recommended method to be used by improving the results.

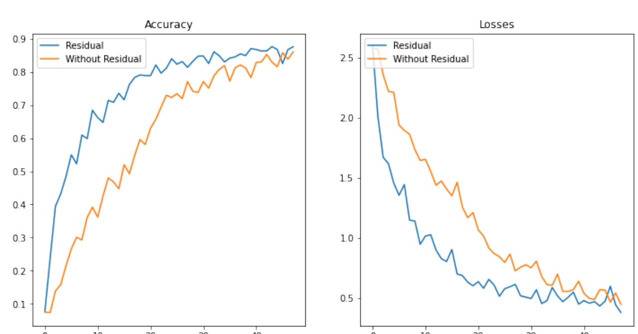

Figure 3: Plot of Accuracy and Loss for Residual Comparision

3.3 Learning Rate Comparision

Deep learning neural networks are trained using a stochastic gradient descent optimization algorithm. Adam optimization is used in these algorithms. The learning rate is a hyperparameter that controls how much the model is updated in response to the estimated error each time the model weights are updated. Too large of this value makes it difficult to reach the minimum error point. If this value is too small,reach the minimum error point in too many steps. This value should be chosen as an average value.

In this algorithm, 3 different learning rate values are used: 0.001, 0.0005, 0.00005

A value of 0.0005 is considered the best option for the model. Since 0.001 is too large, it is difficult to reach the minimum. Since the value of 0.00005 is small for the model, it may reach the minimum value in more steps.

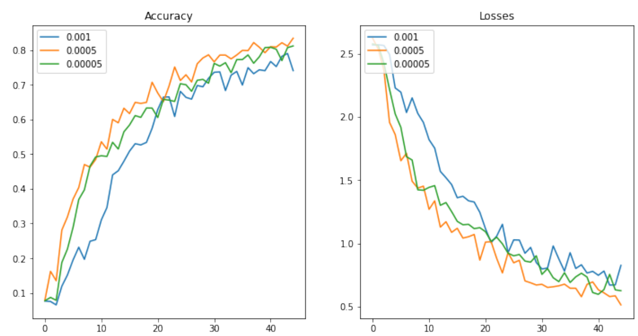

Figure 4: Plot of Accuracy and Loss for Learning Rate Comparision

3.4 Batch Size

There are 3000 data in our train data and 750 data in our validation data. I use this data in smaller sets to train the model. These sets are called batches. In this algorithm, two batch size values, 64 and 128, were determined.

As a result of the analysis, it is observed that the value of 64 is more successful than the value of 128. A smaller value of 64 may give better results for a while, but getting smaller may have a negative effect on this model.

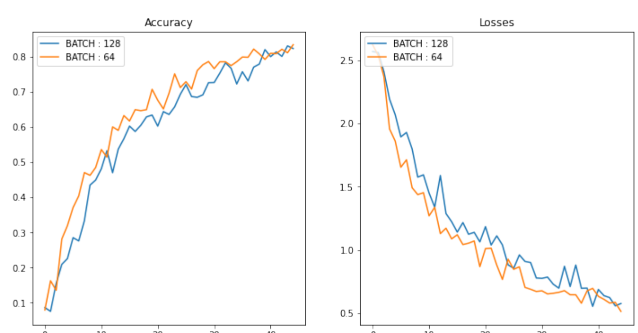

Figure 5: Plot of Accuracy and Loss for Batch Size Comparision

3.5 Dropout Best Model

It is assumed that removing some links within the network will improve training performance and prevent overfit. This process is called dropout. The dropout layer is given a ratio greater than 0 and less than 1. This value is the probability value.

Under normal circumstances, it is said that the dropout function will increase success. Since our model did not have an overfit problem and the train and validation performances were similar, this value had a negative effect on our model. If we develop our model further, the overfit problem will arise, and one of the best options to fix this would be the dropout technique.

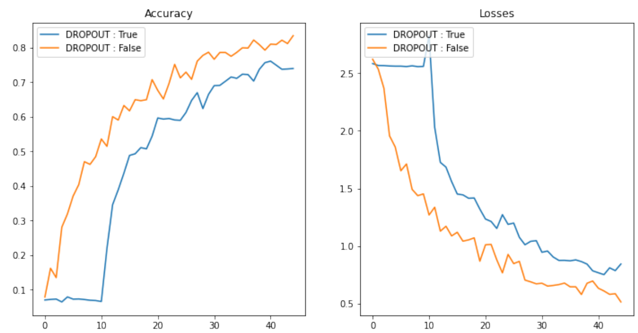

Figure 6: Plot of Accuracy and Loss for Dropout Comparision

3.6 Best Model Analysis

The best option for the learning rate value was 0.0005. The best value for the batch size value is 64. The value of 50 was used as the epoch number. RELU activation function is used in the models because of its fastness and high performance, solving the vanishing gradient problem.

In all these analyses, the above hyperparameters were chosen as the main model. Dropout technique had a negative effect for this epoch number. Residual connections, on the other hand, increase success.

In the matrix below, it is observed that the classes "Tomato, Radish, Bean" are not predicted correctly enough. In order to develop these classes, it is necessary to increase the epoch value and optimize the hyperparameters. For values, residual links can be added to the algorithm and the model can be developed.

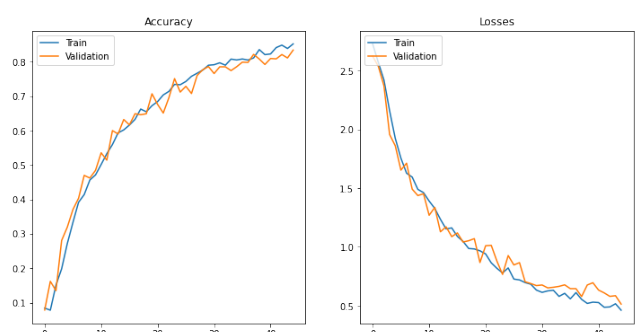

Figure 7: Plot of Accuracy and Loss for Best Model

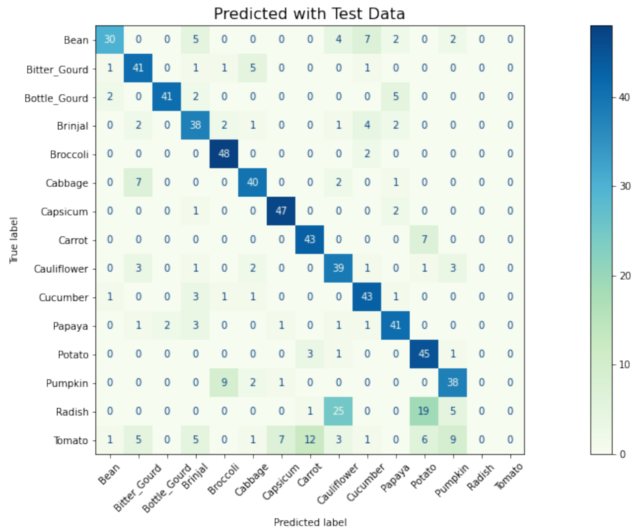

Figure 8: Confusion Matrix for Best Model

4 Part 2 Transfer Learning with CNN's

In this part of the assignment, I used and fine-tuned the pretrained ResNet-18 convolutional neural network and finetune this network to classify the sample images. Applying fine-tuning enables us to prepare powerful and fast models by training pre-trained models with our own data. This greatly increases our performance.

I rearranged all layers before training the network, except for the FC layer. I synced the last FC layer with class count. Likewise, operations were performed by changing Fc and two layers. And finally, train the model with its own data. And finally, I updated the trained weights of the model using my own data on the last layer. Thus, I adjust the model according to our own data.

Learning rate is 0.001 and our epoch number is 50. Model trained with these parameters gave a higher score than all the other models in part 1.

Due to the Transfer Learning technique, our pre-train model started with high success and low loss value. It can be observed in the graph that high success has been achieved in this model.

Because the pre-train model was trained with more comprehensive data, the model we changed to the last layer gave better results than the other. But with the right hyperparameters and optimization, the last layer and additional two-layer training can yield better results.

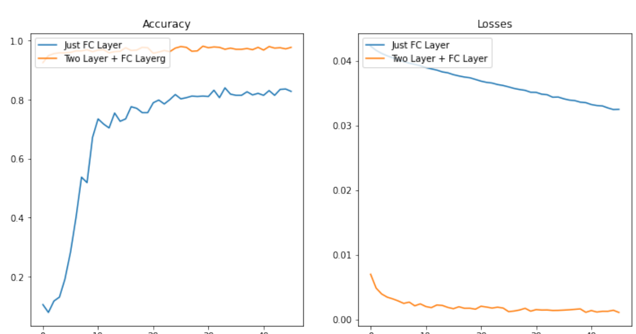

Figure 9: Plot of Accuracy and Loss for Pre-trained Model

In Part 1, the "Bean" class was not classified well enough. Here we see that we have improved it. A stronger model should be prepared for the "Radish, Tomato" classes.

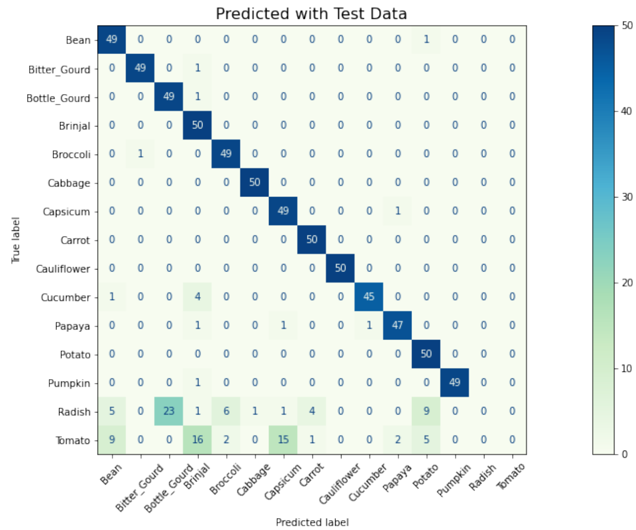

Figure 10: Confusion Matrix for Best Pre-Trained Model

5 Conclusion

In all these analyses, the CNN architecture was examined in depth. There are used models that are coded from scratch and pre-trained. Learning rate, epoch, batch size, dropout technique, residual links are analyzed and results are given. Judging by the overall results, it can be said that using pre-trained models is the best option.

All this work was done primarily on Colab using CPU. In this process, determining the correct hyperparameters and training the model required a lot of time. Later, GPU was used on Colab Pro. Even in this process, when we consider an epoch time and the number of models, the epoch number is 50. It has been determined that increasing the number of epochs will increase the success, but will make it difficult in terms of time.

6 References

[1] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[2] https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/

[3] Deep Learning by Aaron Courville, Ian Goodfellow, and Yoshua Bengio