VGG16 Extension on CIFAR classification

Tyvon Factor-Gaymon, tf18wj, 6580310 and Peter Fung, pf18np, 6509830

***Abstract*—As technology continues to progress, so to will its impact on global society. The rise in AI and its widespread use will allow humans to advance civilization at an unprecedented rate. In order to further develop AI, the networks that compose them should be understood. The factors that influence them should be studied as well so that the networks can be optimized and efficiently perform their tasks. This report examines the factors involved in making an efficient image recognition neural network by extending the VGG16 model to learn about the CIFAR-10 dataset. It will convey how well the network performs through various experiments by adjusting many factors, including, but not limited to, using different learning rates, number of batches, epochs, and augmentation. The use of augmentation in the experiments was computationally expensive, so the maximum number of epochs was limited throughout the experiment. It was found that the use of augmentation would benefit the network the most since CIFAR-10 is a relatively small dataset.**

I. Introduction

In this day and age, the improvement of artificial intelligence (AI) is a major technological advancement that currently has a significant impact on society. It can be thought of as an extension of the industrial revolution (IR) where the machines used in automation are now capable of much more due to being augmented with AI. Also, its applications are numerous and it serves as a catalyst to help humans achieve tasks that were previously unimaginable. It may be premature to assume that this is the spark of another revolution but it is important to recognize the fact that AI is still relatively new. One such AI system that is currently dominating the global scene is the use of the natural language processing AI, ChatGPT. It is capable of taking in information from many different resources and articulating an answer that can facilitate the inquirer's understanding of the content.

An artificial neural network is a system that is based on biological neurons. Each network is composed of these interconnected neurons to process and transmit information. The goal of this system is to learn from the input data by modifying the strengths of the connections between the neurons. This enables the system to recognize patterns and make predictions based on these patterns. The reason the system attempts to learn these patterns is so that when new data is introduced to the system, it can accurately make predictions about them. There are many different types of artificial neural networks with many different variations that excel at performing specific tasks. In order to process more complex data, deep neural networks are used. What makes them "deep" is that they have at least two hidden layers of neurons.
A deep neural network that is of interest in this study is the convoluted neural network. It is commonly used to identify patterns in images. It involves applying filters to input images and then transforming it into a feature map in order to learn important patterns in the image. It uses pooling layers to down-sample the feature maps to reduce their dimensionality and improve the network.

Regarding neural networks, there are many factors involved in the processing of the data that influence the overall performance of the system. Such factors include, but are not limited to, the complexity of the data to be processed, the number of hidden layers, the number of nodes/neurons in a layer, the activation function used, momentum functions, and learning rates. In order to optimize the learning process and increase performance, these parameters should be modified. There are no set

of parameters that are generally best for all AI networks. This means that the parameters should be adjusted and experimented with to find the best settings to increase performance. The main goal in building one of these networks is to get it to generalize well to new data. That means that the network should be able to accurately classify or make predictions about new data. This report will analyze various experiments that seek to extend a preexisting neural network to be able to optimally perform well on a specified task.

II. Objective

The goal is to use the VGG16 convoluted neural network (CNN) that has been pre-trained on ImageNet to initialize the weights of the convolutional layers. It then extends this model by adding new fully connected layers that are trained on the CIFAR-10 dataset, allowing the model to accurately predict classes for smaller images. The idea behind using these frozen pretrained weights is to ensure that the features remain the same during training. By incorporating additional layers, the model can then learn and adapt to accurately classify the smaller images into specific categories.

Trained using Dataset CIFAR-10 using the VGG16 architecture.

The model is trained to classify images into 10 different categories.

III. Experimental Setup

This section describes the parameters and procedures used for this experiment. Here, the purpose of this section is to list all the parameters and procedures covered by this experiment for the sake of allowing the reader to easily replicate any results gathered during this experiment.

A. Dataset used

The dataset used for this experiment is a CIFAR-10 dataset consisting of 60,000 images. All these images are coloured images and are consistent in sizing of 32x32 resolution size. All 60,000 images are further categorized into 10 categories with 6000 images in each category. Out of all 60,000 images in the dataset, 50,000 images are used to train the deep learning model, and the other 10,000 images are used to validate the model during training.

The dataset used for training will consist of 5,000 images from each category. Though, the training dataset is further broken up into multiple 'batches' of training sets, such that each 'batches' will contain images from each category in a random order. Here, it is important to note that a training 'batch' may contain more images from a category than other batches. Though, the amount of images in each batch is consistent.

The dataset used for validation will contain the other 10,000 randomly selected images from each category.

B. Framework used

The framework used for this experiment is TensorFlow. Before beginning this experiment, multiple deep learning frameworks such as Microsoft Azure, Pytorch, and NVidia TAO toolkit were considered and heavily discussed. The discussion consisted of evaluating the advantages and disadvantages of using each framework. Ultimately, TensorFlow was chosen for this experiment due to its variety of benefits such as its scalability for augmented datasets, the availability of a variety of pre-trained models, adaptability with other libraries such as numpy, matplotlib, Keras, and finally its support for using GPU hardware. TensorFlow's ability to use modern GPU hardware is crucial for this experiment, as it exponentially sped up the training process of each set of experiments.

C. Deep Learning System

The deep learning model implemented for this experiment is the VGG16 model. This model is a pre-trained convolutional neural network that has a total of 16 layers, and is a popular option for tasks that involve image recognition and classification. The core advantage of using this network is that

the model has been originally trained across a large pool of data consisting of over a million images and 1000 different classes. Given the scope of this experiment, the VGG16 model has been chosen as an appropriate model to train the chosen dataset of 50,000 images. Furthermore, compared to other deep learning networks, the VGG16 model is considered to be relatively simple to work with, available on TensorFlow, and has a high-accuracy on a wide range of image classification tasks.

D. Parameters

1) Learning Rate: For neural networks, the learning rate is responsible for controlling the amount of change in the model in response to the error rate after each iteration of training datasets. This is achieved by updating the weights of the network in accordance to the learning rate after iterating through a training batch. Implementing a proper learning rate will provide great benefits to the networks such as helping the network converge faster on a better solution. Implementing an inadequate learning rate can hinder the network either due to leading the system to a slow convergence, or worse yet; never reaching convergence and instead fluctuating around the optimal solution. This conveys the importance of experimenting with different learning rates and observing which learning rate improves performance.

For these experiments, there will be two types of learning rates implemented. The first is a dynamic learning rate, where the learning rate will decrease by a factor based on the current epoch and the total number of epochs. This means that the learning rate will decrease at specific points in training. The second is a cyclic learning rate, where the learning rate will fluctuate over a range of predefined learning rate values. Though, the cyclic learning rate will be applied for later experiments for the purpose of fine-tuning the model in an attempt to achieve better results.

In the first set of experiments, the dynamic learning rate is implemented by having the initial learning rate decremented by a fixed value over each epoch. The formula used for determining the fixed value can be found as: (Starting learning rate - end learning rate)/number of epochs. For this formula, the end learning rate will need to be predetermined. So, for the sake of this experiment the end learning rate will always be defined as the value 0.1.

In the second set of experiments, the cyclic learning rate is implemented by having the initial learning rate increase and decrease depending on the epoch. The function responsible for the cyclic learning rate will first determine the cycle number based on the epoch number, then determine the position of cycle within the current epoch, and will then finally determine the learning rate by calculating the linear interpolation of the minimum and maximum learning rate based on the second step. This method is vastly different compared to the dynamic learning rate method, since the learning rate will either gradually increase or decrease based upon the position of the cycle within the current epoch.

2) Initial Weights: The initial weights used for these experiments were the pre-trained weights from the VGG16 CNN. The initial weights allow the extending CNN to be able to recognize features such as objects in images. This means that the extending network is able to recognize patterns previously trained by the VGG16 network. This effectively facilitates the extending network's ability to detect low level features such as edges, shapes, and textures. The CNN used in this report is effectively building upon the VGG16 network. The weights are fine-tuned and adjusted according to the CIFAR-10 dataset to learn high-level features specific to that dataset.

3) Predefined Variables: This section covers the list of predefined variables used for this experiment. Predefined variables are necessary for this experiment as it is used to speed up the training process and ensure consistency across different sets of experiments. Here, the predefined variables can be found listed below:

1) Input_shape: the shape of each images input

into the system is of 32x32 resolution.
2) The top layer of the pre-trained VGG16 model is not loaded.
3) The initial weights of the model is pre-trained using the 'image net' dataset.
4) The number of output classes is 10.

4) Batch Size: The batch size represents the number of samples that will be processed through the network before the weights are updated. The CNN will receive batches of input data and process it to calculate the loss of each batch in order to determine the average loss over all batches. Through back-propagation, the gradient of this average loss is used to update the weights in the network. Using a smaller batch size causes the network to update more frequently and may be more likely to converge faster. Larger batches will lead to more stable updates but can be computationally expensive.

5) Epochs: Determining the number of epochs is important as it can easily lead to over-training or under-training the neural network. If the number of epochs is set too low, then the network may not have had sufficient time to learn the patterns. At the other extreme, if the number of epochs is set too high, then the model will memorize the data and will not be able to generalize well. For the purposes of this study, the number of epochs used in the experiments were in the range of 60 to 200.

6) Adam Optimizer: Adam refers to an optimization algorithm that is used to update the weights and biases in the CNN. It keeps track of past gradients and past squared gradients for the purpose of using them to adjust the learning rate for all the weights and biases. It uses the knowledge about gradients from previous iterations to modify the current weights. Adam also makes use of bias correction and momentum to increase performance and mitigate oscillations.

7) Loss Function: The loss function used in this CNN is the categorical cross entropy function. It's usually used for multi-class problems because it measures the difference between the predicted probability distribution and the actual probability distribution of the classes. It does this to estimate the accuracy of predicting one class. Its main goal is to minimize the difference between the predicted and actual probability distributions. This is done by adjusting the weights and biases during training.By fine tuning these weights and biases, the network is able to learn the patterns to reduce this difference. Ultimately, it brings the predicted distribution closer to the actual distribution, allowing for more accurate predictions.

8) Augmentation: Augmentation is a technique used to artificially expand the size of a training set by creating new samples out of the original ones. This is done through applying transformations out of input data to create variations out of them. Transformations include, flipping, rotating, shifting, scaling, and cropping images. Its goal is to provide a more diverse set of input to improve the network's ability to generalize data. Not all experiments data will have augmentation applied to them.

IV. Results

This section encompasses the results regarding each set of experiments. The Results are categorized into two main sections, non-augmented experiments and augmented experiments. Each section will then be further categorized depending on the batch sized for each experiment.

A. Non Augmented Experiments: batch size 32

B. Non Augmented Experiments: batch size 128

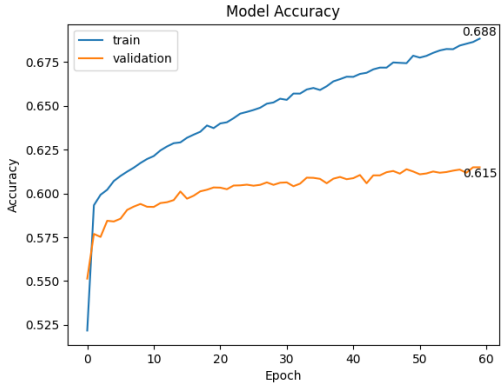

Fig. 1: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 32. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

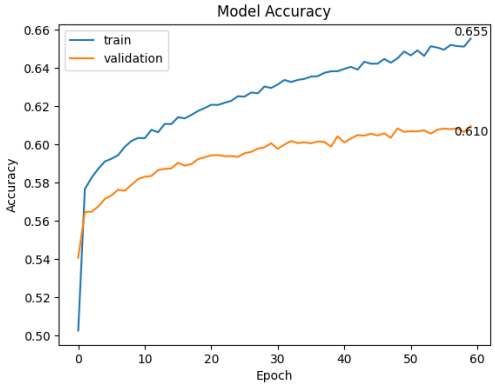

Fig. 3: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 128. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

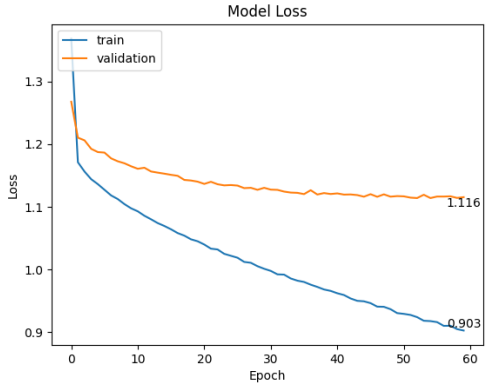

Fig. 2: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 1.

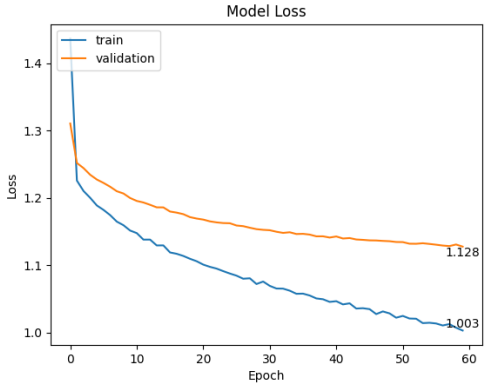

Fig. 4: This graph shows the training set and validations set convergence curves for an experiment that used a batch size of 128. It is part of the same experiment as 5.

C. Augmented Experiments

These data sets have been augmented have had various transformations and modifications applied to the original images. The purpose of augmentation is to introduce diversity into the dataset and prevent over-fitment of data.The various augmentations done to the data include: rotation, width shift, height shift, and horizontal flip.

1) Augmented data set using batch size of 32:

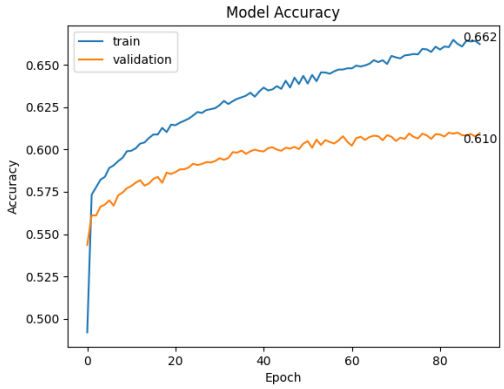

Fig. 5: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 128. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

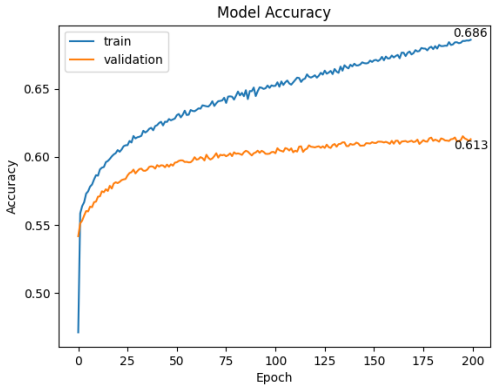

Fig. 7: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 256. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

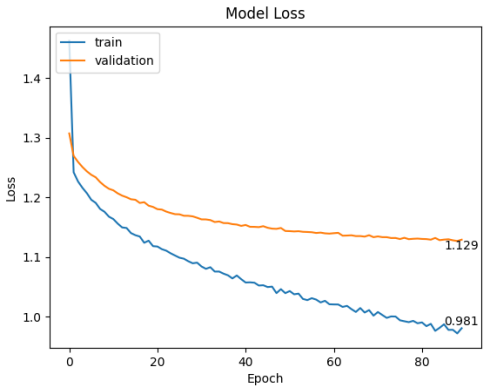

Fig. 6: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 5.

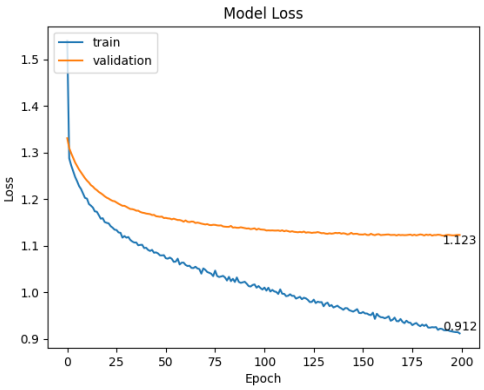

Fig. 8: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 7.

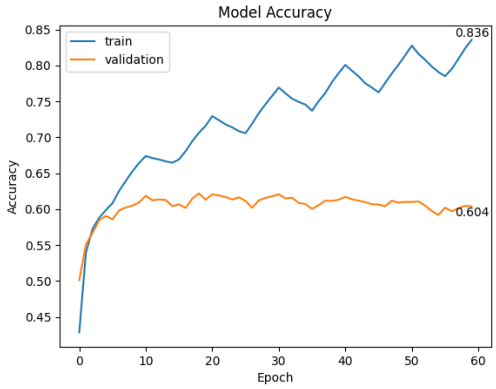

Fig. 9: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 32. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used a cyclical learning rate that fluctuated between 0.001 and 0.0001 every 5 epochs

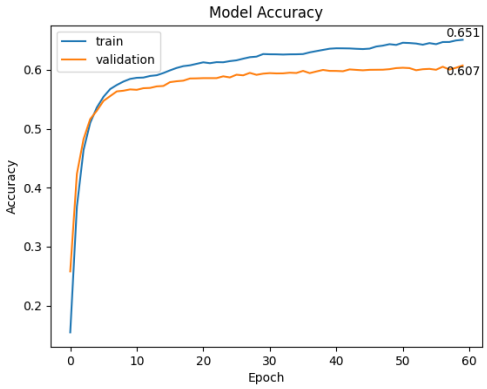

Fig. 11: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 32. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used a cyclical learning rate that fluctuated between 0.0001 and 0.00001 every 5 epochs

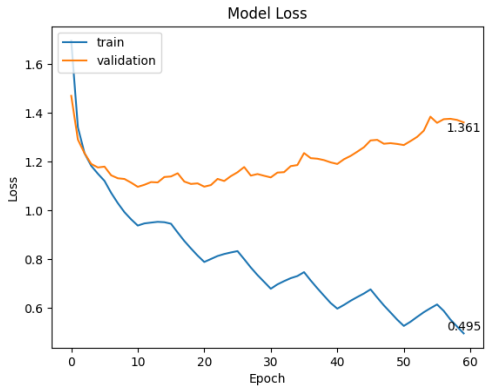

Fig. 10: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 9.

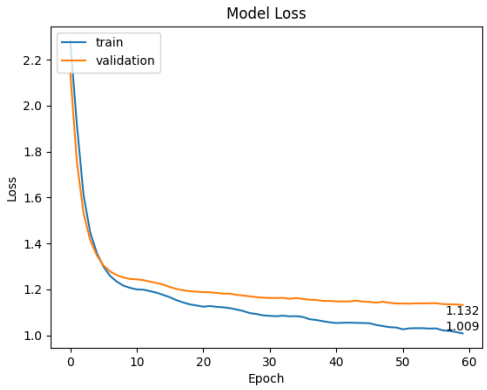

Fig. 12: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 11.

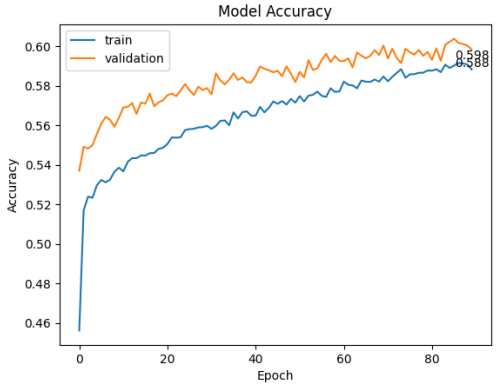

Fig. 13: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 32. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

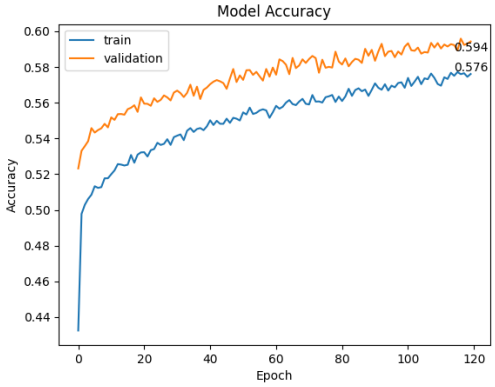

Fig. 15: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 128. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

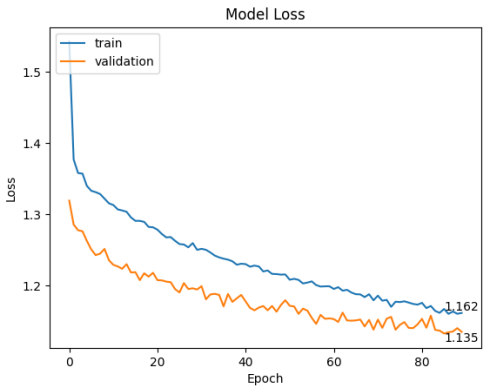

Fig. 14: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as Fig 13.

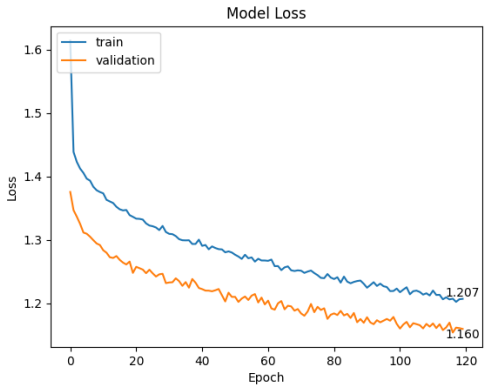

Fig. 16: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as 15.

2) Augmented data set using batch size of 128:

3) Augmented data set using batch size of 256:

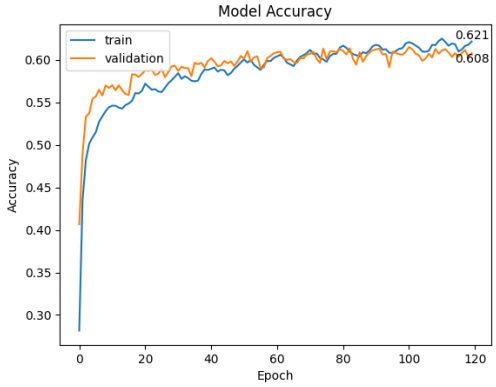

Fig. 17: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 128. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used a cyclical learning rate that fluctuated between 0.0001 and 0.00001 every 5 epochs

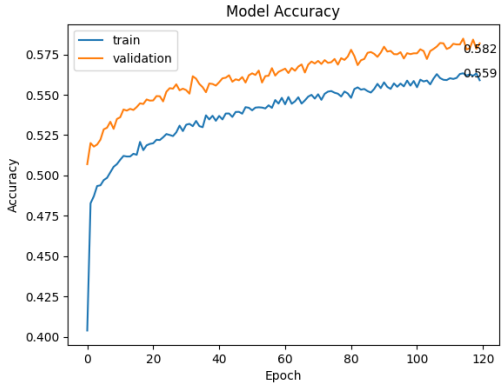

Fig. 19: This graph shows the training set and validation set convergence curves for an experiment that used a batch size of 256. It conveys how accurate the model was able to classify the smaller images into the correct categories. This experiment used an initial dynamic learning rate decay value of 0.001 and ended with using 0.0001

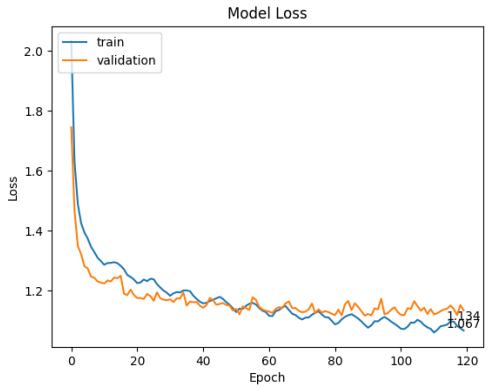

Fig. 18: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as 19.

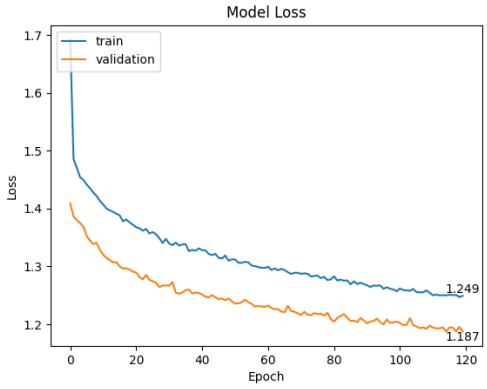

Fig. 20: This graph shows the loss values for both the training set and the validation set over a set amount of epochs. It is part of the same experiment as 17.

V. Discussion

With the exception of the last experiment, figures 9 and 10, all of the other experiments yielded similar results. When it came to the model accuracy, the validation set quickly plateaued around an accuracy of 60%. A wide range of epochs were chosen for these experiments so that a suitable optimally trained network could be produced. As can be seen from figures 1-8, given a batch size of either 32, 128, or 256, and a dynamic

learning decay rate, the highest accuracy rate that the validation set could produce was around 61%. For these non-augmented experiments, there were not too many factors that that could improve this network. If more epochs were used, then the network would have overfitted to the training data. Figure 9 and 10 used a cyclic learning rate which is evidenced by the shape of the graphs. Using this learning rate did not improve the model as it accelerated the rate of over-fitting the data. It can be seen that the validation loss is increasing and the training loss is decreasing which would indicate that the model is memorizing the training set and is not able to generalize new data well. It was interesting to see the use of this cyclic learning rate caused the network to quickly diverge. This early divergence suggests that the network is more sensitive to this type of learning rate and it is learning too quickly. As seen in Fig. 7 and Fig.8, the learning rate used for those graphs are relatively low. And so, as shown in both figures, the 'jumps' are not as drastic, and instead appears more like one of the above graphs with a dynamic learning rate implemented.

Figures 13 - 20 are graphs that model augmented input data. It can be seen from viewing these graphs that it is less prone to overfitting. For all figures but 19 and 20, the validation set is performing well and has better values than their corresponding training set values. This means that the network at this point is generalizing better and is not memorizing the data. Figure 19 and 20 show that the network is performing well using augmentation and the cyclic learning rate. The network is not as sensitive when it is augmented. Also, the validation accuracy and loss values are similar to the training values, indicating that the network is not overfitting and is generalizing the data. The training rate values are similar to the validation values With that said, more epochs should have been used here to see if the validation set values could be higher. Their values are comparable to the non-augmented validation set values.

It is important to note that the training times were far more lengthy than when training the non-augmented data. This is because the use of augmentation could have increased the complexity of the original input data to be processed. Augmentation could have introduced many new patterns and variations that was previously unseen by the model, and so it could require more training time. Also, this is much more computationally expensive because new inputs are created out of the original input data, thereby increasing the total amount of input to be processed by the network. Due to the limited hardware used in this study, the total number of epochs used in the augmented experiments had to be limited. The optimal values found here were around 61%. This means that the model was capable of predicting the correct classification with an accuracy rate of approximately 61% This number could have been improved if the augmented experiments were given enough epochs. Augmentation seemed to benefit it the most, making it more resistant to overfitting. This is due to the CIFAR-10 dataset being a relatively small.

VI. Conclusion

In conclusion, the non-augmented experiments using a dynamic learning rate has shown to have yielded similar results to each other regardless of batch size and number of epochs. In terms of accuracy, the validation data-set commonly plateaus around an accuracy of 60% across all graphs that depict a non-augmented dataset using dynamic learning. Furthermore, in terms of loss, the validation set is seen to generally plateau at around the value of 1.128 across all graphs depicting a non-augmented data set using dynamic learning. Since both validation accuracy and validation loss is similar across all graphs, this suggests that the model is generalizing the data well, especially in regards to the validation data. Interestingly enough, using a cyclic learning rate yields graphs that depicts the validation loss as increasing, and training loss as decreasing. This separation suggests that the model is memorizing the training set instead of generalizing, hence the increase in validation loss.

In addition, experiments involving augmented data sets have been seen to have an overall positive influence on the model. The resulting graphs of each experiment have shown that the model is less sensitive to over-fitting, as the augmenting the data set will produce more data and therefore encourage diversity within the data set. This can be evident by graphs depicting the validation set, where the validation loss is lower than that of the non-augmented data set. This suggests the augmented data set performs better on the test sets used. Furthermore, results depicted in the graphs suggests that using the augmented data set is substantially more beneficial than using the non-augmented data set. As mentioned in the paragraph above, the graphs depicting the non-augmented data set appears to start plateauing near the end. This fact suggests that given enough epochs, the model using the non-augmented data set will start to memorize the training set and thus the issue of over-fitment will soon occur. However in contrast, the model using the augmented data set doesn't appear to have the same problem, as the validation set continues to lower after each epoch. This suggests that the model generalizes the images in the data set much better compared to them model using a non-augmented dataset.