Final Project

I. Introduction

This convolutional neural network (CNN) is being developed to determine the type of infection of a given patient by assessing their chest x-ray. Viral infections disrupt the lives of many people across the globe. Over 31 million cases of COVID-19 were reported between January 2020 and April 2021 [1]. During 2018-2019, it is estimated that 35.5 million people got sick with influenza [2]. Similarly, bacterial infectious infect a significant portion of the human population. A study performed by Meatherall et al. in Calgary found that Klebsiella pneumoniae had an incidence of 7.1 per 100,000 between 2000 and 2007 [3]. Bacterial and viral infections cause similar symptoms but have drastically different methods of treatment. Frequently, bacterial infections are treated with antibiotics [4]. Antibiotics do not work with viral infections. Medical professionals try to minimize the incorrect usage of antibiotics because excessive use of antibiotics increases antibiotic resistance [5]. Therefore, correct diagnoses are essential to ensure proper care is being administered to the patient and prevent unnecessary antibiotics usage. This CNN would allow physicians to act with more confidence when diagnosing a patient.

The main challenges in solving this problem are the unbalanced dataset and the visual similarity between the viral and bacterial infections. 67.91% of the dataset has the "viral" label, while 67.91% of the dataset has the "bacterial" label. Also, 93% of the dataset has the "pneumonia" label. The labeling disparity can pose an issue. There is a possibility that the network will learn to predict viral infections and pneumonia more frequently because there are more viral infections and pneumonia data points. Furthermore, multiple infectious diseases in the dataset, such as Chlamydophila, Staphylococcus, and Herpes, have three or fewer total images. The lack of images means that there is a possibility that the validation or test dataset contains a label that the model has never seen. Lastly, the visual similarity between the infections will make it challenging to develop a highly accurate network. All of these challenges must be overcome to create an accurate classifier.

II. Literature Survey

A variety of networks have been tested on chest x-rays and COVID-19 specific datasets, and multiple studies suggest VGG is a high-performing architecture. Different versions of architectures such as Xception, Inception, ResNet, Visual Geometry Group (VGG), and DenseNet have been tested on chest x-rays [6]. Multiple studies have found that VGG networks perform exceptionally well with chest x-rays and, more specifically, with COVID-19 x-rays. HamedBehzadi-khormoujia et al. found that VGG16 performed better than DenseNet121, ChestNet, ChestNet2, and PyramidCNN on pediatric chest x-rays. The VGG16 model achieved an accuracy of 94.5% and an AUC of 99 [7]. Also, the model had the best sensibility and sensitivity scores on the pediatric chest x-ray dataset. A study performed by Bansal and Sridhar, VGG 19 performed identically to the VGG 16 model when classifying COVID-19 images [6]. Dansana et al. evaluated VGG-19's ability to conduct an early diagnosis of COVID-19. They found that they show highly satisfactory performance with a validation accuracy of 91% [8]. Bressem et al. compared different deep learning architectures for the classification of chest radiographs [9]. They found that CNN with fewer convolutional layers can achieve a higher area under the precision-recall curve (AUPRC) scores. VGG-19 had high scores for AUPRC and place under the receiver operating characteristic curve (AUROC). The multitude of studies praising VGG and VGG-19, in particular, influenced the creation of this CNN.

Multiple studies suggest that residual blocks perform well in solving machine learning tasks. Dai et al. found that the models incorporating residual blocks performed better than his original models that did not include residual blocks [10]. A different study conducted by Duan et al. found a ResNet-based model structure was effective in health monitoring [11]. These papers influenced the decision to test the model with residual blocks.

III. Methods

Three separate networks were created, and their performance on the chest x ray dataset was tested. The first network closely follows the implementation of VGG-19. The second network included residual blocks within the modified VGG-19 network. The third network was the replica of the highly effective network used to classify handwritten digits. The three different networks varied in complexity to allow comparison between deep networks.

A. First Implemented Network: Modified VGG-19

The first implemented network closely follows the architecture of VGG-19. The number of channels for each convolutional layer and all of the linear layers were limited to decrease the number of parameters in the model and ensure that the model trained quickly and does not require too much memory. The network has four separate groups of convolutions before the linear layers. Within the first two groups, there were two different convolutional layers, each followed by ReLU activation. The following two groups had four separate convolutional layers, each followed by a ReLU activation. All of the groups conclude with a two-dimensional max pooling. The result of the final group enters the linear layers after adaptive averaging and flattening. There are three linear layers, and each layer is followed by ReLU activation and dropout, excluding the final linear layer (Fig. 1).

B. *Second Implemented Network: Modified VGG-19 with Residual Blocks*

The second network added residual blocks to the modified VGG-19 network [12]. The first two residual blocks contained two separate convolutions and a single ReLU activation between the convolutions. The following two residual blocks contained four separate convolutions with a single ReLU activation between each convolutional layer. After each residual block, the output underwent two-dimensional max pooling. Following the four residual blocks, enters the linear layers. The linear layers of this network are the same as the modified VGG-19 network (Fig. 2).

C. *Third Implemented Network: MNIST CNN*

The third network tested was the network utilized to classify the MNIST dataset. The MNIST network was chosen because it is known to perform well on MNIST data, and it is a relatively simple model. The input enters three different convolutional layers that have channel sizes of 6, 16, and 120. Each convolutional layer is followed by two-dimensional max pooling. Adaptive averaging and flattening follows the three layers of convolution. The resulting tensor enters the linear layers. There are two total linear layers in this network. The first linear layer is followed by ReLU activation and dropout. The second linear creates the output tensor that contains the classifications (Fig. 3).

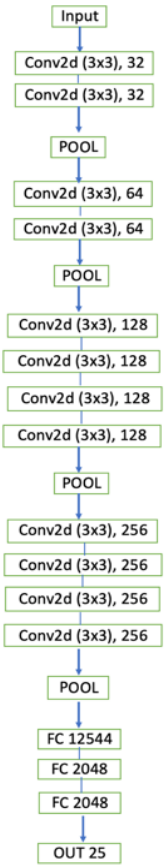

Fig.1: Diagram of the modified VGG-19 network used to classify chest x-ray images.

IV. Experiments

The dataset is composed on 535 total images. The largest label is pneumonia, which is present in 93.46% of the data. The second largest label is viral at 67.91%, while COVID-19 is the third largest label at 60.12%. Chlamydophila is the smallest label in the entire dataset at 0%. Aspergillosis, Aspiration, H1N1, MRSA, and Staphylococcus were the next fewest labels at 0.31% of the total dataset. The other labels appeared in 1-8% of the total images.

The training set had 60% of the total images, while the validation set had 20% and the testing set had 20%. A comparison was made between augmented data and non-augmented data to determine how data augmentation affects the results of the model. Data augmentation was performed by randomly translating the image horizontally or vertically by a randomly determined amount. Training was performed with all 25 labels and with a subset of 18 labels. The subset was created by removing labels that appeared three or less times in the total dataset. This decision was performed in order to determine if the model would train better if it did not have to predict labels that rarely occur.

The metrics used for validation are accuracy and Area under the ROC Curve (AUC). Accuracy is the percentage of the total label predictions that are labeled correctly. A prediction is considered correct if every single label of an image is correctly identified. The AUC metric will provide information about the true positive and false positive rates in the model. High AUC scores indicate that the model is capable of distinguishing between classes. Models were evaluated based on their AUC score and the accuracy percentage. A quality model would have a high AUC score, and a high accuracy percentage.

Multiple experiments were completed to determine the best overall network and the best hyperparameters for the best network. As shown in Fig. 4, the work flow involved the importation of the data from the GitHub page. Afterwards the data was split into the training, validation, and testing datasets. The first experiment was performed on the modified VGG-19 architecture described in Fig. 1. This experiment compared different learning rates and different amounts of weight decay for the stochastic gradient and the Adam optimizer. Additionally, the experiment compared the performance of the different models with or without data augmentation. Following the evaluation of the models on the validation dataset, the process would be repeated with the residual block VGG-19 (Fig. 2) and the MNIST CNN (Fig. 3).

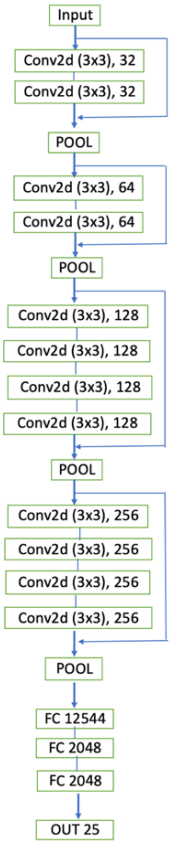

Fig. 2: Residual Block VGG-19. Diagram of the modified VGG-19 network with the addition of residual blocks used to classify chest x-ray images.

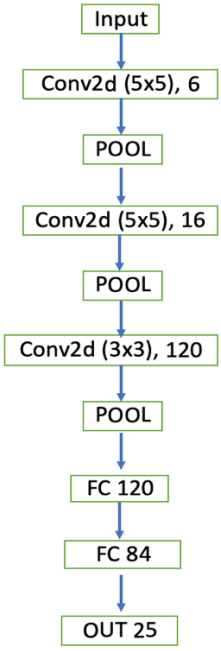

Fig. 3: MNIST CNN. Diagram of the CNN used to classify MNIST diagrams. This network will be used to classify chest x-ray images for this project.

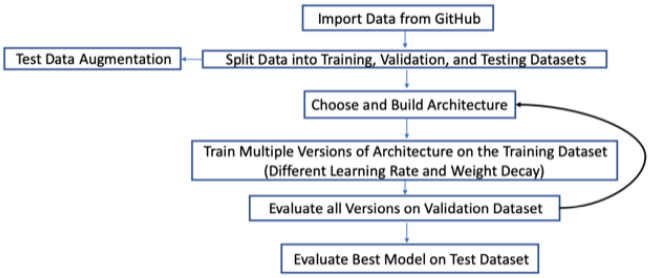

Fig. 4: Modified VGG-19. Diagram of the process used to determine a quality architecture for the problem

A. First Experiment

The first experiment compared four different learning rates, and three different weight decays for the modified VGG-19 network. The tested learning rates were: 10^-2, 10^-3, 10^-4, and 10^-5 [13]. The tested weight decay values were: 10^0, 10^-2, and 10^-4. The study performed by C. Anusha and P. S. Avadhani influenced the learning rate and weight decay values that were selected for all of the experiments. All possible combinations of these learning rates and weight decay values were tested for ten epochs. Adam was used as the optimizer for this experiment. The data was not augmented for this testing.

The combination of learning rate = 0.01 and weight decay = 1 displayed a trend across multiple combinations. The mean AUC oscillated between 0.825 and 0.99 throughout the first ten epochs for most learning rates and weight decay combinations (Fig. 5). The validation dataset had the larger mean AUC for the majority of epochs for most combinations. The loss experienced an exponential decay for the majority of combinations (Fig. 6). Furthermore, most combinations experienced dramatic oscillation in the percentage of correct predictions. As shown in Fig. 7, the accuracy peaks at 66% for the second epoch, to fall to 0% on the third epoch. Following this drop, the accuracy returns to 66% for the fourth epoch. This rise and fall are repeated twice in the span of ten epochs.

Some combinations did not resemble the results found in the majority of combinations. The network with a learning rate of 0.00001 and a weight decay of 1 did not have an exponential decrease in loss (Fig. 8). This combination experienced an increase in loss before beginning to plateau after seven epochs. The increasing loss did not negatively affect the peak accuracy for the combination. The network reached 66% within the first epoch and maintained the level of accuracy for the remainder of the training. Other combinations that had increasing losses were the network with a weight decay of 1 and a learning rate of 0.0001. Similarly, the combination of weight decay = 1 and learning rate = 0.0001 reached 66% for the first epoch and remained constant for the duration of testing (Fig. 9).

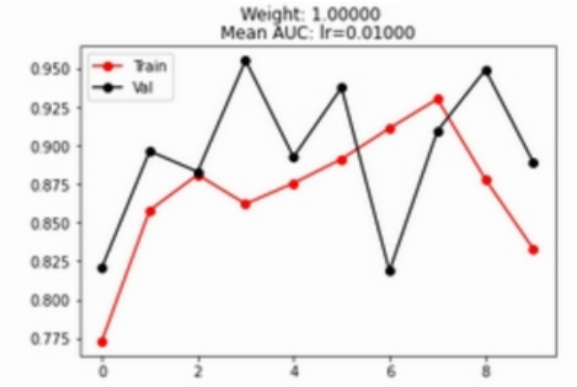

Fig. 5: Mean AUC comparison for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.01.

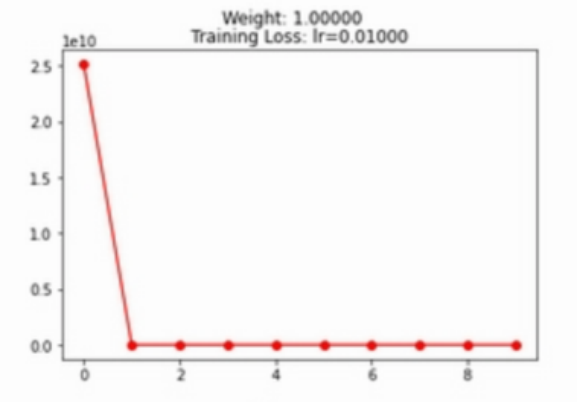

Fig. 6: Training Loss for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.01.

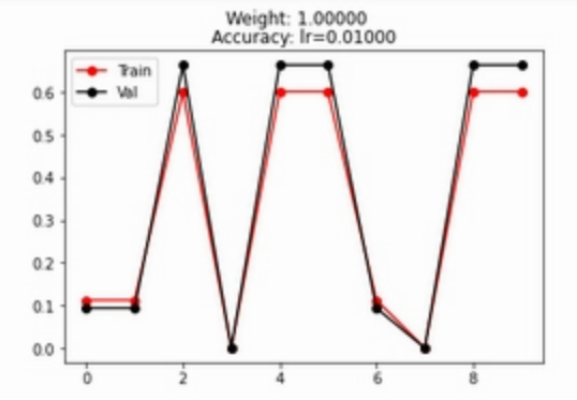

Fig.7: Accuracy comparison for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.01.

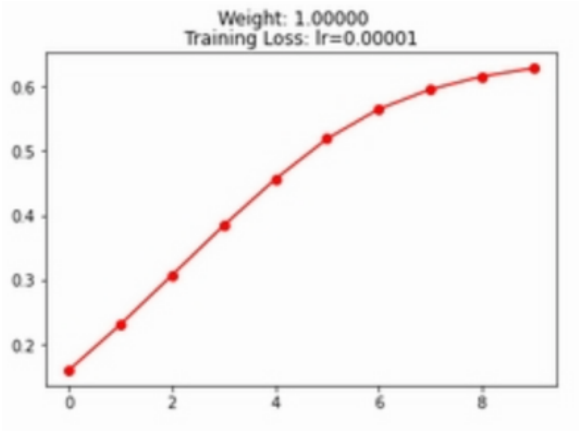

Fig. 8: Training Loss for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.00001.

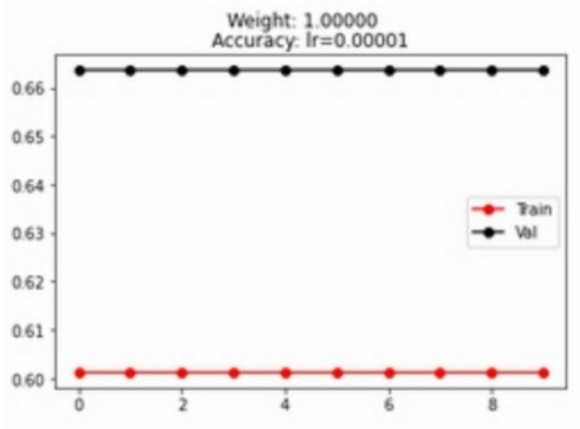

Fig. 9: Accuracy comparison for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.01.

Looking at the best-performing combinations, it is evident that three learning rates and two-weight decay values perform the best. The learning rates of 0.001, 0.0001, and 0.00001 have mean AUC values above 0.98 while being 66.35% accurate on the validation set (Fig. 10). The high-performing weight decay values are 0.01 and 0.0001. Out of the top seven performers, four of the models have a weight decay = 0.01.

Looking at the poor-performing combinations, it is evident that the learning rate of 0.01 and a weight decay of 1 are the worst combinations. Out of the nine networks with the lowest mean AUC, seven had a weight decay of 1, and eight had a learning rate of 0.01 (Fig. 11). The lowest mean AUC scores were between 0.8 and 0.9, relatively low to the other AUC scores used for different tasks.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.00100	4.0	0.984848	0.0100	0.663551
0.00100	2.0	0.988485	0.0001	0.663551
0.00010	6.0	0.981212	0.0100	0.663551
0.00010	3.0	0.988485	0.0001	0.663551
0.00001	3.0	0.980606	0.0100	0.663551
0.00001	7.0	0.981212	0.0100	0.663551
0.00001	6.0	0.983030	0.0001	0.663551

Fig. 10: Models with the top seven mean AUC scores for the modified VGG-19 model. The optimizer is Adam.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.010	0.0	0.820606	1.0000	0.093458
0.010	1.0	0.896364	1.0000	0.093458
0.010	2.0	0.882984	1.0000	0.663551
0.010	4.0	0.892596	1.0000	0.663551
0.010	6.0	0.818656	1.0000	0.093458
0.010	9.0	0.889091	1.0000	0.663551
0.010	0.0	0.884848	0.0100	0.000000
0.010	8.0	0.880758	0.0001	0.663551
0.001	8.0	0.873050	1.0000	0.000000

Fig. 11: Models with the bottom nine mean AUC scores for the modified VGG-19 model. The optimizer is Adam.

B. Second Experiment

The second experiment compared four different learning rates, and three different weight decays with the stochastic gradient descent (SGD) optimizer. The data was not augmented for this testing. The same learning rates and weight decay values were used for this experiment. All possible combinations of these learning rates and weight decay values were tested for 15 epochs.

The combination of learning rate = 0.01 and weight decay = 0.0001 displayed a trend that was seen across multiple combinations for the SGD optimizer. The mean AUC oscillated between 0.91 and 0.99 throughout the first ten epochs for the majority of learning rate and weight decay combinations (Fig. 12). Not a single epoch dropped below 0.919 for the SGD optimizer. This consistency was not as present for the Adam optimizer, which fell below 0.90 for many epochs through the entire testing process. Similar to the Adam optimizer, the validation dataset had the larger mean AUC for the majority of epochs for most combinations. Most of the models had a drop in total loss for each epoch, as exemplified in Fig. 13. In contrast to the Adam optimizer, most of the models started at the peak accuracy of 66% and remained at the mark. The accuracy plot for learning = 0.1 and weight decay = 0.0001 shows the common trend for most combinations (Fig. 14).

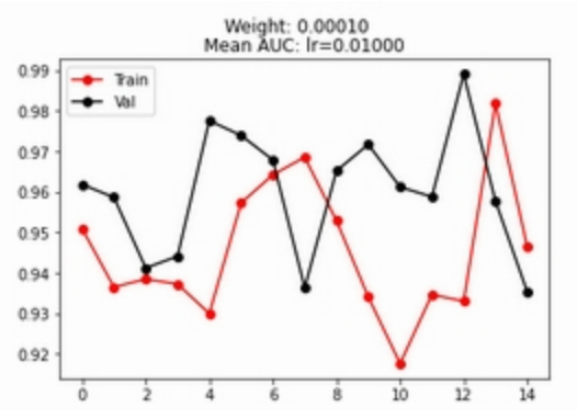

Fig. 12: Mean AUC comparison for the modified VGG-19 model across 10 epochs. This model had a weight decay of 0.0001 and learning rate of 0.01.

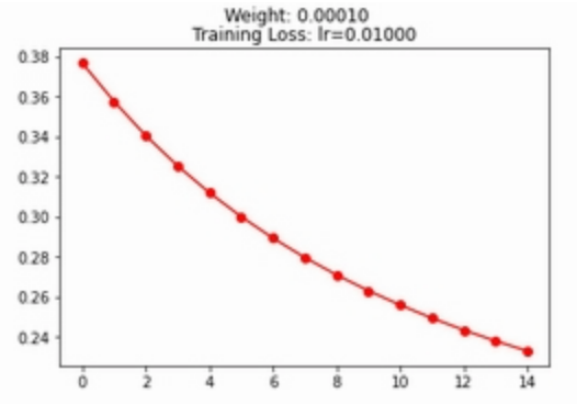

Fig. 13: Training Loss for the modified VGG-19 model across 10 epochs. This model had a weight decay of 0.0001 and learning rate of 0.01.

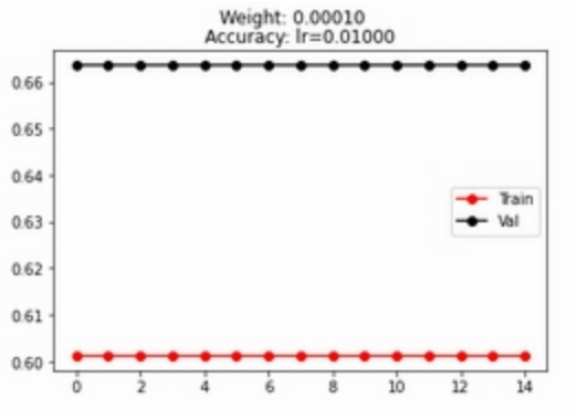

Fig. 14: Accuracy comparison for the modified VGG-19 model across 10 epochs. This model had a weight decay of 0.0001 and learning rate of 0.01.

Some combinations did not resemble the results found in the majority of combinations. Most of the networks with a weight decay of 1 (Fig. 15) did not exponentially decrease in loss. These combinations experienced an increase in loss before beginning to plateau after seven epochs. The increasing loss did not

negatively affect the peak accuracy for the combination. The networks reached 66% within the first epoch and maintained at the level of accuracy for the remainder of the training.

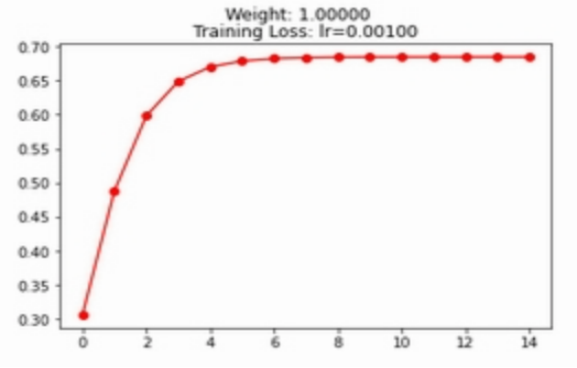

Fig. 15: Training Loss for the modified VGG-19 model across 10 epochs. This model had a weight decay of 1 and learning rate of 0.001.

Looking at the best performing combinations, it is evident that there is one learning rate and one weight decay that provides consistently high mean AUC scores and accuracy percentages. The learning rate, 0.0001, appears in seven of the top 12 performing models (Fig. 16). Meanwhile, the weight decay value of 0.0001 appears six times. Other learning rates, such as 0.0001, appear three times in the top models. The weight decay value of 0.0001 does not have a competitor.

Looking at the poor-performing combinations, it is evident that the learning rate of 0.01 and a weight decay of 1 are the worst combinations. Out of the seven networks with the lowest mean AUC, six had a weight decay of 1, and seven had a learning rate of 0.01 (Fig. 17). The lowest mean AUC scores were between 0.91 and 0.93, which are relatively larger than the worst AUC scores of the model using the Adam optimizer.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	12.0	0.989091	0.0001	0.663551
0.00010	2.0	0.986061	0.0100	0.663551
0.00010	7.0	0.983636	0.0001	0.663551
0.00010	11.0	0.983636	0.0001	0.663551
0.00001	2.0	0.989091	1.0000	0.663551
0.00001	8.0	0.980738	1.0000	0.663551
0.00001	3.0	0.988636	0.0100	0.663551
0.00001	7.0	0.983182	0.0100	0.663551
0.00001	7.0	0.991515	0.0001	0.663551
0.00001	10.0	0.981818	0.0001	0.663551
0.00001	12.0	0.980606	0.0001	0.663551

Fig. 16: Models with the top 11 mean AUC scores for the modified VGG-19 model. The optimizer is SGD.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	2.0	0.930909	1.0000	0.663551
0.01000	3.0	0.927490	1.0000	0.663551
0.01000	7.0	0.933155	1.0000	0.663551
0.01000	9.0	0.939447	1.0000	0.663551
0.01000	13.0	0.919545	1.0000	0.663551
0.01000	14.0	0.937424	1.0000	0.663551
0.01000	3.0	0.937523	0.0100	0.663551

Fig. 17: Models with the bottom seven mean AUC scores for the modified VGG-19 model. The optimizer is SGD.

C. *Third Experiment*

The third experiment compared the performance of the two optimizers with augmented data. Four different learning rates and three different weight decays were tested for the Adam optimizer and SGD optimizer. All possible combinations of these learning rates and weight decay values were tested for ten epochs.

The Adam optimizer performed better with the augmented dataset than with the original dataset. The highest mean AUC for the Adam Optimizer was 0.988 (Fig. 18). When training with the data augmented dataset, the mean AUC reached 0.99. The maximum accuracy achieved with the augmented data was not larger than the maximum achieved with non-augmented data. Furthermore, the models trained with the augmented data did not reach mean AUC levels below 0.88 (Fig. 19). The models trained on augmented data contrast with the first Adam optimized models, which dropped to 0.83 for the mean AUC score.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	9.0	0.981818	1.0000	0.663551
0.00100	3.0	0.985455	0.0100	0.000000
0.00100	6.0	0.982424	0.0001	0.663551
0.00010	0.0	0.984848	0.0100	0.663551
0.00001	3.0	0.986667	1.0000	0.663551
0.00001	7.0	0.990303	1.0000	0.663551
0.00001	7.0	0.984717	0.0001	0.663551
0.00001	9.0	0.980817	0.0001	0.663551

Fig. 18: Models with the top eight mean AUC scores for the modified VGG-19 model. The optimizer is Adam.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01	3.0	0.892095	1.00	0.663551
0.01	4.0	0.880007	1.00	0.663551
0.01	2.0	0.888636	0.01	0.000000

Fig. 19: Models with the bottom three mean AUC scores for the modified VGG-19 model. The optimizer is Adam.

The SGD optimizer performed better with the augmented dataset than with the original dataset. The highest mean AUC for the SGD optimizer was 0.989 (Fig. 20). When training with the data augmented dataset, the mean AUC reached 0.992. The maximum accuracy achieved with the augmented data was not larger than the maximum achieved with non-augmented data. Furthermore, the models trained with the augmented data did not reach mean AUC levels below 0.935 (Fig. 21). The models trained with augmented data contrast with the first SGD optimized models, which dropped to 0.919 for the mean AUC score.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	3.0	0.982424	0.0100	0.663551
0.01000	3.0	0.983636	0.0001	0.663551
0.00100	3.0	0.980606	0.0100	0.663551
0.00001	1.0	0.983030	1.0000	0.663551
0.00001	2.0	0.992121	1.0000	0.663551
0.00001	4.0	0.985455	1.0000	0.663551
0.00001	3.0	0.985455	0.0001	0.663551
0.00001	4.0	0.983636	0.0001	0.663551

Fig. 20: Models with the top eight mean AUC scores for the modified VGG-19 model. The optimizer is SGD.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.00010	4.0	0.935250	0.01	0.663551
0.00001	0.0	0.933333	0.01	0.663551
0.00001	2.0	0.936917	0.01	0.663551

Fig. 21: Models with the bottom three mean AUC scores for the modified VGG-19 model. The optimizer is SGD.

D. Fourth Experiment

The fourth experiment compared the performance of the models looking at a smaller number of classifiers. For this experiment, labels that had three or fewer occurrences were removed. This removal left the dataset with 18 classes for the classifier. Four different learning rates and three different weight decays were tested for the Adam optimizer and SGD optimizer. All possible combinations of these learning rates and weight decay values were tested for 10 epochs. Augmented data was used for this experiment.

The results of this experiment provided evidence that the smaller number of classes does not impact model accuracy. As shown in Fig. 22, the largest accuracy for this experiment was 66.3%. However, the smaller number of classes allowed the models to reach a mean AUC of 0.99 more frequently.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	5.0	0.991358	0.0100	0.000000
0.01000	5.0	0.982948	0.0001	0.663551
0.00100	0.0	0.986420	1.0000	0.663551
0.00100	9.0	0.984645	1.0000	0.663551
0.00010	2.0	0.988889	0.0001	0.663551
0.00010	4.0	1.000000	0.0001	0.663551
0.00001	1.0	0.992824	1.0000	0.663551
0.00001	7.0	0.988889	1.0000	0.663551
0.00001	8.0	0.992593	1.0000	0.663551
0.00001	8.0	0.980247	0.0001	0.663551
0.00001	9.0	0.982716	0.0001	0.663551

Fig. 22: Models with the top 11 mean AUC scores for the modified VGG-19 model. The optimizer is Adam.

E. Fifth Experiment

The fifth experiment compared four different learning rates, and three different weight decays for the modified VGG-19 network with residual blocks. The tested learning rates were: 10^-2, 10^-3, 10^-4, and 10^-5. The tested weight decay values were: 10^0, 10^-2, and 10^-4. All possible combinations of these learning rates and weight decay values were tested for ten epochs. Adam and SGD were used as the optimizers for this experiment. The data was augmented for this testing.

For the Adam optimizer training, the residual block performed equal to the modified VGG-19. Fig. 23 shows the results for the model that provided the best mean AUC. These results were common among the different combinations of learning rate and weight decay. The mean AUC oscillated throughout the first ten epochs for most learning rate and weight decay combinations. The validation dataset had the larger mean AUC for the majority of epochs for most combinations. The loss experienced an exponential decay for the majority of combinations (Fig. 24). The accuracy fluctuated throughout the ten epochs (Fig. 25). This trend was not present for the modified VGG-19 without residual blocks.

Looking at the best performing models, all of the models reach a mean AUC of 0.98 (Fig. 26). In contrast to the first modified VGG-19 network, many models with a high mean AUC score do not correspond with the highest accuracy percentages. The maximum accuracy percentage is the same as the residual model and the first modified VGG-19.

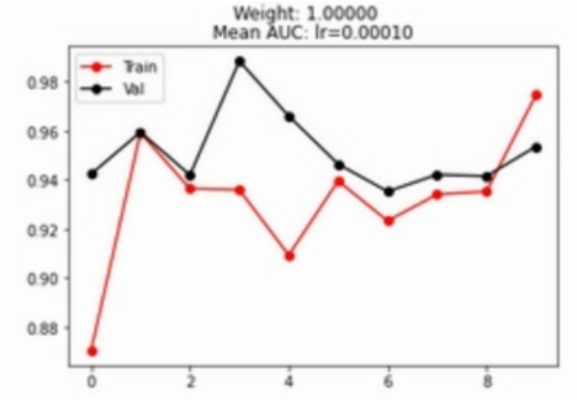

Fig. 23: Mean AUC comparison for the modified VGG-19 model with residual blocks across 10 epochs. This model had a weight decay of 1 and learning rate of 0.0001.

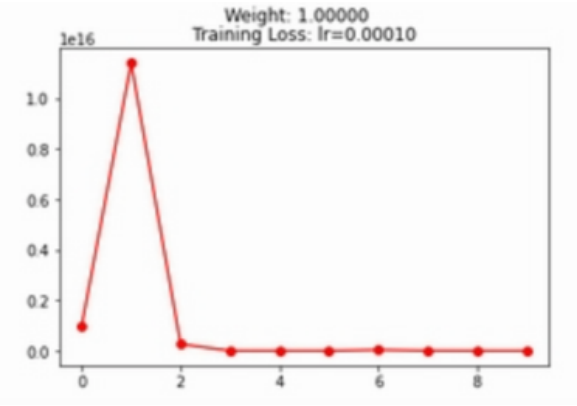

Fig. 24: Training Loss for the modified VGG-19 model with residual blocks across 10 epochs. This model had a weight decay of 1 and learning rate of 0.0001.

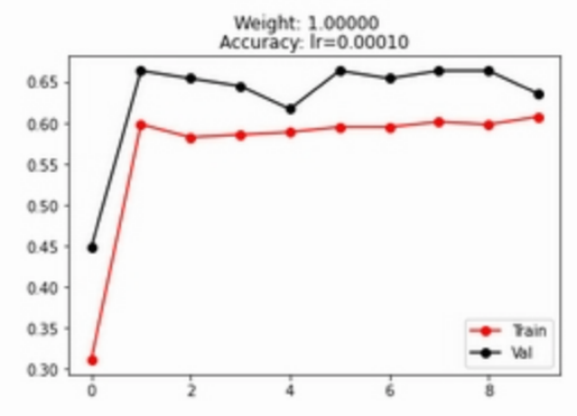

Fig. 25: Accuracy comparison for the modified VGG-19 model with residual blocks across 10 epochs. This model had a weight decay of 1 and learning rate of 0.0001.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	3.0	0.980580	0.0001	0.654206
0.00010	3.0	0.988485	1.0000	0.644860
0.00001	8.0	0.980606	1.0000	0.654206
0.00001	9.0	0.983333	1.0000	0.663551
0.00001	5.0	0.981212	0.0001	0.663551
0.00001	7.0	0.987273	0.0001	0.654206
0.00001	9.0	0.983030	0.0001	0.663551

Fig. 26: Models with the top 11 mean AUC scores for the modified VGG-19 model with residual blocks. The optimizer is Adam.

For the SGD optimizer training, the residual block performed equal to the modified VGG-19. Looking at the best performing models, all of the models reach a mean AUC of 0.98 (Fig. 27). One model was able to get 0.992 for its mean AUC score. Similar to the first modified VGG-19, all high mean AUC scores correspond with the maximum accuracy. The maximum accuracy percentage for the residual model is the same for the the modified VGG-19.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	8.0	0.982635	0.0100	0.000000
0.01000	4.0	0.982424	0.0001	0.663551
0.01000	6.0	0.987879	0.0001	0.663551
0.00100	0.0	0.983030	1.0000	0.663551
0.00100	6.0	0.984242	1.0000	0.663551
0.00100	7.0	0.980606	0.0001	0.663551
0.00010	5.0	0.984848	0.0100	0.663551
0.00001	1.0	0.992121	1.0000	0.663551
0.00001	6.0	0.983636	1.0000	0.663551
0.00001	1.0	0.985455	0.0001	0.663551

Fig. 27: Models with the top 10 mean AUC scores for the modified VGG-19 model with residual blocks. The optimizer is SGD.

F. Sixth Experiment

The sixth experiment compared four different learning rates, and three different weight decays for the MNIST CNN. The tested learning rates were: 10^-2, 10^-3, 10^-4, and 10^-5. The tested weight decay values were: 10^0, 10^-2, and 10^-4. All possible combinations of these learning rates and weight decay values were tested for ten epochs. Adam and SGD were used as the optimizer for this experiment. The data was augmented for this testing.

For the Adam and SGD optimizer training, the MNIST CNN performed equal to the modified VGG-19 and residual block VGG-19. Looking at the best performing models, all models reach a mean AUC of 0.98 for Adam optimizer (Fig. 28). The MNIST CNN was able to reach a mean AUC of 0.98 for the

SGD optimizer (Fig. 29). The maximum accuracy percentage is the same for the MNIST model and the residual and first modified VGG-19.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	7.0	0.984848	0.0100	0.663551
0.01000	3.0	0.986667	0.0001	0.663551
0.00100	8.0	0.987484	1.0000	0.093458
0.00100	4.0	0.980152	0.0001	0.663551
0.00100	8.0	0.980606	0.0001	0.663551
0.00010	0.0	0.981818	1.0000	0.663551
0.00010	8.0	0.981818	1.0000	0.663551
0.00010	5.0	0.987879	0.0001	0.663551
0.00001	2.0	0.980606	1.0000	0.663551
0.00001	3.0	0.989091	1.0000	0.663551
0.00001	0.0	0.981686	0.0001	0.663551
0.00001	3.0	0.981212	0.0001	0.663551

Fig. 28: Models with the top 12 mean AUC scores for the modified VGG-19 model with residual blocks. The optimizer is Adam.

Learning Rate	Epoch	Mean AUC	Weight Decay	%Acc
0.01000	7.0	0.984848	0.0100	0.663551
0.01000	3.0	0.986667	0.0001	0.663551
0.00100	8.0	0.987484	1.0000	0.093458
0.00100	4.0	0.980152	0.0001	0.663551
0.00100	8.0	0.980606	0.0001	0.663551
0.00010	0.0	0.981818	1.0000	0.663551
0.00010	8.0	0.981818	1.0000	0.663551
0.00010	5.0	0.987879	0.0001	0.663551
0.00001	2.0	0.980606	1.0000	0.663551
0.00001	3.0	0.989091	1.0000	0.663551
0.00001	0.0	0.981686	0.0001	0.663551
0.00001	3.0	0.981212	0.0001	0.663551

Fig. 29: Models with the top 12 mean AUC scores for the modified VGG-19 model with residual blocks. The optimizer is SGD.

G. Seventh Experiment

The final experiment evaluated the residual model after 50 epochs. The chosen learning rate was 0.00001, and the weight decay value was 0.0001. These values were chosen because they performed well and had a decreasing loss in the ten-epoch experiment. As shown in Figure 30, the mean AUC oscillated throughout 50 epochs. The validation mean AUC was larger for the majority of epochs. The loss decreased exponentially during the experiment (Fig. 31). Furthermore, the accuracy oscillated as much as the mean AUC (Fig. 32). When tested on the test dataset, the model had an accuracy of 57.009% and a mean AUC of 0.974.

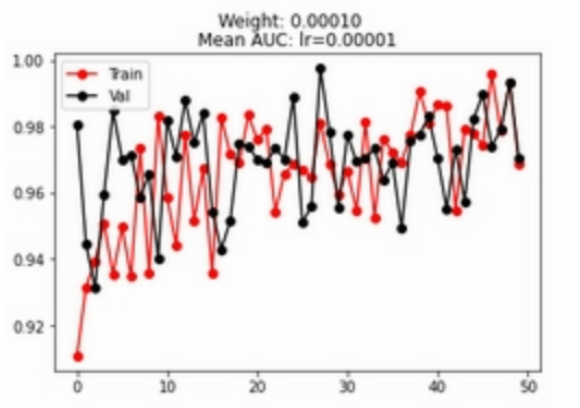

Fig. 30: Mean AUC comparison for the modified VGG-19 model with residual blocks across 50 epochs. This model had a weight decay of 0.0001 and learning rate of 0.00001.

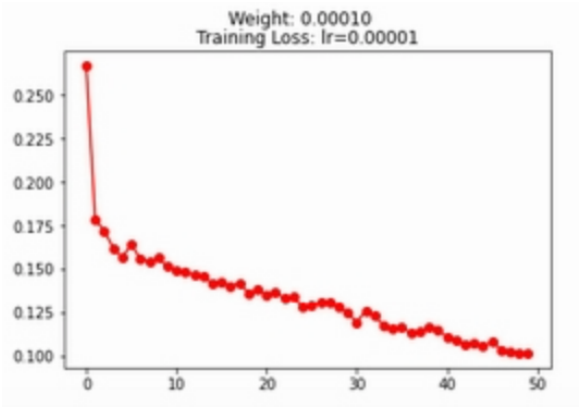

Fig. 31: Training Loss for the modified VGG-19 model with residual blocks across 50 epochs. This model had a weight decay of 0.0001 and learning rate of 0.00001.

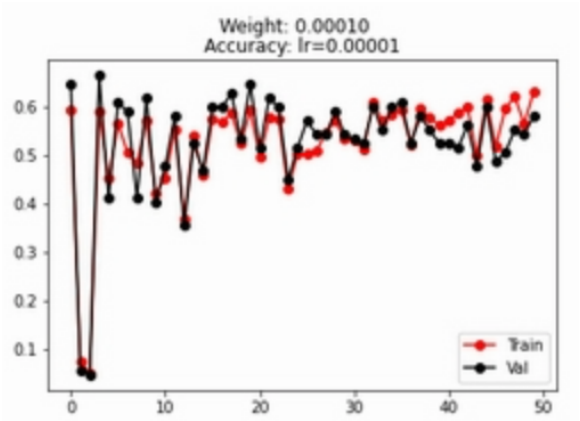

Fig. 32: Accuracy comparison for the modified VGG-19 model with residual blocks across 50 epochs. This model had a weight decay of 0.0001 and learning rate of 0.00001.

V. Conclusions

One lesson that I learned from this project is that every model has different optimal parameters. One learning rate may work for a certain optimizer and learning rate, but it will perform poorly for a different optimize and learning rate. The same conclusion can be made about the weight decay. There were multiple models that performed poorly with the weight decay equal to 1. However, one of the better performing models for a different optimizer had a weight decay of 1. I believe it would be more beneficial to perform more research on different optimizers to determine the types of problems that they are designed to help solve. This might help reduce the possibilities that should be tested.

Another lesson that I learned was that the simple CNN can perform as good as more complex CNNs. In this project, the MNIST CNN performed equal to the more complex CNNs, such as the modified VGG-19 and the VGG-19 with the residual blocks. Furthermore, MNIST was close to matching the maximum mean AUC scores. This project was evidence that complex networks do not by default result in a drastically better predictor.

One future modification might be to include the full number of channels in the VGG-19 models. Other modifications could be to test different optimizers or to utilize a optimizer scheduler to reduce the learning rate during training.

References

[1] CDC, "COVID Data Tracker Weekly Review | CDC," May 07, 2021. https://www.cdc.gov/coronavirus/2019-ncov/covid-data/covidview/index.html (accessed May 08, 2021).

[2] CDC, "Estimated Influenza Illnesses, Medical visits, Hospitalizations, and Deaths in the United States — 2018–2019 influenza season | CDC," Jan. 08, 2020. https://www.cdc.gov/flu/about/burden/2018-2019.html (accessed May 08, 2021).

[3] B. L. Meatherall, D. Gregson, T. Ross, J. D. D. Pitout, and K. B. Laupland, "Incidence, Risk Factors, and Outcomes of Klebsiella pneumoniae Bacteremia," *Am. J. Med.*, vol. 122, no. 9, pp. 866–873, Sep. 2009, doi: 10.1016/j.amjmed.2009.03.034.

[4] MedlinePLus, "Bacterial Infections: MedlinePlus," Jun. 28, 2018. https://medlineplus.gov/bacterialinfections.html (accessed May 08, 2021).

[5] CDC, "Antibiotic Do's & Don'ts | Antibiotic Use | CDC," Jan. 31, 2020. https://www.cdc.gov/antibiotic-use/do-and-dont.html (accessed May 08, 2021).

[6] N. Bansal, "Classification Of X-ray Images For Detecting Covid-19 Using Deep Transfer Learning," May 2020, doi: 10.21203/rs.3.rs-32247/v1.

[7] H. Behzadi-khormouji *et al.*, "Deep learning, reusable and problem-based architectures for detection of consolidation on chest X-ray images," *Comput. Methods Programs Biomed.*, vol. 185, p. 105162, Mar. 2020, doi: 10.1016/j.cmpb.2019.105162.

[8] D. Dansana *et al.*, "Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm," *Soft Comput.*, 2020, doi: 10.1007/s00500-020-05275-y.

[9] K. K. Bressem, L. C. Adams, C. Erxleben, B. Hamm, S. M. Niehues, and J. L. Vahldiek, "Comparing different deep learning architectures for classification of chest radiographs," *Sci. Rep.*, vol. 10, no. 1, p. 13590, Dec. 2020, doi: 10.1038/s41598-020-70479-z.

[10] D. Dai, L. Yu, and H. Wei, "Parameters Sharing in Residual Neural Networks," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1393–1410, Apr. 2020, doi: 10.1007/s11063-019-10143-4.

[11] J. Duan, T. Shi, H. Zhou, J. Xuan, and S. Wang, "A novel ResNet-based model structure and its applications in machine health monitoring," *JVC/Journal Vib. Control*, vol. 27, no. 9–10, pp. 1036–1050, May 2021, doi: 10.1177/1077546320936506.

[12] J. Naranjo-Alcazar, S. Perez-Castanos, I. Martin-Morato, P. Zuccarello, and M. Cobos, "On the performance of residual block design alternatives in convolutional neural networks for end-to-end audio classification." Accessed: May 08, 2021. [Online]. Available: https://urbansounddataset.weebly.com/urbansound8k.html.

[13] C. Anusha and P. S. Avadhani, "Optimal Accuracy Zone Identification in Object Detection Technique-A Learning Rate Methodology," *Int. J. Eng. Adv. Technol.*, no. 9, pp. 2249–8958, 2019, doi: 10.35940/ijeat.A2258.109119.