Capstone Project 2: Chest X-Ray Image Classification Using Deep Learning

Milestone Report 2

Jonathon Poage
Springboard Data Science Career Track
05/30/2019

Table of Contents

Introduction

Pneumonia is the leading infectious cause of death for children worldwide. According to the World Health Organization (WHO), pneumonia caused 920,136 deaths of children under the age of 5 in 2015.[1] Pneumonia affects children everywhere, but fatalities are most common in developing areas of the world. Fatality rates could potentially be reduced if patients are diagnosed and referred for proper treatment efficiently.

The two leading causes of pneumonia are bacterial and viral pathogens. Viral pneumonia is treated with supportive care, and bacterial pneumonia requires treatment with antibiotics. Low cost treatments for pneumonia are available. The issue is that fast and accurate diagnosis of pneumonia can be challenging in facilities that have limited resources.

Diagnosis of pediatric pneumonia may involve one or more medical tests. According to the National Heart, Lung, and Blood Institute (NHLBI), the best diagnostic test involves inspecting chest radiograph (X-ray) images for signs of inflammation in the lungs.[2] This project aims to develop methods to facilitate diagnosis of pneumonia from chest X-ray images.

This project's GitHub repository can be found here.

The Capstone Milestone Report 1 can be found here.

Goals of Analysis

This project aims to develop a classification model to predict the presence or absence of pneumonia in patients, based on chest X-ray images. In addition, the model will distinguish between bacterial pneumonia and viral pneumonia. The ultimate goal of this project is to provide an automated, fast, and accurate method for interpreting chest X-ray images, in order to facilitate diagnosis of pediatric pneumonia.

Data

The raw data was obtained for this project by downloading the file ZhangLabData.zip from the following website: Mendeley.[3] The raw data set contains thousands of anterior-posterior chest radiograph images from Guangzhou Women and Children's Medical Center. Kermany et al[4] describe the methods they used to collect and prepare the images in the raw data set.

The raw data was preprocessed for this analysis using Python. The image files were split into train, validation, and test sets. Feature and target arrays were created from the image data and

prepared to be used as input for the classification models during the analysis. See the Milestone Report 1 for more details about the data preprocessing that was performed for this project.

Analysis

The analysis for this project involved three parts. The analysis framework is discussed below. Note that additional follow up analysis may be performed after the submission of this milestone report.

The first part of the analysis examined properties of the images in general. The results of this part of the analysis were discussed in the Milestone Report 1.

The second part of the analysis applied deep learning techniques to develop a set of multi-class classification models. The models predict the probabilities that a chest X-ray image indicates the presence of bacterial pneumonia, the presence of viral pneumonia, or the absence of pneumonia.

The third part of the analysis evaluated the predictive performance of the most promising models on a test set. The predictive performances of the models were evaluated based on various scoring metrics, such as recall and F_1 score. The practical significance of the best performing model was assessed based on its ability to predict the presence or absence of pneumonia as well as its ability to distinguish between bacterial and viral pneumonia.

Model Creation

Three Convolutional Neural Networks (CNNs) were developed for this analysis. Each CNN model consists of a convolutional base followed a classifier. Each model takes an array of image pixel values as input and produces a class probability distribution as output. The models were trained using Keras with a TensorFlow backend.

The first model was custom built for this project. The second and third models utilized transfer learning by adapting the pre-trained VGG16 model's convolutional base.[5] The second model used the convolutional base of the VGG16 model as a feature extractor by fixing the pre-trained weights. The third model fine tuned the convolutional base of the VGG16 model, by fixing the pre-trained weights of the lower convolutional layers while allowing the weights in the upper convolutional layers to be trained.

The code containing the functions that were used to create the models for this analysis can be found here.

Classifiers

A neural network (NN) classifier was custom built for this project. All three models used this NN classifier. Time permitting, additional alternative classifiers may be developed for this project.

Neural Network Classifier

The NN classifier takes input from the convolutional base and produces a class probability distribution as output. The NN is composed of two fully connected (FC) layers, each followed by a dropout layer, and an output layer. The hidden FC layers use ReLU activation, and the output layer uses softmax activation. Figure 1 shows the architecture of the NN classifier.

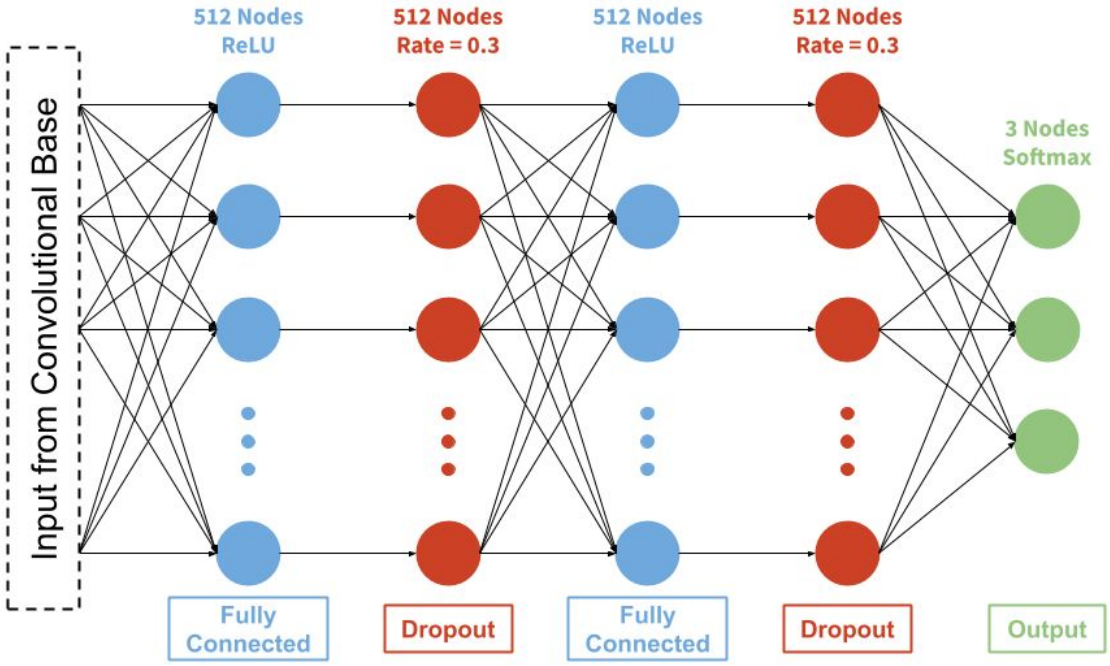

Figure 1: Architecture of the Neural Network classifier.

Convolutional Bases

The convolutional bases used for the models are described in the following sections.

Base 1: Convolutional Neural Network

Convolutional base 1 was built from scratch for this analysis. The base consists of four convolutional layers, each paired with a max pooling layer, followed by a flatten layer. The convolutional layers all use a 3 x 3 kernel and ReLU activation.The pooling layers all use a 2 x 2

pool size. Figure 2 shows the architecture of convolutional base 1.

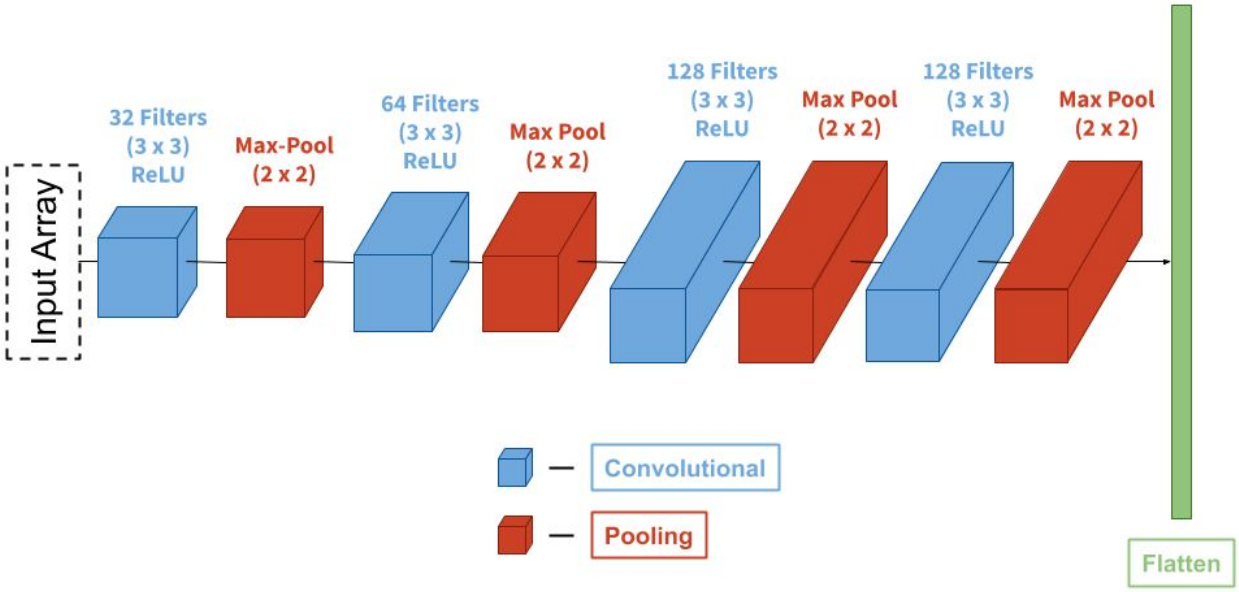

Figure 2: Architecture of the custom CNN convolutional base.

Base 2: VGG16 with Fixed Weights

Convolutional base 2 utilizes the VGG16 model's base with weights pre-trained on ImageNet. The VGG16 model was obtained using Keras.[6] The top layers that serve as VGG16's classifier were excluded, and a flatten layer was added to the remaining base. The weights of this convolutional base were fixed during model training. Convolutional base 2 serves as a feature extractor for the classifier.

Figure 3 shows the architecture of convolutional base 2.

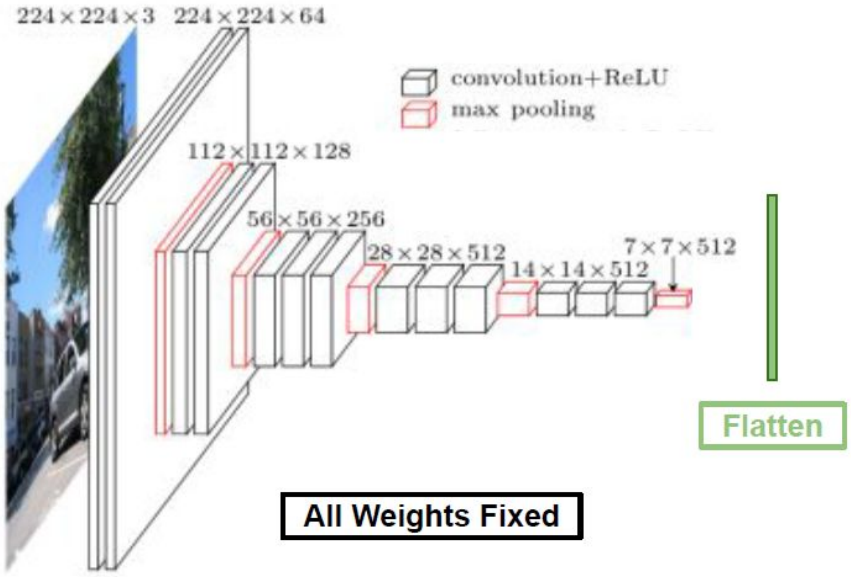

Figure 3: Architecture of convolutional base 2. All weights in the convolutional base were fixed during training. The original, unmodified source image was obtained from heuritech.[7]

Base 3: VGG16 with Fine-Tuning

Convolutional base 3 utilizes the VGG16 model's base initialized with weights pre-trained on ImageNet. The VGG16 model was obtained using Keras. The top layers that serve as VGG16's classifier were excluded, and a flatten layer was added to the remaining base. The weights of the bottom three convolutional blocks were fixed during model training. The weights of the top two convolutional blocks were not fixed, and were updated during training.

Figure 4 shows the architecture of convolutional base 3.

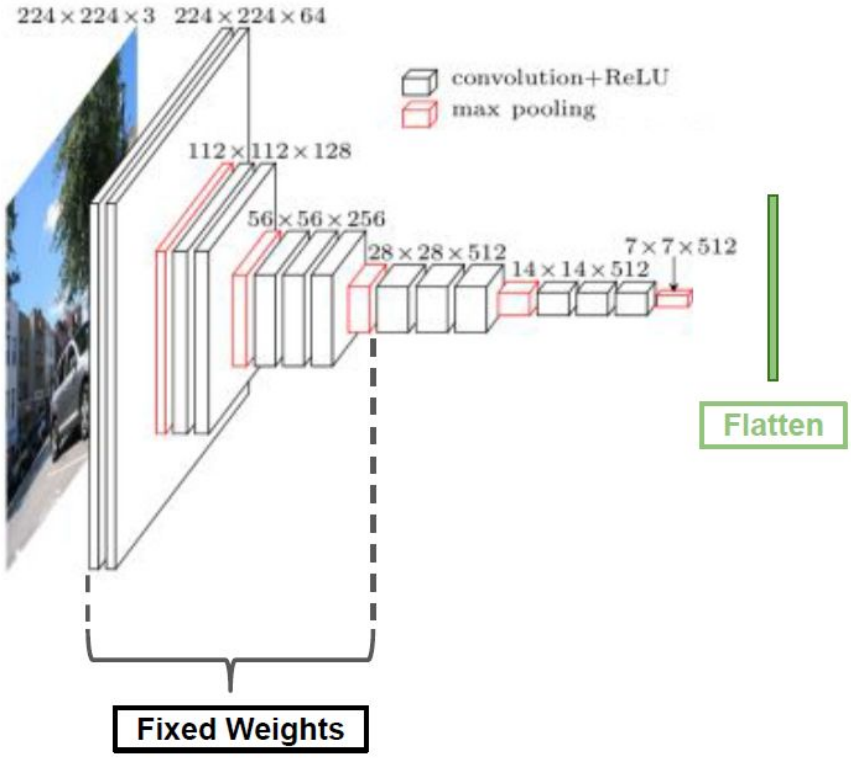

Figure 4: Architecture of convolutional base 3. All weights in the bottom three convolutional blocks were fixed during training. The original, unmodified source image was obtained from heuritech.[7]

Models

The models were created for this analysis by combining the convolutional bases with the NN classifier. The set of models is summarized in table 1.

Table 1: Summary of the models used in this analysis. The convolutional base and classifier are specified for each of the models.

Model	Convolutional Base	Classifier
Model 1	CNN	NN Classifier
Model 2	VGG16 with Fixed Weights	NN Classifier
Model 3	VGG16 with Fine-Tuning	NN Classifier

Model Fitting

The models were fit using Keras during the analysis. The fits were performed using the Adam optimization algorithm. Categorical Cross-Entropy was used as the loss metric, though accuracy scores were evaluated as well at the end of each epoch.

The following scripts were used to fit the models for this analysis: model 1, model 2, and model 3. These model fitting scripts use functions from two additional scripts, found here and here.

Data Preparation

The image data in the train and validation sets was transformed before fitting each model. For model 1, the image pixel values were rescaled from the range [0, 255] to the range [0, 1]. For models 2 and 3, the feature arrays were transformed using Keras's vgg16.preprocess_input function.

Training data was generated in batches from the transformed image data in the train set. The training data was generated with image augmentation in order to improve the model's ability to generalize to unseen data. Validation data was generated in batches from transformed image data in the validation set. No image augmentation was performed on the validation data.

Anterior-posterior chest X-ray images show a view of the patient from a particular orientation, so care was taken to restrict the transformations performed during the image augmentation. For example, horizontal and vertical reflections were not permitted, while minor rotations were permitted.

Fit Results

The training histories are plotted for each of the three models. The training history plots show the training and validation accuracy and loss scores that were evaluated at the end of each epoch.

Figure 5 shows the training history for model 1.

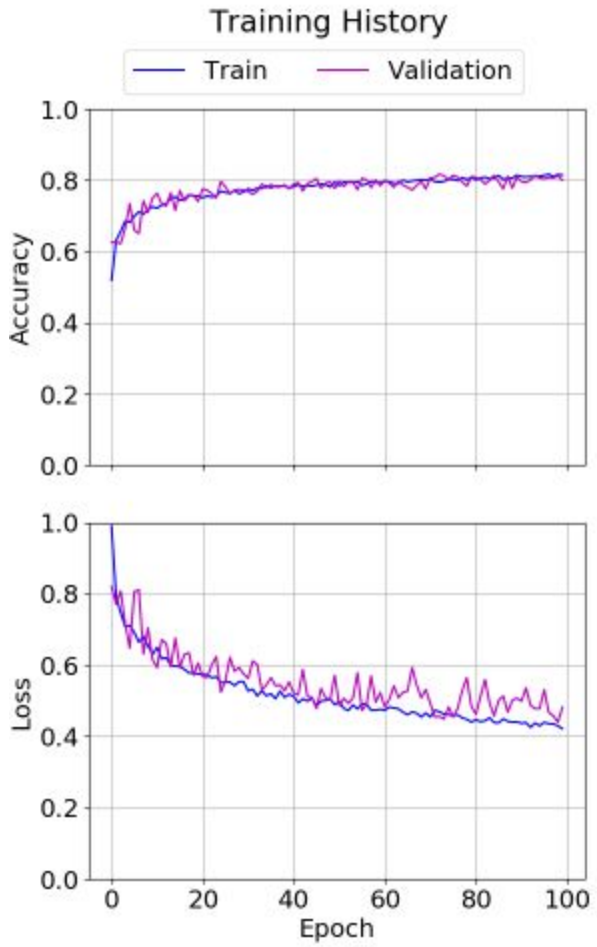

Figure 5: Training history for model 1. The train and validation accuracy scores are comparable, as are the loss scores.

The training history for model 1 indicates that the model generalizes well to unseen data. The train and validation accuracy scores reached values close to 0.8, and the loss scores reached values around 0.4 to 0.5.

Figure 6 shows the training history for model 2.

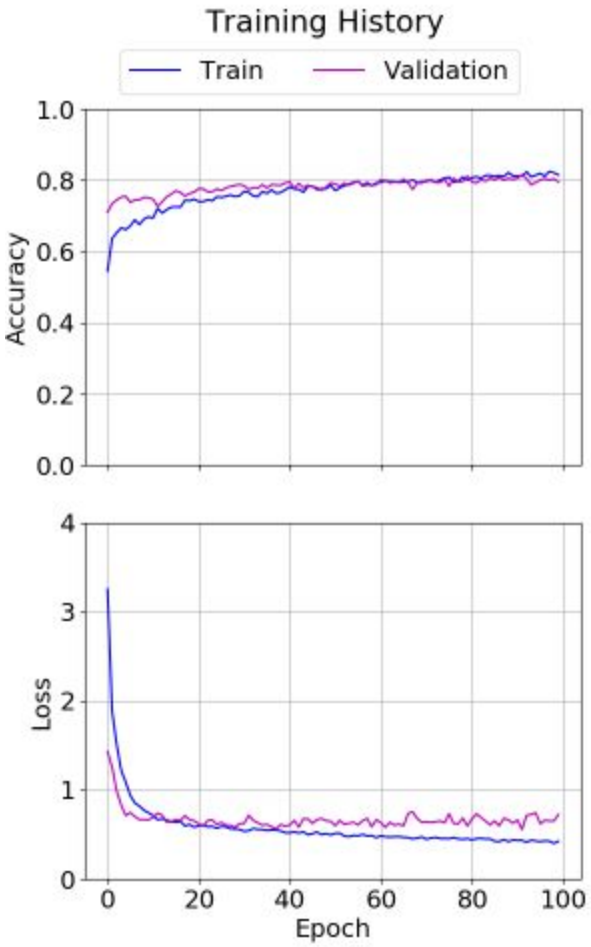

Figure 6: Training history for model 2. The train and validation accuracy scores are comparable at later epochs. The train and validation loss scores are comparable as well, but the train loss scores are slightly lower than the validation loss scores at later epochs.

The training history for model 2 indicates that the model generalizes fairly well to unseen data. The train and validation accuracy scores reached values close to 0.8. The train loss scores reached values around 0.4, while the validation loss scores reached values around 0.6.

Figure 7 shows the training history for model 3.

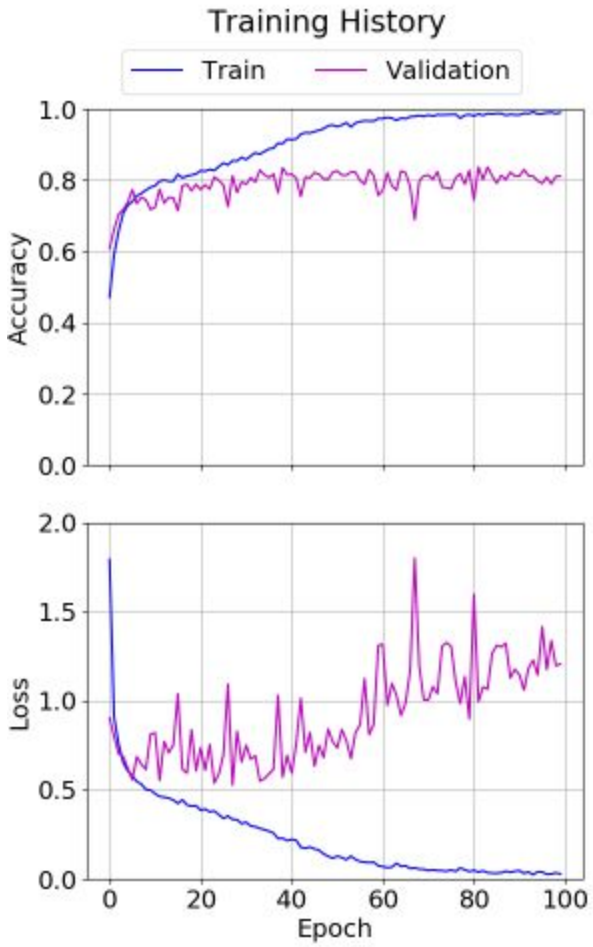

Figure 7: Training history for model 3. The train accuracy scores are higher than the validation accuracy scores for most of the epochs. Similarly, the train loss scores are lower than the validation loss scores for most of the epochs. The training history indicates that model 3 has been overfit to the training data.

The training history for model 3 indicates that the model does not generalize very well to unseen data. There are large discrepancies between the train and validation accuracy scores as well as between the train and validation loss scores at later epochs.

Fit Summary

Models 1 and 2 show similar performance in terms of validation and loss scores. Model 3 shows similar performance in terms of validation accuracy, but worse performance in terms of validation loss scores.

Model 3 appears to have been overfit to the training data. Model 3 did outperform models 1 and 2 in terms of train accuracy and loss scores. It is possible that a model that

utilizes VGG16's base with fine tuning could outperform models 1 and 2, if the appropriate adjustments are made to the convolutional base and the model fitting procedure.

Model Evaluation

The predictive performances of models 1 and 2 were evaluated using the image data in the test set. The predictive performance of model 3 was not evaluated, because the model seems to have been overfit to the training data. The results of the model evaluations are discussed in the following sections.

The image data in the test set was transformed before evaluating each model, using the same transformations that were applied to the image data in the train and validation sets before fitting the models.

The script used to evaluate the models for this analysis can be found here. This model evaluation script uses functions defined in another script found here.

Evaluation Criteria

The predictive performance of each model was evaluated by considering all of the following metrics:

1. Bacterial pneumonia recall score.
2. Proportion of viral pneumonia images misclassified as normal.
3. Weighted average F_1 score.

The following discussion describes the motivation for using the metrics listed above to evaluate the predictive performance of the models.

The test data set was imbalanced. The model evaluation criteria must be suitable for imbalanced classes.

Certain types of misclassification would have more severe impacts on patients than other types of misclassification. Misclassification of bacterial pneumonia images would have the most severe impact on patients, because bacterial pneumonia requires the most intensive treatment. Misclassification of viral pneumonia images as normal would also have a severe impact on patients. The model evaluation criteria must prioritize these severe types of misclassifications.

The less severe types of misclassification are also of practical significance. Misclassification of viral pneumonia images as bacterial pneumonia would inconvenience patients by subjecting them to unnecessary treatment, as would misclassification of normal images. The model evaluation criteria must assess each model's overall ability to correctly classify images.

Model 1 Evaluation

Figure 8 shows the confusion matrix for model 1.

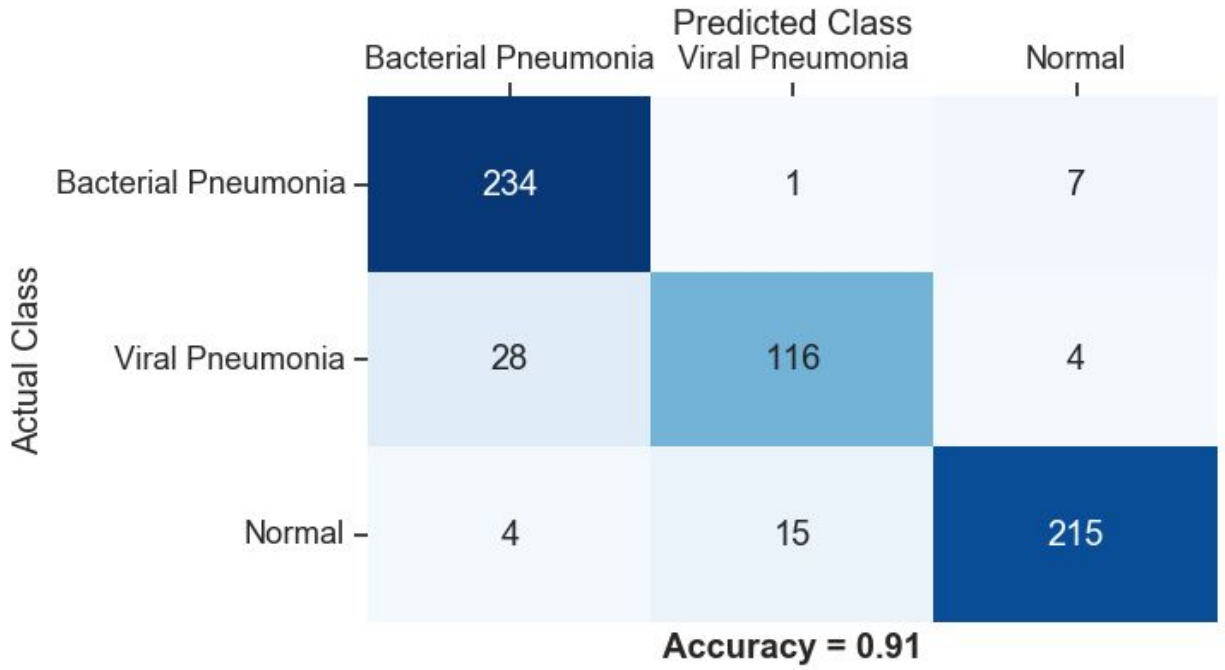

Figure 8: Confusion matrix for model 1. The accuracy score indicates that the model correctly classified 91% of the images in the test set.

The upper right corner of the confusion matrix displays the three misclassification types that most severely impact patients. This region of the confusion matrix for model 1 contains 12 instances of misclassified images. In particular, 4 viral pneumonia images were misclassified as normal.

The confusion matrix also shows that 18.9% of the viral pneumonia images were misclassified as bacterial pneumonia. This indicates that model 1 had relative difficulty classifying viral pneumonia images.

Table 2 shows a classification report for model 1.

Table 2: Classification report for model 1.

	f1-score	precision	recall	support
Bacterial Pneumonia	0.92	0.88	0.97	242
Viral Pneumonia	0.83	0.88	0.78	148
Normal	0.93	0.95	0.92	234
Micro Average	0.91	0.91	0.91	624
Macro Average	0.89	0.90	0.89	624
Weighted Average	0.90	0.91	0.91	624

The bacterial pneumonia recall score indicates that model 1 correctly classified 97% of the bacterial pneumonia images. The weighted average F_1 score of 0.90 indicates fairly strong overall predictive performance. The viral pneumonia F_1 and recall scores were much lower than the corresponding scores for the bacterial pneumonia and normal classes, corroborating the notion that model 1 had relative difficulty classifying viral pneumonia images.

Model 2 Evaluation

Figure 9 shows the confusion matrix for model 2.

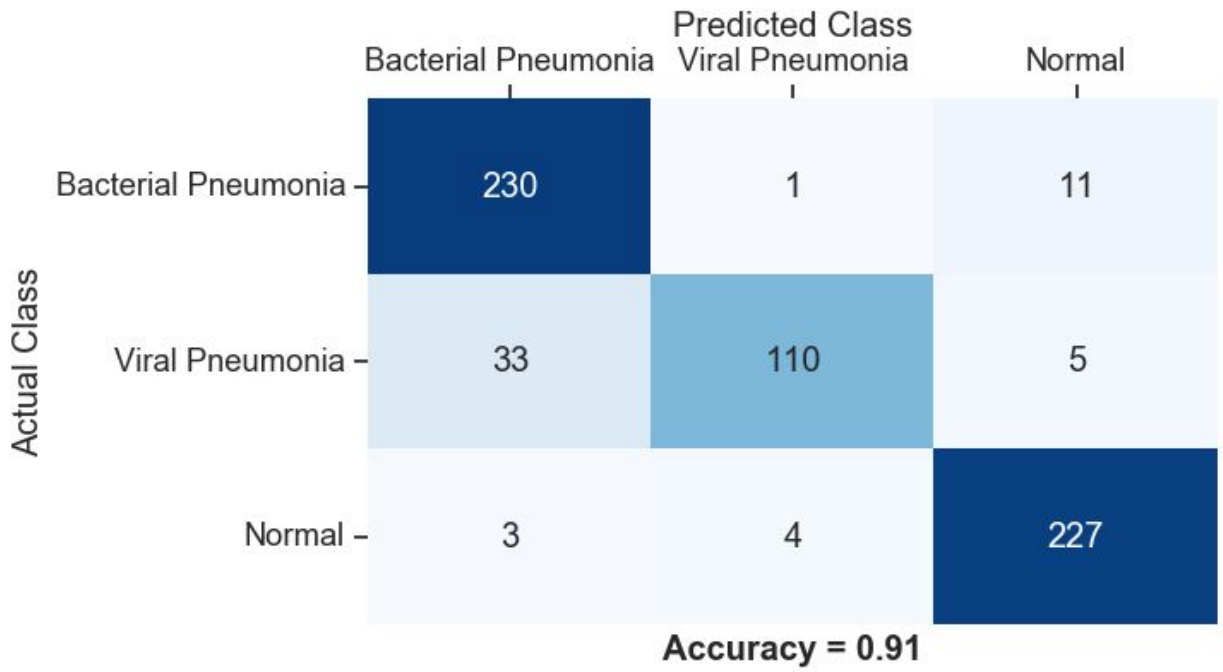

Figure 9: Confusion matrix for model 2. The accuracy score indicates that the model correctly classified 91% of the images in the test set.

The upper right corner of the confusion matrix displays the three misclassification types that most severely impact patients. This region of the confusion matrix for model 2 displays 17 instances of misclassified images. In particular, 5 viral pneumonia images were misclassified as normal.

The confusion matrix also shows that 22.3% of the viral pneumonia images were misclassified as bacterial pneumonia. This indicates that model 2 had relative difficulty classifying viral pneumonia images.

Table 3 shows a classification report for model 2.

Table 3: Classification report for model 2.

	f1-score	precision	recall	support
Bacterial Pneumonia	0.91	0.86	0.95	242
Viral Pneumonia	0.84	0.96	0.74	148
Normal	0.95	0.93	0.97	234
Micro Average	0.91	0.91	0.91	624
Macro Average	0.90	0.92	0.89	624
Weighted Average	0.91	0.91	0.91	624

The bacterial pneumonia recall score indicates that model 2 correctly classified 95% of the bacterial pneumonia images. The weighted average F_1 score of 0.91 indicates fairly strong overall predictive performance. The viral pneumonia F_1 and recall scores were much lower than the corresponding scores for the bacterial pneumonia and normal classes, corroborating the notion that model 2 had relative difficulty classifying viral pneumonia images.

Model Comparison

Overall, models 1 and 2 showed relatively similar predictive performance. Model 1 had a slightly better recall score for bacterial pneumonia, as well as 4 instances of viral pneumonia images misclassified as normal compared to the 5 instances for model 2. Model 2 showed a slightly higher weighted average F_1 score.

Model 1 showed slightly better predictive performance regarding the first two evaluation criteria, while model 2 showed slightly better predictive performance regarding the third criterion. Both models demonstrated admirable predictive performance, and the choice of best model ultimately depends on the prioritization of the evaluation criteria. The first criterion was prioritized for this analysis, and so model 1 was selected as the best model.

Summary

The analysis for this project has been completed. The raw data has been obtained and processed. The general image properties were examined in order to evaluate typical

characteristics of chest X-rays for each image class. Three CNN classification models were created and evaluated. The best model in terms of predicted performance was identified according to a set of evaluation criteria.

The next steps for the project are described in the following section.

Next Steps

An in-depth analysis will be conducted on the best performing model in order to evaluate its ability to distinguish pneumonia images from normal images, without specifying the type of pneumonia. The model's predictive performance for this binary classification task will be evaluated using various scoring metrics, such as accuracy, ROC AUC, and average precision. The practical significance of the best performing model will be assessed based on its ability to predict the presence or absence of pneumonia as well as its ability to distinguish between bacterial and viral pneumonia.

Time permitting, the analysis may include further follow up analysis. The extended analysis could potentially include visualizations of occlusion maps. Occlusion maps could be overlayed over sample images to visually indicate the regions that contributed the highest importance to the deep learning algorithm. The extended analysis could also introduce and evaluate new models. The full scope of the project will be determined after the submission of this milestone report.

The final report for this project will contain a summary of the results of the full analysis. The final report will describe the practical significance of the results with respect to the goals of the project, and provide actionable recommendations.

Reference List

1. Pneumonia. World Health Organization Web site. https://www.who.int/news-room/fact-sheets/detail/pneumonia. Published November 7, 2016. Accessed April 30, 2019.

2. Pneumonia. National Heart, Lung, and Blood Institute Web site. https://www.nhlbi.nih.gov/health-topics/pneumonia. Accessed April 30, 2019.

3. Kermany D, Zhang K, Goldbaum M. Large dataset of labeled optical coherence tomography (OCT) and chest X-ray images. Mendeley Web site. https://data.mendeley.com/datasets/rscbjbr9sj/3. Published June 1, 2018. Accessed April 13, 2019.

4. Kermany DS, Goldbaum M, Cai W, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*. 2018;172(5):1122-1131. doi:10.1016/j.cell.2018.02.010.

5. Karen S, Andrew Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. Paper presented at: 3rd International Conference on Learning Representations; May 7-9, 2015; San Diego, CA. https://arxiv.org/pdf/1409.1556.pdf. Accessed May 30, 2019.

6. Applications. keras.io. https://keras.io/applications/. Accessed May 1, 2019.

7. A Brief Report of the Heuritech Deep Learning Meetup #5. heuritech.com. https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/. Published February 29, 2016. Accessed May 28, 2019.