|  |
| --- |
| **Detecting Bacterial and Viral Pneumonia in Chest X-Rays Using Deep Learning Models** |

**Michael Morton**

Department of Computer Science

Georgia State University

Atlanta, GA 30269

mmorton11@student.gsu.edu

**Abstract**

In 2017, pneumonia was responsible for 15% of the annual deaths for children under the age of five [4]. The large number of these deaths could have been prevented by better detection methods and access to the correct treatment. This project uses deep learning binary and categorical image classification on chest x-rays in order to determine the most accurate and efficient model for identifying the presence of bacterial and viral pneumonia in chest x-rays. Three different models were developed over the course of this project. Model #1 and Model #2 are binary image classification models that predict the presence or absence of pneumonia in chest x-rays. Model #3 is a categorical image classification model that predicts if a chest x-ray either contains bacterial or viral pneumonia or is healthy. Model #1 was able to achieve a test accuracy of 94%, while having a harder time correctly identifying chest x-rays containing pneumonia. Model #2 obtained a test accuracy of 97%, while slightly over predicting the number of x-rays containing pneumonia. Model #3 achieved a test accuracy, and it had the most difficulty identifying x-rays that contained viral pneumonia. Overall, all three models were able to achieve test accuracies over 90%. The goal of this project is to help radiologist with (1) accurately diagnosing pneumonia and (2) differentiating bacterial and viral pneumonia in chest x-rays.

**1 Background Information**

Pneumonia is an infection that causes an inflammation of the air sacks in one or both lungs. This inflammation is caused when the air sacks become filled with fluid or purulent material, which causes the symptoms of pneumonia. These symptoms include chest pain when breathing or coughing, confusion or changes in mental awareness, cough with phlegm or pus, fatigue, fever, sweating or shaking chills, reduced body temperature, nausea, vomiting, diarrhea, and shortness of breath. These symptoms will vary for each patient, but patients that are in high-risk groups should seek medical attention immediately as pneumonia can become a life-threatening condition. Patients in these high-risk groups include adults over the age of 65, children under the age of 2, people with underlying health conditions or compromised immune systems, and people receiving chemotherapy. There are multiple agents that can cause pneumonia, but the main causes are bacteria and viruses. The different types of pneumonia are named for the agent that causes the infection, so these two types of pneumonia are known as bacterial pneumonia and viral pneumonia [3]. These different forms of pneumonia produce distinct images in chest x-rays. The differences in these x-ray images will be the basis for the deep learning image classification in this project.

Before I began pursuing a degree in computer science, I was a laboratory scientist. I have Bachelor of Science degrees in Biology and Environmental Health from the University of Georgia. I intended to use this project as an opportunity to link my old career with my new career. This project gave me the chance to blend my vast biological knowledge with my interest in machine learning, allowing me to gain more experience using deep learning techniques.

**1.1 Problem and Goal**

There were 2.6 million deaths caused by pneumonia in 2017 [1]. Of these deaths, 808,920 were children under the age of five. This accounts for nearly 15% of all deaths for children under the age of five. The World Health Organization (WHO) estimates that 45,000 of the 808,920 deaths were caused due to household air pollution [4]. By creating a diagnostic that is more efficient, a great number of these deaths could be reduced. The goal of this project is to create and optimize multiple deep learning models that will aide radiologists in making more accurate detections of pneumonia in chest x-rays. These models will also distinguish the source of the pneumonia, either bacterial or viral, allowing doctors to treat their patients more efficiently.

**1.2 Proposed Deep Learning Solution**

In order to carry out this project, I found an image dataset on Kaggle entitled “Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images.” From this dataset, I used the portion that contains the chest x-rays. This portion of the dataset contains 5856 images of chest x-rays. The dataset is then subdivided into testing and training datasets. The training dataset contains 1349 images of chest x-rays of normal (healthy) lungs and 3883 images of chest x-rays of lungs infected with pneumonia. The pneumonia x-rays are labeled as either bacteria or virus, depending on the cause of the pneumonia. The testing dataset contains 234 images of chest x-rays of normal (healthy) lungs and 390 images of chest x-rays of lungs infected with pneumonia [2]. Using the training and testing datasets, I created multiple deep learning models to determine the most efficient and accurate way for radiologist to detect and classify bacterial and viral pneumonia in chest x-rays.

This project relies on supervised deep learning techniques. The main problem for this project is using deep learning to perform image classification. In order to carry out this deep learning image classification, I implemented multiple models consisting of convolutional neural networks (CNN). The multiple CNN contain different types and numbers of hidden layers. By comparing the accuracies and losses of the different models, I was able to determine the most accurate and efficient model for predicting the presence of pneumonia in chest x-rays.

**2 Methodology**

The methodology can be broken down into three main categories: dataset creation and image preprocessing, creation of binary classification models, and creation of categorical classification models.

**2.1 Dataset Creation and Image Preprocessing**

The first task I had to undertake was creating the training and testing datasets and preprocessing the images. Since the dataset I found was already set up for binary image classification, I did not have to alter the training and testing datasets. To be able to perform categorical image classification, I had to create training and testing datasets that were comprised of three categories: bacteria, virus, and normal. For the training dataset, the bacteria dataset consists of 2538 bacterial pneumonia chest x-rays, the virus dataset consists of 1345 viral pneumonia chest x-rays, and the normal dataset consists of 1349 healthy chest x-rays. For the categorical testing dataset, the bacteria dataset consists of 242 bacterial pneumonia chest x-rays, the virus dataset consists of 148 viral pneumonia chest x-rays, and the normal dataset consists of 234 healthy chest x-rays. Now that all the training and testing datasets were created, I had to preprocess and normalize the chest x-rays within the datasets.

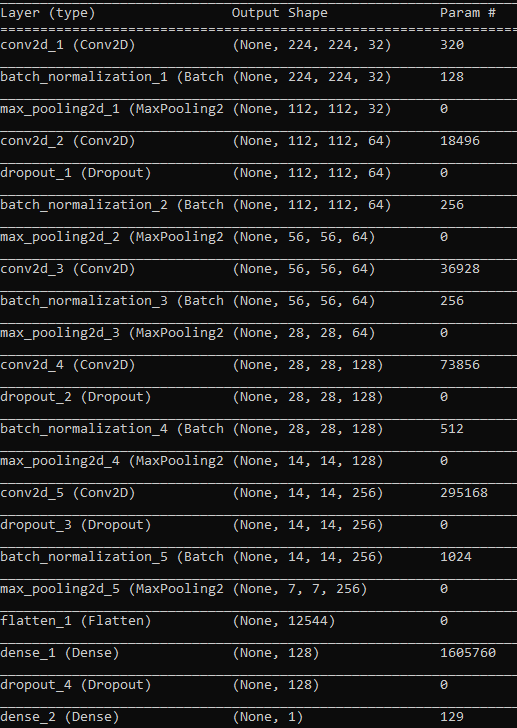
The first set of image preprocessing was generating data from each of the x-ray images. To carry out this task, I used ImageDataGenerator() from the keras preprocessing library. The settings that I used for preprocessing were rescale = 1. /255., rotation\_range = 30, zoom\_range = 0.2, width\_shift\_range=0.1, height\_shift\_range=0.1, and horizontal\_flip = True. These setting ensured that any added to the x-rays was changed to pure black and white. They also allowed for some images to be flipped horizontally so that there would be equal training on the images regardless of the orientation of left and right on the x-ray. I then changed the size of each image to 224 by 224 so that each x-ray would be a constant size throughout training and testing.

Next, it was time to create the actual training and testing image datasets used in the models. The binary image classification datasets were created using flow\_from\_directory() with the following settings: class\_mode = 'binary', color\_mode = "grayscale", target\_size = (224, 224), batch\_size = 32, shuffle = True, and seed = 42. The same settings were used for the categorical image classification models except that class\_mode was changed to ‘categorical.’ The training and testing batches were created and now it was time to create and implement the CNN models.

**2.2 Creation of Binary Classification Models**

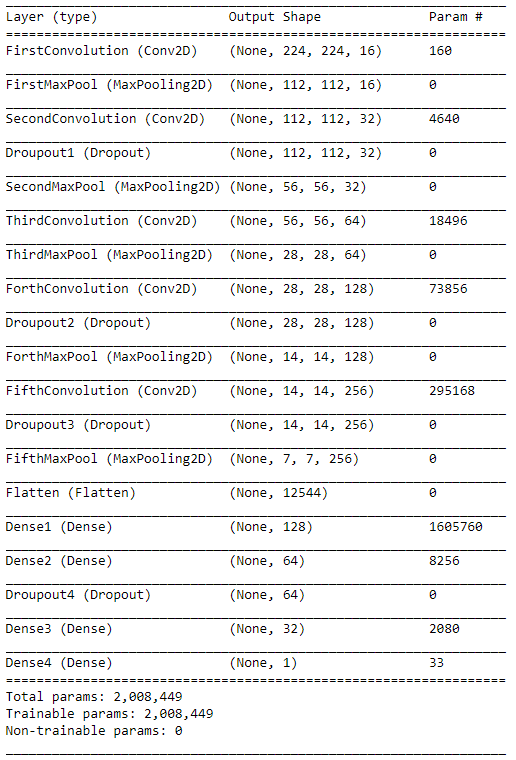
In order to determine how well each model was preforming, I had to create a baseline model that the other models can be compared against. I first created a baseline model for binary classification of the chest x-rays. This model was used to predict the presence or absence of pneumonia in the chest x-rays. A summary of the baseline binary model is shown in Figure 1. The model contains multiple convolutional, max pooling, batch normalization, dropout, and dense layers. Running this model multiple times, I was able to come up with a baseline accuracy of 84.375% and a baseline loss of 0.5709 for binary image classification.

Figure 1: Baseline Binary CNN Model Summary



Now that I had a baseline binary model, I was able to make systematic alterations to the model in order to optimize its accuracy and loss. The first changes I made were to change the batch size and the number of epochs for training and testing. I determined that these changes had little to no positive increases on the accuracy of the models. Next, I decided to remove some of the layers to determine what impact it would have on the model. I removed the batch normalization layers and the model increased in accuracy and decreased in loss. From here, I fine-tuned the model by adding more dense layers and adjusting the batch size and number of epochs to produce an optimized binary classification model (Model #1). Figure 2 shows a summary of Model #1. For training and testing, Model #1 used the same original dataset with a batch size of 32 and 20 epochs. The convolutional layers used a Relu activation function, and the compilation layer used the RMSprop optimizer and binary cross entropy loss function. I was happy with the overall performance of Model #1, but I thought that could increase the accuracy of the model by making a few small changes.

Figure 2: Model #1 Summary Figure 3: Model #2 Summary

Table

Description automatically generated

I first started by changing the order of some of the layers, but this did not produce a noticeable increase in results. Next, I added a fifth dropout layer and began to see an increase in performance. The only other change that I could think of making was to use a different optimizer for the compilation layer of the model. I decide to change the optimizer to Adam, and on my first run of the model, I achieved terrible results. I then realized that I needed to change the loss function to be more compatible with the new optimized model. I changed the loss function to categorical cross entropy, and I achieved my best results yet. After fine-tuning the batch size and number of training and testing epochs, I came up with my best performing binary image classification model (Model #2). An overview of Model #2 can be seen in Figure 3. I placed Figure 2 and Figure 3 side by side so the differences in the models would be easier to observe. For training and testing, Model #2 used the same original dataset with a batch size of 32 and 16 epochs. The convolutional layers used a Relu activation function, and the compilation layer used the Adam optimizer and a sparse categorical cross entropy loss function. I was finally satisfied with my binary classification model and was ready to move on to creating my categorial image classification model.

**2.3 Creation of Categorical Classification Model**

The first step in creating my categorical classification model was changing the program to run using my categorical training and testing datasets that I described in Section 2.1. After I had the program retrieving the correct datasets needed to produce the training and testing datasets for the model, I had to change how the x-rays were labeled. I decided to label the x-rays as either Class 0 for bacterial pneumonia, Class 1 for viral pneumonia, and Class 2 for normal or healthy chest x-rays. Now that the training and testing datasets were created and labeled correctly, it was time to change my model so that it would work with categorial classification. The first change I had to make was to change the output on the last dense layer to 3 so that there would be 3 outputs moving into the compilation layer. Next, I set the compilation optimizer to Adam and the loss function to categorical cross entropy. I ran the model and the training and testing worked, but when it tried to calculate the accuracy and loss of the model, the program ran into an issue. I realized that by changing the labeling on the x-rays to Class 0, Class 1, and Class 2, the integers were not able to be compared to the floats that the testX dataset was producing for its labels. I quickly solved this problem by changing the testX labels to integers and then using those integers for the accuracy calculation. I ran the program again, and to my relief, it ran all the way through. Now I had a baseline categorical classification model that I could use to create my optimized categorical image classification model (Model #3).

Figure 4: Model #3 Summary

Table

Description automatically generated

In order to create Model #3, I went through the same procedures that I used in optimizing both Model #1 and Model #2. I started by changing the batch size and number of epochs. Next, I changed the order and number of layers in the CNN. Lastly, I tried using different optimizers for the compilation layer. For all this testing, I took my best result and used that model as Model #3. An overview of Model #3 is shown in Figure 4. For training and testing, Model #3 used the same categorical dataset with a batch size of 32 and 20 epochs. The convolutional layers used a Relu activation function, and the compilation layer used the Adam optimizer and a categorical cross entropy loss function.

**3 Experimental Results**

I was pleased with the overall success of my models. Once they were optimized, all three of the models were able to achieve a test accuracy of over 90% and a loss under 0.60. Each model contains their own strengths and weakness, and next, I will go through the detailed results for each model.

**3.1 Experimental Results for Model #1**

Model #1 was able to achieve a testing accuracy of 94% and a loss of 0.509. The highest achieved training accuracy for Model #1 was 96% with a loss of only 0.104. Table 1 shows a more detailed breakdown of the performance of Model #1.

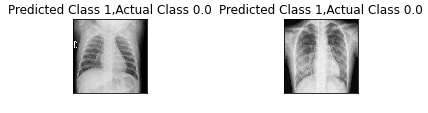
Table 1: Detailed Classification Report for Model #1

Table

Description automatically generated

Table 1 shows that Model #1 was better at classifying the Normal x-rays over the Pneumonia x-rays; this is evident in the F1-scores. Table 1 also show that Model #1 had 100% precision for Class 0, meaning that all of the x-rays classified as Class 0 actually belonged to Class 0. It also shows that Class 1 has 100% recall, which means that the model was able to successfully classify all of the Class 1 x-rays as Class 1. These results suggest that Model #1 was more likely to classify an x-ray as Class 1, even if the actual class for the x-ray was Class 0. An example of this can be seen in Figure 5.

Figure 5: Model #1 Incorrect Classifications



Looking closely at the x-rays in Figure 5, it appears that both x-rays contain a fair bit of background noise that may have masked the presence of pneumonia from the model. This issue may be alleviated by training with a larger dataset or training with a dataset that is slightly more augmented.

**3.2 Experimental Results for Model #2**

Model #2 was able to achieve a testing accuracy of 97% and a loss of 0.165. The highest achieved training accuracy for Model #2 was 95% with a loss of 0.133. Table 2 shows a more detailed breakdown of the performance of Model #2.

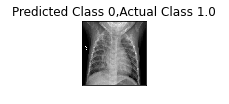
Table 2: Detailed Classification Report for Model #2

Table

Description automatically generated

According to the results in Table 2, Model #2 was slightly better at classifying x-rays that contained images of healthy lungs over those that were infected with pneumonia. Table 2 also shows that Model #2 was able to achieve 100% recall when predicting images as Class 0. This means that the model was able to classify all the x-rays belonging to Class 0 as Class 0, but since the precision was 92%, this means that the model was slightly inclined to classify an image as Class 0 even though the image should belong to Class 1. An example of this is present in Figure 6. Looking back at Table 2, we can see that Model #2 achieved 100% precision for Class 1. This means that all of the x-rays that the model classified as Class 1 actually did belong to Class 1. The lower recall for Class 1 can be explained by the fact that the model was more likely to classify an image as Class 0, as discussed earlier.

Figure 6: Model #2 Incorrect Classification



Looking closely at Figure 6, there appears to be a circular mass at the lower left-hand side of the spinal column. This would indicate pneumonia, especially in adolescents, but upon closer inspection, it appears that the “mass” is just a distortion in the image and is just a portion of the lower rib cage. I believe that using a higher resolution image would have been beneficial in this case. If the model would have had a higher resolution image, I believe that it would have most likely correctly classified the x-ray as Class 1.

**3.3 Experimental Results for Model #3**

Model #3 was able to achieve a testing accuracy of 91% and a loss of 0.572. The highest achieved training accuracy for Model #3 was 90% with a loss of 0.477. Table 3 shows a more detailed breakdown of the performance of Model #3.

Table 3: Detailed Classification Report for Model #3

Table

Description automatically generated

The results in Table 3 show that Model #3 was the best at predicting x-rays containing bacterial pneumonia and struggled predicting x-rays containing viral pneumonia. Starting with Class 0, Model #3 was able to correctly identify all the x-rays that belonged to Class 0. The model as achieved 93% accuracy for Class 0, meaning that 93% of the images that Model #3 assigned to Class 0 actually belonged to Class 0. Moving on the Class 1, Table 3 shows that all the images that Model #3 assigned to Class 1 indeed belonged to Class 1, but the recall of 70% means that the model was only able to find 70% of the x-rays that belonged to Class 1. Examples of these incorrect classifications of Class 1 can be seen in Figure 7. Finally, looking at Class 2, Table 3 shows that Model #3 was able to correctly label all of the Class 2 images as Class 2, but it was only able to achieve a precision of 80% meaning 20% of the images that Model #3 classified as Class 1 did not belong to Class 1.

Figure 7: Model #3 Incorrect Classifications

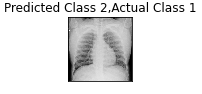
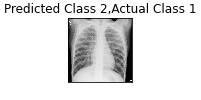


Figure 7 shows the images that Model #3 incorrectly classified. The first two images are clear and show little distortion, meaning that Model #3 was just wrong when identifying these two images. I believe that this incorrect classification can be attributed to the fact that Class 1 had the smallest number of training examples, 1345, compared to the 2538 training images for Class 0. By increasing the number of training images for Class 1, I believe that Model #3 will show an increase in precision when classifying x-rays as Class 1. Looking at the third image in Figure 7, it is smaller and not as clear as the previous images. I believe that these two factors lead to the misclassification of this image, and that I should have removed this image from the testing dataset.

**3.3 Summary of Experimental Results for All Models**

Table 4 shows a brief overview of all three of the models. It also shows the associated training and testing accuracies and losses for each of the models. For the table, it is evident that Adam was the most successful optimizer and that the training and test accuracies for all of the models were 90% or higher.

Table 4: Experimental Results Summary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Classification Type** | **Compilation Optimizer** | **# Epochs** | **Training Accuracy** | **Training Loss** | **Test Accuracy** | **Test Loss** |
| **Model #1** | Binary | RMSprop | 20 | 96% | 0.10 | 94% | 0.51 |
| **Model #2** | Binary | Adam | 16 | 95% | 0.13 | 97% | 0.17 |
| **Model #3** | Categorical | Adam | 20 | 90% | 0.48 | 91% | 0.57 |

**4 Conclusion**

Overall, I feel that I was able to accomplish the goals set forth by this project. I feel that I was able to completely accomplish my first goal of creating a binary image classification model for detecting the presence or absence of pneumonia in a chest x-ray. I was able to create a model that is 97% accurate, and I believe that this model would greatly help radiologist seed up their detection of pneumonia in x-rays. For my second goal of creating a model to predict both the presence or absence of pneumonia in chest x-rays and then classify the pneumonia as either bacterial or viral, I feel that I was somewhat successful. For this classification model, I was able to achieve an accuracy of 91%, but the model does seem to have an issue classifying viral pneumonia. Overall, I am very pleased with the models that I have created and the scores they were able to achieve.

In the future, I would like to keep fine tuning my categorial image classification model and create a new model 4-class categorical classification model that takes pneumonia caused by COVID-19 into account. Aside from fine tuning the parameters of my model, I believe that either finding more datasets containing chest x-rays of viral pneumonia or using image augmentation to create more training images would be beneficial in helping to increase the accuracy of my 3-class categorical model. Once I have increased the accuracy to over 95%, I would like to use this model to create a new 4-class categorical classification model that classifies a chest x-ray as normal, bacterial pneumonia, COVID-19 pneumonia, or viral pneumonia. I believe that a model like this would be very beneficial in the current climate of the world, and it would give doctors a quick way to either diagnosis or rule out COVID-19 infection.

**References**

[1] Dadonaite, B., & Roser, M. (2019). Search. Retrieved February 12, 2021, from <https://ourworldindata.org/search?q=pneumonia>

[2] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), “Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images”, Mendeley Data, V3, doi: 10.17632/rscbjbr9sj.3

[3] Pneumonia. (2020, June 13). Retrieved February 12, 2021, from https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204#:~:text=Pneumonia%20is%20an%20infection%20that,and%20fungi%2C%20can%20cause%20pneumonia.

[4] Pneumonia. (2019, August 2). Retrieved February 12, 2021, from <https://www.who.int/news-room/fact-sheets/detail/pneumonia>

**Acknowledgments**

Richard Han and his article “Pneumonia Diagnosis using Deep Learning” were very helpful in understanding the basic process of image classification using chest x-rays.

Article link: <https://towardsdatascience.com/pneumonia-diagnosis-using-cnns-bfd71e3c05>