



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

tiaplagata / **dsc-phase-4-project**

forked from [learn-co-curriculum/dsc-project-template](#)

Phase 4 Project CNN Model to Detect Pneumonia in Chest X-Rays

View license

☆ 0 stars 49 forks

☆ Star

👁 Watch ▼

Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master ▼



This branch is 7 commits ahead of learn-co-curriculum:master.

Pull request Compare



tiaplagata ...

10 seconds ago



[View code](#)

README.md



Welcome! In this project, I built a Convolutional Neural Network to predict

pneumonia in chest x-rays!

DETECTING PNEUMONIA IN CHEST X-RAYS

Tia Plagata

Navigation

- [Project Overview](#)
- [Preprocessing](#)
- [Model Analysis](#)
- [Future Work](#)

Important Links

- [Link to Data](#)
- [Slideshow Presentation](#)
- [Non-Technical Video Presentation](#)
- [Jupyter Notebook Full Modeling Process](#)

Project Overview

XYZ Healthcare System is looking to hire a data scientist to build a machine learning model to classify their radiology department's chest x-ray images and screen them for pneumonia.

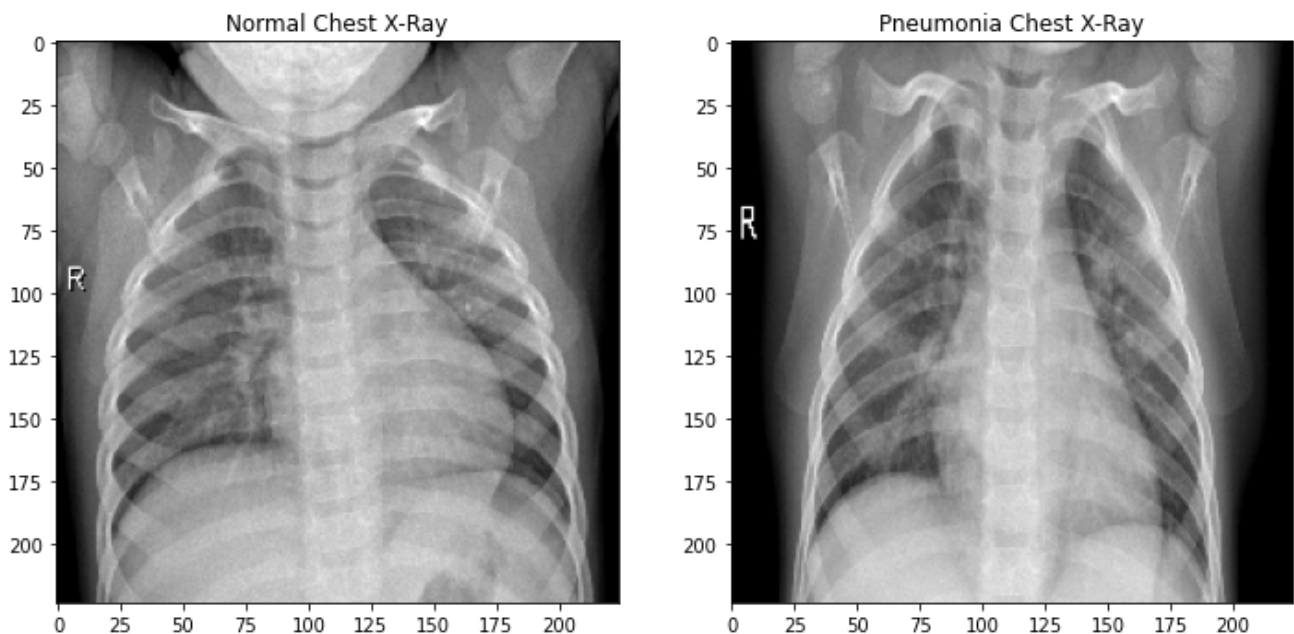
To help XYZ Healthcare System with this problem, I built a Convolutional Neural Network to classify the chest x-ray images as pneumonia or non-pneumonia. This way, XYZ Healthcare System can streamline their radiology department and improve their efficiency.

Scope & Data Used

This project used the Mooney Pneumonia dataset, which can be found on kaggle via [this link](#). This dataset included 5856 unique images of chest x-rays depicting normal or pneumonia x-rays.

Preprocessing

In order to preprocess the images, I used Keras' ImageDataGenerator to rescale the images' pixels (to a scale of 0 to 1), and resized the images to 224 x 224 px. Even though the x-rays seem to be in grayscale, their actual sizes have all 3 layers of a colored photograph (rgb). Thus, I decided to leave them alone instead of converting them to grayscale. Below are two of the preprocessed images.



Model Analysis

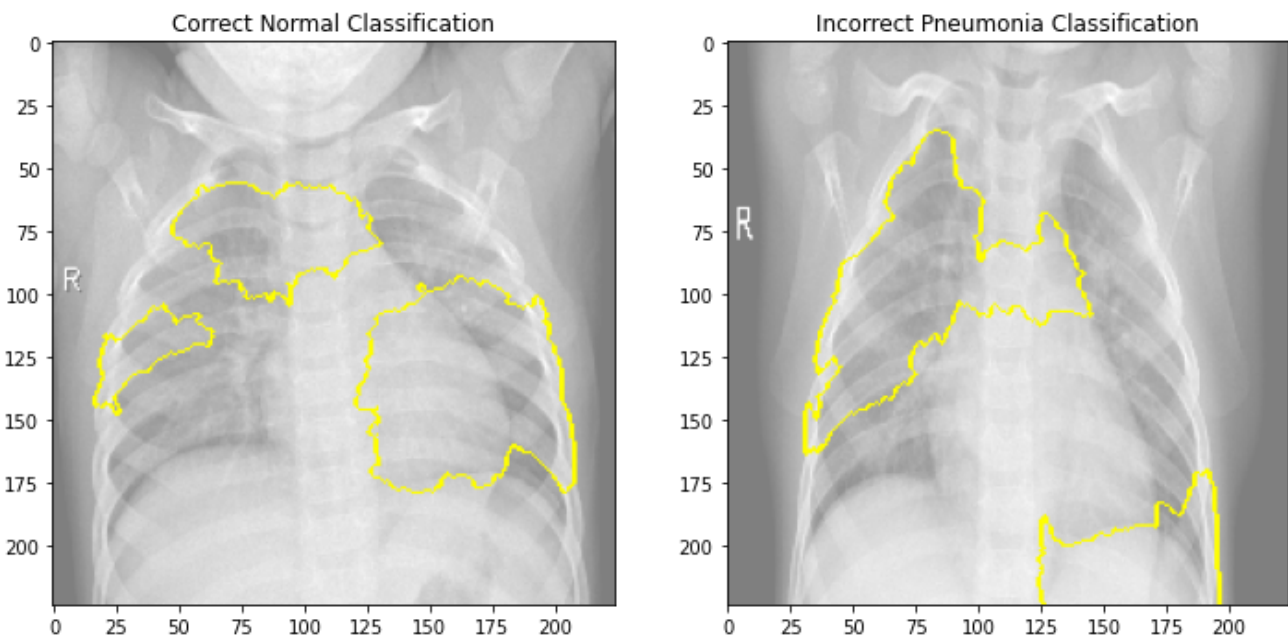
My final model was a Convolutional Neural Network with 82% accuracy. It has 6 blocks which include 10 convolutional layers, 5 max pooling layers, and 3 fully-connected layers.

Metric Used

I used accuracy to score this model because both false negatives and false positives have a cost in this situation. For example, we want to reduce the rate of false negatives because we do not want anyone with pneumonia to go untreated. However, we would not want too many false positives either because we wouldn't want to prescribe steroids or antibiotics to patients that did not need it. In this use-case, even with the model's prediction, the x-ray will be looked over with human eyes before making a final diagnosis. Therefore, we want the overall accuracy of the model to be high to make this process the most efficient for the technician or doctor reviewing the x-ray. The more accurate the model, the less time they need to spend in review.

Feature Importances

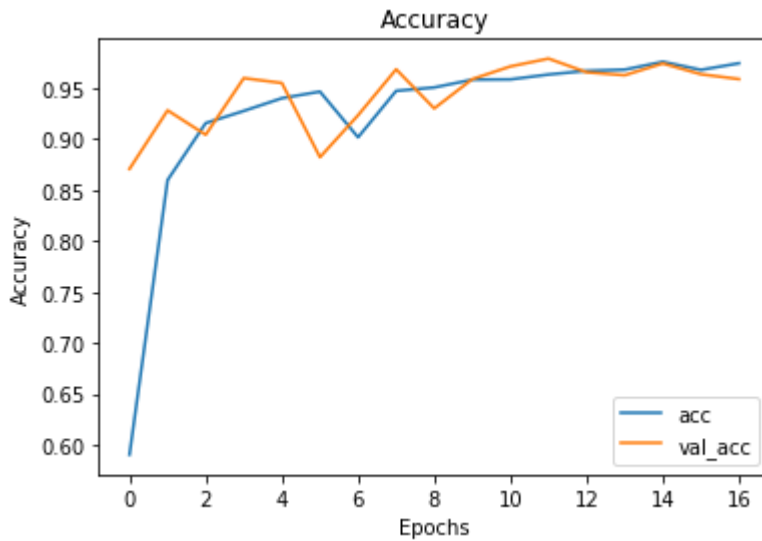
Using the Lime package, I was able to depict the features (edges) in the image that the model found most important in making its prediction. The image on the left was correctly identified as a non-pneumonia. The features that led to this accurate prediction are outlined in yellow. The image on the right was incorrectly identified as non-pneumonia when it was actually pneumonia. The features that led it to this inaccurate prediction are outlined in yellow.



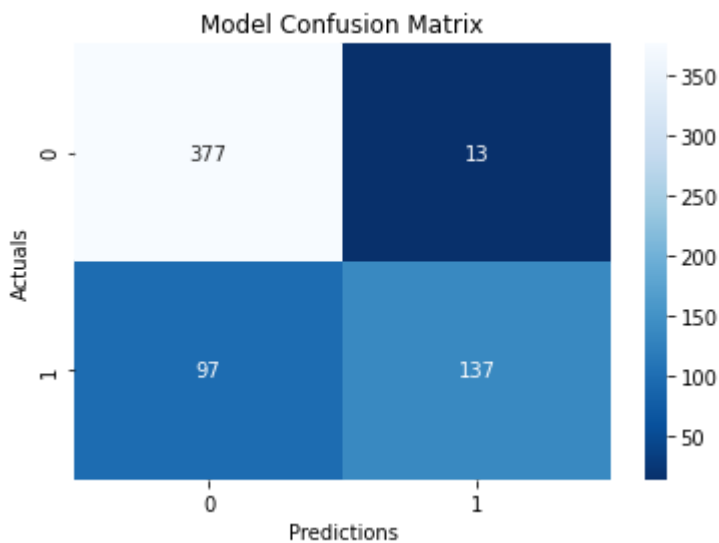
From the image on the right, we can see that the model picked up a piece of the diaphragm as an important feature in making its prediction. Because of this, the high-intensity pixels from the diaphragm may be adding noise to the model.

Model Validation

Looking at the final model's validation and training curves, we can see that the model converges around 97% accuracy.



The final model had an accuracy score of 82% when predicting new data (test set). The confusion matrix from the test set is depicted below. The model is much better at correctly identifying the normal class, and still struggles with correctly identifying the pneumonia class. However, it is overall predicting most of each class correctly.



Model Conclusion and Recommendation

In conclusion, this model is not perfect. It could use some more tuning to increase its accuracy score. However, it will be very useful in improving the efficiency in XYZ Healthcare System's radiology department.

I recommend using this model to provide a preliminary prediction (normal or pneumonia), then having the doctor or x-ray technician review the image to provide a final diagnosis. Even with 82% accuracy, this model will speed up the review process, and free up the doctors' and technicians' time so they can focus on other important tasks.

Future Work

If I had time to explore further, I would do the following techniques to improve this model's performance:

- Try to retrain the model using smaller images (ex. 64x64 px or 32x32 px) to see if it helps eliminate noise
- Look at more preprocessing methods to remove the diaphragm and thus remove noise
- Freeze and unfreeze specific layers of the transfer learning models to fine-tune them
- Do more research on other methods to tune cnns specifically with chest x-ray/pneumonia classification

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● **Jupyter Notebook** 100.0%