

 [anna-dang](#) / [dsc-phase-4-project](#)

forked from [learn-co-curriculum/dsc-phase-4-project](#)

<> Code

 Pull requests

 Actions

 Projects

 Wiki

 Security

 Insights

 main ▾

...

[dsc-phase-4-project](#) / [README.md](#)



**anna-dang** Update README.md

 History



2 contributors



Raw

Blame



86 lines (49 sloc) 3.62 KB

Flatiron Data Science Program

Module 4 Project - Neural Networks

December 15th, 2020

# Computer Vision

*X-Ray Image Classification*



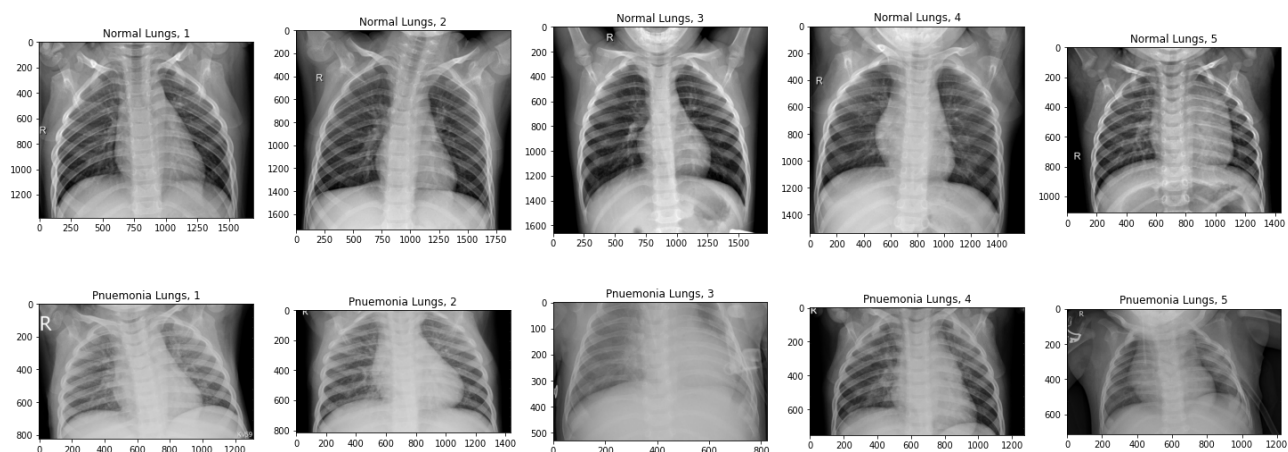
## Overview

Objective: Build a model that can classify whether a patient has pneumonia, given a chest x-ray image.

Repository Contents:

- images : images for display through this analysis
- project\_4\_notebook.ipynb : Google Colab notebook containing full analysis/modeling
- README.md : project summary and contents
- Mod04\_presentation.pdf : presentation slides with comments

## Data



"Chest X-ray images (anterior-posterior) were provided from pediatric patients of Guangzhou Women and Children's Medical Center, Guangzhou, China. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system."

In the above images, some opacity can be observed in the bottom row of pneumonia lungs. This is due to the X-ray picking up the fluid filled air sacs due to the infection.

Sources: [original study source](#), [data download source/Kaggle competition](#)

## Classifier

Full analysis: [Colab Notebook](#)

For this project I wanted to optimize the model to arrive at as few false predictions as possible. False positives could be costly in resources for the provider and patient, or result in unnecessary treatment. False negatives could allow illness to be missed and treatment delayed. Thus I selected **accuracy** and the **AUC** (area under the ROC curve) as target metrics for a balanced model, for both penalize false predictions.

The final model was trained on images with the following preprocessing:

- Images were resized to 124 x 124 pixels, with 3 RGB color channels
- Pixel values were normalized to a 0-1 scale
- To prepare the model to discern noise, four data augmentations were used: rotation, vertical/horizontal shifting, and zoom
- The imbalanced data set (75% pneumonia vs. 25% normal X-rays) was corrected by applying class weights

The final model architecture was a convolutional neural network (CNN) with 3 convolution blocks (convolution, drop out, pool).

Model: "sequential\_15"

| Layer (type)                  | Output Shape          | Param # |
|-------------------------------|-----------------------|---------|
| conv2d_64 (Conv2D)            | (None, 128, 128, 128) | 3584    |
| dropout_8 (Dropout)           | (None, 128, 128, 128) | 0       |
| max_pooling2d_55 (MaxPooling) | (None, 64, 64, 128)   | 0       |
| conv2d_65 (Conv2D)            | (None, 62, 62, 64)    | 73792   |
| dropout_9 (Dropout)           | (None, 62, 62, 64)    | 0       |
| max_pooling2d_56 (MaxPooling) | (None, 31, 31, 64)    | 0       |
| conv2d_66 (Conv2D)            | (None, 29, 29, 32)    | 18464   |
| dropout_10 (Dropout)          | (None, 29, 29, 32)    | 0       |
| max_pooling2d_57 (MaxPooling) | (None, 14, 14, 32)    | 0       |
| flatten_15 (Flatten)          | (None, 6272)          | 0       |
| dense_30 (Dense)              | (None, 32)            | 200736  |
| dense_31 (Dense)              | (None, 1)             | 33      |
| Total params: 296,609         |                       |         |
| Trainable params: 296,609     |                       |         |
| Non-trainable params: 0       |                       |         |

Resulting in performances scores of:

- Accuracy - 89.77%
- Recall - 94 %
- Precision - 82 %
- AUC - 0.88

---

### Classification Report for Test Data:

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0.0          | 0.90      | 0.82   | 0.86     | 176     |
| 1.0          | 0.90      | 0.94   | 0.92     | 293     |
| accuracy     |           |        | 0.90     | 469     |
| macro avg    | 0.90      | 0.88   | 0.89     | 469     |
| weighted avg | 0.90      | 0.90   | 0.90     | 469     |

---

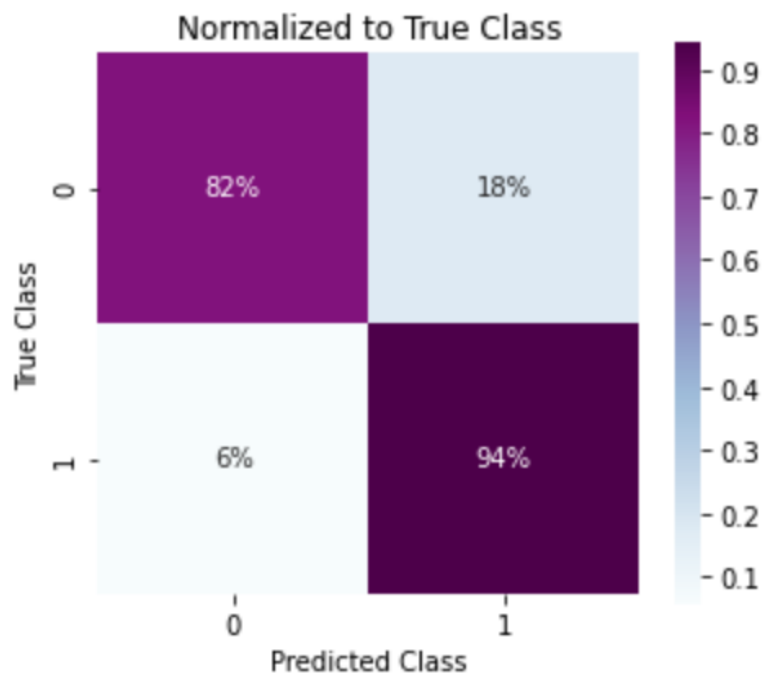
15/15 [=====] - 0s 7ms/step - loss:  
 Loss of the model is - 0.2689375579357147  
 15/15 [=====] - 0s 7ms/step - loss:  
 Accuracy of the model is - 89.76545929908752 %

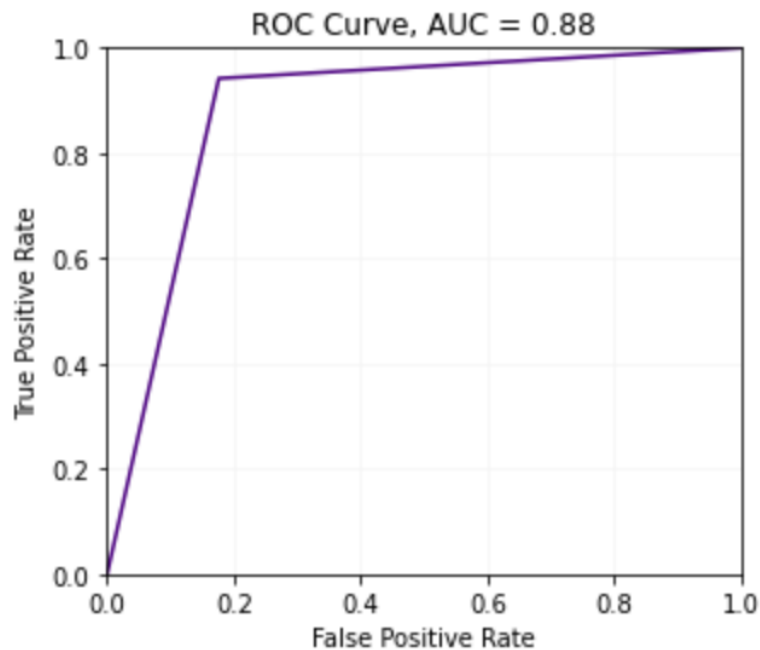
---

Correct: 421, 89.77%  
 Incorrect: 48, 10.23%

---

a:





## Reccomendations

- Continue collecting labeled images to progressively train the model.
- Store image data at 128 x 128 to conserve storage memory (this is up to a 10% reduction in original image size).
- Use the model to improve efficiency of Xray review.

## Future Work

This is a supervised learning task and thus performance is based on the quality of the dataset used.

- Collect more labeled images or continue data augmentation to increase the quantity of images in the training set.
- Try transfer learning - use an established x-ray classifier and build model on top of that.
- Progressively resize the model input image size to find the smallest possible input size without sacrificing performance.

**Thank you for viewing my project!**

Please review the full analysis in the [Colab Notebook](#) or view my presentation [slideshow](#).