COMPUTER VISION

Presented by Anna D'Angela | December 15th, 2020

Phase 4 Project - Neural Networks

BUSINESS CASE

Even for a trained radiologist, it is challenging to examine chest X-rays for markers of pneumonia. There is a need to improve the diagnostic accuracy.

A medical imaging company is looking to increase efficiency for its technicians by building a classifier to aid them in detecting pneumonia from chest X-ray images.

Competition Data Hosted by Kaggle
Project for Flatiron School, Online Data Science Program

X-RAY CLASSIFICATION OF PEDIATRIC PNEUMONIA

"Pediatric pneumonia is responsible for the deaths of more than **800,000** young children worldwide each year."

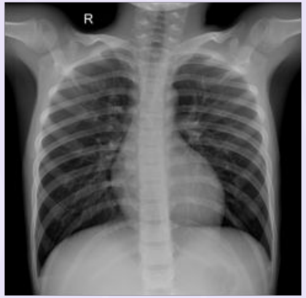
—United Nations Children's Fund (UNICEF)

EXAMPLE X-RAYS

Normal Lungs

Normal Lungs

Normal Lungs



Normal Lungs

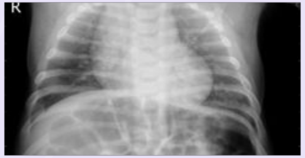
Normal Lungs

Pnuemonia Lungs

Pnuemonia Lungs

Pnuemonia Lungs

Pnuemonia Lungs



Pnuemonia Lungs

CLINICAL EFFICIENCY

PATIENT VISIT

Physical exam and chest x-ray if symptoms present

RAPID ANALYSIS

Radiologist is assisted in diagnosis by the model

TREATMENT

A swift start to treatment leads to faster recovery

OBJECTIVE

Create an X-ray image classification tool to assist in diagnosing cases of pediatric pneumonia

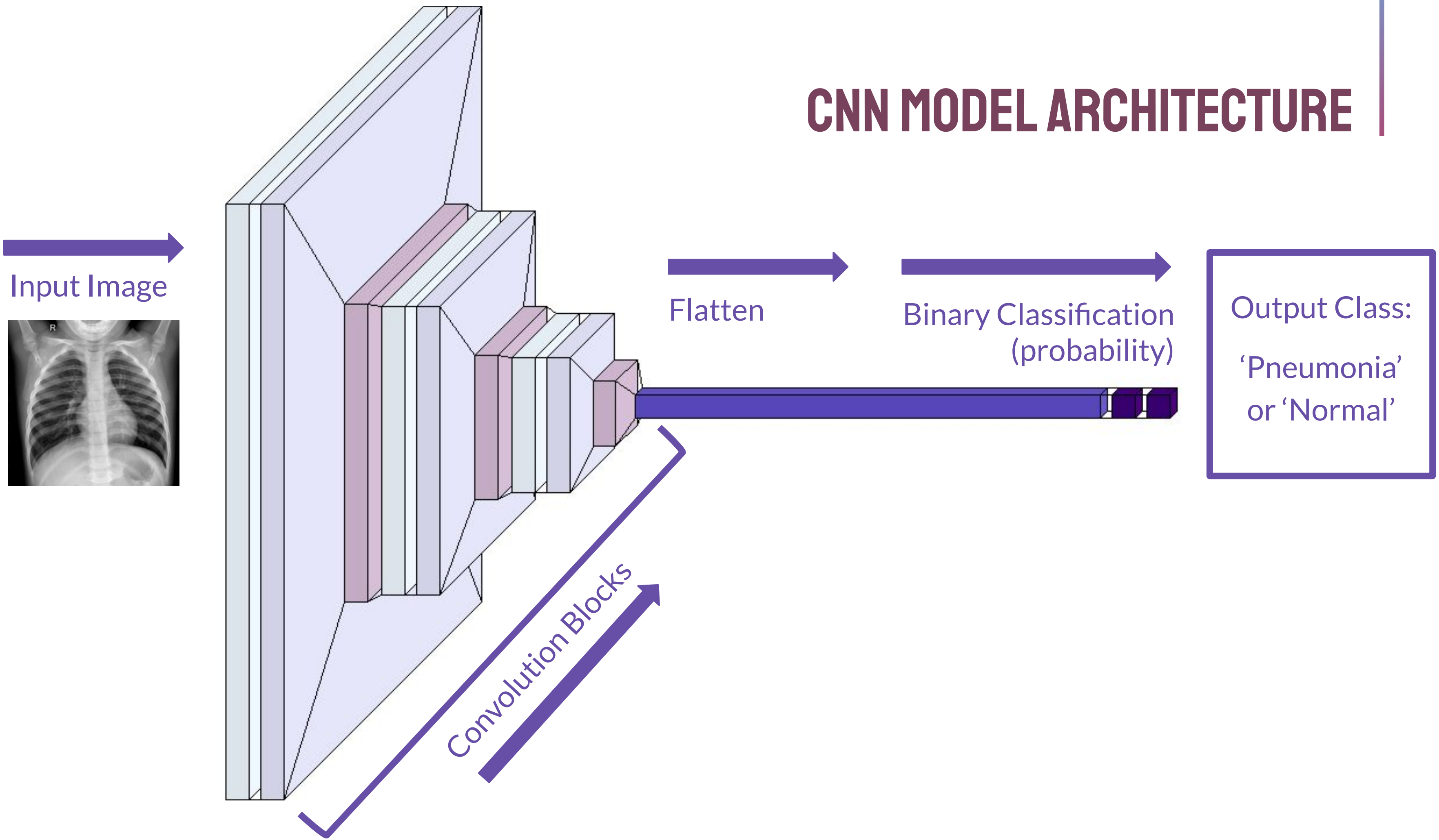
METHOD

Use a convolutional neural network (CNN) to find patterns in X-rays to predict class label

SUCCESS CRITERIA

Optimize for accuracy to limit false predictions

CNN MODEL ARCHITECTURE



89.77%

MODEL DIAGNOSTIC ACCURACY

MODEL PERFORMANCE

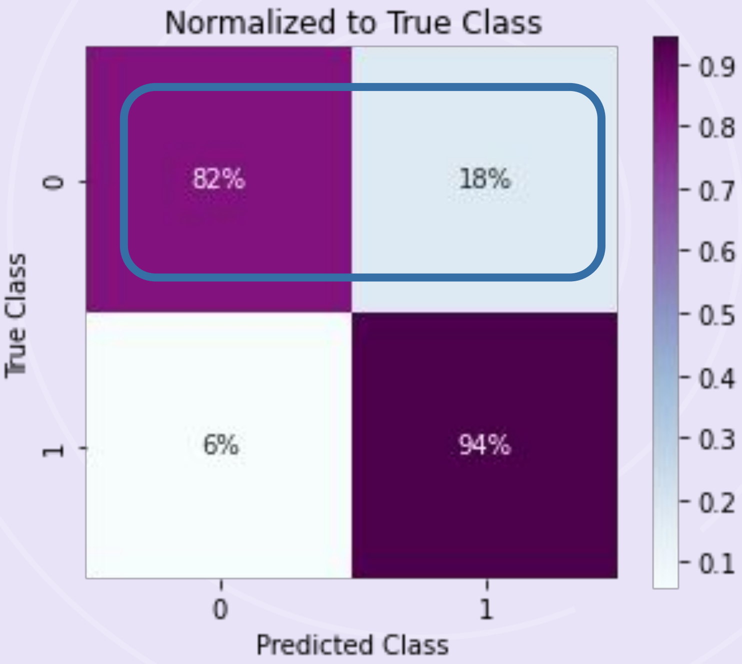
90% Accuracy

94% Recall

82% Precision

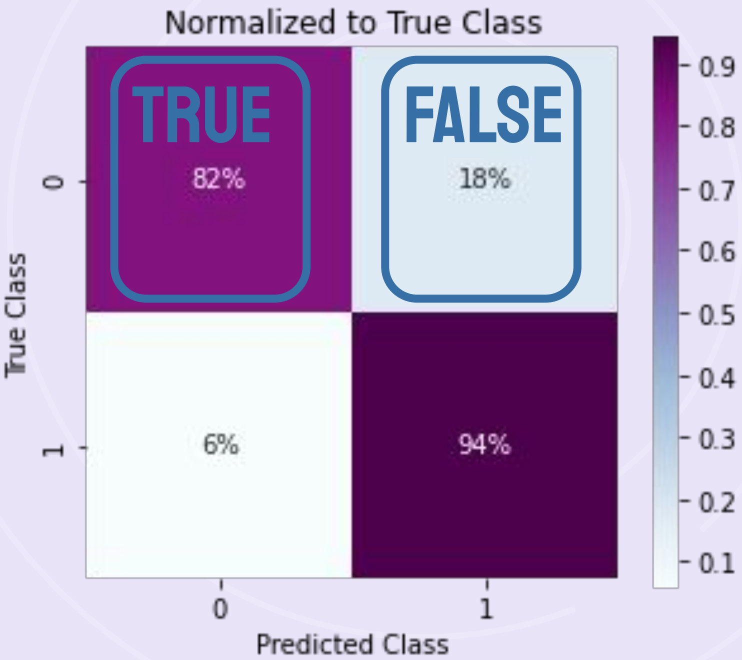
0.90 F1 Score

PREDICTIONS



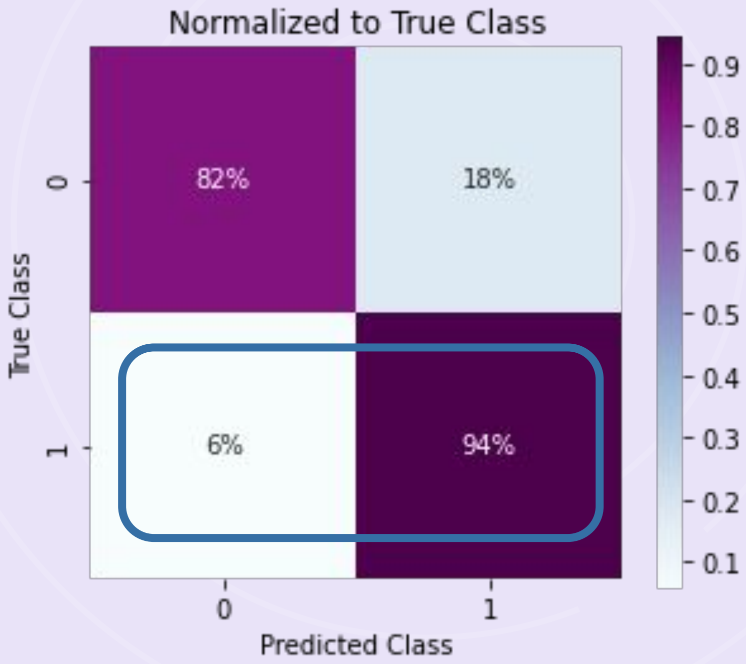
'NORMAL' / NEGATIVE CASES

PREDICTIONS



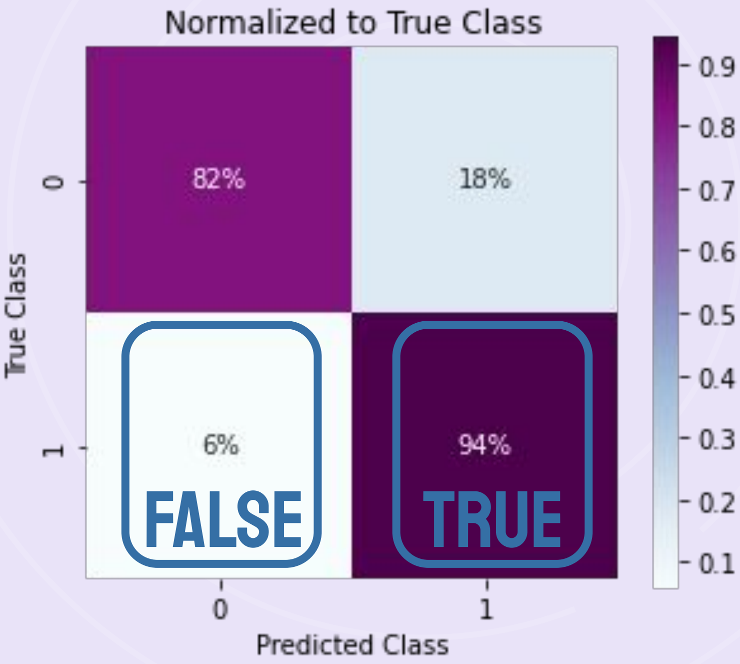
'NORMAL' / NEGATIVE CASES

PREDICTIONS



‘PNEUMONIA’ / POSITIVE CASES

PREDICTIONS



'PNEUMONIA' / POSITIVE CASES

USER RECOMMENDATIONS

- **Model use:** Use the model as a pre-screening tool to improve efficiency of X-ray review, but do not replace human classification.
- 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).

FUTURE WORK

MODEL TUNING / IMPROVEMENTS

- Increase the quantity of images in the training set.
- Utilize transfer learning to improve the base model.
- **User experience:** Build an application to receive an X-ray as input and output a prediction.

THANK YOU!

Please find my full analysis on
GitHub: @anna-dang

Anna D'Angela | Detroit, MI

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

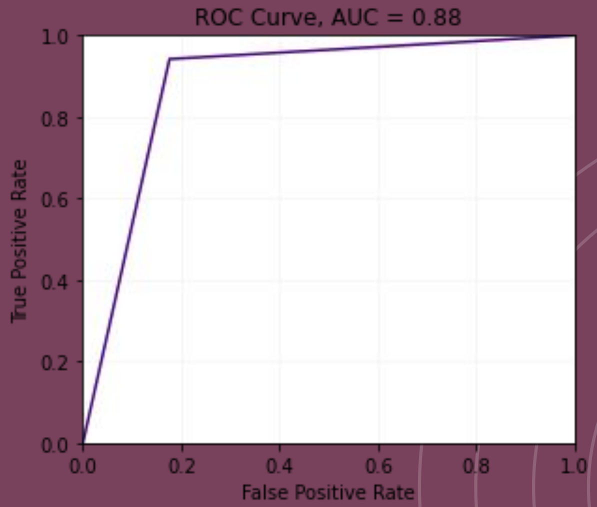
APPENDIX (i)

CLASSIFICATION REPORT

```
----------------------------------------------------------------
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%
----------------------------------------------------------------
```

ROC CURVE / AUC

A perfect binary classifier has an 'Area Under Curve' (AUC) of 1, with ROC curve hugging the top left corner



MODEL STRUCTURE

APPENDIX (ii)

IMAGE PREPROCESSING

- Images were resized to 124 x 124 pixels, with 3 RGB color channels

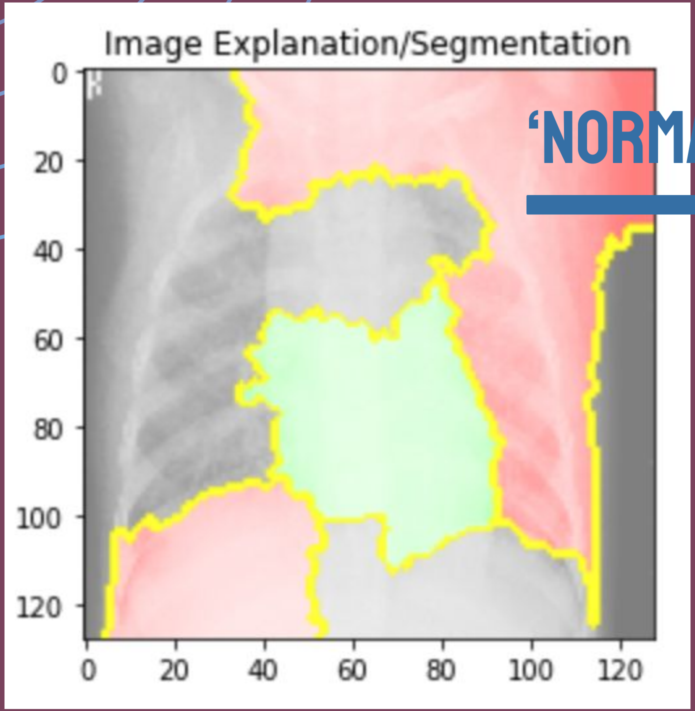
- Pixel values were normalized to a 0-1 scale

- To prepare the model to discern noise, four data augmentations were used: rotation, vertical/horizontal shifting, and zoom

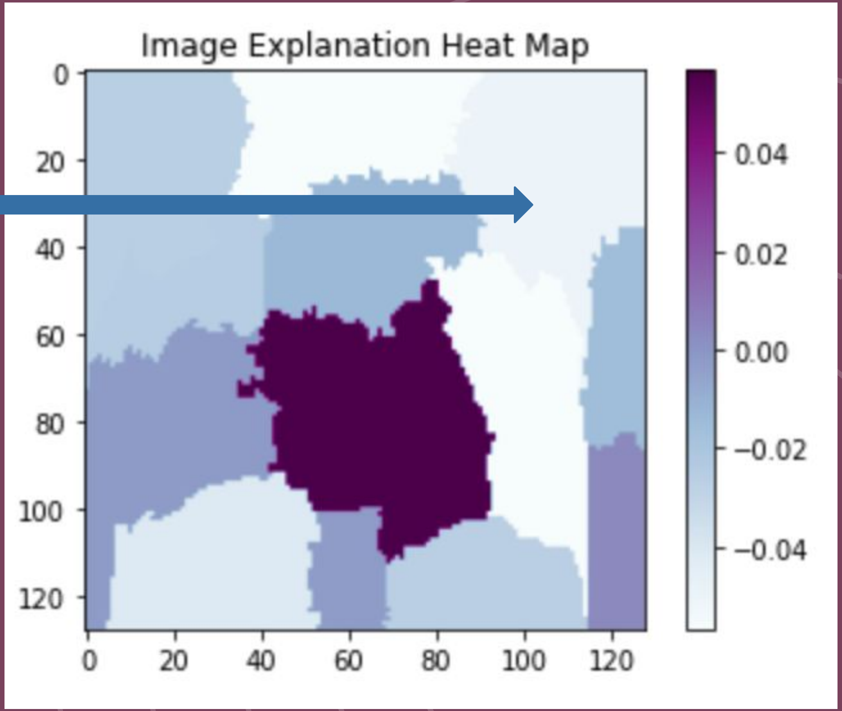
- The imbalanced data set (75% pneumonia vs. 25% normal X-rays) was corrected by applying class weights

```
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
_________________________________________________________________
```

APPENDIX (iii)
IMAGE EXPLAINER (LIME)



‘NORMAL’



APPENDIX (iii)
IMAGE EXPLAINER (LIME)