Capstone Project–Interim Report
Pneumonia Detection

1. Introduction

The problem consists of binary classification of chest X-rays on three different classes of lung opacities such as opacity, no opacity and not normal. The major issue is dissimilarity in quality X-rays in terms of brightness, resolution and position region of interest. To model such task, we describe our algorithm that can detect the visual signal for pneumonia in medical chest radiographs, and output either pneumonia positive or negative, and if positive it also returns predicted bounding boxes around lung opacities.

To be more specific, we are building a Deep Learning model that can detect bounding boxes around the region where the pneumonia related symptoms appear in CXR (Chest X-Ray). It is an Object Detection problem.

The dataset consists of CSV labelled data and chest radiograph (CXR) images. The CSV has patient id's with XY coordinates of center of bounding box along with height and width of box. The CSV file also contain class label/target variable whether the patient has pneumonia or not.

2. Exploratory Data Analysis

Note: After having run some simple codes in our system (which is not GPU enabled, and also Kaggle GPU seems more appropriate) we found out that the data given to us as a Google Drive file [https://drive.google.com/drive/folders/1GYAe8hZB8Si5YSW0akNXBpsELRicE4Hp] is EXACTLY the same as that is here: [https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge].

[https://github.com/rahultheogre/Capstone-RSNA-Pneumonia-Detection/blob/main/rsna-pneumonia-detection-eda.ipynb]

SUMMARY

- RSNA - CXR Dataset contains 30227 X-ray images in DICOM format.
- There are three classes with 31.61% lung opacity, 39.11% -no lung opacity, 29.28% normal images.
- In the target class, there are 31.61% of pneumonia class, 68.38% of non-pneumonia images.
- Bounding boxes for patients having pneumonia are defined in the train labels file. There are 9555 positive patients in this file. Each X-ray image has metadata associated with it. It gives information about the patient, the view position etc. 3543 duplicate entries suggest presence of different X-ray views for the same patient.
- Number of images in train set is: 26684
  Number of patients in csv file as per their Id is: 26684
  Number of images in test set is: 3000

TRAIN LABELS FILE

1) Each record in the train_labels table contains
   1. patientId - A patientId. Each patientId corresponds to a unique image.

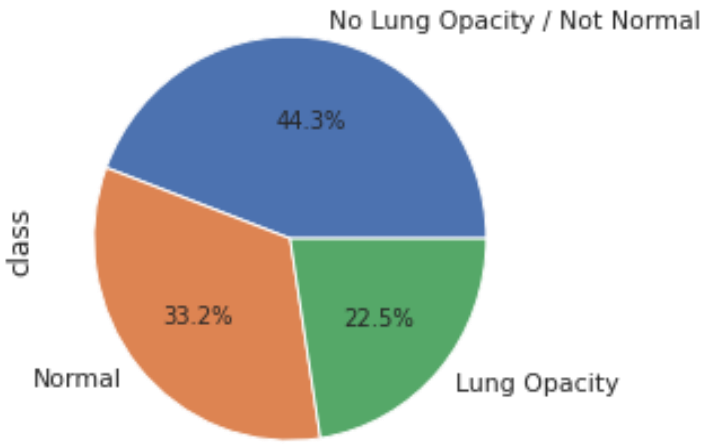
2. x - the upper-left x coordinate of the bounding box.
3. y - the upper-left y coordinate of the bounding box.
4. width - the width of the bounding box.
5. height - the height of the bounding box.
6. Target - the binary Target, indicating whether this sample has evidence of pneumonia. (Either 0 or 1 for absence or presence of pneumonia, respectively)

2) There are no duplicate records.
3) There are many NaN values in four columns. But they seem to follow a trend. For each record in which we have any of value in the tuple (x, y, width, height) as NaN, the other 3 will also be NaN.
4) This seems plausible. x and y are values of a tuple. Together they stand for a location, and width and height also form a pair. All four, together, define the bound of the abnormality in on the lung. All will be NaN together, or none will. If the values in the tuple (x, y, width, height) is NaN, then the value in the column 'Target' is definitely 0, that is, the patient is not pneumonic. 5) Target, as per data dictionary, stands for whether the patient is pneumonic or not. And in this study, we are looking for the same through an analysis of images.
5) The number of unique Target values when the feature has value NaN turn out to be 1, which imply all patients for whom the record shows absence of 'abnormality', will be non-pneumonic. So, the missing values in the dataset have a reason. There are empty non-existent values conditioned on Target being 1 or zero. In other words, the Target column is dependent on the tuple of these four features. We don't need it. We will remove it.

TRAIN CLASS TABLE

1) Number of Duplicated records in train_class file: 3543
2) Each record in the train_class table contains: patientId, and class.
3) There are no missing values.
4) The number of records in the two tables: train_class and train_labels is same: 30227
5) Number of unique patientId values in train_class: 26684. Number of unique patientId values in train_label: 26684.
6) Number of patientIds are same in both the train_labels dataset and train_class. So data is consistent across the two excel files.

DISTRIBUTION OF CLASSES



Meaning of classes:

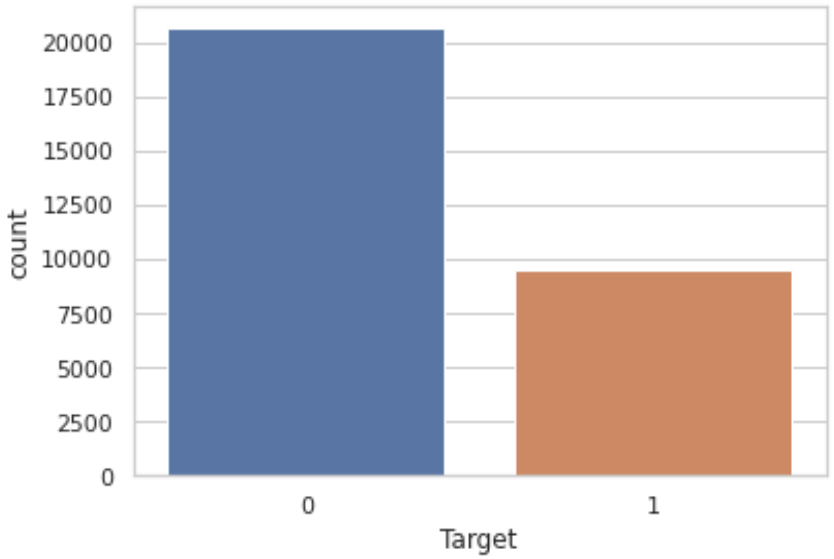
- 'Normal' indicates that the lung x-ray is normal, and so is the lung.

- 'Lung Opacity' confirms that the the lung has opacity and is indicative of pneumonia.
- 'No Lung Opacity / Not Normal' indicates that while pneumonia was determined not to be present, there was nonetheless some type of abnormality on the image and oftentimes this finding may mimic the appearance of true pneumonia.

The third label presents the primary challenge. These images have some abnormality, but they are not pneumonic. Our machine learning algorithm should be able to read this into the images, and not get fooled. In other words, it should be able to correctly classify the abnormal images into 'with pneumonia' and 'without pneumonia'.

As the pie chart indicates, there's a relatively uniform split between the three classes, with nearly 2/3rd of the data comprising of no pneumonia (either completely normal or no lung opacity / not normal). Compared to most medical imaging datasets, where the prevalence of disease is quite low, this dataset has been significantly enriched with pathology.

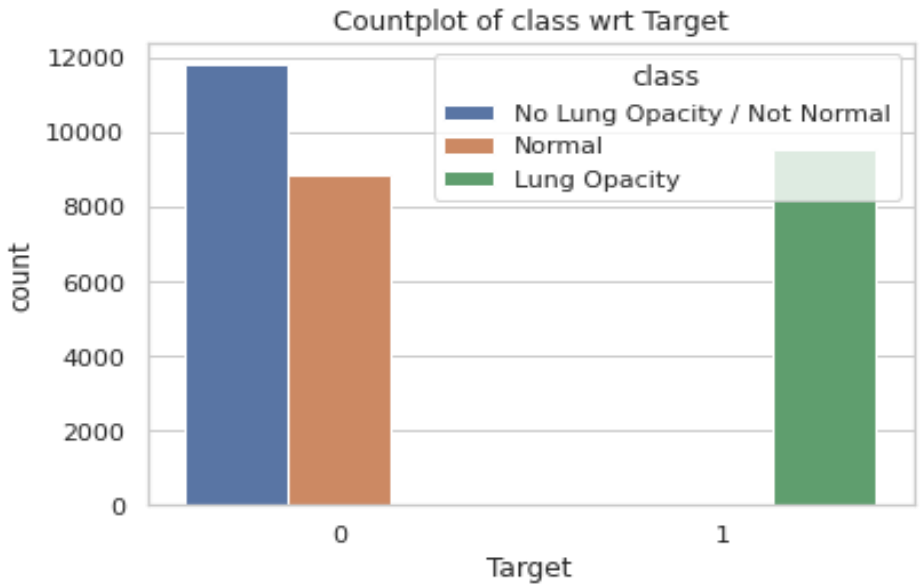
DISTRIBUTION OF TARGET VALUES



There are 9555 records with Target = 1 and same number of records with class = Lung Opacity. We don't need to analyze the data with code to know that they correspond to the same patientId. If the x-ray of a patient has region(s) of lung opacity, then they are pneumonic.

Now we merge the two DataFrames we have into one: train_meta. And check its info and a few random samples to assert everything is alright. A visual inspection confirms our ideas about 'Target' and 'class' and their relationship. But still, to be on the safe side, we group the data according to Target and class, and see if our ‘feelings’ were right with a count plot.

RELATIONSHIP OF TARGET AND CLASS FEATURE



There is no incongruence. If class = Lung Opacity, then Target = 1. And if Target = 1, class is necessarily Lung Opacity. In other words, all the 'abnormalities' with class = No Lung Opacity/Not Normal come under the category of Target=0.

DICOM IMAGES

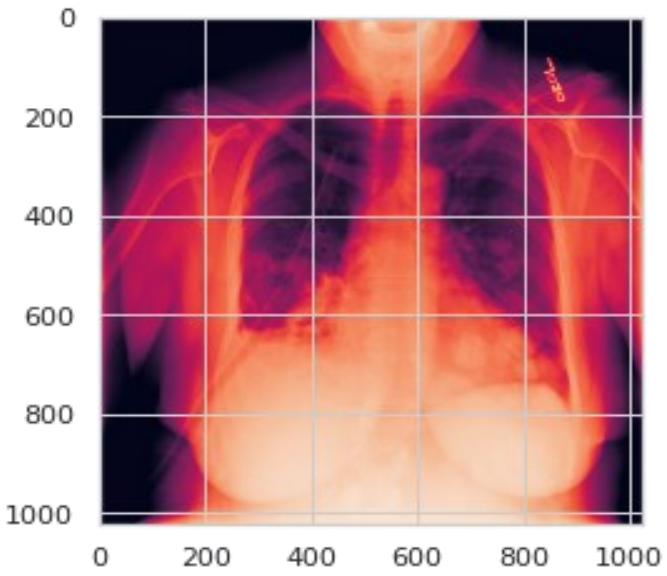
All the CXR are given in DICOM format. DICOM stands for Digital Imaging and Communications in Medicine. It is a standard file for communicating and managing medical imaging information and related data. The meta information saved in a sample DICOM file (of patientId: 0004cfab-14fd-4e49-80ba-63a80b6bddd6) is as follows:

```
(0002, 0000) File Meta Information Group Length  UL: 202
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: Secondary Capture Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 1.2.276.0.7230010.3.1.4.8323329.28530.1517874485.775526
(0002, 0010) Transfer Syntax UID                 UI: JPEG Baseline (Process 1)
(0002, 0012) Implementation Class UID            UI: 1.2.276.0.7230010.3.0.3.6.0
(0002, 0013) Implementation Version Name         SH: 'OFFIS_DCMTK_360'
-------------------------------------------------
(0008, 0005) Specific Character Set              CS: 'ISO_IR 100'
(0008, 0016) SOP Class UID                       UI: Secondary Capture Image Storage
(0008, 0018) SOP Instance UID                    UI: 1.2.276.0.7230010.3.1.4.8323329.28530.1517874485.775526
(0008, 0020) Study Date                          DA: '19010101'
(0008, 0030) Study Time                          TM: '000000.00'
(0008, 0050) Accession Number                    SH: ''
(0008, 0060) Modality                            CS: 'CR'
(0008, 0064) Conversion Type                     CS: 'WSD'
(0008, 0090) Referring Physician's Name          PN: ''
(0008, 103e) Series Description                  LO: 'view: PA'
(0010, 0010) Patient's Name                      PN: '0004cfab-14fd-4e49-80ba-63a80b6bddd6'
(0010, 0020) Patient ID                          LO: '0004cfab-14fd-4e49-80ba-63a80b6bddd6'
(0010, 0030) Patient's Birth Date                DA: ''
(0010, 0040) Patient's Sex                       CS: 'F'
(0010, 1010) Patient's Age                       AS: '51'
(0018, 0015) Body Part Examined                  CS: 'CHEST'
(0018, 5101) View Position                       CS: 'PA'
(0020, 000d) Study Instance UID                  UI: 1.2.276.0.7230010.3.1.2.8323329.28530.1517874485.775525
(0020, 000e) Series Instance UID                 UI: 1.2.276.0.7230010.3.1.3.8323329.28530.1517874485.775524
(0020, 0010) Study ID                            SH: ''
(0020, 0011) Series Number                       IS: '1'
(0020, 0013) Instance Number                     IS: '1'
(0020, 0020) Patient Orientation                 CS: ''
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
```

(0028, 0010) Rows US: 1024
(0028, 0011) Columns US: 1024
(0028, 0030) Pixel Spacing DS: [0.14300000000000002, 0.14300000000000002]
(0028, 0100) Bits Allocated US: 8
(0028, 0101) Bits Stored US: 8
(0028, 0102) High Bit US: 7
(0028, 0103) Pixel Representation US: 0
(0028, 2110) Lossy Image Compression CS: '01'
(0028, 2114) Lossy Image Compression Method CS: 'ISO_10918_1'
(7fe0, 0010) Pixel Data OB: Array of 142006 elements

VISUAL OF THE IMAGE

(patientId: 0004cfab-14fd-4e49-80ba-63a80b6bddd6)



Inferences from DICOM file:

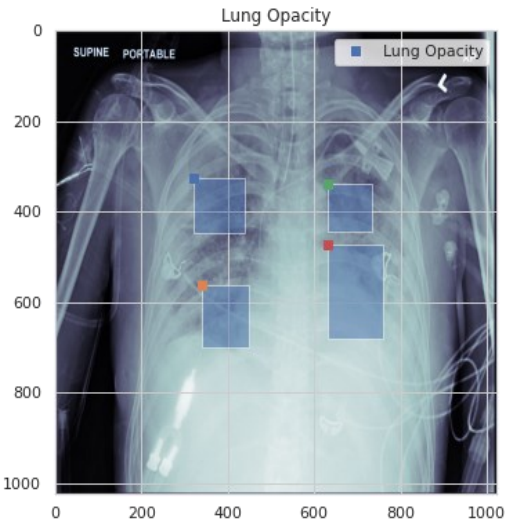
1) Most of the standard headers containing patient identifiable information have been anonymized (removed) so we are left with a relatively sparse set of metadata. The primary field we will be accessing is the underlying pixel data.
2) We don't have the patient's name which is the same as their id. Also, there is no physician name and study id, nor is there the DOB of the patient.
3) There are 1024 rows and columns in the image- which is the standard size of all the images.
4) It is monochrome, i.e. grayscale.
5) Every image takes up 8 bits of data.

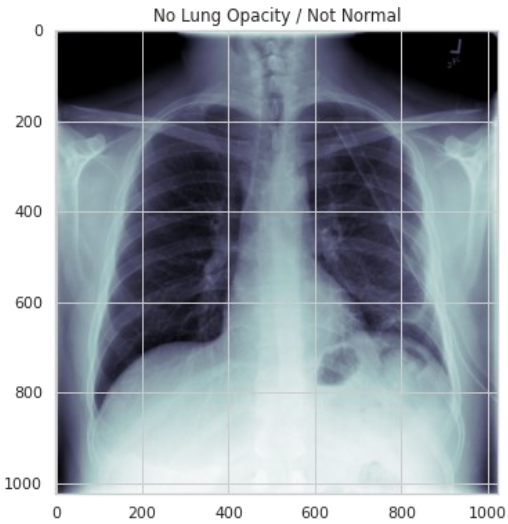
BOUNDING BOXES

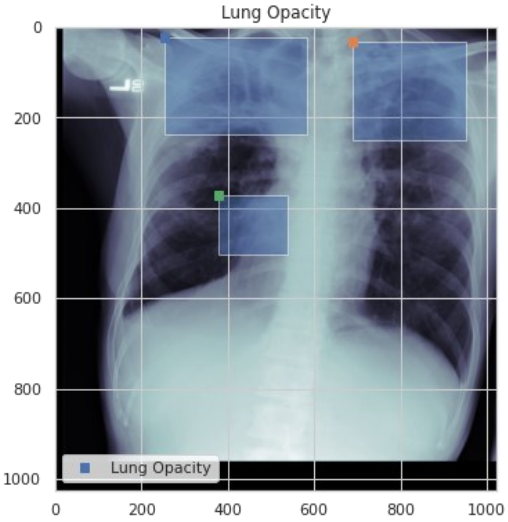
Total number of bounding boxes: 30227
Distribution of number of boxes in the lungs of each patient:

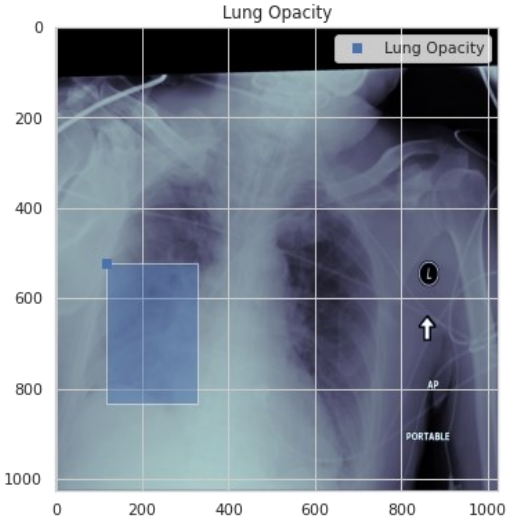
	boxes	patients
0	1	23286
1	2	3266
2	3	119
3	4	13

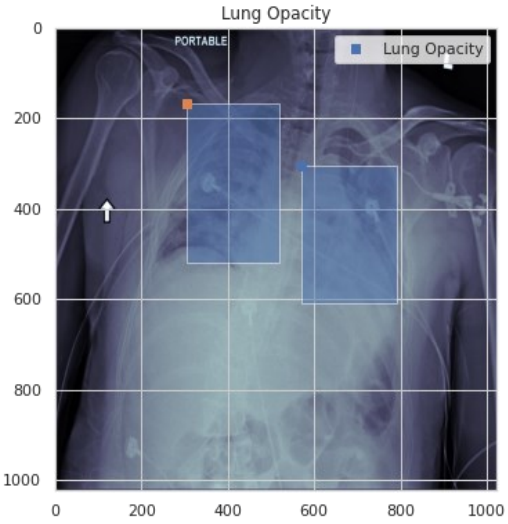
Visualizing Images with bounding boxes¶

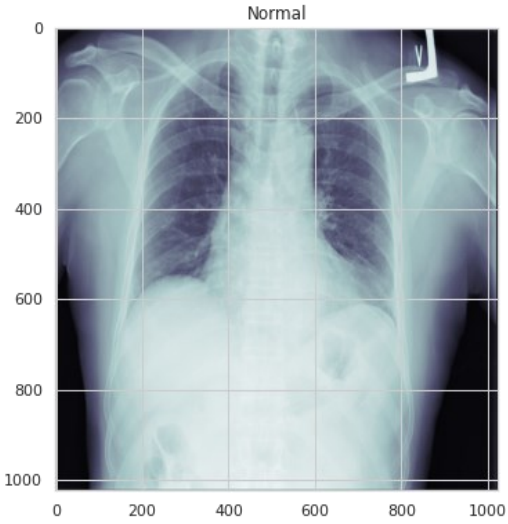












3. Base Model and Architecture

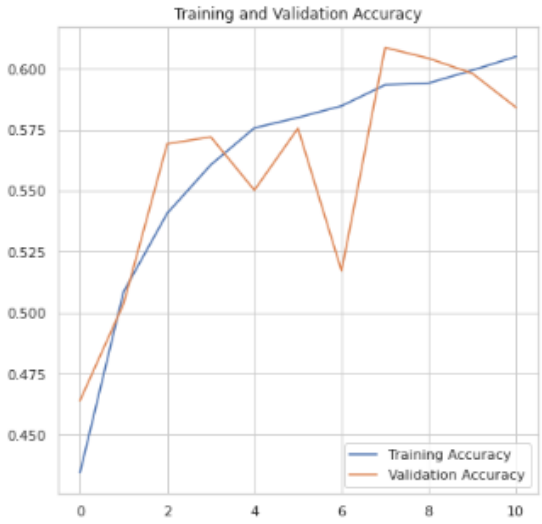
We performed a basic CNN classification on our dataset.
[https://github.com/rahultheogre/Capstone-RSNA-Pneumonia-Detection/blob/main/pneumonia-detection-with-cnn-basic.ipynb]
The three classes were the classes provided to us. The architecture was:

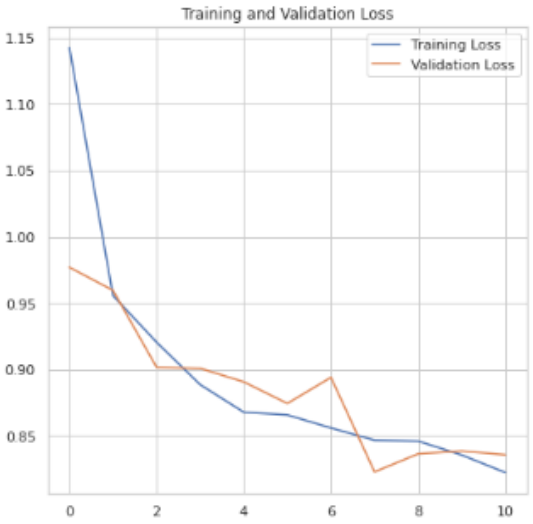
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_20 (Conv2D)           (None, 128, 128, 32)      896
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 128, 128, 32)      9248
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
dropout_11 (Dropout)         (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 64, 64, 64)        18496
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 64, 64, 64)        36928
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 32, 32, 64)        0
_________________________________________________________________
dropout_12 (Dropout)         (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 32, 32, 128)       73856
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 32, 32, 128)       147584
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 16, 16, 128)       0
_________________________________________________________________
dropout_13 (Dropout)         (None, 16, 16, 128)       0
_________________________________________________________________
global_max_pooling2d_2 (Glob (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 256)               33024
_________________________________________________________________
dropout_14 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_5 (Dense)              (None, 3)                 771
=================================================================
Total params: 320,803
Trainable params: 320,803
Non-trainable params: 0
_________________________________________________________________
```

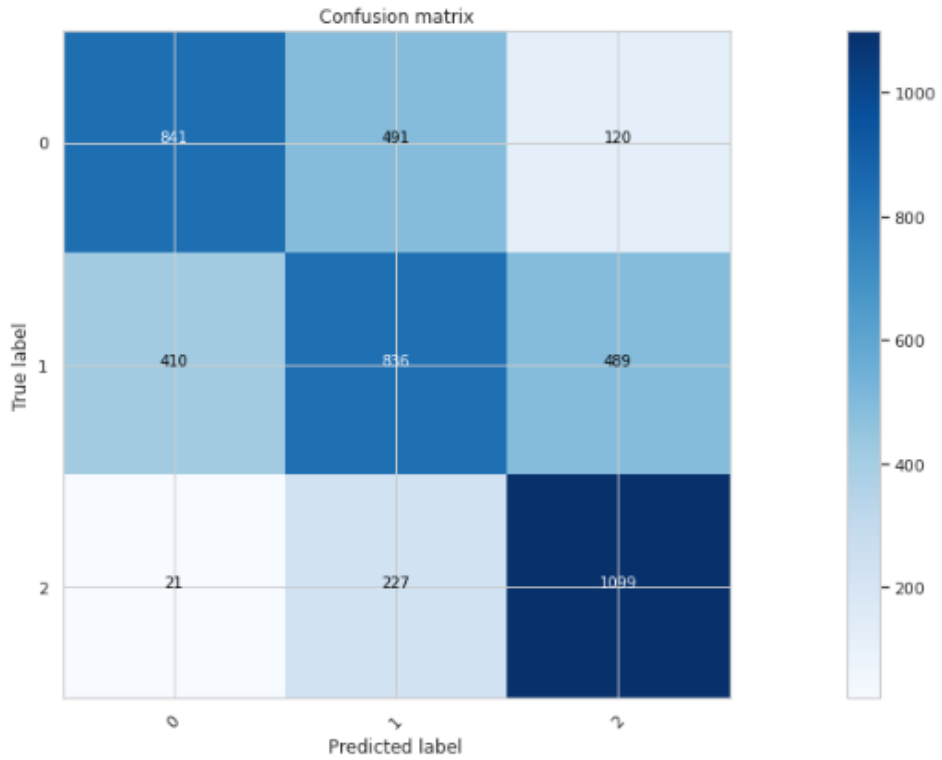
```
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

callbacks = [
    ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=4),
    EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
]
```

- BATCH SIZE = 16
- EPOCHS = 20

4. Early Results







	Method	accuracy	Test Score	1_precision	1_recall	1_f1-score	1_support
0	CNN	0.604925	0.612263	0.537967	0.481844	0.508361	1735

Fine tuning may have helped reduce the overfitting, but the accuracy is not worthy of attention. We do need to shift to transfer learning, and use pretrained weights of popular architectures. We think we will try our hand at VGG16, Inceptionv3 and ResNet50. We will also try to look at the problem from a segmentation perspective- something beyond the scope of the current curriculum and try to use Masked RCNN on the dataset.

5. References

1. Chartrand G, Cheng PM, Vorontsov E, et al. Deep learning: a primer for radiologists. *RadioGraphics* 2017; 37:2113–2131 [Crossref] [Medline] [Google Scholar]
2. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vis* 2015; 115:211–252 [Crossref] [Google Scholar]
3. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in neural information processing systems 25 (NIPS 2012).* San Diego, CA: Neural Information Processing Systems Foundation, 2012 [Google Scholar]
4. Prevedello LM, Halabi SS, Shih G, et al. Challenges related to artificial intelligence research in medical imaging and the importance of image analysis competitions. *Radiol Artific Intell* 2019; 1:e180031 [Crossref] [Google Scholar]
5. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *IEEE Proceedings: 30th IEEE Conference on Computer Vision and Pattern Recognition.* Los Alamitos, CA: IEEE, 2017:3462–3471 [Google Scholar]
6. Shih G, Wu CC, Halabi SS, et al. Augmenting the National Institutes of Health chest radiograph dataset with expert annotations of possible pneumonia. *Radiol Artif Intell* 2019; 1:e180041 [Crossref] [Google Scholar]

7. RSNA Pneumonia Detection Challenge: overview. Kaggle website. www.kaggle.com/c/rsna-pneumonia-detection-challenge. Accessed July 17, 2019 [Google Scholar]
8. Halabi SS, Prevedello LM, Kalpathy-Cramer J, et al. The RSNA pediatric bone age machine learning challenge. *Radiology* 2019; 290:498–503 [Crossref] [Medline] [Google Scholar]
9. Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks. In: *2017 IEEE International Conference on Computer Vision (ICCV).* Piscataway, NJ: IEEE, 2017:764–773 [Google Scholar]
10. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. arXiv website. arxiv.org/abs/1602.07261. Published February 23, 2016. Accessed May 18, 2018 [Google Scholar]
11. Chollet F. Xception: deep learning with depthwise separable convolutions. arXiv website. arxiv.org/abs/1610.02357. Published October 7, 2016. Accessed May 18, 2018 [Google Scholar]
12. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. arXiv website. arxiv.org/abs/1608.06993. Published August 24, 2016. Accessed May 18, 2018 [Google Scholar]
13. ImageNet website. ImageNet. www.image-net.org/. Accessed February 20, 2019 [Google Scholar]
14. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: *2017 IEEE International Conference on Computer Vision.* Los Alamitos, CA: IEEE, 2017:2999–3007 [Google Scholar]
15. Dai J, Li Y, He K, Sun J. R-FCN: object detection via region-based fully convolutional networks. arXiv website. arxiv.org/abs/1605.06409v2. Published May 20, 2016. Accessed February 22, 2019 [Google Scholar]
16. Hu H, Gu J, Zhang Z, Dai J, Wei Y. Relation networks for object detection. arXiv website. arxiv.org/abs/1711.11575v2. Published November 30, 2017. Accessed February 22, 2019 [Google Scholar]
17. Code for 1st place solution in Kaggle RSNA Pneumonia Detection Challenge. GitHub website. github.com/i-pan/kaggle-rsna18. Accessed July 17, 2019 [Google Scholar]
18. Ng A. Convolutional neural networks. Coursera website. www.coursera.org/learn/convolutional-neural-networks. Accessed December 27, 2018 [Google Scholar]
19. Howard J. Cutting edge deep learning for coders: part 2. Onwards fast ai website. course18.fast.ai/part2.html. Accessed January 26, 2019 [Google Scholar]
20. Gaiser H. Keras implementation of RetinaNet object detection: keras-retinanet. github.com/fizyr/keras-retinanet. GitHub website. Accessed January 19, 2019 [Google Scholar]
21. 3rd Place solution for RSNA Pneumonia Detection Challenge. GitHub website. github.com/pm-cheng/rsna-pneumonia. Accessed March 2019 [Google Scholar]
22. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* Los Alamitos, CA: IEEE, 2009:248–255 [Google Scholar]
23. Pan I, Agarwal S, Merck D. Generalizable inter-institutional classification of abnormal chest radiographs using efficient convolutional neural networks. *J Digit Imaging* 2019 Mar 5 [Epub ahead of print] [Google Scholar]
24. Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Med* 2018; 15:e1002683 [Crossref] [Medline] [Google Scholar]
25. Chang K, Balachandar N, Lam C, et al. Distributed deep learning networks among institutions for medical imaging. *J Am Med Inform Assoc* 2018; 25:945–954 [Crossref] [Medline] [Google Scholar]
26. RSNA Pneumonia Detection Challenge: private leaderboard. Kaggle website. www.kaggle.com/c/rsna-pneumonia-detection-challenge/leaderboard. Accessed July 17, 2019 [Google Scholar]
27. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arxiv.org/abs/1512.03385. arXiv website. Published December 10, 2015. Accessed April 14, 2019 [Google Scholar]