Great Learning

Capstone Project Final Report -Pneumonia Detection using Computer Vision

This document deals with detailed information on the Capstone Project Final Report covering the Scope, Executive Summary, Architecture,EDA, Model Building and Inference, Final Results.

AIML - Post Graduate Program – Group 3A

18-Mar-2020

Table of Contents

Capstone Project Final Report –Pneumonia Detection using Computer Vision

AIML - Post Graduate Program – Group 3A

18-Mar-2020

GIT Repo Link : https://github.com/niteshnagreddy/Group3A-Capstone.git

1. Executive Summary

This project represents a culmination of the Ten modules of the AI and ML Specialization offered by Great Lakes Executive Learning and University of Texas at Austin via Great Learning. The Pneumonia Detection prediction model is built based on the basics of Computer Vision Technique techniques learned throughout the specialization. The project focuses on to build Prediction model on Pneumonia Detection

- Build a pneumonia detection model starting from basic CNN and then improving upon it.
- Train the model. To deal with large training time, save the weights so that you can use them when training the model for the second time without starting from scratch.
- Test the model and report as per evaluation metrics - IOU - Intersection over Union
- Build models on SSD, Mask R CNN, YoloV3 for the Pneumonia Detection
- Set different hyper parameters, by trying different optimizers, loss functions, epochs, learning rate, batch size, check pointing, early stopping etc..for these models to fine tune them
- Evaluate metrics for these models along with your observation on how changing different hyper parameters leads to change in the final evaluation metric.
- Deploy the Best Predicting Model on to Google Cloud Platform

2.Analysis

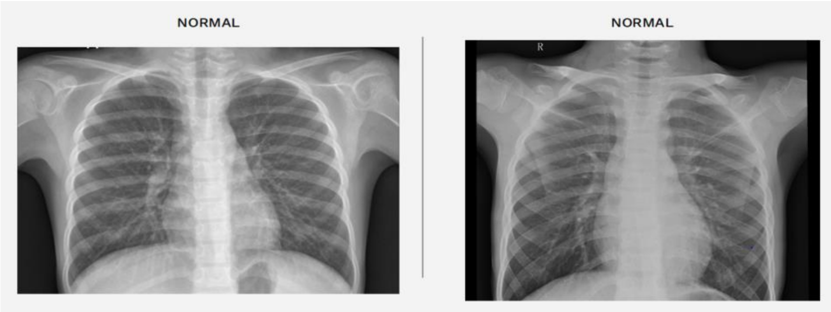
Exploratory Data Analysis

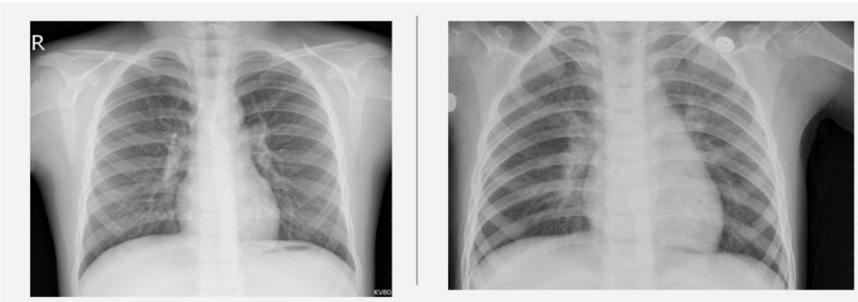
The corpora given comprises X-RAY of Lung images of very large datasets from Kaggle competition - of more than1000 and above DICOM images with file size of over 4 GB.

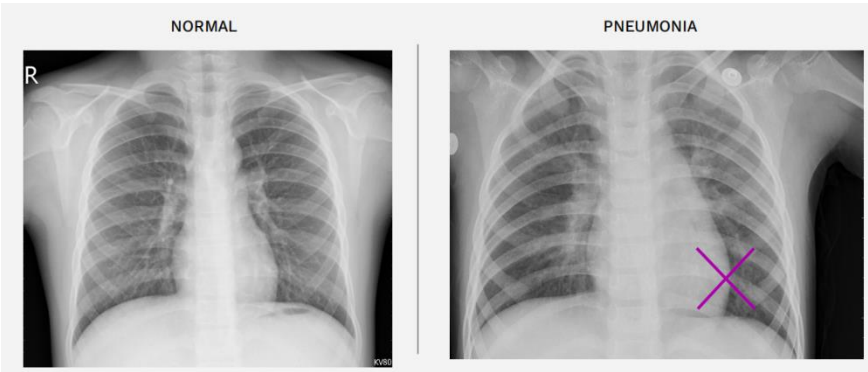
Reference Jupyter notebook- EDA_PneumoniaDetection.ipynb

Kaggle dataset - quick look

- 5’863 X-Ray images
- pediatric patients 1-5 years old
- labeled by several specialists







In order to be accomodated within my system limitations (and in keeping with the approach recommended) a sample of the corpus was selected for study in order to build and train the prediction model.

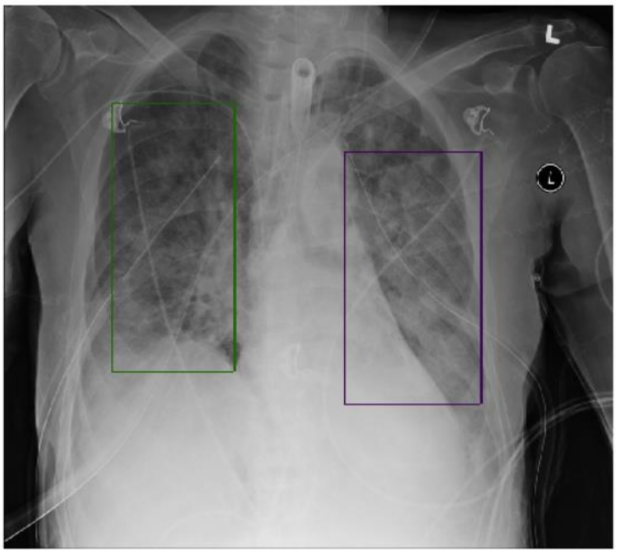
It is very important to understand the data in DICOM files before we work on Prediction Models.

```
import pydicom
dcm_file = '../content/RSNAdata/data/stage_2_train_images/%s.dcm' % pat_choose

dcm_data = pydicom.read_file(dcm_file)

print(dcm_data)
```

```
(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: 1.2.276.0.7230010.3.1.4.8323329.1556.1517874291.545552
(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: AP'
(0010, 0010) Patient's Name                      PN: '00f08de1-517e-4652-a04f-d1dc9ee48593'
(0010, 0020) Patient ID                          LO: '00f08de1-517e-4652-a04f-d1dc9ee48593'
(0010, 0030) Patient's Birth Date                DA: ''
(0010, 0040) Patient's Sex                       CS: 'M'
(0010, 1010) Patient's Age                       AS: '58'
(0018, 0015) Body Part Examined                  CS: 'CHEST'
(0018, 5101) View Position                       CS: 'AP'
(0020, 000d) Study Instance UID                  UI: 1.2.276.0.7230010.3.1.2.8323329.1556.1517874291.545551
(0020, 000e) Series Instance UID                 UI: 1.2.276.0.7230010.3.1.3.8323329.1556.1517874291.545550
(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: [0.139, 0.139]
(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 143458 elements
```

Understanding the data from the DICOM files is imperative to being able to ensure one's conceptualization of bounding boxes on the arrays from those files. We need to visualize those boxes in order to augment the knowledge regarding the visual aspects of pneumonia:

```
Patient ID: 00f08de1-517e-4652-a04f-d1dc9ee48593
```




```
patientId    00f08de1-517e-4652-a04f-d1dc9ee48593
x                                             181
y                                             184
width                                         206
height                                        506
Target                                          1
class                                Lung Opacity
Name: 19, dtype: object
```

Overview: Data present in the form of pydicom images and the pixel array is extracted out of the pydicom file and the annotations are used from the CSV supplied to us which consists of the bounding box dimensions.

Created a metadata of the model names and the annotation objects so that the data set can be prepared accordingly based on the training on sample or whole data

Approach planned is to evaluate different models like SSD, YOLO, MASK R CNN on the sample data to see which architecture is performing better compared to the others based on accuracy

Finalized model would be trained on full dataset to do improvements on hyper parameters and save the best model.

3. Methodology (Step by step walkthrough of solution)

Model Building

Approach 1- Bounding Box Predictor with CNN

Reference jupyter notebook- BoundingBoxPredictor(CNN).ipynb

Step 1: Load the trainingdataSample , BoundingdataSamples and testdatasample and parse through Patient ID

Step 2: Create Classification labels on “Pneumonia” and “LungOpacity” and load images with label “Pneumonia” and perform masking using CV2.

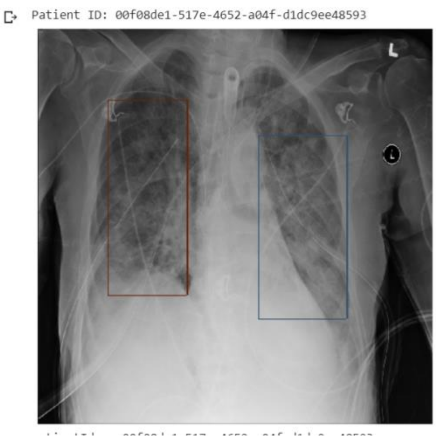
Step 3: Perform Image annotations over the Training images

Step 4: Plot the masked images and set Detector configurations

Step 5: Load the pre-trained CNN models with all layers and batched dataset.

Step 6: Load the weights on the CNN model and get colors for Class ID =1

Step 7: Once the model is loaded, plot and visualize the model detection output



Approach 2- Bounding Box Predictor using SSD

Reference jupyter notebook- PneumoniaDetection_SSD_Nitesh.ipynb

Step 1: Load the trainingdataSample , BoundingdataSamples and testdatasample and parse through Patient ID

Step 2: Create Classification labels on “Pneumonia” and “LungOpacity” and load images with label “Pneumonia” and perform masking using CV2.

Step 3: Perform Image annotations over the Training images
Step 4: Plot the masked images and set Detector configurations

Step 5: Load the pre-trained Mobilenet SSD models with all layers and batched dataset.

Step 6: Load the weights on the SSD model and get colors for Class ID =1

Step 7: Once the model is loaded, plot and visualize the model detection output

Approach 3 - Bounding Box Predictor using Mask R CNN

Reference jupyter notebook- PneumoniaDetection_Maskrcnn_Nitesh.ipynb

Step 1: Load the sample data for the train and test

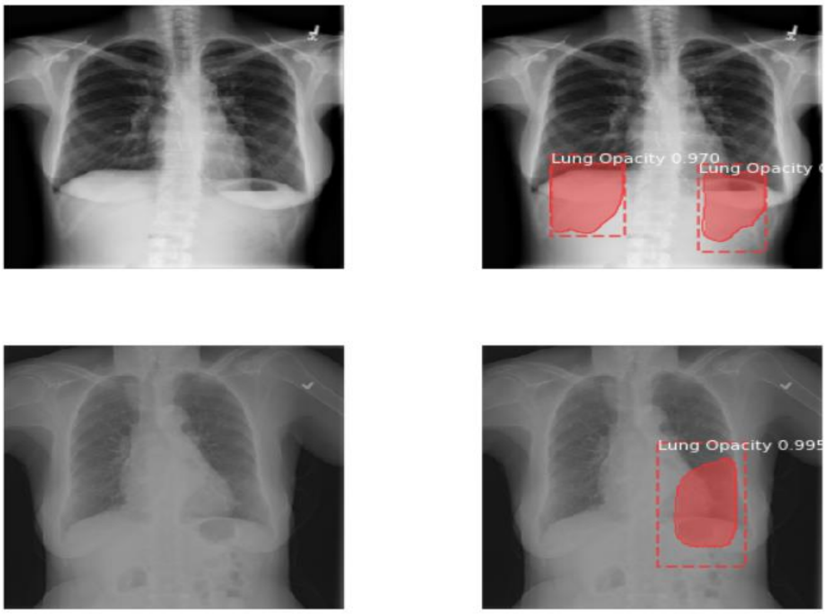
Step 2: Create Classification labels on "Pneumonia" and "LungOpacity" and load images with label "Pneumonia".

Step 3: Parse the dataset to create a generator object and prepare the dataset (annotations dictionary and the generator object)
Step 4: Plot the masked images and set Detector configurations

Step 5: Load the pre-trained Mask RCNN models with all layers.

Step 6: Prepare the configuration to be used by the model

Step 7: Train the model on the detector dataset prepared using the training data and validate on the test data



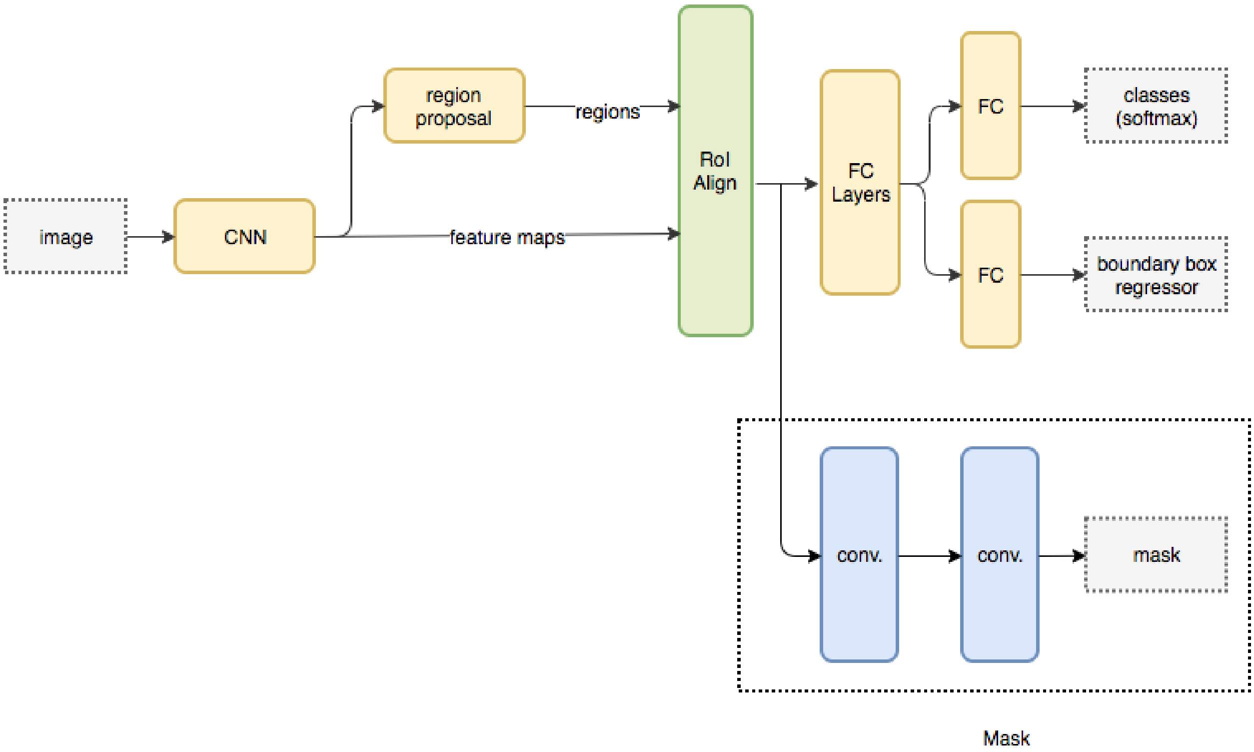
4. Architecture

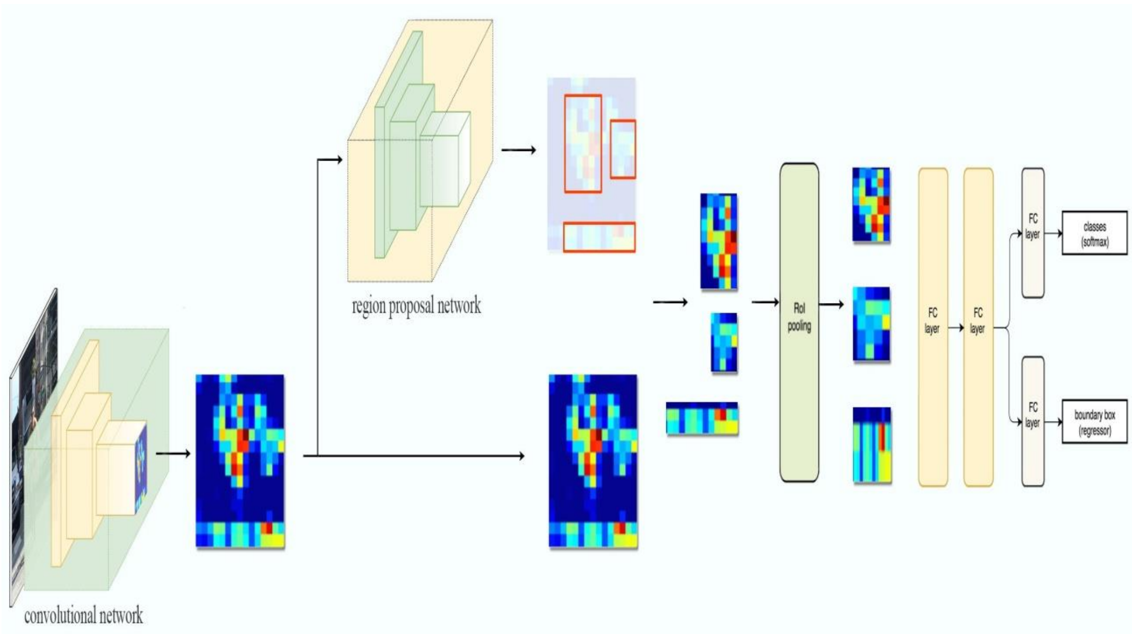
Mask R CNN Architectue

The Faster R-CNN builds all the ground works for feature extractions and ROI proposals. At first sight, performing image segmentation may require more detail analysis to colorize the image segments. By surprise, not only we can piggyback on this model, the extra work required is pretty simple. After the ROI pooling, we add 2 more convolution layers to build the mask.

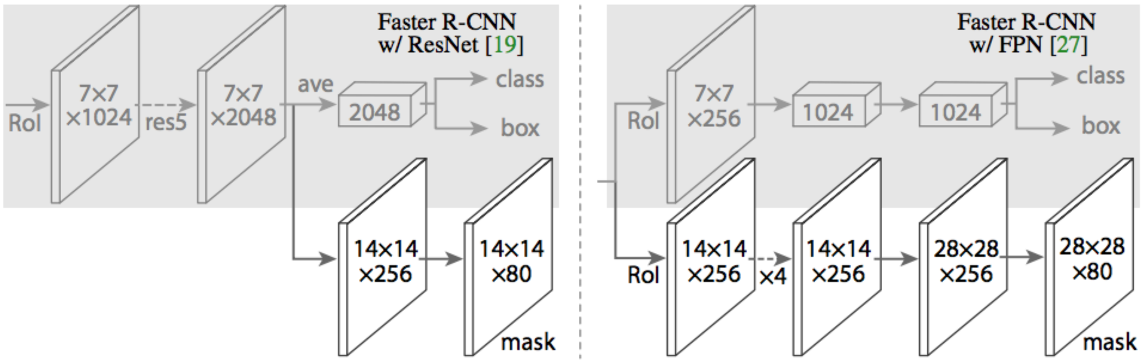
Reference Archives

https://arxiv.org/pdf/1703.06870.pdf





The Mask R-CNN paper provides one more variant (on the right) in building such mask.



5. Results

Model Evaluation and Validation

IOU - Intersection over union has been set as a benchmark against all 4 Models (CNN, SSD, Mask R CNN, Yolo V3). Objects in an Image/Frame are detected with a simple box plotted around them.This task of plotting a box around the Object can be called bounding boxes.The bounding box is nothing but (x-y) coordinates of the object in the image. These co-ordinates uniquely defined objects in the Image.Now, the bounding box for an Object in Image is primarily hand labeled and can be called as Primary Boundary Box.The Deep Learning model predicts a bounding box around the Object which can be called Predicted Boundary Box.

IOU can be computed as Area of Intersection divided over Area of Union

Areas of Intersection

Areas of Union

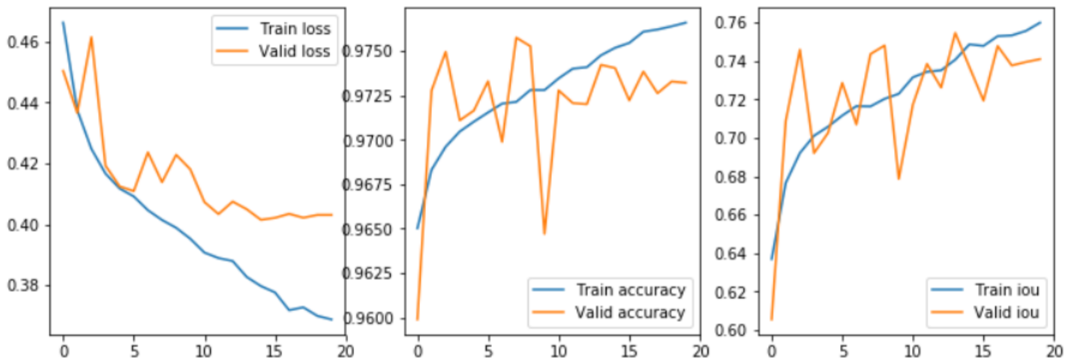
The model output for predicted bounding box is extremely unlikely to be as an exact primary bounding box in reality. Therefore to measure how accurate is the object identified in the Image/Frame we can make use of metric IOU.

This gives us an option to consider the object detected is complete or not. The IOU is a simple way of evaluation of our training model +bounding box with its performance on the testing set.

General Threshold for the IOU can be 0.5. This can vary from problem to problem. Normally **IOU>0.5 is considered a good prediction.**

IOU is an important metric in deciding the object prediction of deep learning models.

Sample IOU plot is shown below



As per our Analysis, our findings on each models shows as below

Model	***IOU Metric***	***Remarks***
CNN	0.65	Bad
SSD	0.69	Good
Mask R CNN	0.89	Best
Yolo V3	0.71	Good

Mask R CNN scored better compared to other models in terms of the IOU metrics.

Comparison to Benchmark:

Mask R CNN model was trained with different hyper parameters and randomized tuning of the hyper parameters were done on the sample data set to evaluate the model performance and benchmark the optimal settings among the trial cases performed.

Configuration was tweaked manually for the different range of hyper parameters and the IOU score was verified. Parameters mentioned below were tweaked for the tuning:

i. BATCH_SIZE (5,8,10)

ii. LEARNING RATE (0.1,0.01,0.001)

iii. STEPS_PER_EPOCH (100,150,225)

For the above combinations, the best model accuracy for the IOU metric was arrived for the parameter values of BATCH_SIZE = 8, LEARNING_RATE = 0.001 and STEPS_PER_EPOCH = 225.

Below is the finalized configuration added for the benchmarked model.

```
Configurations:
BACKBONE                       resnet50
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     8
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        3
DETECTION_MIN_CONFIDENCE       0.7
DETECTION_NMS_THRESHOLD        0.1
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 8
IMAGE_MAX_DIM                  256
IMAGE_META_SIZE                14
IMAGE_MIN_DIM                  256
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [256 256   3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.001
LOSS_WEIGHTS                   {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0,
'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               3
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           pneumoniaFull
NUM_CLASSES                    2
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        1000
POST_NMS_ROIS_TRAINING         2000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (32, 64, 128, 256)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                225
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           32
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               50
WEIGHT_DECAY                   0.0001
```

Visualization:

Input image and bounding box prediction for random samples of data were checked

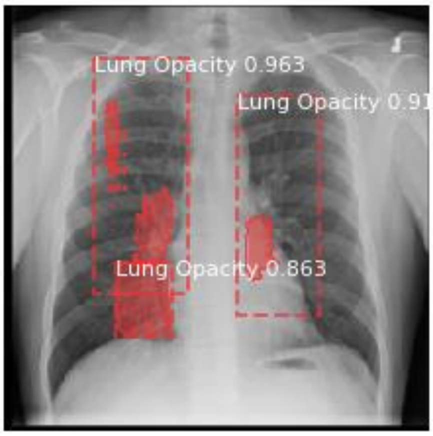
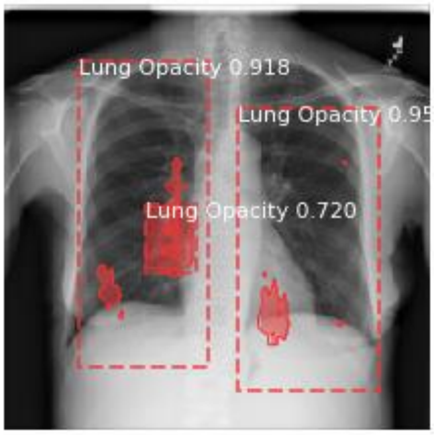



Figure above shows the input image and the mask created by the model.

As can be observed, the model is able to identify the minute intricacies of the shaded areas of lung opacity and the shading along with the bounding box is drawn.

Another sample shown above is representative of the very intricate shading of the areas where the area around the ribs is highlighted as being opaque.

Implications on business:

- Solution achieved currently helps the radiologist to identify the areas of opacity to assess the degree of pneumonia which the patient is affected with in order to visually communicate it to the doctors for further investigations.
- Secondly, it also helps speed up the process of medical diagnosis to treatment so that the patient can be relieved of his symptoms as the treatment can be started much earlier.
- Thirdly, the confidence level can be tweaked up based on the potential risk that can be allowed by the model error and the current confidence allowed is 80%.

Limitation of solution:

- X-Ray films having the opaque area not belonging to the chest may be identified as lung opacity
- Typically, minute opacities present in few cases and fall under the model error may not be predicted correctly
- Cases pertaining to defect in X-Ray image due to which a non opaque region is seemingly looking like an opaque area, model tends to classify it and draw a mask as lung opacity.

Closing Reflections:

Learnings:

- Dealing with DICOM images and working with pydicom library for the data pre-processing
- Hands on working knowledge of various algorithms used for bounding box detection and how to customize the layers to match the requirements.
- Fine – tuning the models and improving the accuracies based on the validation set by tuning the hyper parameters to improve the model performance.
- Plotting pydicom images and adding masks over it to visualize the image with masks / bounding box

What can be improved?

- Model developed is using the configuration and minimalistic performance tuning is done on the hyper parameters due to the lack of the computing resources and computing time.
- Model can be improved further more if the randomized grid search is implemented to find the optimal values of hyper parameters and the validation data is used as a basis to finalize the hyper parameters

What can be done differently next time:

- Model tuning can be more effective
- Randomized grid search CV could be implemented
- Other metrics of model accuracy could be explored and evaluated
- More epochs and lower learning rates could be tried out based on the availability of time and computing resources.