Final Project

Ryspayev Vladislav

I. Introduction

The project was mainly aimed at detecting pneumonia disease by observing patients' images and locating them by predicting bounding boxes. Several difficulties were faced like creating a generator/preprocessor for images, defining metrics for images and overall working with medical images

II. EDA

A. *First look at the data*

First of all, I observed that a number of records is larger than unique rows. It became evident that some patients had 2 bounding boxes of pneumonia instances. Looking at the distributions of lung conditions Fig1 and target labels Fig2, the inadequate imbalance was noted. It was balanced after further training because CNN works better with balanced data.

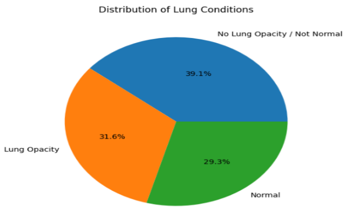

Fig. 1. Distribution of Lung Conditions.

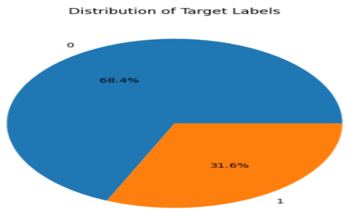

Fig. 2. Distribution of Target Labels.

Data before and after balancing classes Fig3

Here is some visuzalization of somebounding boxes Fig4

B. *Some Thoughts*

The data set is pretty large so data generator would be useful for training the model by batches

III. Building Model

The CNN model was built. The structure of the model is: Initial Convolution Layer: The model starts with a standard 2D convolution layer (Conv2D) that processes the input image. Downsampling and Residual Blocks: The model then alternates between downsampling layers and a series of residual blocks, repeated num-downsampling-layers times. Each downsampling layer consists of Batch Normalization, LeakyReLU activation, a 1x1 Conv2D layer for channel adjustment, and a MaxPooling layer to reduce the spatial dimensions by half. This process effectively compresses the input, focusing on the most important features while reducing computation for deeper layers. Each residual block further processes the data using two sets of Batch Normalization, LeakyReLU activation, and 3x3 Conv2D layers. The output of the second Conv2D layer is added to the original input of the residual block (a shortcut connection), which helps in mitigating the vanishing gradient problem and allows deeper models to be trained effectively. After the downsampling and residual blocks, the model applies another set of Batch Normalization and LeakyReLU. An UpSampling2D layer then upscales the output back to the original input size. The factor for upsampling is 2 times num-downsampling-layers, reversing the spatial dimension reduction done by the MaxPooling layers.

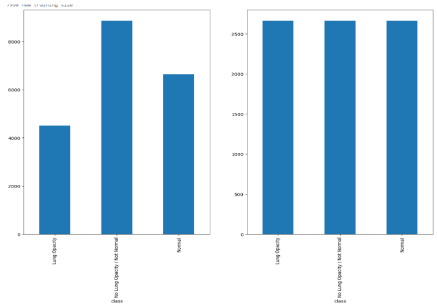

Fig. 3. Classes balancing.

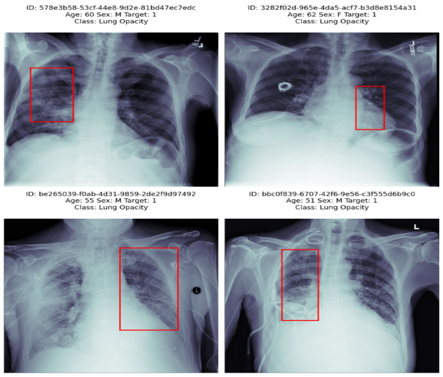

Fig. 4. Lungs.

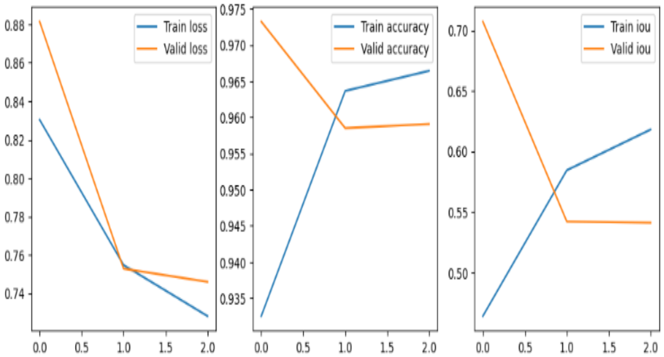

Fig. 5. Metrics Plot.

A. The Image Processor

The ImageProcessor class in the code is a custom utility for handling medical image data, specifically for training deep learning models in Keras. It's designed to load and preprocess DICOM images, a common format in medical imaging, and organize them into batches suitable for model training. This includes resizing images, applying data augmentation (like flipping), and generating corresponding masks for lung abnormalities if available. The class supports both training and prediction modes, making it versatile for the entire model lifecycle. Its integration with Keras through the Sequence interface ensures efficient and effective processing of large medical datasets, crucial for training robust deep learning models in the medical imaging field.

B. The problem

The model trained for a long time, each epoch took around 20 minutes, so the number of epochs was reduced to 3. This may lead to poor performance of the model

IV. Results

A. After the training

The following plots of loss, accuracy, IoU was obtained Fig5.

B. Predictions

After predicting on the Kaggle data test set, the scores of 0.03627- private, and 0.01527 - public. Here are some comparisons on the fig 6. We see that the model makes two predictions because some regions are easily can be confused with pneumonia regions.

V. Some thoughts

The performance of the model should be improved by increasing the number of epochs, by using large pre-trained models like YOLO, Mask R-CNN, and so on. However, training took too long and the project had a deadline, so I was not able to train it for enough time. The use of pre-trained models was beyond of my interests because I primarily wanted to implement and build the model myself

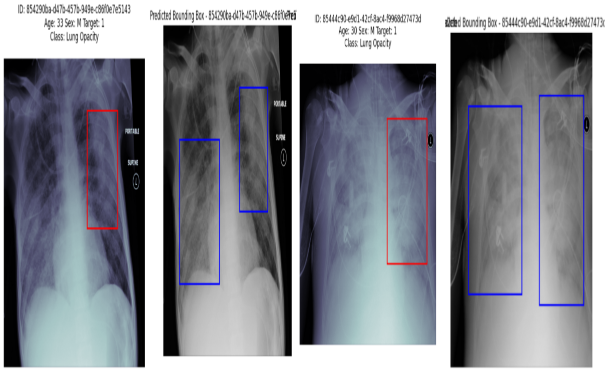
ID: 854290ba-d47b-457b-949e-c86f0e7e5143
Age: 33 Sex: M Target: 1
Class: Lung Opacity
PORTABLE
SUPINE
Predicted Bounding Box - 854290ba-d47b-457b-949e-c86f0e7e5
PORTABLE
SUPINE
ID: 85444c90-e9d1-42cf-8ac4-f9968d27473d
Age: 30 Sex: M Target: 1
Class: Lung Opacity
L
Predicted Bounding Box - 85444c90-e9d1-42cf-8ac4-f9968d27473
L

Fig. 6. Bounding boxes actual vs predicted.