Lung Nodule Detection

Project Report

By
Ayush Kumar

Abstract

Lung Carcinoma, commonly known as Lung Cancer is an infectious lung tumour caused by uncontrollable tissue growth in the lungs. Current diagnostic methods include biopsies and imaging. Early detection of lung cancer (detection during the earlier stages) significantly improves the chances for survival. In this project I want to explore the task of detecting lung cancer given patients CT scans of lungs. I used LUNA 16 dataset for this problem along with the nodule annotations. I performed a combination of pre-processing techniques, lung segmentation and nodule detection on the data. I used various image processing techniques like CLAHE (Contrast Limited Adaptive Histogram Equalization), threshold, erosion, dilation etc. for segmentation of lungs from ct scans. For nodule detection, I trained a 2D U-NET convolutional neural network architecture.

Introduction

Lung Cancer is most dominant reason for cancer related deaths across the globe. The uncontrollable division of undesirable cells in the lung region can be classified as Lung Cancer. As their growth progresses, the abnormal cells form tumours and interfere with the normal functioning of the lungs. Since there are no apparent signs or symptoms of early lung cancer, the clinical diagnosis of lung cancers are often late, making treatment expensive and ineffective. Early diagnosis of cancer is critical for providing victims with the best treatment and the possibility of a revival. Early Detection of Lung cancer in its early stages using CT scans would help save many lives but analysing the scans of the majority is an immeasurable burden for radiologists. I have used a Deep Convolutional Neural Network (DCNN) and numerous preprocessing techniques to buildup the exactness of the automated prediction of Lung nodules using CT Scans.

I used LUNA16 (LUng Nodule Analysis 2016) dataset that contains 888 CT Scans. It also contains annotations which were collected during a two-phase annotation process using 4 experienced radiologists.

The evaluation function adopted in this study is dice coefficient loss, which is usually used in image segmentation tasks. The final model was able to achieve the dice score of 0.81 .

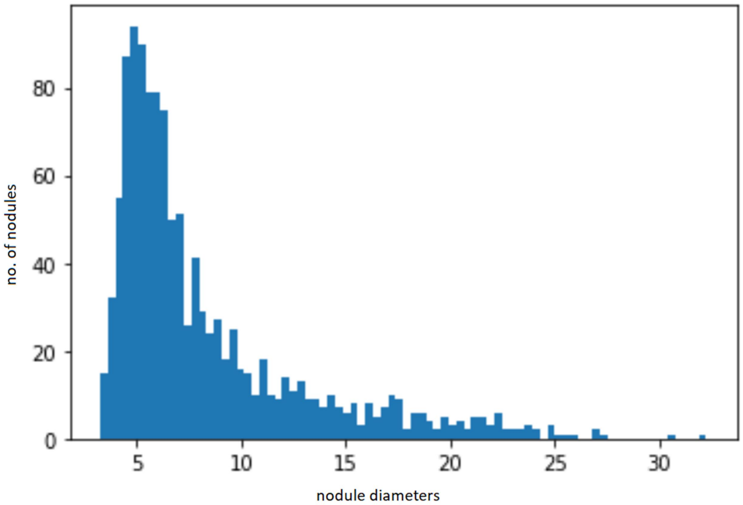
Methods

1. Data Description

I used the LUNA16 dataset which consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans with marked-up annotated lesions. This dataset has 888 patient cases, where each case includes CT scans and annotations of nodule position. The annotation file is a csv file that contains one finding per line. Each line holds the SeriesInstanceUID of the scan, the x, y, and z position of each finding in world coordinates; and the corresponding diameter in mm. The annotation file contains 1186 nodules.

The complete dataset is divided into 10 subsets that should be used for the 10-fold cross-validation. All subsets are available as compressed zip files. In each subset, CT images are stored in MetaImage (mhd/raw) format. Each .mhd file is stored with a separate .raw binary file for the pixeldata.

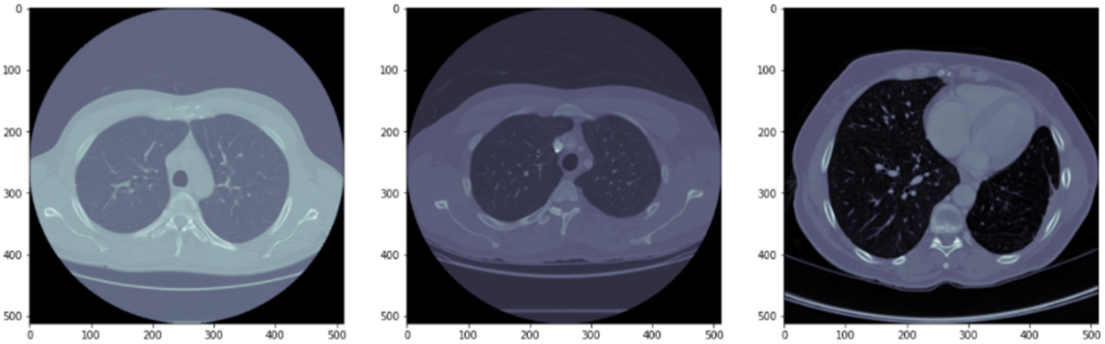
The nodules present has diameters from 3mm to 33mm. Below graph shows distribution of diameters of nodules present.



2. Pre-processing

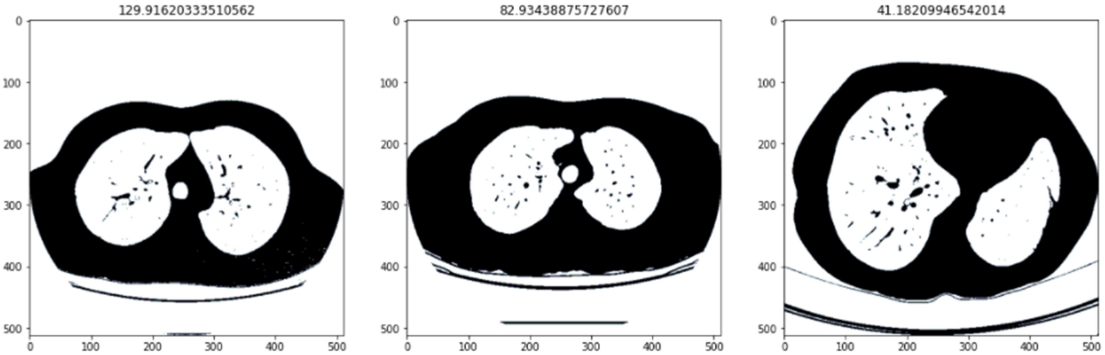
I used *SimpleITK* python library to read the mhd files from the dataset and extract lung images. Each CT scans is a 3D medical image consisting of multiple slices representing different cross-sectional views of lungs.

After extracting lung images from mhd files, pixel values were normalized. Then based on annotations for that ct scan respective slides were took out where nodules were present.

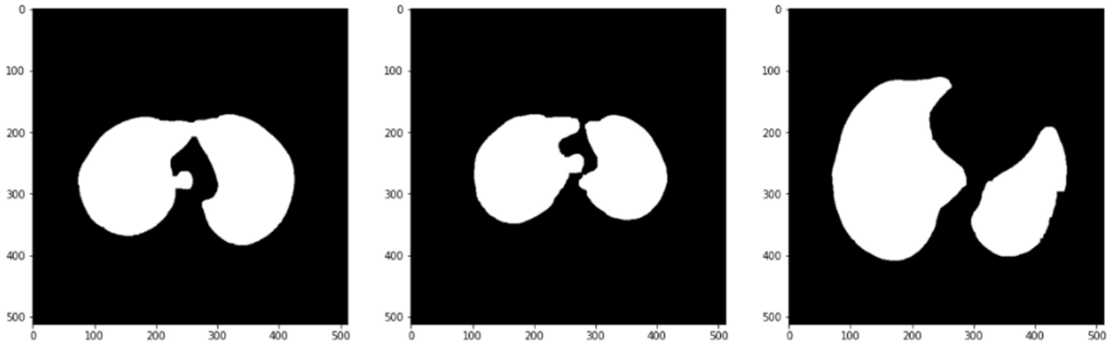


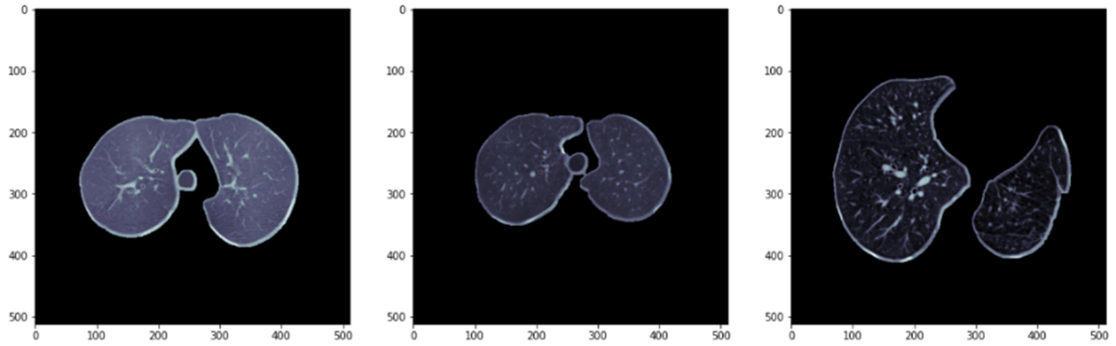
I used *OpenCV* library for applying various image processing techniques. First I used CLAHE (Contrast Limited Adaptive Histogram Equalization) to balance the contrast in CT images, then applied threshold to create binary image of the lungs for the purpose of segmenting lungs. Hard coded threshold value didn't give good result because of different brightness levels of different ct scans. So I used KMeans clustering algorithm to cluster bright pixels (tissues) and dark pixels (empty space) in the image into two clusters and the taking mean of the centroids of two clusters was used as threshold value.

The numerical values shown on top of below figure are the different threshold values for respective images.



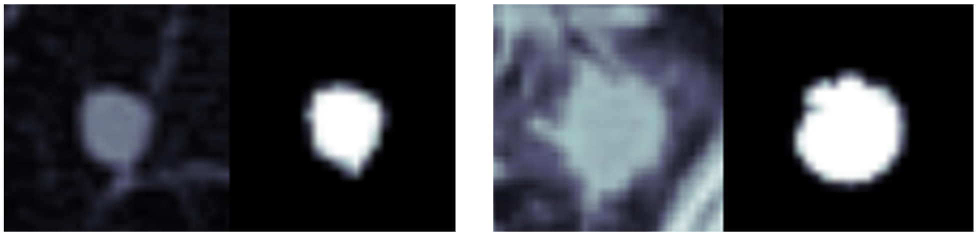
After thresholding, several erosion and dilation we applied to remove blood vessels, filling holes etc. Then labeled each region and obtained the region properties, the background region was removed by removing regions with a box that is too large in either dimension also, the lungs are generally far away from the top and bottom of the image, so any regions that are too close to the top and bottom are removed this does not produce a perfect segmentation of the lungs from the image, but it is surprisingly good considering its simplicity.





Then applying the final mask to the lungs image gets us the ROI (lungs).

For the extraction of nodule masks, I first got the nodule location from annotations file and extracted the circular area at the nodule location and then to get the mask of exact shape as of nodule, I applied binary threshold with same threshold values as obtained for lungs segmentation above.



3. Training

I used a 2D UNet convolutional neural network architecture which is mainly used for image segmentation. U-net is an encoder-decoder deep learning model which is known to be used in medical images. It is first used in biomedical image segmentation. U-net contained three main blocks, downsampling, upsampling, and concatenation.

The whole code was implemented using Tensorflow in python.

The dice coefficient loss is selected as the loss function. Dice coefficient as is often used in medical image segmentation.

$$\text{Dice Coefficient} = \frac{2|X \cap Y|}{|X| + |Y|}$$

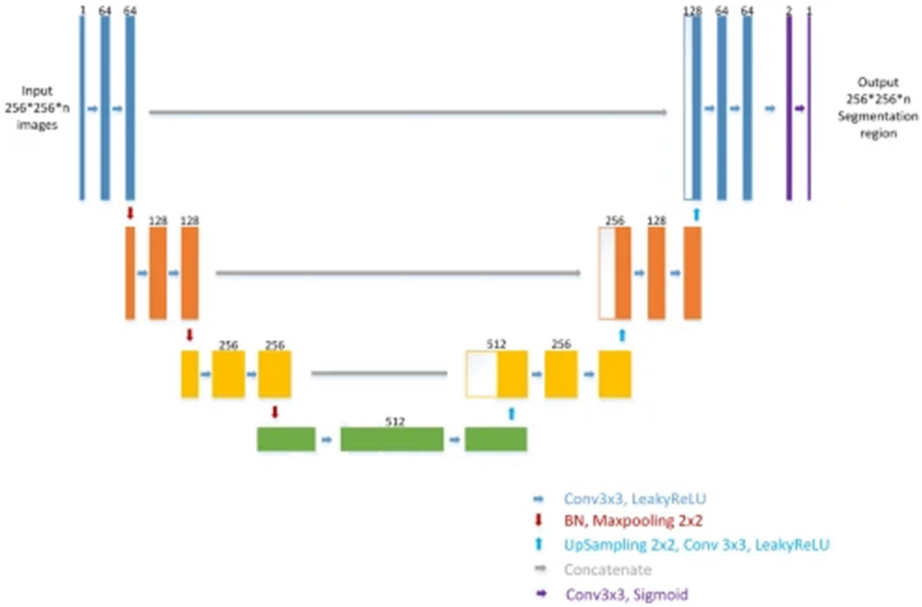
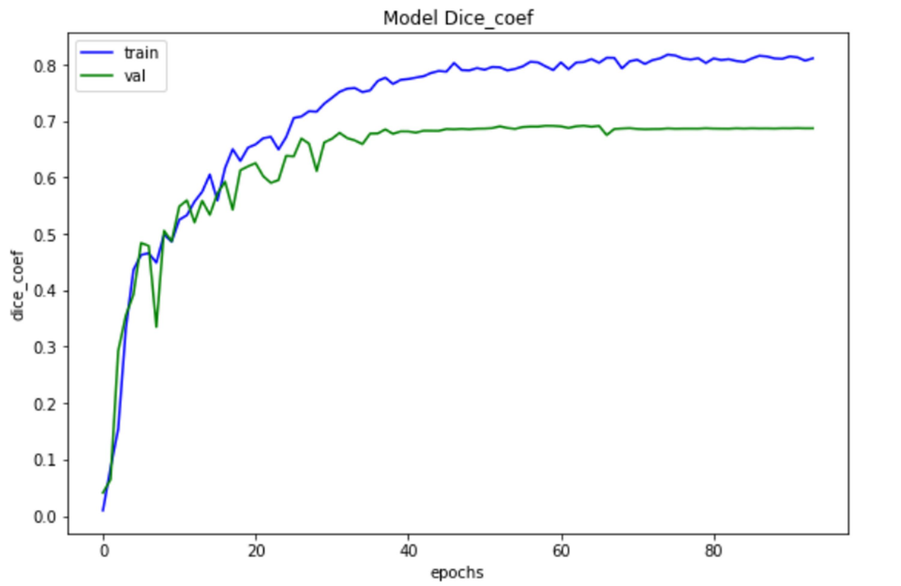


Fig. U-Net architecture

I also used various kinds of callbacks like EarlyStopping, ReduceLROnPlateau, ModelCheckpoint to get best result from training.

4. Result

This model was able to achieve a dice score of 0.81 in training data and 0.68 on test data. The model was trained for 94 epochs.



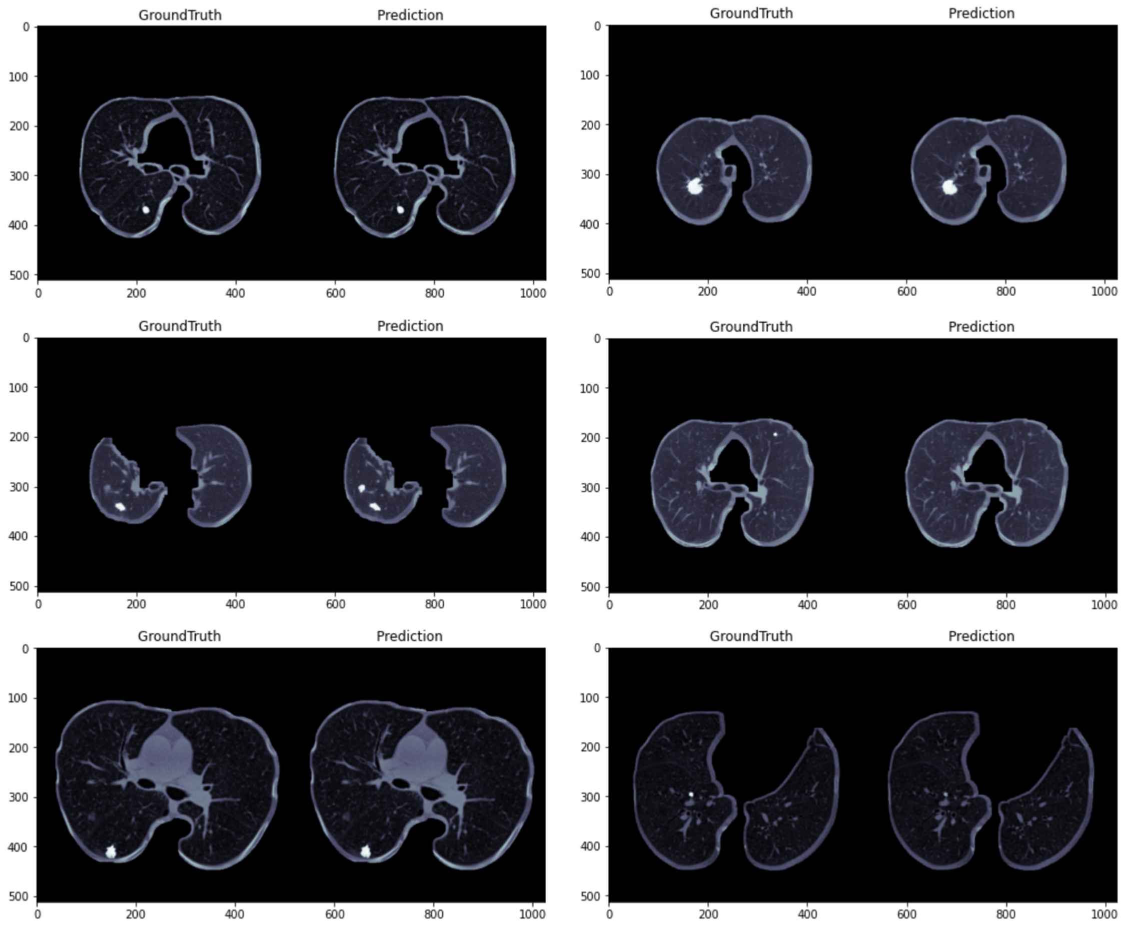


Fig. GroundTruth vs Predictions

References

i. Automatic detect lung node with deep learning in segmentation and imbalance data labelling, https://www.nature.com/articles/s41598-021-90599-4 (2021)

ii. Automated Lung Nodule Detection and Classification Using Deep Learning Combined with Multiple Strategies, https://ieeexplore.ieee.org/document/9182258 (2019)

iii. Automated lung tumor diagnosis in medical image data - methods, challenges and perspectives, https://youtu.be/iAQyBGfYlwc (2018)