Predicting Tuberculosis Related Lung Deformities from CT Scan Images Using 3D CNN

Anup Pattnaik, Sarthak Kanodia, Rahul Chowdhury, and Smita Mohanty

PricewaterhouseCoopers US Advisory, Mumbai, India
{anup.a.pattnaik, sarthak.p.kanodia, rahul.r.chowdhury, smita.u.mohanty}@pwc.com

Abstract. CT scan of lung has become an invaluable tool in the diagnosis of tuberculosis. However, analysis of 3-D image data is time consuming and relies heavily on trained expertise. As an attempt to automate this approach without compromising accuracy advanced AI algorithms have been explored to draw clinically actionable hypothesis. The approach comprises of detailed image processing, followed by feature extraction using tensor flow and 3-D CNN to further augment the metadata with the features extracted from the image data and finally perform 6 class binary classification using random forest. On the test dataset the method resulted in an overall mean AUC of 0.6.

Keywords: 3D-CNN · Neural Networks · Deep Learning · Medical Imaging · CT · Tuberculosis · ImageCLEF

1 Introduction

Tuberculosis is a very common and contagious disease caused by the bacteria Mycobacterium tuberculosis and is one of the top 10 causes of death worldwide, according to the World Health Organization (WHO). In 2017, 10 million people were diagnosed with TB and 1.6 million people died from the disease. Medical imaging plays a very important part in diagnosing TB and determining the medical course.

The ImageCLEF 2019 Tuberculosis task [1] from ImageCLEF 2019 [2] consists of 2 subtasks - CT report and severity scoring, both of which focus on improving and automating existing TB diagnosis by analysing medical image through the application of state of the art deep learning techniques and developing a framework capable of extracting necessary information; minimising human intervention throughout the process. The aim of the severity scoring subtask was

to assess TB severity score on a scale of 1 (critical) to 5 (very good), whereas that of the CT report subtask was to generate an automatic report based on CT image which would include the following information in binary form: Left lung affected, right lung affected, presence of calcifications, presence of caverns, pleurisy, lung capacity decrease. This article describes the solution provided by PwC to the CT report subtask using 3-D convolutional neural network.

2 Datasets

For ImageCLEF 2019 tuberculosis task, both the CT report and severity scoring subtasks use the same dataset containing 335 chest 3-D CT scans of TB patients - of which 218 are used for training and 117 are used for testing. The image data is supported by clinically relevant metadata on the following attributes in binary form: disability, relapse, symptoms of TB, comorbidity, bacillary, drug resistance, higher education, ex-prisoner, alcoholic, smoking.

The 3-D CT images for individual patients were provided in the form of 2-D slices with dimension of 512×512 pixels and number of slices varying from 100 to 150. The CT images were stored in NIFTI file format which stores raw voxel intensities in Hounsfield units (HU) along with the image metadata viz. image dimensions, voxel size in physical units, slice thickness etc.

Automatic extracted masked image [3] of the lungs were also provided along with the image data.

3 Methodology

The solution to the CT report task is based on a two-stage approach: data pre-processing and modeling. In the data pre-processing stage, the CT scan images and mask images were processed to be able to fed in directly to the neural network architecture developed.

In the pre-processing stage, the CT scan images were resized, slices were concatenated to maintain consistency across patients (20 slices per patient were generated) and segmented. The mask image data along with the CT images were used to extract lung volume. The detailed methodology for pre-processing is explained in detail in the next subsection. The processed scans were then passed through a neural network and features were extracted. The extracted features, lung volume and patient attribute metadata were then fed into a machine learning classifier and trained on the training dataset.

3.1 Image Pre-processing

The training image data provided comprised of 3-D CT scanned images of 218 TB patients with slice size of 512×512 pixels and variable number of slices i.e. the images were of size (512, 512, z) where z is the number of slices in the CT scan and it varied depending on the resolution of the scanner.

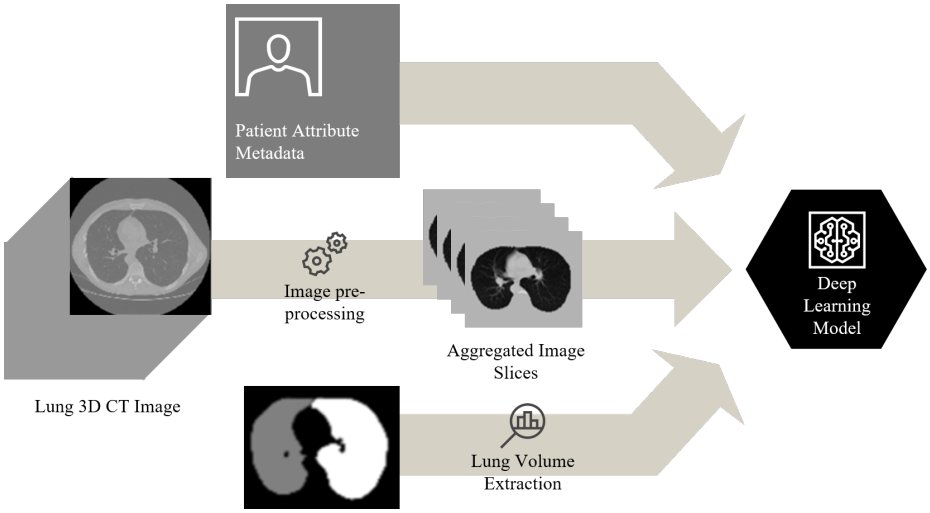

Fig. 1. Overall approach flow.

Due to limitation of computation power and to make the framework computationally efficient, the large 3-D images were pre-processed before feeding into a convolution network architecture. The pre-processing of images involved resizing, concatenation and segmentation, to reduce size, have uniform number of slices per patient and find the regions within each image that are more probable of having deformities.

Resizing The first step of image pre-processing was to downsize the CT images to 150×150 pixels using OpenCV as shown in Fig. 2. [4]

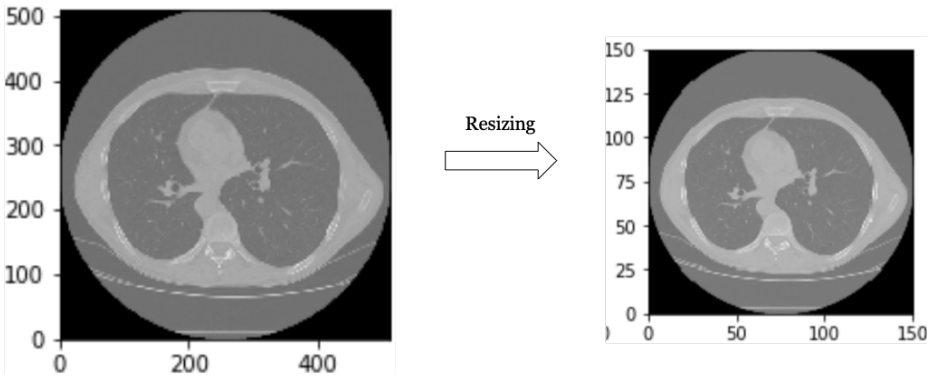

Fig. 2. a. Original 512×512 pixels image. b. Resized image of 150×150 pixels.

Concatenation To handle the problem of non-uniformity in depth in terms of number of slices (ranging from about 100 to 150 per patient) i.e. to have constant number of slices for each patient, concatenation was used to create 20 slices per patient as shown in Fig. 3 below. [4]

The vector representation of consecutive slices were very similar to each other, so in order to maintain heterogeneity between slices, and also to reduce the number of slices, multiple consecutive slices were concatenated. The number of slices to concatenate was determined by dividing the total number of slices for each patient by 20 (e.g. if a patient had 100 slices, every 5 consecutive slices were concatenated to generate 20 slices overall) and then the average value of the concatenated slices was considered as the value of the new slice.

For cases where number of processed slices were less than 20, the last slice was appended more than once (in case of 19 or 18 slices). 6 patients were dropped entirely which had less than 18 slices.

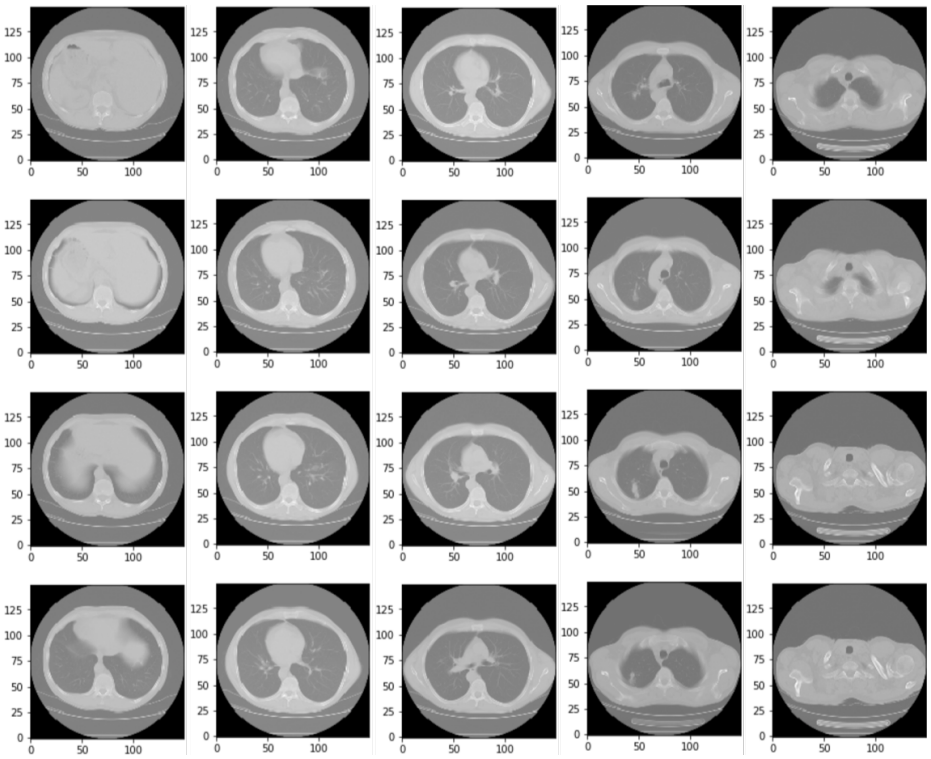

Fig. 3. Concatenated slices of 3-D CT scan of a patient arranged in a vertical sequence.

Custom Mask Generation Apart from the 3-D CT scanned images, masks of the lung images were also provided which are segmented images of the original lung image, created by cutting off the left and right lung fields from the lung parts and removing the surrounding noise. The masks could be used to get the areas of interest within the original image which would then be fed to the deep learning architecture. Since the masked images played a crucial part in the entire classification process, custom masks were generated by performing the steps mentioned in the Segmentation section below. It allowed modification and fine tuning the region of interest within the lungs by changing the hyperparameters involved in the segmentation process. It also served as a benchmark to compare the results obtained by using the masked images that is part of the dataset provided.

Segmentation The next step in pre-processing was segmentation of lung structures because the regions of interests lies inside the lungs. In the CT scans, the lungs are the darker regions whereas the brighter regions inside the lungs are blood vessels or air. The lung structures were segmented from each slice of the CT scan image and it was tried not to lose the possible region of interests attached to the lung wall.

The segmentation of lung structures is a very challenging problem because: homogeneity is not present in the lung region, pulmonary structures have similar densities, scanners and scanning protocols for capturing images are usually different. For getting the segmented lungs 8 steps were carried out using scikit-image package. [5]

Conversion to binary image In the first step of segmentation, concatenated slices of images were converted into binary images as shown in Fig. 4a. Typical radiodensities of various parts of a CT scan are shown in Table 1. A thresholding of -600 HU was applied to segment lung parenchyma. [5]

Table 1. Typical radiodensities of various substances (in HU) in a CT scan

Substance	**Radiodensity (HU Range)**
Air	-1000
Lung	-400 to -600
Nodule	-150
Fat	-60 to -100
Water and Blood	0
Soft Tissue	+40 to +80
Bone	+400 to +1000

Removing the blobs connected to the border In order to filter the noise obtained as a result of segmentation, and to be able to classify the images properly, the

regions which were connected to the border of the image were removed as shown in Fig. 4b.

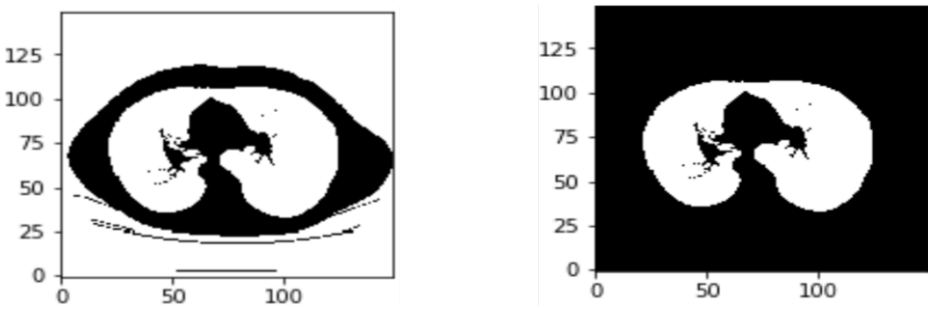

Fig. 4. a. Binary image; b. Image obtained after removing border blobs.

Labelling Connected regions of integer array of the images were labelled, as given in Fig. 5a, using skimage.measure label function. Two pixels are connected when they are neighbors and have the same value. In 2-D, they can be neighbors either in a 1 or 2-connectivity sense.

Labels with 2 largest areas were kept i.e. both lungs and components which had area less than lungs were removed as shown in Fig. 5b.

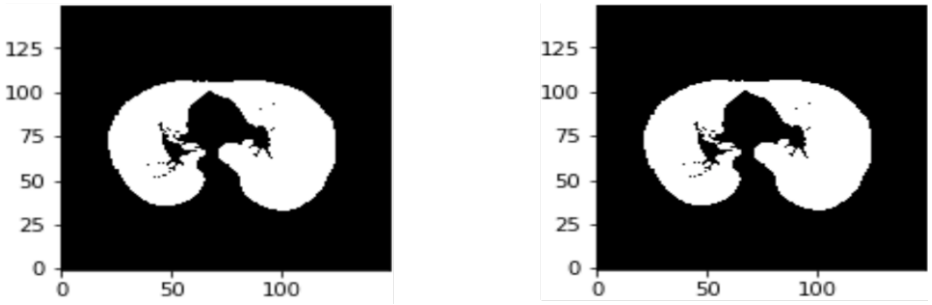

Fig. 5. a. Labelled image; b. Labelled image keeping 2 largest areas.

Erosion Operation Erosion operation (with a disk of radius 2) was performed to separate the lung nodule attached to the blood vessels. It was done using binary_erosion() which sets a pixel at (i,j) to the minimum over all pixels in the neighborhood centered at (i,j). Erosion shrunk bright regions and enlarged dark regions as given in Fig. 6a.

Closure Operation Closure operation (with a disk of radius 10) [6] was performed to keep nodules attached to the lung wall. It was done using binary_closing() which can remove small dark spots (i.e. pepper) and connect small bright cracks. This tended to close up (dark) gaps between (bright) features as shown in Fig. 6b.

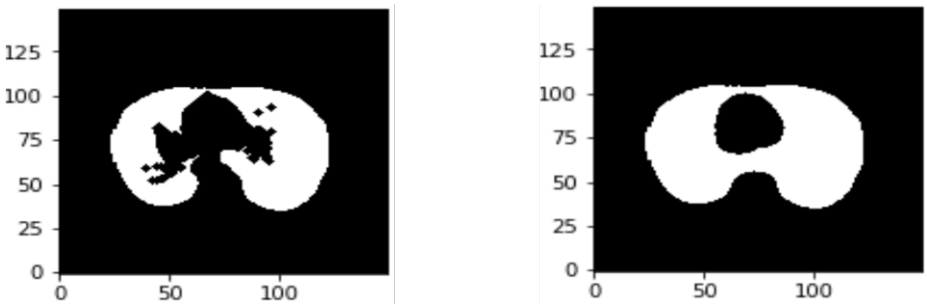

Fig. 6. Image after: a. erosion operation; b. closure operation.

Filling in the small holes inside binary mask Sometimes due to imperfections in the binary conversion identified by the optimal thresholding, a set of background regions (black pixels), lying completely within the foreground regions or region of interest (white pixels), are formed in binary image . This regions known as holes might be useful. To take this region into account, binary_fill_holes() function was used to fill them and image as shown in Fig. 7a was obtained.

Superimposing The last step in the segmentation involved superimposing the binary mask generated on the input image to obtain the region of interest (see Fig. 7b) which can be fed as an input to a CNN model.

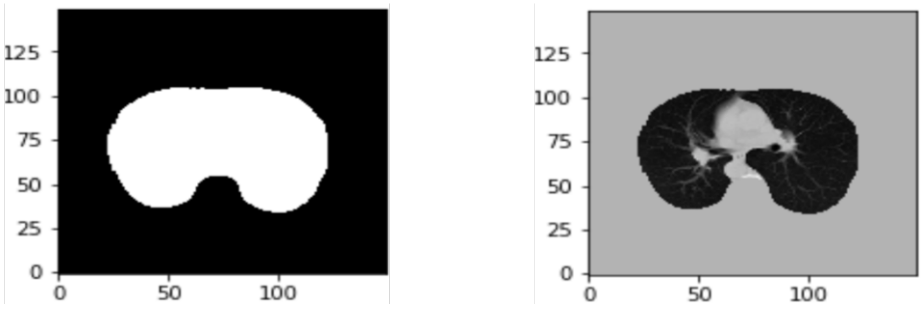

Fig. 7. Image after: a. filling small holes; b. superimposing on the input image

3.2 Lung Volume Extraction

As additional features the total left and right lung volumes were quantified. This was done using image semantic segmentation. Semantic segmentation is a process of classifying each pixel of an image to an appropriate class. This was carried out using the Deeplab v3 [7], which in-turn is based on Googles MIT licensed Tensorflow library within a framework of deep convolutional networks. Deeplab v3 has won several image segmentation competition including PASCAL VOC 2012. [7] Two innovations in particular have contributed immensely on improving the overall accuracy by Deeplab. One, is a layer of conditional random fields on top of CNN that aids in smoothing the predictions. Second, is atrous convolutions which are novel dilated convolution steps that allow to capture more context around an object without increasing the input data to the CNN model. [8]

CT images from the training data (150 training, 50 validation) were used as input (all slices included) along with the masks provided. Pixels labels for left lung, right lung and background was extracted from the mask images. CT image along with the masks were converted into Tensorflow format (tfrecords) using Deeplab v3 code. One innovation that reduced the training time significantly was transfer learning which was supported by Deeplab v3 code. A pretrained model (named Xception, trained by Google for their winning algorithm for PASCAL VOC 2012 competition [9]) was used, where the last 2 layers was removed and retrained using CT scan image data along with the masks. Training was carried out for 200,000 iterations and model was evaluated every 10,000 iterations. The final model gave a mean intersection-over-union of 0.879 for the validation data. The model generated, was used to predict one of the 3 classes for every pixel, furthermore total volume of left and right lung was estimated by considering the total number of pixels belonging to each class across all slices for an individual.

3.3 Model Flow

Post the pre-processing and noise removal of the 2-D images of the patients, we obtained 20 slices per patient. Each patient was now represented by a sequence of 20, 150*150 pixel images and a vector with 12 features obtained from the provided patient attribute metadata and additional lung volume features that was generated using the image mask data along with the CT images.

All the pre-processing steps were performed on the 218 patients train data and it was split in the ratio of 70:30 to create the training and validation sets.

Deep Learning Architecture A 3-D CNN model was used to process the sequence of images of the patients. The model was implemented using Keras version 2.2.4 with the support of GPUs, which helped reduce the training time of the neural network significantly. The 3-D CNN model training ran on an Ubuntu server machine equipped with 2 NVIDIA Tesla P4 GPU accelerator. The GPUs were accessed using Google Cloud Platform.

The sequence of images was passed on to the neural network as a tensor of shape (number of images per batch, 150, 150, 20, 1). The tensor was passed through a sequence of Conv3D, MaxPooling3D, BatchNormalization, Dropout and dense layers to finally produce a 6×1 vector for each of the patients in the training set. The 6×1 vector contained confidence scores for the 6 anomalies that are being detected. The neural network was trained to minimise the sum of loss across all the classes.

The model did not perform well in terms of accuracy and was generating same confidence score for all patients in the validation set for multiple classes. This indicated that the training data was not sufficient to train a neural network with 12 layers and it was necessary to reduce the size of the model architecture. Additionally since the model was trained on minimizing the sum of losses of 6 classes, it was observed that the number of epochs required and hyperparameters tuned for converging to global minima did not match with the individual convergence of 2 classes.

The second approach involved building a neural network with the same architecture but instead of training the neural network on minimizing the sum of losses across 6 classes, the network was trained on minimizing the loss of individual classes. The idea was to build 6 different neural networks for binary classification of the 6 classes with the same architecture but tuned with different set of hyperparameters. This ensured that all the classes were trained until they converged. The overall sum of losses was better than the previous model. However, this model predicted the confidence scores of all the patients in the test set very close to each other and the scores were always within a limited range. This again suggested that there was insufficient training data to train this model.

Finally, the size of the neural network was reduced to 8 layers and trained to minimize the sum of losses across all classes. The model performed better than the previous two models and the confidence score had significant variance across patients in the validation set. The model was fed with mini batches of data containing the 20 slices of images. The model was trained for 100 epochs with batch size fixed at 15. The train and validation sets converged to a cross entropy loss of 3.05 and 3.32 respectively within 100 epochs.We used SGD (Stochastic Gradient Descent) as the optimizer with a learning rate of 10^{-4} and decay rate of 10^{-6}. Gradient Descent is an optimization algorithm, based on a convex function, that tweaks its parameters iteratively to minimize a given function to its local minimum. Stochastic Gradient updates the parameters for each training example, one by one. This makes SGD faster than other widely used optimizers like Batch Gradient Descent. We also tried using the mask data given to us instead of creating it from the input data image through first 2 steps of segmentation mentioned in section 3.1.3 above. The idea was to leverage the mask data provided to us keeping the same final approach as mentioned above and to have a comparative analysis of the results obtained through the 2 approaches before finalising the best approach.

The deep learning model had several hyperparameters and multiple iterations were run to tune the parameters. The activation function used was *tanh* as it

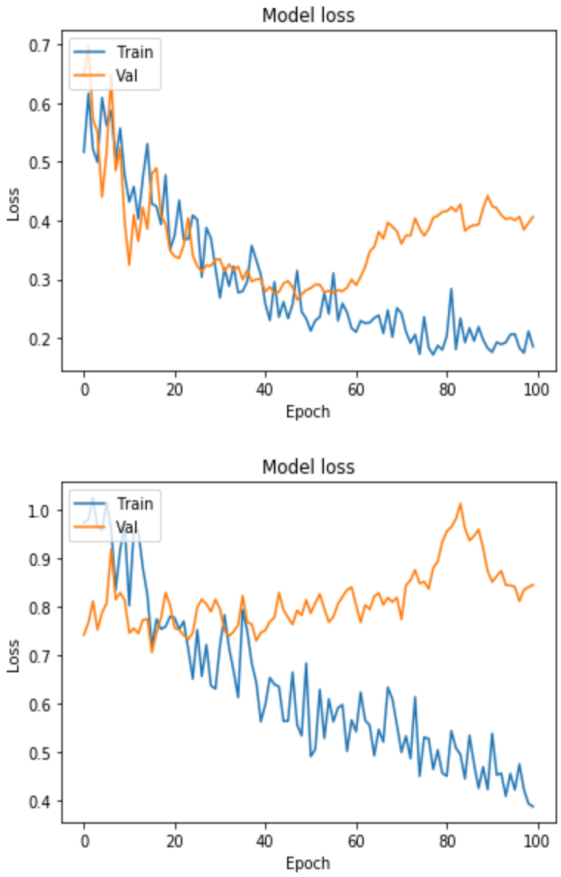

Fig. 8. Graphs showing the convergence of the training and validation sets for the 2 classes (Pleurisy and Caverns) which did not converge with the global minima.

usually performs well for a binary classification. The kernel and pool size was kept as (8,8,8); relatively larger sizes were used as the data needed to shrink before passing on to the dense layer. The stride size was fixed at (4,4,4) for the same reason. SGD with learning rate e^{-04} and decay rate e^{-06} was used. To compensate the low learning rate values, the model was trained on 100 epochs. Multiple combination of the number of neurons were tried in the 2 dense layers (dense_1 and dense_2) and the numbers were fixed at 256 for dense_1 layer and 16 for dense_2 layer. The dropout ratio was tuned at 0.6 to ensure the model does not overfit.

Machine Learning Classifier The deep learning model was trained on the 180 patients in the training dataset and the parameters were tuned. The penultimate

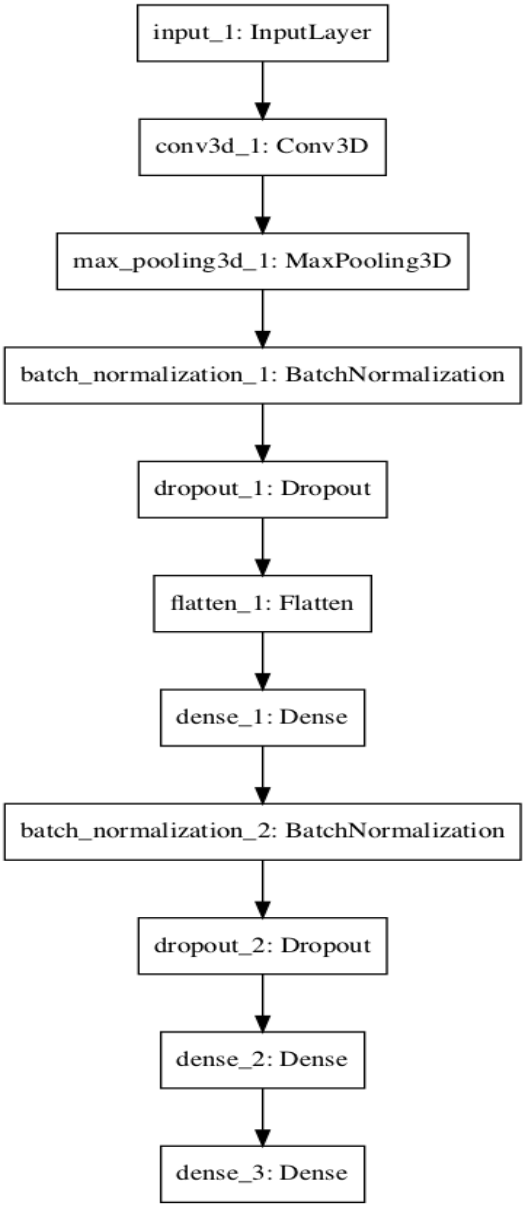

Fig. 9. Neural network architecture of single binary classification.

dense layer was a vector of shape (16,). This vector was appended with the metadata feature vector of shape (12,) and the resultant vector was trained on a couple of Machine Learning Classifiers.

Fig. 10. Graph showing the training and validation convergence of the model trained on minimizing the loss across all classes.

The deep learning model was trained to minimize the sum of losses across classes and the deep learning model itself classified the patients across classes. Using the ML classifiers helped us to take into consideration both the metadata and the 16 element vector which was generated as the penultimate layer of the neural network. The 16 element vector carried the features from the model which was trained to minimize the sum of losses. The features from the deep learning model were leveraged and appended with the metadata to input the resultant vector to a machine learning classifier.

2 machine learning classifiers: Support Vector Classifier and Random Forest Classifier were tried. Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. GridSearchCV was used to tune the parameters of the model - the regularization parameter C and Gamma parameter. The regularization parameter was used to specify the extent to which misclassifying each training example had to be avoided. The gamma parameter defines how far the influence of a single training example reaches.

Random forest is an ensemble algorithm. Ensemble algorithms combine more than one algorithm of same or different kind for classifying. Random forest classifier creates a set of decision trees from randomly selected subset of training set. GridSearchCV was used to tune the parameters of the model- n_estimators, max_depth and max_features. n_estimators is the number of decision trees considered for making the decision, max_depth is the maximum number of levels in each decision tree and max_features is the maximum number of features considered for splitting a node. Random forest performed relatively better on the data as compared to SVC.

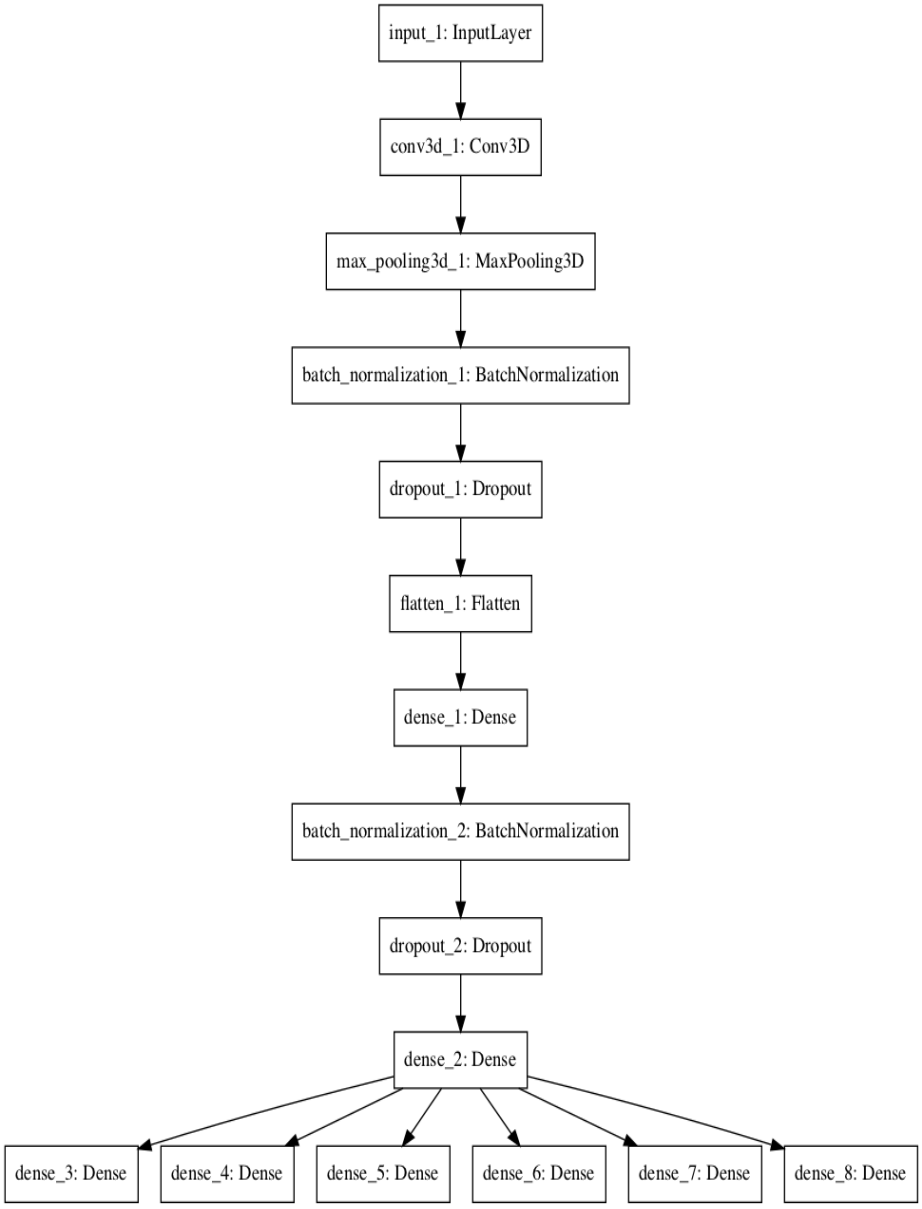

Fig. 11. Neural network architecture of the model which does multi label binary classification.

4 Results

AUC values from the different approaches that was tried are provided below. The values reported here are on the validation set of 32 patients.

Table 2. Model performance from various approaches on validation set

Methods	Accuracy	Mean AUC	Min AUC
8 layer multi label binary classification	0.74	0.634	0.51
6 separate NN for binary classification	0.72	0.58	0.44
12 layer multi label binary classification	0.68	0.59	0.42
Multi label classification using mask data	0.73	0.61	0.36

8 layer multi label classification was the final model that was used for the task. That model performed the best on the validation set with a mean AUC of 0.634. The next best model in terms of mean AUC was the multi label classification in which the mask data was leveraged instead of segmenting the original image. One of the drawbacks of that model was that it performed very poorly on classifying patients for the Caverns class. The 6 individual neural net models and the 12 layer multi label classification model both produced similar metrics on the validation set, giving out a mean AUC of 0.58 and 0.59 respectively.

The results from the 2 submitted runs on the test set are provided below:

Table 3. Model performance from submitted runs on test set

Methods	Mean AUC	Min AUC
8 layer multi label binary classification	0.6	0.472
6 separate neural networks for binary classification	0.554	0.427

5 Results Analysis

3 patients (Patient 117,140 and 190) from the validation set, with greater lung defects were analysed separately to test the predictions from the model and compare it with the actual flags present in the metadata for 6 classes. Table below lists down the model prediction probabilities and actual flag in metadata for these 3 patients. It can be inferred that the developed model performs well in classifying first 5 classes but caverns prediction can still be improved. Fig. 11 shows image slices for each of these patients with the defects.

6 Conclusion and Future Work

Due to time and computation resource limitations during the course of the challenge, a lot of possible enhancements to the existing model could not be explored. Some of the enhancements to the pre-processing stage that can be considered as immediate next steps are leveraging the mask image data to get better segmentation and classification output, finding lung nodule candidates and regions of interest from the segmented lungs, augmenting data to take care of class imbalance present in some categories, adding annotations to the data slices, and

Table 4. Comparison of predicted probability with actual class

	Patient No. 117		Patient No. 140		Patient No. 199	
Deformity Class	Predicted	Actual	Predicted	Actual	Predicted	Actual
Left Lung Affected	0.94	1	0.95	1	0.85	1
Right Lung Affected	0.75	1	0.88	1	0.96	1
Lung Capacity Decrease	0.47	0	0.63	1	0.04	0
Calcification	0.13	0	0.12	0	0.1	0
Pleurisy	0.02	0	0.05	0	0.03	0
Caverns	0.79	0	0.86	1	0.68	0

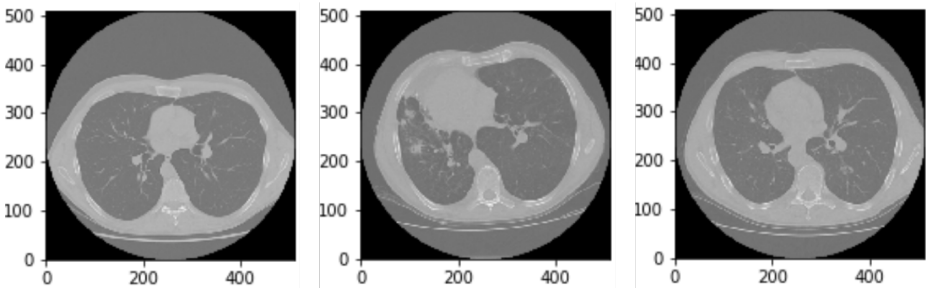

Fig. 12. Image slices of patients: a. 117; b. 140; c. 199.

considering weights while aggregating slices instead of a simple average. Exhaustive hyperparameter training that could not be explored due to computing limitations, could be explored further.

Based on recent literature reviews on image analysis using deep learning and looking at multiple submissions to the challenge using deep learning algorithms, it shows that deep learning holds great promise. Exploring other deep learning architectures and evaluating the impact on the model results remains to be seen.

References

1. Yashin Dicente Cid, Vitali Liauchuk, Dzmitri Klimuk, Aleh Tarasau, Vassili Kovalev, Henning Müller: Overview of ImageCLEFtuberculosis 2019 - Automatic CT-based Report Generation and Tuberculosis Severity Assessment,CLEF 2019 Working Notes. CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073, http://ceur-ws.org/Vol-2380/
2. Bogdan Ionescu, Henning Müller, Renaud Péteri, Yashin Dicente Cid, Vitali Liauchuk, Vassili Kovalev, Dzmitri Klimuk, Aleh Tarasau, Asma Ben Abacha, Sadid A. Hasan, Vivek Datla, Joey Liu, Dina Demner-Fushman, Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Minh-Triet Tran, Mathias Lux, Cathal Gurrin, Obioma Pelka, Christoph M. Friedrich, Alba García Seco de Herrera, Narciso Garcia, Ergina Kavallieratou, Carlos Roberto del Blanco, Carlos Cuevas Rodríguez, Nikos Vasillopoulos, Konstantinos Karampidis, Jon Chamberlain, Adrian Clark, Antonio

Campello: ImageCLEF 2019: Multimedia Retrieval in Medicine, Lifelogging, Security and Nature In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), Lugano, Switzerland, LNCS Lecture Notes in Computer Science, Springer (September 9-12 2019), Vol 2380.
3. Yashin Dicente Cid, Oscar A. Jiménez-del-Toro, Adrien Depeursinge, and Henning Müller: Efficient and fully automatic segmentation of the lungs in CT volumes. In: Goksel, O., et al. (eds.) Proceedings of the VISCERAL Challenge at ISBI. No. 1390 in CEUR Workshop Proceedings (Apr 2015).
4. Kaggle Data Science Bowl Article, http://www.kaggle.com/sentdex/first-pass-through-data-w-3d-convnet. Last accessed 26 May 2019
5. Sasidhar B, Ramesh Babu D R, Ravi Shankar M, Bhaskar Rao N: Automated Segmentation of Lung Regions using Morphological Operators in CT scan. Proceedings of the International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September 2013 1114 ISSN 2229-5518.
6. Eng. Michael Samir Labib Habib: A Computer Aided Diagnosis (CAD) System for the detection of pulmonary nodules on CT scans. Systems and Biomedical Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt, 2009.
7. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume:40, Issue:4, Apr 2018).
8. Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam: Rethinking Atrous Convolution for Semantic Image Segmentation. https://arxiv.org/abs/1706.05587.
9. Francois Chollet: Xception: Deep Learning with Depthwise Separable Convolutions. Google, Inc.