

Automated Classification of Lung Tuberculosis Using 3D Deep Convolutional Neural Networks

Sushaanth. Srinivasan¹, Sharvesh. Shankar¹, Nitheesh Kumar. N¹, Sabarivasan. Velayutham¹, Thejas. N¹, Vikash Anand. N¹, Lekshmi. Kalinathan¹ and Prabavathy. Balasundaram¹

¹Department of CSE, SSN College of Engineering, Rajiv Gandhi Salai, Chennai, Tamil Nadu, India

Abstract

Automated TB and disease classification is a dire need in these times, as traditional diagnostic procedures are inefficient. Existing literature is focused on TB identification and classification using 2D images. As 3D images contain extra depth information which helps in more accurate modelling of the disease cavern, an approach using the 3D-CNN model has been proposed to classify the type of caverns present in lung CT scans in order to ensure prompt treatment.

Keywords

Tuberculosis, Deep Learning, 3D CNN Classification, Cavern

1. Introduction

Tuberculosis (TB) is a bacterial infection caused by Mycobacterium Tuberculosis. It usually affects the lungs and can then spread to other parts of the body such as the brain and the spine. TB cavern has three classes which represent its property of having *thick walls*, *foci around* and the *presence of calcification*.

The traditional diagnostic procedures include skin tests, blood tests, imaging modalities, sputum testing and culture test. The results of a sputum smear requires several days, while the results of a culture needs several weeks. This reduces the diagnostic efficiency and frequently delays the isolation of infectious individuals. These tests have a low sensitivity as well. TB diagnosis, especially in smear-negative patients, can be extremely difficult.

If TB is not diagnosed properly, due to its highly transmittable nature, it spreads from one person to the next through the air. Hence, it is a dangerous disease, and if not treated rightly, can be fatal. According to the World Health Organization, a total of 1.6 million people died in the year of 2020 due to Tuberculosis.

Existing research presented in Table 1 has been applied on 2D images. However, 3D images provide better insights when compared to 2D images. The training process for 3D images would be time consuming when compared to the training process on 2D images. Hence, the inference

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

✉ sushaanth19113@cse.ssn.edu.in (Sushaanth. Srinivasan); sharvesh19101@cse.ssn.edu.in (Sharvesh. Shankar); nitheesh2010343@ssn.edu.in (N. Kumar. N); sabarivasan2010624@ssn.edu.in (Sabarivasan. Velayutham); thejas2010679@ssn.edu.in (Thejas. N); vikashanand2010015@ssn.edu.in (V. Anand. N); lekshmik@ssn.edu.in (Lekshmi. Kalinathan); prabavathyb@ssn.edu.in (Prabavathy. Balasundaram)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).


 CEUR Workshop Proceedings (CEUR-WS.org)

Table 1

Existing work along with their methodology and performance

Existing work	Methodology used	Performance
Automated TB classification using ensemble of deep architectures. Multimedia Tools and Applications [3]	Ensemble classifier using AlexNet, GoogleNet and ResNet to classify 2D CXR images	Accuracy - 88.24% Area under Curve - 0.93
Simple Neural Network based TB Classification. [4]	A simple shallow neural network is employed with three layers to classify 3D CT-images	Validation Accuracy- 20% Testing Accuracy - 22.1%
Tuberculosis (TB) detection system using deep neural networks. Neural Computing and Applications [5]	Model learns from the pre-trained weights of Inception V3 and classifies the data using support vector machine (SVM) from the transferred knowledge	Accuracy - 95.05%
Application of a convolutional neural network using transfer learning for tuberculosis detection. [6]	ConvNet model that uses VGG16 to classify 2D CXR images	Accuracy - 80% without applying augmentation, Accuracy - 81.25% with application of augmentation
A deep learning approach for the classification of TB from NIH CXR dataset. [7]	A custom-built CNN architecture to classify 2D CXR images	Accuracy - 92.5%
Analysis of Tuberculosis (TB) on X-ray Image Using SURF Feature Extraction and the K-Nearest Neighbor (KNN) Classification Method.[8]	SURF Feature Extraction and the K-Nearest Neighbor (KNN) Classification to classify 2D X-ray image	Average Accuracy - 73%
Diagnosing tuberculosis using deep convolutional neural network. [9]	Deep Convolutional Neural Network (CNN) to classify 2D CXR images	Validation Accuracy - 87.1%

can be made that a model using a 3D dataset is more reliable. Therefore it is crucial to come up with new deep learning solutions that can detect TB based on 3D images and provide higher accuracy.

2. Task and Dataset Description

The dataset used for this task is from ImageCLEF 2022[1]. The goal of the *Caverns Report task* [2] is to predict 3 binary features of caverns namely, *Has Thick Walls*, *Has Foci Around* and *Has Calcification*.

A single 3D image is provided for each of the 60 patients. Each 3D image contains around 100 slices of 2D images of 512x512 pixels. All the CT images are stored in NIFTI file format with .nii.gz file extension (g-zipped .nii files). This file format stores raw voxel intensities in Hounsfield units (HU) as well the corresponding image metadata such as image dimensions, voxel size in physical units and slice thickness.

Two versions of automatically extracted masks of the lungs were provided for each CT image. This data is available along with the CT images of the patients. The first version of segmentation is able to provide masks in an accurate manner, but tends to miss features in severe TB cases where large abnormal regions of lungs are present. On the contrary, the second segmentation provides rough bounds, but includes lesion areas.

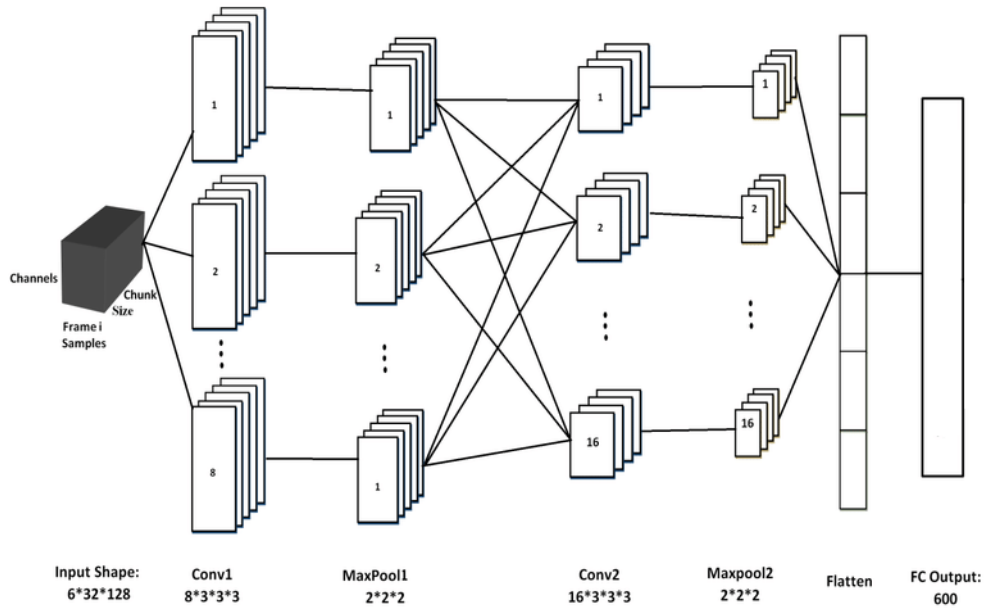


Figure 1: 3D-CNN Architecture

3. Techniques Used

The 3D-CNN architecture is frequently used for a stack of 2D images, particularly for medical images, as it can assess the positions of defects in the time domain. During the convolution step, the 3D-CNN generates a 3D activation map. This is required for time and volumetric context. To calculate the representation of elements at a low level, a three-dimensional filter is employed for the 3D convolution of the dataset.

Convolution and pooling layers connect only to local regions around each input in the CNN, which is a version of the classic neural network. CNNs are sparsely connected to hierarchical representation of the input, allowing them to process images from general forms to edge details. The 3D-CNN is a development of 2D-CNNs that captures discriminating features in both the spatial and temporal dimensions by dividing hierarchical 3D visual information into small cubes rather than 2D patches.

Figure 1 illustrates the notion of a 3D convolution. The input 3D image is split into 2D slices. The process of 3D convolution begins with the 2D slices, with x_{ij} denoting the j^{th} receptive field on the i^{th} slice. Local receptive fields are constructed on 2D slices to generate 2D features y_{ij} , which are convolved along the temporal direction i to generate feature vector y_j at the convolution layer. A 3D pooling layer follows the same steps as a 2D pooling layer. However, it takes the maximum or average values at each step. The CNN is ready for hierarchical learning from motifs to edges after repeated convolution and pooling operations until the picture information is adequately compressed.

The sequences of signals will be fed into the final hidden layer, where a fully connected neural network will learn the image's attributes and generate the output as data sequences.

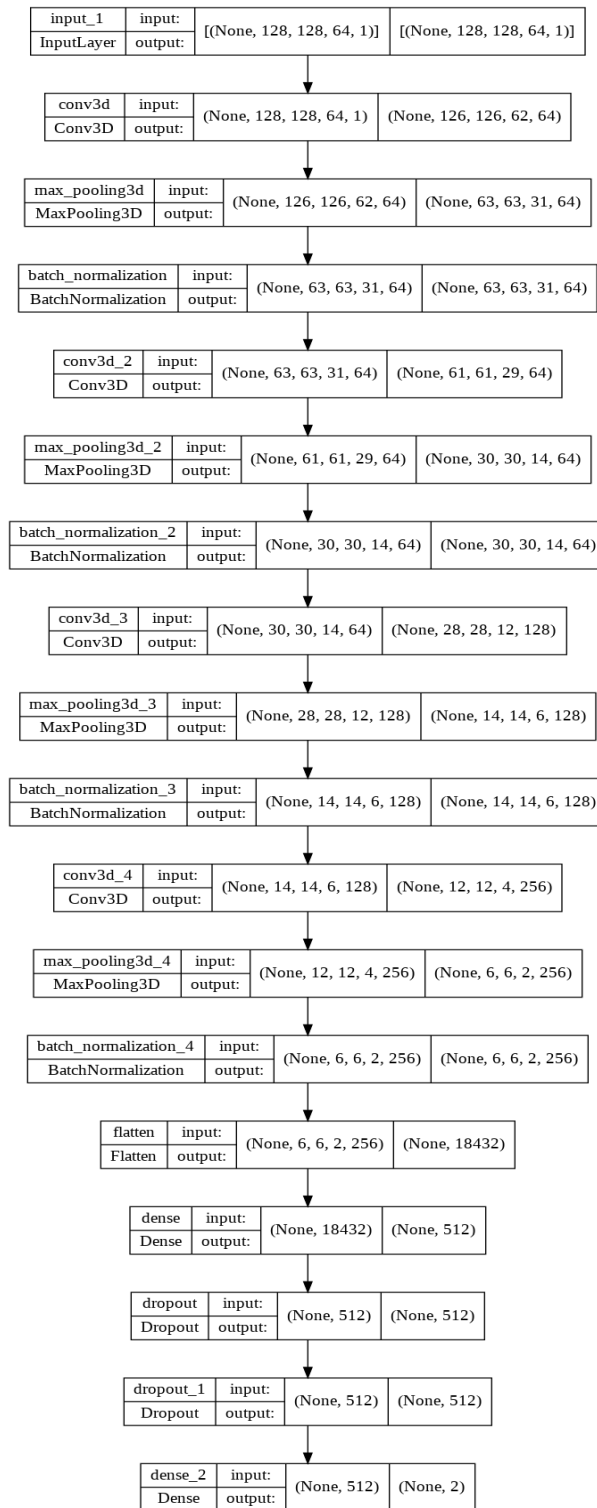


Figure 2: Proposed 3D-CNN Architecture

4. Implementation

The model was trained on an Intel i7 core CPU, Samsung 1TB SSD and an Nvidia GeForce GTX 2060 Super GPU System. Google Colaboratory was also used which uses a 12GB Nvidia Tesla K80 GPU. The Deep Learning framework Tensorflow was used along with tools such as Anaconda, Jupyter Notebook Environment, CUDA, Numpy, Pandas and Sci-Kit Learn.

A TB classification system was built which has been implemented as multiple binary classifiers for each of the three classes; namely *Has thick walls*, *Has foci around*, and *Has calcification*. The procedure for the implementation of the system is given as follows:

- The labels were separated by class in order to implement multiple binary classifiers for the three classes. The separated labels were stored in a Numpy array.
- The given dataset for the TB Task consists of 3D images. Each 3D image was split into a set of 2D image slices. Each slice was resized to 128x128 pixels using Spline Interpolated Zoom and Inter Cubic Interpolation.
- The training images were loaded into Numpy arrays in order to train the model.
- The training data was normalized as it calibrates the different pixel intensities into a normal distribution which makes computation efficient and helps the model converge faster.
- The normalized data was split into training and testing sets.
- During the learning process, the learning rate is automatically reduced if the loss remains the same for a predefined number of epochs. This callback monitors the validation loss and helps prevent stagnation.
- If the model shows no improvement in the reduction of validation loss after a predefined number of epochs even after reduction of the learning rate, the training is automatically stopped.
- The architecture of the model is as shown in Figure 2.
- K-fold cross validation was performed to estimate the prediction of the model on unseen data and is a preventive measure for overfitting. This ensures that the proportion of the feature of interest is the same across the original data, training and the test set, which gives a more accurate estimate of the performance of the model.
- After training, the model was made to predict values for the test data.

5. Result and Analysis

The model was trained on a small scale dataset of five patients and then upscaled to sixty patient records. The model has attained an accuracy of 75% on training data and 60% on validation data. The area under the ROC curve for the validation data is 0.78 for the class *Has Calcification*, 0.74 for the class *Has Foci Around*, and 0.65 for the class *Has Thick Walls*.

For the classes *Has Foci Around* and *Has Calcification*, the most of the ROC curve is present towards the left of the line denoting the random classifier but it intersects the line. The validation accuracy is very less compared to the training accuracy and the validation loss is much greater

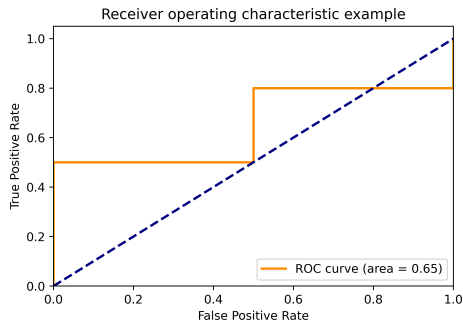
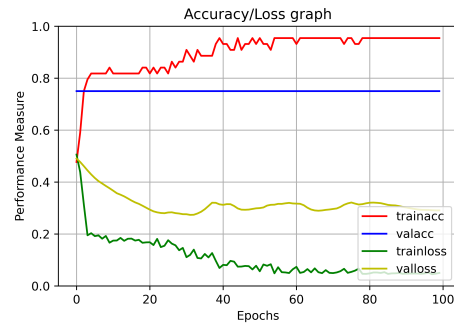


Figure 3: (a) roc-thickwalls (validation data)



(b) trainval-acc-loss-thickwalls

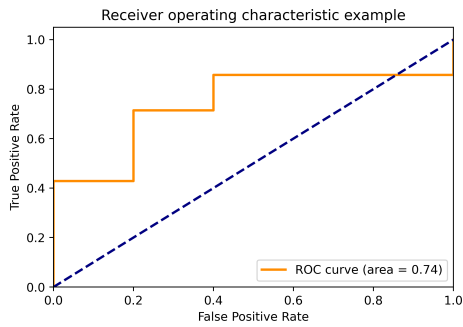
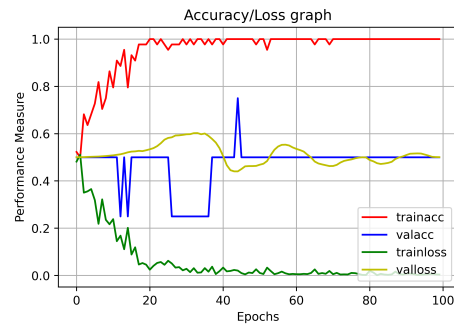


Figure 4: (a) roc-foci around (validation data)



(b) trainval-acc-loss-foci around

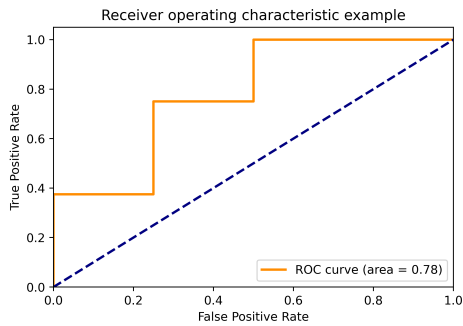
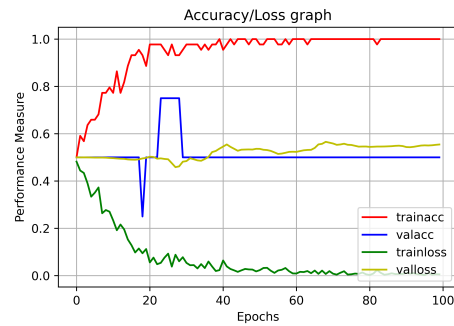


Figure 5: (a) roc-calcification (validation data)



(b) trainval-acc-loss-calcification

than the training loss. Therefore it can be inferred from the figures 3, 4 and 5 that the model might be overfitting the data. This is due to the imbalance between the positive and negative binary classes present in the separated datasets for the classes *Has Foci Around* and *Has Calcification*. However, for the class *Has Thick Walls*, the ROC curve is present towards the left side of the line denoting the random classifier as well as does not intersect it. The training and validation accuracy plot lines are close to each other and the same is true for loss values as well. This

shows that the model has correctly fitted the data. Thus, it can be inferred from this graph that the model performs well on this class as the positive and negative samples in the dataset for this class are balanced.

The following table describes the probability of the three classes predicted by the model.

Table 2

Probability of having Thick Walls, Foci Around and Calcification for first 15 patients

Image	Thickwalls	Foci Around	Calcification
TST_00	0.9978415	0.8283372	0.9964127
TST_01	0.99999547	0.34543616	0.83471966
TST_02	0.999368	0.69200265	0.9884094
TST_03	0.9999981	0.8152172	0.9991033
TST_04	0.999992	0.38898218	0.9299915
TST_05	0.9999901	0.4413950	0.8158201
TST_06	0.9999924	0.73378474	0.9003653
TST_07	0.99952185	0.72178733	0.6093098
TST_08	0.9999995	0.30586368	0.9996055
TST_09	0.9991689	0.98074776	0.9726256
TST_10	0.99997044	0.9738817	0.96510625
TST_11	0.929831	0.5132923	0.99828976
TST_12	0.9966671	0.99855965	0.995698
TST_13	0.9999987	0.839497	0.915082
TST_14	0.99993134	0.93050194	0.79498357
TST_15	0.9987043	0.7850096	0.8628713

6. Conclusions

In summary, a Deep 3D Convolutional Neural Network model structure has been proposed to accurately predict the classification of the cavern present in CT Scans of Tuberculosis in the Lung region. In order to increase the predictions of the model, the parameters and the hyperparameters of the 3D-CNN model were tuned. The optimal model resulted in a training accuracy of 75%. The model was tested for the test data of 16 patients and achieved a mean AUC of 0.536.

Acknowledgments

We thank the CSE department of SSN College of Engineering for letting us utilize the GPU machine extensively to implement this task.

References

- [1] B. Ionescu, H. Müller, R. Peteri, J. Rückert, A. Ben Abacha, A. G. S. de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, S. Kozlovski, Y. D. Cid, V. Kovalev, L.-D. Ştefan, M. G. Constantin, M. Dogariu, A. Popescu, J. Deshayes-Chossart,

- H. Schindler, J. Chamberlain, A. Campello, A. Clark, Overview of the ImageCLEF 2022: Multimedia retrieval in medical, social media and nature applications, in: Experimental IR Meets Multi-linguality, Multimodality, and Interaction, Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022), LNCS Lecture Notes in Computer Science, Springer, Bologna, Italy, 2022.
- [2] S. Kozlovski, Y. Dicente Cid, V. Kovalev, H. Müller, Overview of ImageCLEF Tuberculosis 2022 - CT-based caverns detection and report, in: CLEF2022 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org <<http://ceur-ws.org>>, Bologna, Italy, 2022.
- [3] H. Rahul, A. Mittal, S. Sofat, Automated TB classification using ensemble of deep architectures, in: Multimedia Tools and Applications Journal. 2019, 78(22): 31515-31532.
- [4] A. Anand, K.R. Anandan, B. Jayaraman, M.T. Thai, Simple Neural Network based TB Classification 2021.
- [5] R. Dinesh Jackson Samuel, B. Rajesh Kanna, Tuberculosis (TB) detection system using deep neural networks, in: Neural Computing and Applications Journal. 2019, 31(5):1533-1545.
- [6] M. Ahsan, R. Gomes, A. Denton, Application of a convolutional neural network using transfer learning for tuberculosis detection, in: IEEE International Conference on Electro Information Technology (EIT). 2019, pp. 427-433.
- [7] S.Z.Y. Zaidi, M.U. Akram, A.Jameel, N.S.Alghamdi, A deep learning approach for the classification of TB from NIH CXR dataset, in: IET Image Processing. 2022, 16(3):787-796.
- [8] R.A. Rizal, N.O. Purba, L.A. Siregar, K. Sinaga, N. Azizah, Analysis of Tuberculosis (TB) on X-ray Image Using SURF Feature Extraction and the K-Nearest Neighbor (KNN) Classification Method, in: Journal of Applied Information and Communication Technologies (JAICT). 2020, 5(2):9-12.
- [9] M. Oloko-Oba, S. Viriri, Diagnosing tuberculosis using deep convolutional neural network, in: Proceedings of International Conference on Image and Signal Processing. 2020, pp. 151-161.