Estimating Severity from CT Scans of Tuberculosis Patients using 3D Convolutional Nets and Slice Selection

Hasib Zunair[1], Aimon Rahman[2], and Nabeel Mohammed[3]

North South University, Dhaka 1229, Bangladesh
hasibzunair@gmail.com[1], irsnigdha@gmail.com[2],
nabeel.mohammed@northsouth.edu[3]

Abstract. In this work, we propose a 16-layer 3D convolutional neural network with a slice selection technique employed in the task of 3D Computed Tomography Image data of Tuberculosis(TB) patients which attained 10 th place in the ImageCLEF 2019 Tuberculosis - Severity scoring challenge. The goal is aimed at estimating TB severity based on the CT image. The best result reported in this work is Area Under the ROC Curve (AUC) of 0.61 and a binary accuracy of 61.5%. Codes for this work can be found at URL.[1]

Keywords: Deep learning · Convolutional Neural Networks · Image Analysis· Computed Tomography.

1 Introduction

Tuberculosis is a potentially serious infectious disease that affects lungs, and sometimes other parts of the body. Mycobacterium tuberculosis, is the bacteria that causes the infection and can spread from one person to another via cough, spit or sneezes. Most infections do not cause any symptoms, which is refered as latent tuberculosis and 10% of them turns to potentially fatal active diseases. People with HIV/AIDS and smokers are more vulnerable to active TB. The symptoms of the infections are cough with blood-containing mucus, night sweats, fever, chills, loss of appetite and severe weight-loss. About one-quarter of the world population has been infected with the disease. In 2010, 1.2 -1.45 million deaths have occurred due to the disease, mostly in developing countries which makes it the second most common cause of death from infectious disease [1]. Tuberculosis is diagnosed by conducting chest X-ray and microscopic examination of bodily fluids. Computed tomography (CT) scan provides

[1] https://github.com/hasibzunair/tuberculosis-severity

more detailed information about the infection than X-ray images [2]. Several techniques have been used to automate the detection of Mycobacterium infection using different machine learning techniques in chest radiographs and CT scan images. Deep learning [14] has also been used to diagnose lung diseases in CT scan images. [4] used deep belief network and convolutional neural network to classify lung nodule in 3D tomography images, where deep beleif network performed better than convolutional neural network and SIFT. [3] shows an ensemble of AlexNet and GoogleNet DCNNs performed best with the AUC score of 0.99 on radiograph images to detect pulmonary tuberculosis. Prognosis of diseases such as chronic obstructive pulmonary disease and acute respiratory disease on smokers lungs has also been predicted using convolutional neural network in computed tomography images [5]. Three different deep learning techniques have also been applied to detect lung cancer from CT scan images using convolutional neural networks(CNN), Deep neural networks(DNN) and Sparse Autoencoders(SAE), where CNN performed best with an accuracy of 84.15%, sensitivity of 83.96%, and specificity of 84.32% [6]. Most computer aided diagnosis has been done for tuberculosis which uses radiographic images [7]-[10]. A challenging part of working with CT scans is the fact that the data points comprises of depth information, which make it not only three dimensional but also computationally expensive to process for images with have large depth size. Several works have been done on detecting 3D objects, for example by integrating volumetric grid representation with a 3D convolutional neural network [11] introduced a network named VoxNet, which is then validated in LiDAR, RGBD, and CAD data. [12] on the other hand, uses 2D representation from various angles of a 3D object and trained a multi view convolutional neural network classifier with view pooling method which performed better than 3D CNN. They show the use of transfer learning [13] which is an effective technique where a model is used which previously learnt a set of features to solve a visual recognition task as a starting point to solve another task.

Training 3D Convolutional Nets on volumetric data is computationally expensive due to depth information which requires additional feature learning and hence results in an increased number of learnable paramters. Moreover, all the data samples do not have same depth size which complicates training. To address this problem, we introduce a novel data partitioning technique which makes the training method not only effective but most importantly feasible. We show, using partial depth information from the each volumetric data point, it is possible to achieve good AUC and accuracy values. We show our approach which achieves 10 th place among a total of 100 participants in the ImageCLEF 2019 Tuberculosis - Severity scoring challenge [16].

2 Experimental Overview

This section provides details of the setup used in the different experiments. A brief description of the data set used for the experimentation will be given,

followed by description of the network architecture. Moreover, details of the training regiment with the metrics are provided.

For replication of this work, it is relevant to mention that all of the experiments were performed on a machine with Windows system with Intel Core(TM) i7-7700 CPU @3.60GHz processor, 32 GB RAM, a single CUDA-enabled NVIDIA GTX 1050 4GB graphical processing unit (GPU), Python 3.6.7, Keras 2.2.4 with Tensorflow 1.12.0 backend, and CUDA compilation tools, release 10.0, V10.0.130 dependencies for GPU acceleration.

2.1 Dataset Description

The dataset for the severity scoring(SVR) task is provided by ImageCLEF Tuberculosis 2019 [15][16]. The dataset consists of a total 335 chest CT scans of Tuberculosis patients in addition with clinically relevant metadata. From the dataset, 218 data points are used for training and the remaining 117 are hold out for the final evaluation. The selected metadata includes the following binary measures: disability, relapse, symptoms of TB, comorbidity, bacillary, drug resistance, higher education, ex-prisoner, alcoholic, smoking. The 3D CT images which were provided have a slice size of 512×512 pixels and a number of slices varying from 50 to 400. All the CT images are stored in NIFTI file format. 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, slice thickness, etc. Figure 1 shows an instance of this.

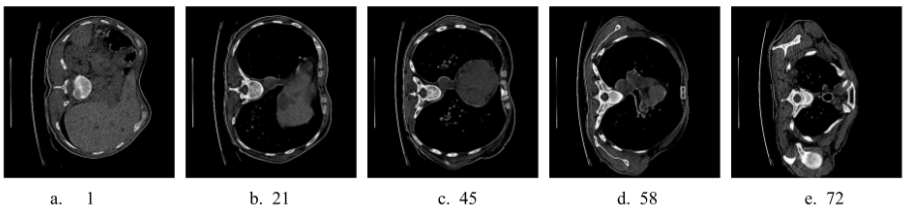


Fig. 1. Slices of 2D image from an arbitrary 3D CT image. The slices are started off from the lower chest progressing to upper chest.

The original severity score which is assigned by a medical doctor is included as training meta-data which are annotated in a scale from 1("critical/very bad") to 5("very good). This grade is converted to "LOW" (scores 4 and 5) and "HIGH" (scores 1, 2 and 3) thereby reducing the task to a binary classification problem as per task descriptions provided by ImageCLEF Tuberculosis 2019[15][16]. In addition to these requirements, the 117 test set labels are hidden, which make it more challenging.

2.2 Data Prerocessing - Slice Selection Technique

In order to prepare the data for training, we first resize individual slices of the 3D input volume to 128×128 with bicubic interpolation.

This is followed by a slice selection technique that we introduce in this work, where a depth size of 32 and 16 - which results in two experimental settings discussed in 1 - are chosen for our final submissions.

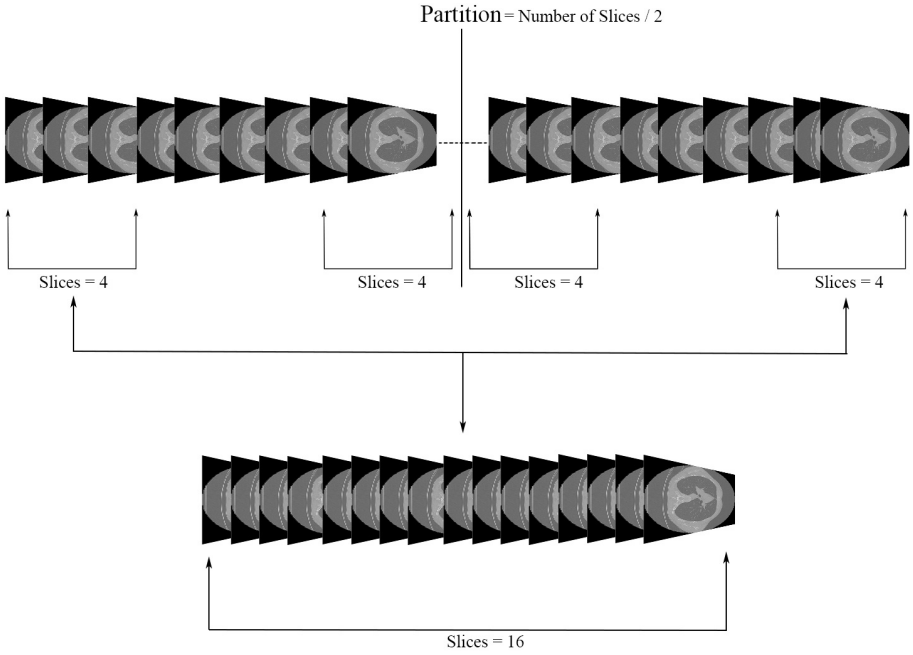


Fig. 2. Slice Selection Technique with a depth size of 16.

In Figure 2, we show a visual representation of the slice selection technique. For a given input volume the first 4 slices, the middle 8 slices and the last 4 slices are extracted. The middle slice of the input volume is obtained by taking the half of the input volume depth. These three sub components are then stacked to reconstruct the desired input volume where, in this case, is a depth size of 16. In other words, the input volume consists of a total of 16 slices. The main motivation behind the proposed technique eliminates the problem of GPU exhaustion during optimization. Since the default input volume consist of large number of slices, it was entire impossible to allocate tensors for computation in our experimental setup discussed at the end of Section 1.

2.3 Configuration of the proposed Convolutional Neural Network

The network used in this work was inspired by the architecture used for real time object recognition by integrating a volumetric occupancy grid representation with a supervised convolutional net named VoxNet [11]. All the 3D input volumes were transformed to $128 \times 128 \times$ *depth*, where *depth* is 32 and 16 for the two different settings, with cubic interpolation for the network input. The network consists of three convolutional layers with 32 filters of size $2 \times 2 \times 2$. After every convolutional layer, a maxpooling layer is added with a stride of 2. Maxpooling layers reduces its size of input to half by taking maximum values from a window of size 2×2. Recitified Linear Units(ReLu) was used as the activation function for both convolutional and fully connected layers. The activation function is governed by Equation 1:

$$a = max(0, x) \tag{1}$$

where, a is the output of the activation for a given input x.

The convolutional blocks in Figure 3a is then followed by a batch normalization layer. The deep learning community has quickly adopted the use of batch normalization as it introduces a form of regularization which restrains the network form simply memorizing the training dataset, which means the network is expected to generalize better on unseen data with use of batch normalization. The output from the batch normalization layer is flattened and passed to a series of fully connected layers with two dense layers having 1028 neurons, one with 512 neurons. Each of the dense layers were connected to a dropout layer which drops the neuron connection with a probability of 40%. The output from the final two dropout layers was followed by a dense layer of 2 neurons. The network architecture is shown in Figure 3.

Softmax activation shown in equation 2 was applied on the last layer to get the probability results for the binary classification problem. The output of the softmax function is equivalent to a categorical probability distribution, it tells you the probability that any of the classes are true. This enables better performance of the model.

$$\sigma(z_j) = \frac{e^{(z_j)}}{\sum_{k=1}^{K} e^{(z_k)}} \, . \tag{2}$$

where z is a vector of the inputs to the output layer (if you have 10 output units, then there are 10 elements in z). And again, j indexes the output units, so $j = 1, 2, ..., K$.

2.4 Training Regiment

Stochastic Gradient descent was used to optimize the weights of the network via backpropagation with a learning rate of 10^{-4} with a momentum of 0.9. Cross-entropy error between the predicted and ground truth was used as the

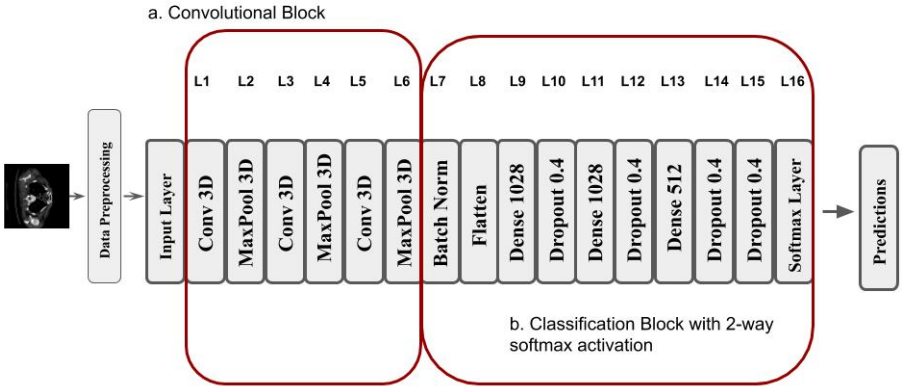


Fig. 3. 16-layer 3D Convolutional Net

loss function show in Equation 3 and the weights were updated using mini-batches for every iteration. The initialization of the weights was done at random and the biases were initialized as zero.

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \epsilon C_c} \log p_{model}[y_i \epsilon_{C_c}] \tag{3}$$

In equation 3, the double sum is over the observations i, whose number is N, and the categories c, whose number is C. The term 1_{y_i} in C_c is the indicator function of the i^{th} observation belonging to the cth category. The p_{model} is the probability predicted by the model for the ith observation to belong to the c^{th} category. When there are more than two categories, the neural network outputs a vector of C probabilities, each giving the probability that the network input should be classified as belonging to the respective category. When the number of categories is just two, the neural network outputs a single probability with the other one being 1 minus the output.

Training was continued for 300 epochs on layers L1 to L16 in Figure 3 with a validation split of 0.1 and the final evaluation was done on the test set. For the final submissions, we only make change in input volume depth size which also required us to change the batch size which results two different settings shown in Table 1. All the other remaining parameters were kept same in the two settings. It is noteworthy that, in both the configurations, the learnable parameters are 23,808,378.

Table 1. Different experimental settings

Name	Input Volume Depth Size	Batch Size
CFG-A	32	4
CFG-B	16	16

2.5 Metrics

The task is evaluated as binary classification problem, including measures such as Area Under the ROC Curve (AUC) and accuracy. An AUC of 1 represents a perfect classification system where True positive rate is 1 and False positive rate is 0. Since the ranking of the techniques will be first based on the AUC and then by the accuracy, AUC is the optimizing metric and the binary accuracy is the satisfying metric.

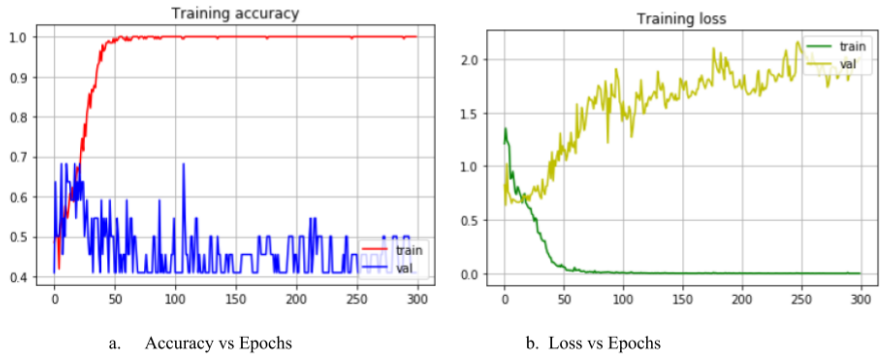
3 Result and Discussion

From Table 2 it can be seen that CFG-A achieves the highest AUC and accuracy values in the experiments conducted in this work. CFG-A, where the network is trained with an input volume of $128 \times 128 \times 32$, this achieves an average test AUC of 0.611 and test accuracy of 61.5%. Also to note, the batch size for this configuration was set to 4, since any larger values resulted in GPU memory exhaustion. This still led to achieve the best result in the set of experiments conducted which is surprising. CFG-B which is trained and evaluated with an input volume of $128 \times 128 \times 16$. From the experiments it can be said that the degradation in CFG-B is due to lower number of slices in the input volume which results in information loss. Even though the batch size was 16 in this setting, the information loss outweighs the impact on the overall performance.

Table 2. Performance on final test set

Name	Area Under ROC Curve	Test Set Accuracy	Best Validation Accuracy	Test-Val Accuracy Margin
CFG-A	**0.611**	**61.5**	68	6.5
CFG-B	0.57	53.8	82.5	28.7

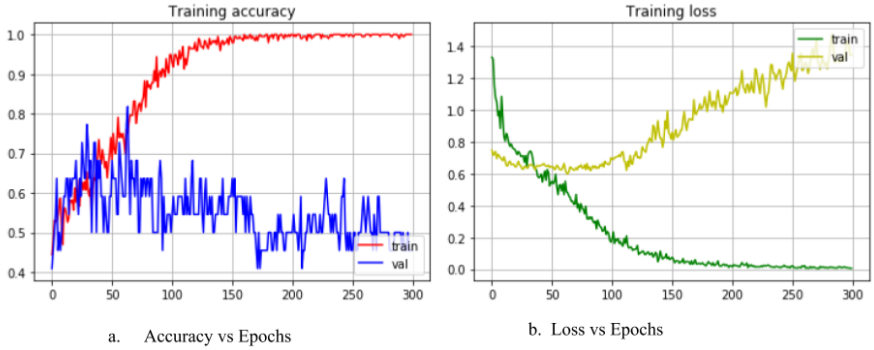
In Figures 4 and 5 we show the training logs for both the configurations. Both CFG-A and CFG-B are trained for 300 epochs. In Figure 4, which portrays the training log for CFG-A, it can be seen that the network starts to overfit only after 25 epochs. CFG-A achieves a highest validation accuracy of 68%, where in the test set an accuracy of 61.5% is achieved which results in test-val accuracy margin of 6.5% between validation and testing set even when the batch size is set to 4.



a. Accuracy vs Epochs b. Loss vs Epochs

Fig. 4. CFG-A Accuracy/Loss graph

In the case of CFG-B which is shown in Figure 5 the network starts overfitting after 60 epochs and achieves a highest validation accuracy of 82.5%. It is surprising that this configuration achieves only a test accuracy of 53.8% which results in a test-val accuracy of 28.7%. From this behaviour we can say that the preprocessing technique employed in CFG-B with depth size of 16 results the test set to not be representative of the validation set. This setup also causes information loss which results in significantly lower performance than CFG-A.



a. Accuracy vs Epochs b. Loss vs Epochs

Fig. 5. CFG-B Accuracy/Loss graph

4 Conclusion

We demonstrate a 3D convolutional neural network with a newly proposed preprocessing technique, slice selection from volumetric data, used in the task to estimate severity based on CT Image of Tuberculosis patients. This work achieved 10 th place with a test AUC of 0.611 and test accuracy of 61.5% in the

ImageCLEF 2019 Tuberculosis - Severity Scoring challenge [15][16]. We show that even without using all the slices from the training set, via slice selection technique, it is possible to achieve certain rather good AUC and accuracy values in the final test set.

5 Future Works

In future works, the results will be further analyzed to gain a better understanding of the reasons behind the results. In addition, various networks architectures will be experimented and further improvements in the proposed slice selection technique will be made, in an attempt to build a robust deep learning model to estimate severity of TB patients.

References

1. Tuberculosis, `https://www.who.int/en/news-room/fact-sheets/detail/tuberculosis`. Last accessed 8 Sept 2018
2. Skoura, E., Zumla, A., Bomanji, J. (2015). Imaging in tuberculosis. International Journal of Infectious Diseases, 32, 87-93.
3. Lakhani, Paras, and Baskaran Sundaram. "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks." Radiology 284, no. 2 (2017): 574-582.
4. Hua, Kai-Lung, Che-Hao Hsu, Shintami Chusnul Hidayati, Wen-Huang Cheng, and Yu-Jen Chen. "Computer-aided classification of lung nodules on computed tomography images via deep learning technique." OncoTargets and therapy 8 (2015).
5. Gonzlez, G., Ash, S.Y., Vegas-Snchez-Ferrero, G., Onieva Onieva, J., Rahaghi, F.N., Ross, J.C., Daz, A., San Jos Estpar, R. and Washko, G.R., 2018. Disease staging and prognosis in smokers using deep learning in chest computed tomography. American journal of respiratory and critical care medicine, 197(2), pp.193-203.
6. Song, QingZeng, Lei Zhao, XingKe Luo, and XueChen Dou. "Using deep learning for classification of lung nodules on computed tomography images." Journal of healthcare engineering 2017 (2017).
7. Hwang, Sangheum, Hyo-Eun Kim, Jihoon Jeong, and Hee-Jin Kim. "A novel approach for tuberculosis screening based on deep convolutional neural networks." In Medical Imaging 2016: Computer-Aided Diagnosis, vol. 9785, p. 97852W. International Society for Optics and Photonics, 2016.
8. Lopes, U. K., and Joo Francisco Valiati. "Pre-trained convolutional neural networks as feature extractors for tuberculosis detection." Computers in biology and medicine 89 (2017): 135-143.
9. Qin, Chunli, Demin Yao, Yonghong Shi, and Zhijian Song. "Computer-aided detection in chest radiography based on artificial intelligence: a survey." Biomedical engineering online17, no. 1 (2018): 113.
10. Vajda, Szilrd, Alexandros Karargyris, Stefan Jaeger, K. C. Santosh, Sema Candemir, Zhiyun Xue, Sameer Antani, and George Thoma. "Feature selection for automatic tuberculosis screening in frontal chest radiographs." Journal of medical systems 42, no. 8 (2018): 146.

11. Maturana, Daniel, and Sebastian Scherer. "Voxnet: A 3d convolutional neural network for real-time object recognition." In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922-928. IEEE, 2015.
12. Su, Hang, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. "Multi-view convolutional neural networks for 3d shape recognition." In Proceedings of the IEEE international conference on computer vision, pp. 945-953. 2015.
13. Pan, S. J., Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
14. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. nature, 521(7553), 436.
15. 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/.
16. 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)