CSE499A CAPSTONE
IN
TUBERCULOSIS TYPE CLASSIFICATION

SENIOR DESIGN PROJECT

Abdullah Al Mahfuj Shaan
ID:1721275042

Mahin Hassan
ID:1711956042

Department of Electrical & Computer Engineering
North South University

Table of Contents

1 Abstract

This report describes the importance of classifying types of Tuberculosis using machine learning algorithms from a dataset and finding out the accuracy by using different model tuning. The dataset was Tuberculosis (TB) is a bacterial infection caused by Mycobacterium tuberculosis, the disease is now a chronic threat and a leading cause of death around the world. Antibiotics will usually treat tuberculosis. However, since different types of tuberculosis need treatment plans, determining the TB type and assessing lesion characteristics are critical real-world activities. The approach was to process 3D CT images by converting them into 2D images in three dimensions while keeping the detail from each dimension. Then use machine learning algorithms to classify each type from the dataset. The novelty abou this project was that except for the pretrained model and data augmentation most part of the project is new. We divided the project into three parts; model building and data augmentation and splitting of the data. By implementing 3D CNN Models, we achieved the primary target of classifying the data.

2 Introduction

Tuberculosis (TB) is a bacterial infection caused by Mycobacterium tuberculosis, a bacteria. The disease is now a chronic threat and a leading cause of death in the world, 130 years since its discovery. This bacteria is most often found in the lungs, but it can also damage other areas of the body.Antibiotics will usually treat tuberculosis. The worst thing that can happen to a TB patient is for the species to become immune to two or three of the normal medications. In comparison to drug-sensitive tuberculosis (DSTB), multi-drug-resistant tuberculosis (MDR-TB) is much more difficult and costly to treat. As a result, early diagnosis of MDR status is critical for effective treatment. Knowing what type it is is extremely important for proper treatment.

The identification of tuberculosis cases from CT scans, classification of the cases into five forms of tuberculosis, and determination of a tuberculosis accuracy score were all important tasks in this project. Working with 3D CT scans needs some experience we did not have, understanding the data took some time. Classifying it into 5 classes and then doing the split was a huge problem that we overcame. Wrong accuracy results were a problem till the end. This report describes the design of the project ,the dataset, methodology, participation and all the data augmentation and the data visualization and then the results section describes the submitted runs and the results obtained for the three sub- tasks. A discussion and conclusion section ends the report. The problem was particularly difficult in some areas.

3 Background

Nowadays the central cause of human tuberculosis is Mycobacterium tuberculosis. Other individuals of the M. tuberculosis complex that can cause tuberculosis incorporate M. bovis, M. microti and M. africanum. In spite of newer modalities for diagnosis and treatment of TB, unfortunately, millions of people are still suffering and dying from this disease. TB is one of the top three infectious killing diseases in the world: HIV/AIDS kills 3 million people each year, TB kills 2 million and malaria kills 1 million [3]. It can afflict persons of any age, although it is more common in persons with compromised immune systems, such as those who have HIV infection. TB infection in healthy persons is generally asymptomatic due to the immune system's ability to keep the pathogenic germs at bay. This bacteria survives and thrives in macrophages, avoiding the patient's serum's natural defense mechanism. There are two phases of tuberculosis infection: asymptomatic latent tuberculosis infection (LTBI) and tuberculosis illness. If left untreated, this illness has a fatality rate of more than 50%.

4 Methodology

4.1 Basic Workflow

The project would follow a linear flow, where the first step would be to acquire a dataset. We would then move on to pre-process the data by making sure the data-set does not contain any duplicates, irrelevant data, and does not contain sections which are almost entirely empty. After the manipulations are applied, we would create our data model by feeding the modified data and train our model to get an insight on the types of TB. Once the model shows confidence and accuracy with regards to providing precise summary and information about included diseases, we would then move on tune the model to achieve better results if possible. Figure below outlines our basic workflow.

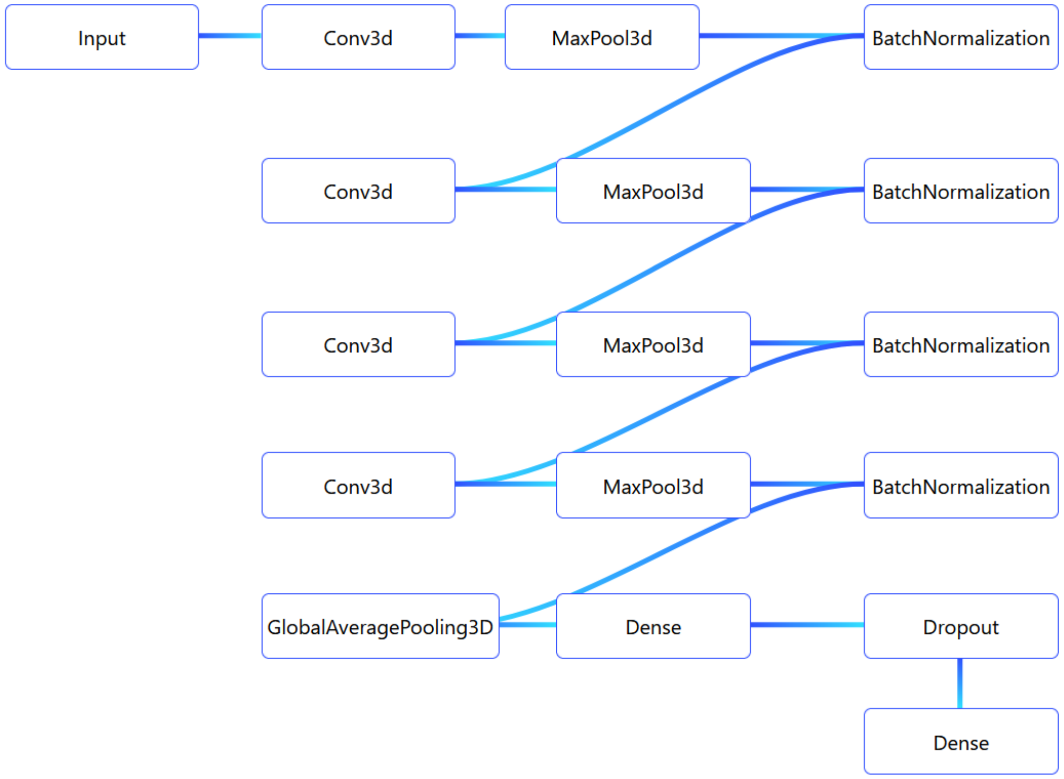


Figure 1: Deep Learning Model for classification of Tuberculosis

4.2 ImageCLEF 2021 Tuberculosis CT Scan Dataset

The dataset contains chest CT scans of 1338 TB patients. 917 images for the Training data set and 421 for the Test set. Some of the scans are accompanied by additional meta-information, which varies on data available for different cases. Each CT-image can correspond to only one TB type at a time.

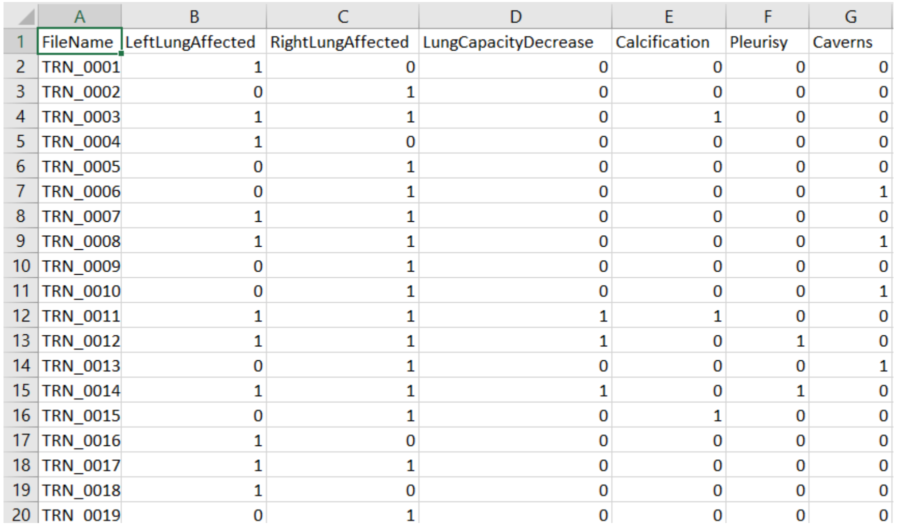
4.2.1 Inside the Dataset

CT Images

The dataset is of 3D CT images which are stored in NIFTI file format with .nii.gz file extension (g-zipped .nii files). This file format has stored 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. Currently, there are various tools available for reading and writing NIFTI files. We have used the NiBabel package for Python.

Masks

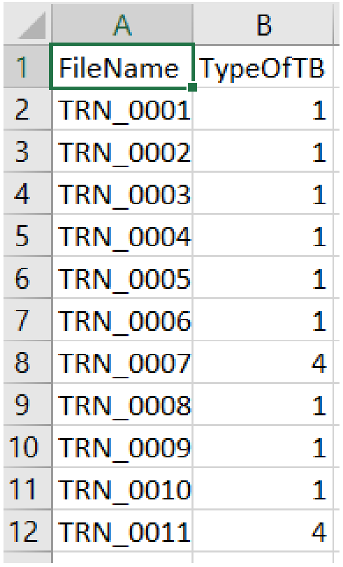
For all the CT images there are two versions of automatically extracted masks of the lungs. These data were downloaded together with the patients CT images. The first version of segmentation has provided more accurate masks, but it tends to miss large abnormal regions of lungs in the most severe TB cases. The second segmentation on the contrary has provided more rough bounds, but behaves more stable in terms of including lesion areas.



	A	B	C	D	E	F	G
1	FileName	LeftLungAffected	RightLungAffected	LungCapacityDecrease	Calcification	Pleurisy	Caverns
2	TRN_0001	1	0	0	0	0	0
3	TRN_0002	0	1	0	0	0	0
4	TRN_0003	1	1	0	1	0	0
5	TRN_0004	1	0	0	0	0	0
6	TRN_0005	0	1	0	0	0	0
7	TRN_0006	0	1	0	0	0	1
8	TRN_0007	1	1	0	0	0	0
9	TRN_0008	1	1	0	0	0	1
10	TRN_0009	0	1	0	0	0	0
11	TRN_0010	0	1	0	0	0	1
12	TRN_0011	1	1	1	1	0	0
13	TRN_0012	1	1	1	0	1	0
14	TRN_0013	0	1	0	0	0	1
15	TRN_0014	1	1	1	0	1	0
16	TRN_0015	0	1	0	1	0	0
17	TRN_0016	1	0	0	0	0	0
18	TRN_0017	1	1	0	0	0	0
19	TRN_0018	1	0	0	0	0	0
20	TRN_0019	0	1	0	0	0	0

Figure 2: Metadata of the TB Type Classification Dataset

Additionally we were provided with labelling data for training the model.



	A	B
1	FileName	TypeOfTB
2	TRN_0001	1
3	TRN_0002	1
4	TRN_0003	1
5	TRN_0004	1
6	TRN_0005	1
7	TRN_0006	1
8	TRN_0007	4
9	TRN_0008	1
10	TRN_0009	1
11	TRN_0010	1
12	TRN_0011	4

Figure 3: Labelling Data for the TB Type Dataset

4.3 Pre-Processing the Data

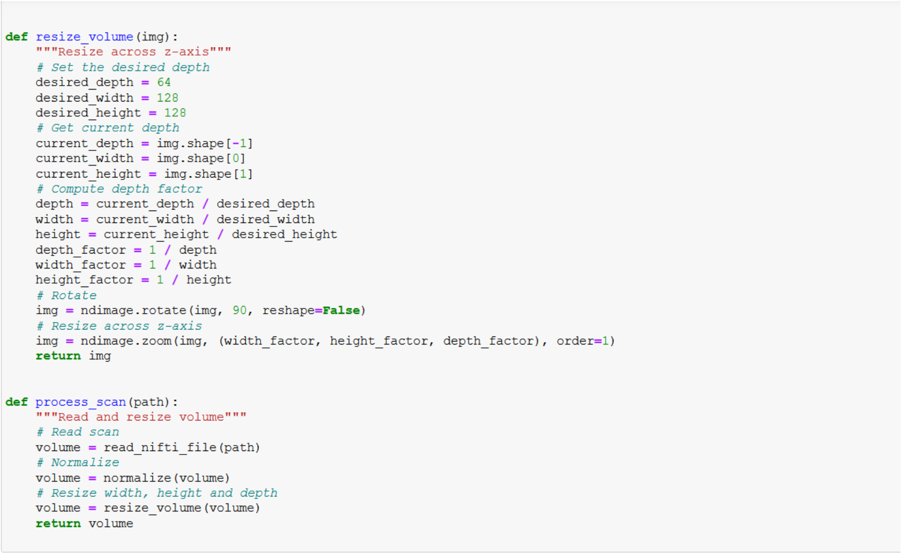
4.3.1 Data Reading and Resizing

The files are provided in Nifti format with the extension .nii. To read the scans, we have used the nibabel package. CT scans store raw voxel intensity in Hounsfield units (HU). They range from -1024 to above 2000 in this dataset. Above 400 are bones with different radio intensity, so this is used as a higher bound. A threshold between -1000 and 400 is commonly used to normalize CT scans.

To process the data, we have done the following:

- Rotated the volumes by 90 degrees, so the orientation is fixed
- Scaled the HU values to be between 0 and 1.
- Resize width, height and depth.

We have defined several helper functions to process the data. These functions will be used when building training and validation datasets.



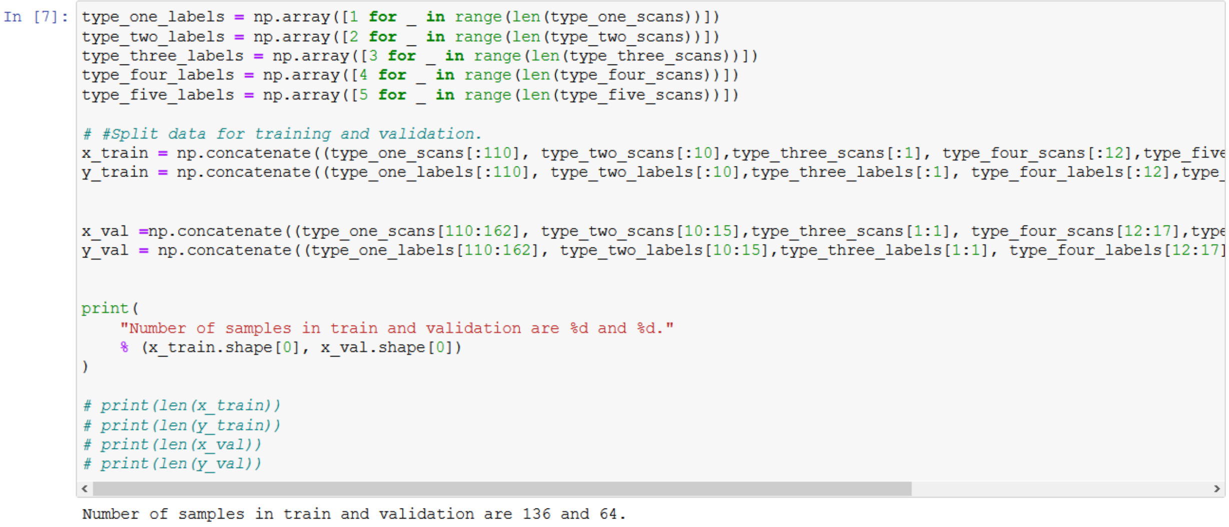
```
def resize_volume(img):
    """Resize across z-axis"""
    # Set the desired depth
    desired_depth = 64
    desired_width = 128
    desired_height = 128
    # Get current depth
    current_depth = img.shape[-1]
    current_width = img.shape[0]
    current_height = img.shape[1]
    # Compute depth factor
    depth = current_depth / desired_depth
    width = current_width / desired_width
    height = current_height / desired_height
    depth_factor = 1 / depth
    width_factor = 1 / width
    height_factor = 1 / height
    # Rotate
    img = ndimage.rotate(img, 90, reshape=False)
    # Resize across z-axis
    img = ndimage.zoom(img, (width_factor, height_factor, depth_factor), order=1)
    return img


def process_scan(path):
    """Read and resize volume"""
    # Read scan
    volume = read_nifti_file(path)
    # Normalize
    volume = normalize(volume)
    # Resize width, height and depth
    volume = resize_volume(volume)
    return volume
```

Figure 4: Data Preprocessing

4.3.2 Splitting the dataset into Train and Validation sets.

Scanned data were loaded from the class directories and assigned labels. Downsampled the scans to have a shape of 128x128x64. Rescaled the raw HU values to the range 0 to 1. Lastly, the dataset was split into train and validation subsets.



```
In [7]: type_one_labels = np.array([1 for _ in range(len(type_one_scans))])
        type_two_labels = np.array([2 for _ in range(len(type_two_scans))])
        type_three_labels = np.array([3 for _ in range(len(type_three_scans))])
        type_four_labels = np.array([4 for _ in range(len(type_four_scans))])
        type_five_labels = np.array([5 for _ in range(len(type_five_scans))])

        # #Split data for training and validation.
        x_train = np.concatenate((type_one_scans[:110], type_two_scans[:10],type_three_scans[:1], type_four_scans[:12],type_five
        y_train = np.concatenate((type_one_labels[:110], type_two_labels[:10],type_three_labels[:1], type_four_labels[:12],type_


        x_val =np.concatenate((type_one_scans[110:162], type_two_scans[10:15],type_three_scans[1:1], type_four_scans[12:17],type
        y_val = np.concatenate((type_one_labels[110:162], type_two_labels[10:15],type_three_labels[1:1], type_four_labels[12:17]


        print(
            "Number of samples in train and validation are %d and %d."
            % (x_train.shape[0], x_val.shape[0])
        )

        # print(len(x_train))
        # print(len(y_train))
        # print(len(x_val))
        # print(len(y_val))

        Number of samples in train and validation are 136 and 64.
```

Figure 5: Splitting into Train and Validation Subsets

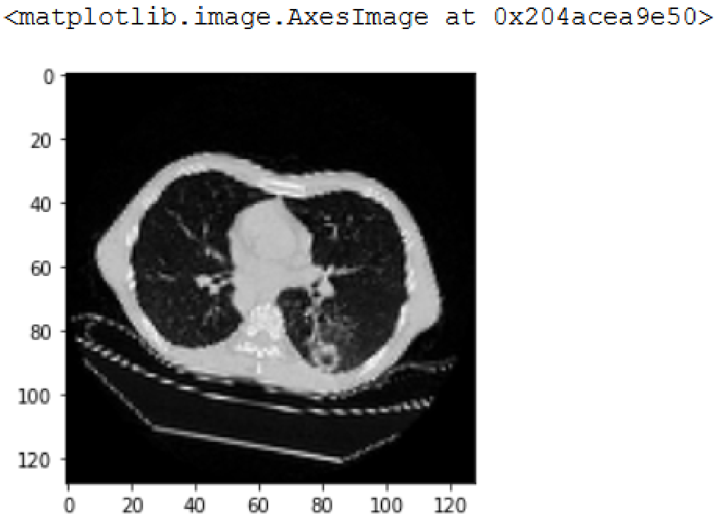
4.3.3 Data Augmentation

The CT scans have been augmented by rotating at random angles during training. Since the data is stored in rank-3 tensors of shape (samples, height, width, depth), we have added a dimension of size 1 at axis 4 to be able to perform 3D convolutions on the data.

The new shape is thus (samples, height, width, depth, 1).

While defining the train and validation data loader, the training data is passed through an augmentation function which randomly rotates volume at different angles. Both training and validation data are already rescaled to have values between 0 and 1.

4.3.4 Data Visualization



```
<matplotlib.image.AxesImage at 0x204acea9e50>
```

Figure 6: Augmented CT Scan

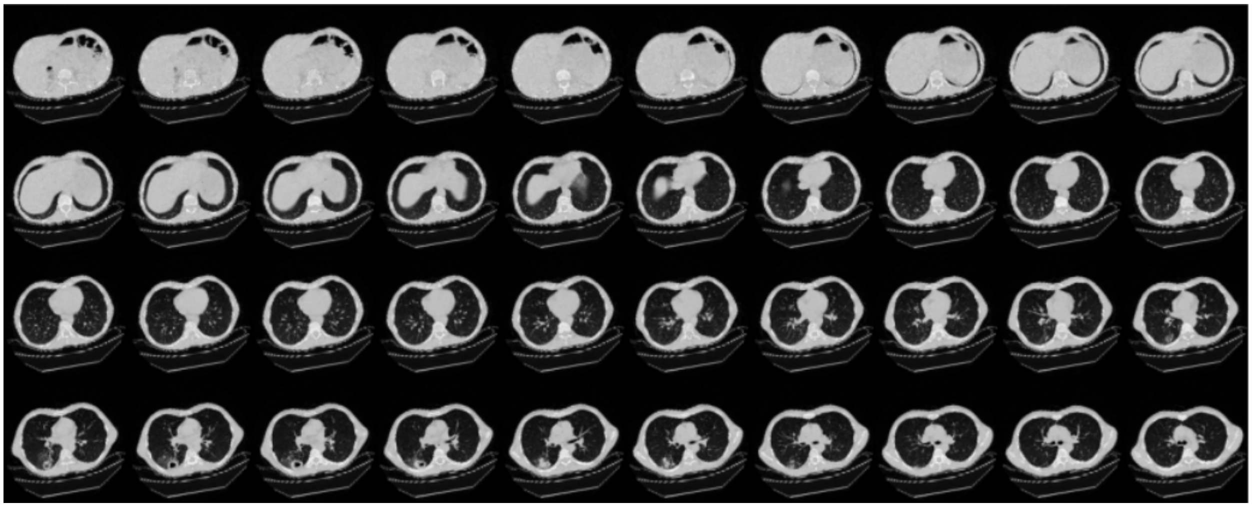


Figure 7: Montage of augmented data

4.3.5 Model Define and Model Training

We created a small 3D CNN Model and trained on a small number of data for test results. We used categorical cross entropy for loss function and Adam as an optimizer. We trained the model on 100 epochs.

5 Results & Discussion

The model accuracy and loss for the training and the validation sets were plotted. Since the validation set is class-balanced, accuracy provides an unbiased representation of the model's performance. But since we could not finish the model and there were a few errors, the results were not satisfactory. The model that we built needs to be tuned again for accurate results. We have used multiclass classification and categorical cross entropy as the loss functions but it did not work well.

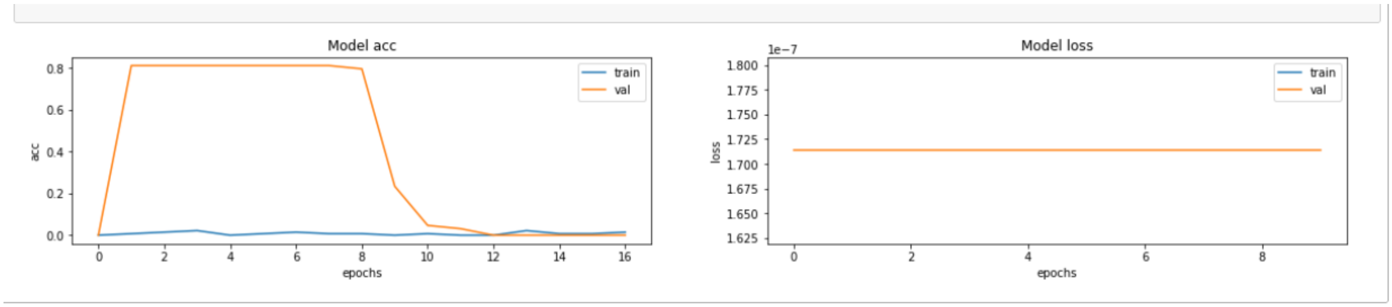


Figure 8: Evaluation of the model

The model needs to be tuned as the accuracy was not at all good enough.

6 Conclusion & Recommendations

In this work, we have analysed and tried to implement 3D CNN Models to achieve our primary purpose. We can now achieve the primary target of classifying the data, but this is just a small achievement for our work. But we have faced a lot of issues to achieve this. To overcome that situation, we will implement the work with proper tuning of the Deep Learning Model. Then we have to test the data on the Test Dataset provided by ImageCLEF. And also read other related works so we can publish a research paper after finding the proper accuracy. We will also be trying to implement UNet Models in our next part of the project.

7 Acknowledgements

References

[1] Zunair, H. (2020, September 23). *Keras documentation: 3D Image Classification from CT Scans*. https://keras.io/examples/vision/3D_image_classification/#loading-data-and-preprocessing

[2] Zunair, H. (2020a, July 26). *Uniformizing Techniques to Process CT scans with 3D CNNs for Tuberculosis Prediction*. ArXiv.Org. https://arxiv.org/abs/2007.13224

[3] Geneva: WHO; 2010. . World Health Organization. Fact Sheet No.104: Tuberculosis. Available from: http://www.who.int/mediacentre/factsheets/fs104/en/print.html . [Google Scholar]