Detecting TB from chest X-Rays in a population of patients living with HIV and diabetes in West Africa

Haozhe Qi [* 1] **Mu Zhou** [* 1] **Anna Paulish** [2]

Abstract

Tuberculosis (TB) is a serious lung disease caused by bacteria, and it remains a big challenge in the global health. Therefore, automatic TB detection is essential. Existing automatic TB detection methods have been developed for images that are of good quality, which limits the use of the method in cases where images are from taken from smartphone camera. To tackle this limitation, we build a TB detection pipeline that enable TB detection from smartphone photos. Additionally, our data present 1) high degree imbalance between positive (with TB) and negative (without TB) examples, 2) complex pathological correlation as many patients also suffer from diabetes and HIV and 3) various image quality. We discuss how these add difficulties to the problem and propose algorithms that tackle them. We show that our proposed method outperform baselines with aforementioned challenges.

1. Introduction

Tuberculosis (TB) is a serious lung disease caused by bacteria called Mycobacterium tuberculosis (WHO et al.). In clinical diagnosis, X-Ray is a widely-used diagnosis tool for detecting TB, and usually it is examined by experienced human readers such as radiologist or clinicians. However, this is a time-consuming and knowledge required task. In recent few years, with the huge success made by deep learning methods in common image classification tasks (He et al., 2015) (Tan & Le, 2019), CNNs (Wong et al., 2021)(Rahman et al., 2020b) and Vision transformer (Duong et al., 2021) have also been applied to several medical image related studies such as TB detection. However, existing TB detection methods (Duong et al., 2021; Pasa et al., 2019; Rahman et al., 2020b; Wong et al., 2021) only perform on well-established data sets, which have 1) good image quality, and 2) balanced classes towards TB and normal. In fact, in real world application, these methods are not robust enough to generalize to other complex dataset, such as our X-Ray images taken by smartphone. In this project, we build a TB detection pipeline with a real-world chest X-Rays dataset collected in West Africa. The main challengings are: 1) the X-Ray images are recaptured by smartphone with different angles and lighting condition, 2) there is a severe class imbalance on the subgroup dataset, 3) the patients may also have other illnesses (HIV, diabetes).

With all the mentioned challenges, our method still obtains a satisfactory performance. Our contributions are as follows:

1) A new TB detection pipeline (containing data preprocessing and augmentation, image classification models and training tricks);

2) A competitive performance models (with 95.7% on F1-score) compared to the existing models (94.0% on F1-score);

3) A label remove method that can automatically remove the labels in the original photos and a label removed dataset that can be used to train learning based label remove methods.

2. Related work

Many solutions have been proposed for TB detection, such as TBCNN (Pasa et al., 2019), TBNet (Wong et al., 2021), ReliableTB (Rahman et al., 2020b), and ViTTB (Duong et al., 2021). TBCNN (Pasa et al., 2019) is a good performance and efficient CNN model, it reduces the computational, memory and power requirements significantly and also preserves the good performance. TBNet (Wong et al., 2021) is a self-attention deep convolutional neural network, which leaverage a machine-driven design exploration strategy to build the self-attention module within the deep neural network. ReliableTB (Rahman et al., 2020b) provides a complete pipeline for TB detection, which utilizes the pre-trained weights from nine different deep CNNs and achieves state-of-the-art performance in TB detection. ViTTB (Duong et al., 2021) is composed of the EfficientNet (Tan & Le, 2019) as backbone and Vision Transforer (Dosovitskiy et al., 2021) as head, which shows the great performance in the TB detection with respect to different quality metrics. Due to the limitation of the page number, we put a more concrete table in the link describing the related studies on TB detection (Betsy Antony, 2017; Pasa et al., 2019; Rahman et al., 2020b; Wong et al., 2021) and others, see Link.

3. Data Processing

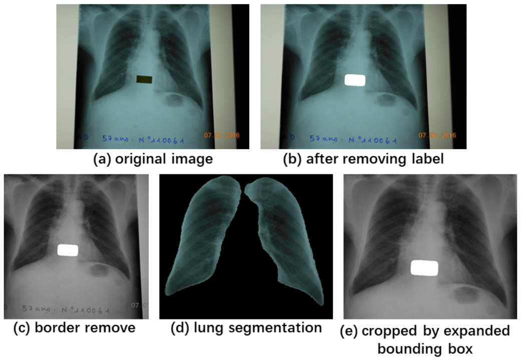

Figure 1. **Data overview**

3.1. Dataset

Our chest X-Rays images are collected from an under-represented population of patients co-infected with diabetes and HIV in West Africa. Figure 1(a) shows our original images from the dataset, which are taken from the smartphone from different perspective. Due to the privacy problem, we will release the data here (Link) but not in the github repo. As seen in the figure, there is a sticky label in each image identifying the category. To reduce the impact of the labels on the model performance, the first step is to remove the labels which is described in section 3.2.

DATA	TB	NOT TB	TOTAL
DIABETES	27	2137	2164
HIV	102	735	837
TB-ONLY	468	-	468
WHOLE DATASET (TOTAL)	597	2872	3469

Table 1: Overview of dataset

Table 1 presents an overview of our dataset, which includes the number of HIV, diabetes data (with and without TB) and the TB only data. From the table, the most challenging part in this dataset is the data imbalance problem occured in diabetes dataset.

3.2. Label remove

As shown in Figure 1(b), we removed the labels on the raw X-Ray images. We improved the original algorithm twice from the host lab (41.5% accuracy) and obtain 90.2% accuracy on label remove. Moreover, we manually remove the labels for images that cannot be detected by the label remove methods to construct the standard dataset for training and evaluation.
Meanwhile, since we manually check and label all the images, the clean dataset can also be used to train learning-based label removal methods in the future.

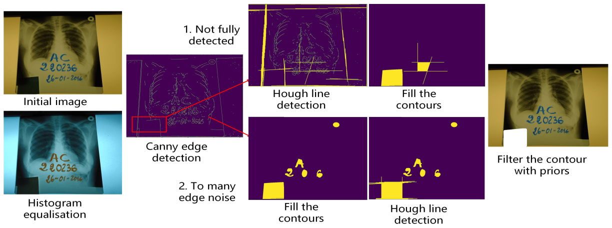

Figure 2. **Label remove overview**

3.2.1. INITIAL SETTING

The initial method provided by the host laboratory has 2 steps: 1) transform the initial image to a binary image using an adaptive threshold, 2) find contours from the binary image and draw rectangles of a suitable size after approximation of the shape of the contour.

3.2.2. IMPROVED METHOD 1

Since the initial method does not provide high accuracy, we improve its performance in the following way (achieve 71.0% accuracy): 1) find a more suitable adaptive threshold parameter to generate binary image, 2) remove noise using morphological transformations, 3) draw a rectangle with additional margins.

3.2.3. IMPROVED METHOD 2

We keep improving the label remove method by replacing the simple threshold segmentation with Canny edge detection (Canny, 1986). We 1) perform histogram equalisation on the initial image and then perform canny edge detection. After that, we 2) use Hough line detection on either edge maps or new contour maps to separately solve two challenging situation, and finally, 3) find all contours and filter the right contour with priors.

3.3. Data preprocessing

3.3.1. BORDER REMOVE

Inspired by the image preprocessing in TBCNN (Pasa et al., 2019), we remove the irregular border and put the lung on the center of image, as shown in Figure 1(c).

3.3.2. LUNG SEGMENTATION

Previous study on TB detection has explored the lung segmentation during data preprocessing step (Rahman et al., 2020b). Motivated by this work, firstly we use a trained lung segmentation model to segment the lung area on our dataset. To prevent the loss of images of the lung area, then we crop the image with the expanding bounding box from the segmented area. The procedure can be seen from the Figure 1(d)1(e).

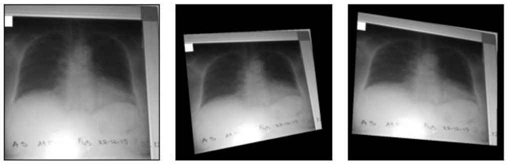

Figure 3. **Data augmentation: image distortion** (a) the original image, (b) the distorted images with different distortion scale (from left to right: 0.2, 0.4, 0.6, 0.8).

3.3.3. Data augmentation

To empower the learning process, we also explore various augmentation techniques, which is widely used in practice to increase the data diversity, such as random rotation, random resized crop, random flip, and center crop. Our original images are taken from the smartphone cameras thus full of noise and distortion. Therefore, we also employ random distortion method to make sure our model is able to perform well with the distorted photos, as shown in Figure 3.

3.3.4. External dataset

Since our own data have quite few TB samples, we also resort to find other related dataset to pretrain our model. In total, we find five different external datasets (Al-Timemy et al., 2021; Ali, 2021; Chauhan et al., 2014; Liu et al., 2020; Rahman et al., 2020a) and combine them together. This gives us 5008 TB samples and 10841 normal samples.

4. Model

To test as many as advanced computer techniques as we can, instead of using existing TB detection method as our baseline, we resort to build our baseline from scratch with deep learning classification library TIMM (Wightman, 2019). TIMM is a frequently updated library that already has $2.5k$ stars on the github. With the help of this library, we successfully build our baseline and test various advanced techniques in network architecture, loss function, optimiser and learning rate schedule.

4.1. Network architecture

Here, we mainly test three network architecture, which are ResNet50 (He et al., 2015), EfficientNet-b2 (Tan & Le, 2019) and Resnext(Xie et al., 2017). Among them, Resnet has been widely used in many areas for a long time, and the other two are recently very popular and advanced methods, so we also want to test their performance on this task.

4.2. Loss function

Here, we mainly use cross entropy loss to train our network. Moreover, Since our data is very unbalanced, in order to solve this problem to some extent, we also try to use focal loss(Lin et al., 2017) and f1 loss to train our network.

4.3. Optimiser

For the optimiser, we mainly use momentum SGD(Sutskever et al., 2013) to optimize our network parameters. In addition, we also test some advanced optimizers including adam (Kingma & Ba, 2014) and lookahead (Zhang et al., 2019).

4.4. Learning rate schedule

Normally, people will use StepLR to schedule the learning rate. But in our experiment, we find we can obtain a better performance using SGDR scheduler (Loshchilov & Hutter, 2016). So we mainly use this.

5. Experiments

5.1. Dataset and evaluation metric

The pipeline is trained and tested on 3 different datasets (Diabetes, HIV and whole dataset), which consist of training, validation and test sets. The positive samples (TB) and negative samples (not TB) are imbalanced (1:4), especially in Diabetes dataset (1:10). Therefore, we choose F1 score as our evaluation metric of the TB detection.

5.2. Comparison models

To compare the model performance with previous works, we tested the trained model from TBCNN (Pasa et al., 2019) and XTBTorch (Upadhyay, 2019) on our dataset. In addition, we also trained the above two networks with our data. Table 4 presents the results on the previous works with our data. When without training, TBCNN and XTBTorch have similar performance in 3 datasets, and both of them are not robust to our data. After training on our data, the performance is improved. However, due to the large imbalance in Diabetes dataset, it is hard to recognize the positive data. Especially, during training with XTBTorch on Diabetes dataset, the model is not able to learn any positive samples.

5.3. Results on the whole dataset

Here, due to the time limitation, we just try to test as many techniques as possible, and we don't combine all the good factors together to generate the best performance. For example, we generate better performance using effcientnet-b2 than resnet50 on the same settings, but as training effcientnetb2 requires a large amount of time, so we just check other useful techniques with resnet50 rather than effcientnetb2. We test various methods related to network architecture, loss function, optimiser and learning rate schedule. Moreover, we also try to use external data to pretrain our model. All the techniques that we test are shown in Table 2, we can see the highest performance that we obtian is 0.958, which outperforms all the comparison methods to a large extent. If we combine all the good factors together (masked data, efficientnetb0, and external data), we believe that we can obtain even higher performance.

Methods	F1 score
Data augmentation	
Initial image	0.949
Cropped image	0.912
W/o border image (default)	0.936
Masked data	**0.957**
Network architecture	
Resnet50 (default)	0.936
Resnext	0.928
Effcientnetb2	**0.954**
Optimizer	
Momentum SGD (default)	**0.936**
Adam	0.910
Lookahead adam	0.829
Learning rate scheduler	
StepLR	0.930
SGDR with top1 score update	0.926
SGDR with f1 score update (default)	**0.936**
External data	
With external data	**0.958**
Without external data (default)	0.936

Table 2: F1 score for whole dataset

5.4. Results on HIV and Diabetes datasets

We then test our method on HIV and Diabetes sub datasets. The performances are shown in Table 3. Our method cannot obtain a good result only using the initial image or the border removed image (default) because these two sebsets are quite small and unbalanced. But with the help of the external data, we improve our performance a lot, especially for the Diabetes dataset, we outperform the comparison methods to a large extent. Moreover, we also try to use f1 loss and Focal loss to relief the unbalance problem. And the f1 loss obtain a better performance on HIV dataset. Importantly, it is surprising that using masked image obtain the best performance on HIV subset. And we think it mainly because masked image get rid of other noise and relieve the learning burden of the model.

Methods	F1 score
HIV	
Initial data	0.687
Default setting	0.643
Masked data	**0.778**
External data	0.702
External data + F1 loss	0.712
External data + Focal loss	0.590
Diabetes	
Initial data	0.0
Default setting	0.0
Masked data	0.012
External data	**0.286**
External data + F1 loss	0.270
External data + Focal loss	0.222

Table 3: **F1 score for HIV and Diabetes datasets** (default setting: all the default mentioned above in Table 2)

Method	Dataset	train	f1-score
tbcnn	Whole	×	0.507
tbcnn	Whole	√	0.934
XTBtorch	Whole	×	0.532
XTBtorch	Whole	√	0.940
Ours	**Whole**	√	**0.958**
tbcnn	Diabetes	×	0.071
tbcnn	Diabetes	√	0.167
XTBtorch	Diabetes	×	0.030
XTBtorch	Diabetes	√	0.0
Ours	**Diabetes**	√	**0.286**
tbcnn	HIV	×	0.260
tbcnn	**HIV**	√	**0.842**
XTBtorch	HIV	×	0.392
XTBtorch	HIV	√	0.771
Ours	HIV	√	0.778

Table 4: Comparison methods on our 3 datasets

6. Discussion

In this project, we try to build a TB detection pipeline with a noisy and unbalanced X-ray dataset. To achieve this, we first remove the label inside the image using our label remove method (90.2% accuracy). After that, we manually label the undetected labels to get the clean data. With this data, we develop our pipeline and try a lot of advanced techniques, which obtain 95.8% accuracy. Moreover, we also do an extensive survey in this literature, train and test state-of-the-art methods on our dataset, which further shows the superiority of our method.

However, due to the time limitation, there are also many other things that we have no time to try, which we list here as the future work. Generally, they can be divided as 3 aspects.

1) The good factors that we have tested can be combined together and more advanced techniques can be tested.

2) More tricks to solve the data unbalance can be applied such as mix-up data augmentation, oversampling training strategy, etc.

3) More literature-related techniques can be applied like bone removal, lung area attention, etc.

References

Al-Timemy, A. H., Khushaba, R. N., Mosa, Z. M., and Escudero, J. An efficient mixture of deep and machine learning models for covid-19 and tuberculosis detection using x-ray images in resource limited settings. In *Artificial Intelligence for COVID-19*, pp. 77–100. Springer, 2021.

Ali, O. Tuberculosis 3000 dataset. https://www.kaggle.com/sindalflekke/tb3000, 2021.

Betsy Antony, N. B. P. K. Lung tuberculosis detection using x-ray images. *International Journal of Applied Engineering Research ISSN*, 2017.

Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. doi: 10.1109/TPAMI.1986.4767851.

Chauhan, A., Chauhan, D., and Rout, C. Role of gist and phog features in computer-aided diagnosis of tuberculosis without segmentation. *PloS One*, 9(11):e112980, 2014.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Duong, L. T., Le, N. H., Tran, T. B., Ngo, V. M., and Nguyen, P. T. Detection of tuberculosis from chest x-ray images: boosting the performance with vision transformer and transfer learning. *Expert Systems with Applications*, 184:115519, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Liu, Y., Wu, Y.-H., Ban, Y., Wang, H., and Cheng, M.-M. Rethinking computer-aided tuberculosis diagnosis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2646–2655, 2020.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2016.

Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D., and Pfeiffer, D. Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization. *Scientific reports*, 9(1):1–9, 2019.

Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., Hamid, T., Islam, M. T., Kashem, S., Mahbub, Z. B., et al. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020a.

Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., Hamid, T., Islam, M. T., Mahbub, Z. B., Ayari, M. A., and Chowdhury, M. E. H. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020b.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Tan, M. and Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 09–15 Jun 2019.

Upadhyay, U. Detecting tuberculosis from x-ray scan using pytorch. https://github.com/udion/XTBTorch, 2019.

WHO et al. Global tuberculosis report 2021; geneva, switzerland, 2021.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Wong, A., Lee, J. R. H., Rahmat-Khah, H., Sabri, A., and Alaref, A. Tb-net: A tailored, self-attention deep convolutional neural network design for detection of tuberculosis cases from chest x-ray images. *arXiv preprint arXiv:2104.03165*, 2021.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. *Advances in Neural Information Processing Systems*, 32:9597–9608, 2019.