



A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection

Erdal Tasci¹ · Caner Uluturk¹ · Aybars Ugur¹

Received: 5 October 2020 / Accepted: 27 May 2021 / Published online: 7 June 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Tuberculosis (TB) is known as a potentially dangerous and infectious disease that affects mostly lungs worldwide. The detection and treatment of TB at an early stage are critical for preventing the disease and decreasing the risk of mortality and transmission of it to others. Nowadays, as the most common medical imaging technique, chest radiography (CXR) is useful for determining thoracic diseases. Computer-aided detection (CADe) systems are also crucial mechanisms to provide more reliable, efficient, and systematic approaches with accelerating the decision-making process of clinicians. In this study, we propose voting and preprocessing variations-based ensemble CNN model for TB detection. We utilize 40 different variations in fine-tuned CNN models based on InceptionV3 and Xception by also using CLAHE (contrast-limited adaptive histogram equalization) preprocessing technique and 10 different image transformations for data augmentation types. After analyzing all these combination schemes, three or five best classifier models are selected as base learners for voting operations. We apply the Bayesian optimization-based weighted voting and the average of probabilities as a combination rule in soft voting methods on two TB CXR image datasets to get better results in various numbers of models. The computational results indicate that the proposed method achieves 97.500% and 97.699% accuracy rates on Montgomery and Shenzhen datasets, respectively. Furthermore, our method outperforms state-of-the-art results for the two TB detection datasets in terms of accuracy rate.

Keywords Tuberculosis detection · Pattern recognition · Deep learning · Fine-tuning · Image processing · Voting · Ensemble learning · Augmentation · CLAHE

1 Introduction

Infectious diseases are disorders caused by many pathogenic microorganisms, such as bacteria, viruses, and parasites [37]. The diseases can be directly or indirectly passed from one person to another. Some infectious diseases can be spread by insects or different animals. Tuberculosis (TB), coronavirus, and malaria are crucial examples of serious contagious diseases.

TB is caused by the bacterium called *Mycobacterium tuberculosis* (MTB) [33]. It often affects the lungs, but TB bacteria can harm other parts of the body such as kidney and brain [6]. This disease is transmitted through the air from person to person. TB disease is a major health threat for the people (particularly adults) of developing countries especially in the African and South-East Asian regions. According to the WHO, 10 million people fell ill with this disease and a total of 1.5 million people died from TB in 2018 [38]. TB is curable and preventable if necessary TB diagnosing, treatments are done and drugs are used properly, otherwise, the disease can be fatal.

Diverse medical imaging modalities like computed tomography scan (CT) and chest X-ray are applied for identifying lung diseases. Chest radiography or chest X-ray (CXR), known as the most common imaging modality, is utilized to detect/diagnose conditions of lung abnormalities, particularly pulmonary TB [27]. CXR is a rapid,

✉ Erdal Tasci
arif.erdal.tasci@ege.edu.tr

Caner Uluturk
caner.uluturk@mobexis.com

Aybars Ugur
aybars.ugur@ege.edu.tr

¹ Computer Engineering Department, Ege University, Izmir, Turkey

essential, highly sensitive, affordable, and primary medical imaging tool for early detection of TB [39].

With the widespread of technological devices, the computer-aided detection/diagnosis (CADe)/(CADx) systems gain importance by providing more accurate, efficient solutions to expert radiologists for speeding up their decision-making process. Nowadays, deep learning approaches (e.g., convolutional deep neural networks (CNN)) beat results for many artificial intelligence-related fields such as image classification, natural language understanding, and speech recognition with the increase in computation power (i.e., GPU and CPU) and data volume [19]. In addition to the deep learning, ensemble learning (e.g., voting mechanisms) fuses the results of various learning models to achieve better predictive performance for machine learning tasks [9]. At the present time, ensemble learning approaches are considered as giving state-of-the-art results for solving machine learning challenges [28, 34].

In this study, we propose an ensemble deep learning method that selects the best pipelines employing different preprocessing, augmentation alternatives, and CNN models for tuberculosis detection. We utilize the voting-based (i.e., soft voting and Bayesian optimization-based weighted voting) ensemble of various fine-tuned CNN models (i.e., InceptionV3 and Xception) with the preprocessing (e.g., Contrast-limited adaptive histogram equalization (CLAHE)) and image data augmentation (e.g., translation, rotation, and scaling) variations for TB detection. Furthermore, we calculate the mean performance values (i.e., accuracy rate (%) and AUC (Area Under ROC Curve)) on different train-test sets and analyze comprehensive experimental results for getting reliable results and improving TB detection performance.

The main contributions of the study are summed up as follows:

- As far as our knowledge, our study includes the first method that uses both variations in preprocessing and image augmentation techniques with various voting schemes to fine-tune different CNN models for performance improvement in TB detection.
- We focus on merging the advantages of the image processing, deep learning, and ensemble learning techniques for image classification on two common TB datasets (namely Montgomery and Shenzhen).
- We introduce a significant time-efficient approach to the fine-tuning process by applying all preprocessing operations as a whole on the images before the fine-tuning process.
- It is aimed to figure out the best combination of models on the fine-tuning process for more efficient and accurate voting operations.

- The extensive effects of the combinations of two image preprocessing types, ten different transformations of image data augmentation, and two pre-trained CNN models are revealed to fine-tune various CNN models for TB detection.
- The performances of voting-based algorithms are measured with a various number of learners in detail.
- We outperform the results of state-of-the-art methods for tuberculosis detection on commonly used two CXR image classification datasets in terms of accuracy rate.

The remaining sections of this study are structured as follows: First, we briefly explain the important previous works and summarize various methods related to tuberculosis detection in Sect. 2. We present CNN models and describe our method in the third section. After the experimental procedures and the evaluation metrics are described, we give the computational results in detail in Sect. 4. Section 5 contains the conclusion, discussion of the results and also possible recommendations for future studies.

2 Related works

In recent years, studies of researchers for TB detection with CADx systems particularly are categorized into two classes: (1) machine learning-based methods (2) deep learning-based approaches [7, 20, 22, 27, 35, 36, 39]. For machine learning-based systems, traditional handcrafted feature extraction methods and different learning models are utilized in this context. For deep learning-based approaches, pre-trained CNNs are used for the deep (learned) feature extraction process. All studies related to this field contribute to the detection/diagnosing process by automation of analyzing CXR images, speeding up operations, increasing the quality of TB detection, and improving the performance.

As conventional handcrafted feature extraction-based TB detection approaches, in [22], a wavelet transform is proposed for TB detection. They acquired thirty line profiles and applied one-dimensional discrete wavelet transform to the profiles to obtain Daubechies coefficients. Then, the coefficients are used as features for identifying TB. In [36], a fully automatic method is introduced to make a decision on CXRs using texture patterns. To this end, the lung fields are divided into parts and analyzed each part individually. Afterward, various texture features (e.g., second, third, and fourth moments) are extracted by applying multi-scale filter banks. Furthermore, k nearest neighbors (K-NN) algorithm is utilized for classifying texture patterns in the range from zero (normal) to one (abnormal).

In [7], the histogram of oriented gradients (HOG), Gabor, gist, and pyramid histogram of oriented gradients (PHOG) features are extracted from images to diagnose TB without segmentation. The results demonstrated that extracted features improved efficiency of discrimination between the TB and non-TB all CXR images than gray level co-occurrence matrix (GLCM) textural features. In [35], the authors develop a set of feature extraction methods (e.g., shape and texture features) with a wrapper-based feature selection strategy to identify normal and TB CXR lung images. They obtain 78.3% accuracy (ACC) and 0.87 AUC values for the Montgomery dataset, 95.57% ACC, and 0.99 AUC values for the Shenzhen dataset.

As deep learning-based methods using CNN models, [11] carried out an ensemble approach by employing three pre-trained CNNs to classify X-ray images of patients for tuberculosis detection. At preprocessing stage, they duplicated every image by horizontally mirroring them and applied histogram equalization or CLAHE to every image. Ensemble of ResNet50, VGG19, and InceptionV3 models are used for classification.

An ensemble of fine-tuned CNNs is also used in another study [18] to classify medical images from the Subfigure Classification dataset from the ImageCLEF 2016 collection. They developed a new feature extractor by fine-tuning CNNs. AlexNet and GoogLeNet CNN architectures are used with softmax and one-vs-one multi-class SVMs classifiers.

Lung region symmetry is also considered to detect pulmonary abnormalities [29]. The study stated that Abnormal Posteroanterior chest radiographs (CXR) tend to reflect changes in lung content (textures), size, and shape. By using that fact, they analyzed lung region symmetry using edge plus texture features and multi-scale shape features. Their classification architecture consists of voting-based combination of multilayer perception neural networks (MLP), bayesian network, and random forest. Montgomery County, Shenzhen, China, India, and New Delhi data collections are used. Their method achieved 91.00% accuracy and 0.96 AUC for abnormality detection.

There is also a study that takes speed and efficiency in first place while preserving accuracy for X-ray tuberculosis screening [23]. They also used visualization capabilities of CNNs by testing saliency maps and gradCAMs as tuberculosis visualization methods. They implemented simple CNN optimized for the problem. Montgomery and Shenzhen datasets are used for experiments.

In [13], computer-aided diagnosis (CAD) system is developed based on deep CNN for tuberculosis screening. They added one extra convolution layer to Alexnet CNN architecture for feature extraction. They used Montgomery, Shenzhen, and the Korean Institute of Tuberculosis

datasets. The effects of transfer learning are also analyzed with experiments.

In [20], the authors introduce three different CNN-based methods for tuberculosis detection. For all schemes, the pre-trained CNNs are fundamentally utilized as feature extractors to detect the disease in this scope. In the first scheme, deep features are extracted from different pre-trained CNNs such as GoogleNet and ResNet. Then, obtained features are given into the support vector machine (SVM) classifier for TB detection. In the second scheme, the bag-of-words (BOW) model and three different deep feature sets from various CNNs in subregions of images with the SVM classifier are used for this purpose. As a final approach, an ensemble of deep feature sets is employed. They achieve values of 82.6% and 84.7% in terms of accuracy, in addition to values of 0.926 and 0.926 in terms of AUC on Montgomery and Shenzhen datasets, respectively.

In [27], both of using handcrafted (i.e., local and global feature descriptors such as GIST and HOG) and deep features are proposed to increase TB visual recognition performance. The authors also make use of a stacking generalization of classifiers to improve accuracy. In this regard, the SVM and the logistic regression (LR) classifiers are used as a base learner and meta learner, respectively. They reach promising values of 87.5% and 93.4% in terms of accuracy, in addition to values of 0.962 and 0.991 in terms of AUC on Montgomery and Shenzhen datasets concerning the results of state-of-the-art methods, respectively.

In [39], hybridization of extracted large-scale of deep features from diverse pre-trained CNNs and handcrafted features is combined with a feature selection algorithm (i.e., particle swarm optimization (PSO)). Then, the features are given into an optimized SVM classifier with the Bayesian optimization algorithm. They also use CLAHE preprocessing method to improve image contrast and quality. Their study achieves state-of-the-art results on commonly used two TB detection image datasets (i.e., Montgomery and Shenzhen). These values are 92.7% and 95.5% in terms of accuracy, and 0.995 and 0.995 in terms of AUC on Montgomery and Shenzhen datasets.

In [34], six voting combination rules are applied (namely weighted probabilities, the product of probabilities, maximum probability, the average of probabilities, minimum probability, and median) for ensemble learning of fine-tuned CNN models on food image recognition datasets for the obesity problem. The author reaches outstanding image classification/recognition results on the three datasets used.

As expressed in the related studies and introduction section, there are many variations and applications of convolutional neural networks in medical image

classification. However, as far as we know, there are no published studies considering and analyzing the results of fusing the various fine-tuned CNN models by focusing on different combinations of augmentation and preprocessing techniques on TB detection.

3 Methods

In this section, firstly, the overview of the proposed method is presented. Then, we describe various image preprocessing techniques (namely CLAHE and image data augmentations), the used pre-trained CNN models (namely InceptionV3 and Xception), fine-tuning of the CNN models. Finally, we also explain the voting-based ensemble learning processes in the following subsections in detail.

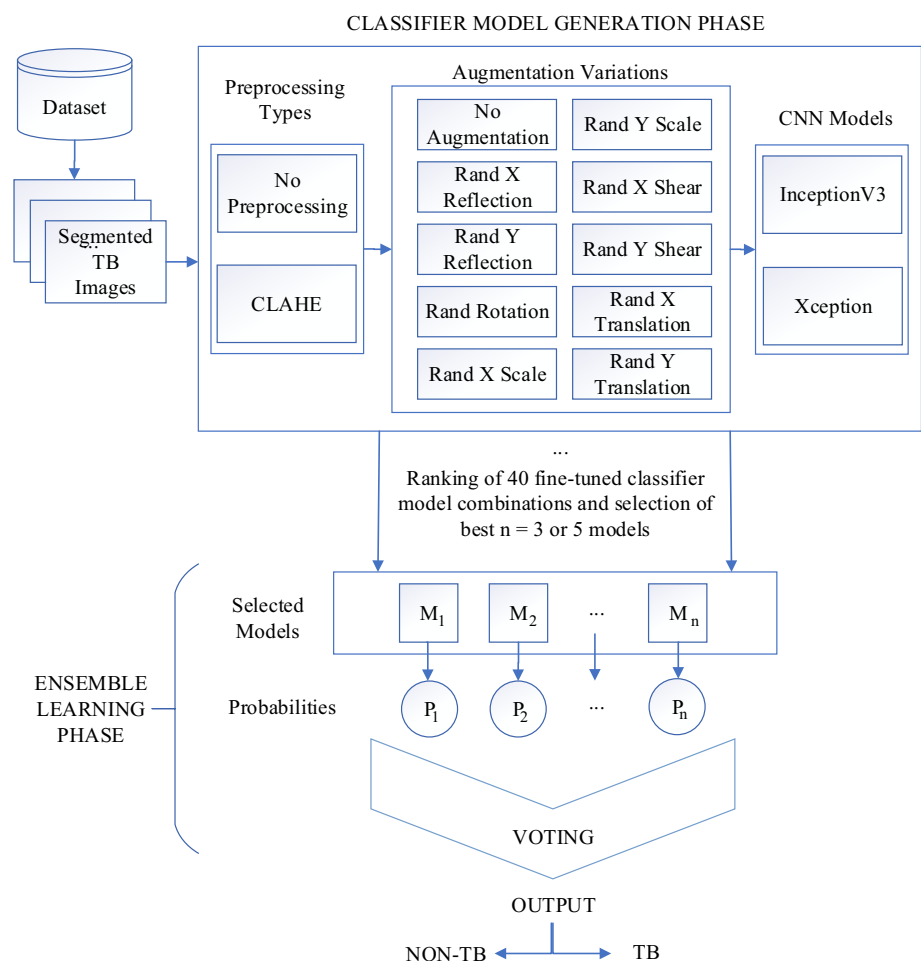
3.1 The overview of the proposed method

The overview of the proposed method that utilizes diverse preprocessing and pre-trained CNN models is shown in Fig. 1. The method consists of two stages: (1) Classifier

model generation based on InceptionV3 and Xception and (2) Ensemble learning with selected models according to their performance values.

First of all, TB CXR datasets and predefined segmented CXR images are obtained and prepared for further operations. Then, the classifier generation (i.e., fine-tuned CNN) phase is implemented with the specified options on the training set of images. The method consists of three main variation types for fine-tuning: (1) preprocessing types (2) transformations for image data augmentation (3) pre-trained CNNs. After the model generation process, 40 fine-tuned classifier combinations are generated, the ranking of all models and selection of best $n = 3$ or five models are yielded according to the performance measure (i.e., accuracy rate). Furthermore, the ensemble of several the best models is employed in the voting-based ensemble learning process to achieve the final classification task on the datasets used. Each stage is described in the following subsections in detail.

Fig. 1 The overview of the proposed method



3.2 Image preprocessing

Image data preprocessing is a series of the operations related to facilitating of further processing process. For example, removing noise of image, improving the quality, image resizing, data augmentation, histogram equalization, and contrast operations (e.g., CLAHE). The preprocessing stage assists the following related stages such as segmentation, model construction, and classification to improve performance.

3.2.1 CLAHE

CLAHE, known as contrast-limited adaptive histogram equalization, refers to a type of adaptive contrast enhancement method. CLAHE, developed by Pizer et al. [25, 26], is based on adaptive histogram equalization (AHE) that computes diverse histograms for each distinct subregion of an image. AHE is useful for enhancing edges of each distinct image regions and improving local contrast.

For CLAHE, the enhancement computation is updated by getting a user-specified maximum clip level value, and thus on the maximum contrast enhancement factor [24]. Then, the neighboring image regions are merged with bilinear interpolation to eliminate artificially stimulated boundaries of regions [43]. This method is especially suitable for medical images to increase image quality and contrast [24, 39]. CLAHE operation on an image of the Montgomery TB dataset is shown in Fig. 2.

3.2.2 Image data augmentation

One of the important image preprocessing techniques is image data augmentation that synthetically increases all of

the size, diversity, and quality of training images without needing additional memory for storage on deep learning applications. As one of the common regularization techniques, data augmentation reduces the risk of model overfitting and poor performance in the process of deep learning model construction [4]. This process is yielded by applying different input transformations that keep corresponding output labels.

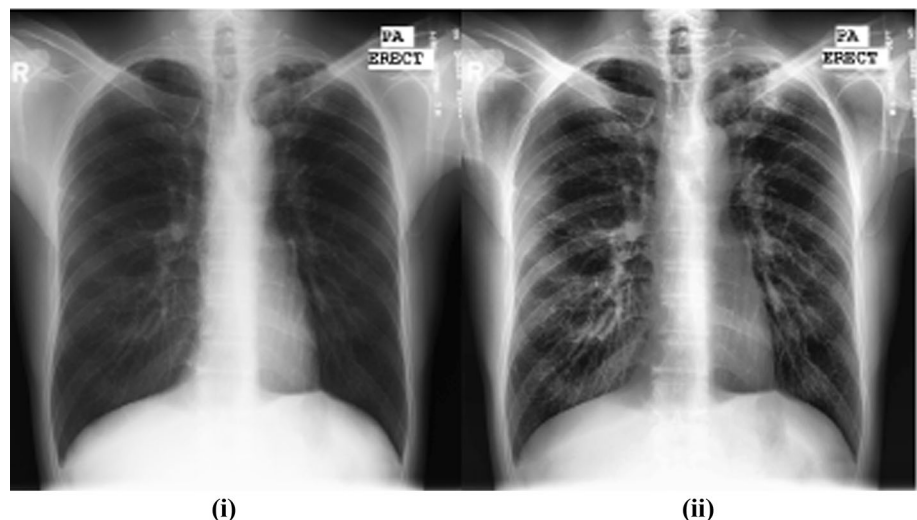
The common transformations for data augmentation can be implemented in various ways: (1) reflection at x or y axis; (2) rotation at some degrees; (3) scaling horizontally or vertically; (4) shearing at x or y axis (5) translation at x or y axis. The task of image data augmentation is used to take into account these several invariances in addition to the image dataset. So the final learning models will perform well despite these challenges [30]. Representative examples of a TB image for the Montgomery dataset [1, 15] are illustrated in Fig. 3.

Image augmentations have also been observed to improve convergence, generalization ability, and robustness of samples and have more advantages compared to other regularization techniques [4, 12]. The limited size of datasets is a particularly widespread case in the field of medical image analysis because of expensive and labor-intensive procedures to collect [30].

3.3 Pre-trained CNN models

A pre-trained CNN is a type of network which is trained on a large-scale benchmark dataset to solve a problem similar to the handling of our related task. InceptionV3 and Xception are two of the examples of pre-trained network models [17]. Pre-trained CNNs allow to use a trained model as a starting point for different analogous problems instead of training a model from scratch. Therefore, they

Fig. 2 Example of the CLAHE operation on an image of the Montgomery dataset (i) Original image (ii) The image after CLAHE operation



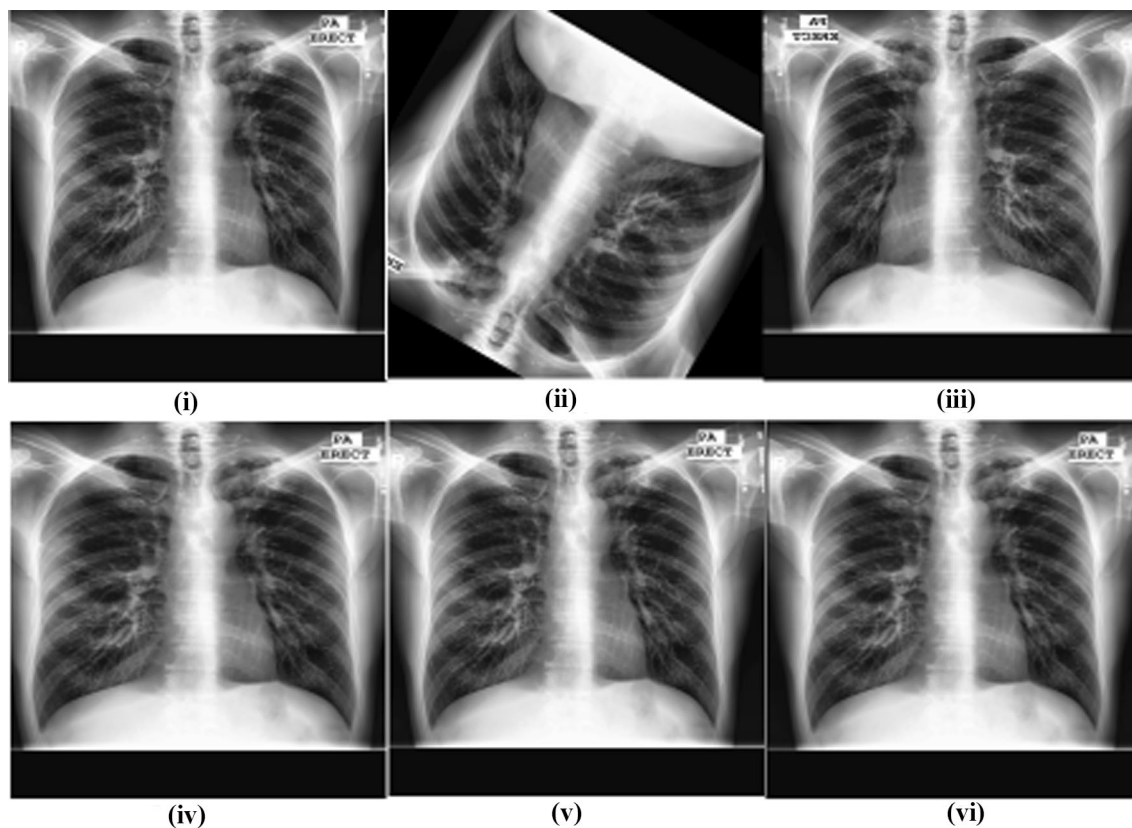


Fig. 3 Examples of the transformation for data augmentation of an image of the Montgomery dataset (i) The image (ii) Rotation (iii) Reflection (iv) Scaling (v) Shearing (vi) Translation

can provide speed, time, and performance efficiency for the corresponding process.

InceptionV3 and Xception are the pre-trained network models trained on millions of images from the ImageNet database [14]. InceptionV3 and Xception networks are 48 and 71 layers deep, respectively, and they require an image with input size of 299-by-299. While Inception considers the problem of representational congestion and yields efficient results with utilizing asymmetric filters and bottleneck layer and replacing large-size filters with small filters [17, 32], Xception gives easier and more efficient results by independently applying cross-channel correlations and spatial correlations [8]. Depth-wise separable convolution is also proposed and the use of cardinality to learn better abstractions is executed for Xception model [17].

3.4 Fine-tuning

In transfer learning, we first train a base model on a primary dataset/problem, and then we reuse the deep features, or transfer them, to a second target model which will be trained on a target dataset/problem as in [41]. Fine-tuning is the most common approach to transfer learning and

improves the generalization ability of the model used [5]. To this end, the weights of the pre-trained CNN models are fine-tuned by continuing the backpropagation operation.

The main approach for fine-tuning is to remove the last fully connected layer of selected pre-trained CNN models and modify them with our new fully connected layer (i.e., the same size as the number of classes in our new dataset). In this study, we used two classes due to TB and non-TB cases of image datasets.

3.5 Soft voting

In soft voting, the probability score-vector is used for vote aggregation instead of class labels in hard/majority voting for classifier ensemble. The output class is determined by the combination rule (e.g., the average of probabilities) used. This approach provides more flexible and fine-grained results than majority voting due to handling probability scores.

3.6 Bayesian optimization-based weighted voting

In the case of weighted voting, the prediction scores are weighted by the classifier's importance level and summed up. Then, the target class with the greatest score (i.e., the sum of the weighted probabilities) wins the vote. Accordingly, the weights of models in the ensemble scheme should change among the different output classes in each classifier according to its performance for getting better results [42]. The weighting process can be observed as an optimization problem to select appropriate weights for each classifier. It can be carried out with various optimization algorithms such as Bayesian optimization.

For the Bayesian optimization-based weighted voting, the Bayesian optimization algorithm is applied to weighting optimization. The Bayesian optimization algorithm is an influential and iterative strategy for finding the extrema of high-cost objective functions globally to evaluate [3, 34]. The optimization techniques are one of the most efficient approaches in terms of the number of function evaluations needed [3]. The Bayesian optimization algorithm is given in Algorithm 1 [21, 34]. As viewed in Algorithm 1, the acquisition function determines new x points for evaluation.

Algorithm 1: Bayesian optimization algorithm.

```
//This algorithm executes the Bayesian Optimization algorithm by
minimizing the objective function f(x)
Get  $x_i$  points randomly within the given variable bounds
Compute the prior over the objective function  $f(x_i)$ .
for  $i=1$ :num of Iterations
    select next  $x_i$  point by maximizing the acquisition function  $a(x)$  for
    evaluation.
    Update the Gaussian statistical model of  $f(x)$  for acquiring a
    posterior distribution
end
return  $x$ 
```

In this study, the weights of the selected fine-tuned CNN models are set in the range of 0–1 for Bayesian optimization-based weighted voting. Each fine-tuned CNN has a class probability score for the related image dataset to classify images. The scores are multiplied with the weights of the CNNs. Then, the summing of products is calculated for the weighting process. The weights are identified randomly with the Bayesian optimization algorithm. Finally, the output class is determined according to the max probability index. The Bayesian optimization algorithm is run 100 iterations with default parameter values. The equation is shown in Eq. (1). In this study, n value can be 3 or 5 as illustrated in Eq. (1).

$$\text{WeightedVoting} = w_1 * CNN_1 + \dots + w_n * CNN_n \quad (1)$$

where w_n and CNN_n represent the weight and the probability score of the selected fine-tuned CNN, respectively.

4 Experimental work

In this section, we describe the experimental procedures, image datasets, and evaluation metrics. Then, we present comprehensive computational results with respect to the performance metrics in the following sections in detail. Finally, we compare and discuss our results with the state-of-the-methods on TB detection problems.

4.1 Experimental process

All the experiments were performed using Matlab R2019b software and a desktop computer with the configuration of Intel®Core i7 8700K CPU with 3.70 GHz, 64 GB RAM, and 8 GB NVIDIA GeForce GTX 1080 GPU Memory. We adjust the train-test ratio as 80-20 for all the experiments (namely fine-tuning and voting schemes). We set the CPU random number generator seed to 1 for fine-tuning of CNN models and Bayesian optimization on all TB image datasets used. We also used 10 different seeds (i.e., 1–10) and train-test split for voting-based ensemble schemes. In the soft voting process, we used an average of the probabilities as the combination rule. Three and five best-fine-tuned models are considered on soft voting and Bayesian optimization-based weighted voting approaches.

As preliminary testing, we aimed to decrease the combination sets thanks to selecting the best two models. To accomplish this, we tried to apply Alexnet, VGGNet, GoogleNet, InceptionV3, and Xception CNN models with various combinations on these datasets in terms of accuracy rate. Then, we decided to select InceptionV3 and Xception as the best two CNN models for the problem based on the success of the results (i.e., accuracy rate) and advanced network models in the literature [17].

In this study, we choose to use CLAHE preprocessing or not to apply preprocessing before image resizing. After that, we apply ten different image data augmentation variations consisting of no augmentation, reflection, rotation, scaling, shearing, and translation for fine-tuning of

Table 1 Types of image data augmentation and parameter values

Transformation	Value
RandXReflection	True
RandYReflection	True
RandRotation	[0 360]
RandXScale	[1 2]
RandYScale	[1 2]
RandXShear	[0 30]
RandYShear	[0 30]
RandXTranslation	[1 3]
RandYTranslation	[1 3]

Table 2 Generated combination sets for the experimental process

Set	Minibatch	CNN Model	Preprocessing	Augmentation
1	64	Inceptionv3	No preprocessing	No augmentation
2	64	Inceptionv3	No preprocessing	No augmentation
3	64	Inceptionv3	No preprocessing	RandXReflection
4	64	Inceptionv3	No preprocessing	RandYReflection
5	64	Inceptionv3	No preprocessing	RandRotation
6	64	Inceptionv3	No preprocessing	RandXScale
7	64	Inceptionv3	No preprocessing	RandYScale
8	64	Inceptionv3	No preprocessing	RandXShear
9	64	Inceptionv3	No preprocessing	RandYShear
10	64	Inceptionv3	No preprocessing	RandXTranslation
11	64	Inceptionv3	No preprocessing	RandYTranslation
12	64	Inceptionv3	CLAHE	No augmentation
13	64	Inceptionv3	CLAHE	RandXReflection
14	64	Inceptionv3	CLAHE	RandYReflection
15	64	Inceptionv3	CLAHE	RandRotation
16	64	Inceptionv3	CLAHE	RandXScale
17	64	Inceptionv3	CLAHE	RandYScale
18	64	Inceptionv3	CLAHE	RandXShear
19	64	Inceptionv3	CLAHE	RandYShear
20	64	inceptionv3	CLAHE	RandXTranslation
21	64	Inceptionv3	CLAHE	RandYTranslation
22	16	Xception	No preprocessing	No augmentation
23	16	Xception	No Preprocessing	No augmentation
24	16	xception	No preprocessing	RandXReflection
25	16	Xception	No preprocessing	RandYReflection
26	16	Xception	No preprocessing	RandRotation
27	16	Xception	No preprocessing	RandXScale
28	16	Xception	No preprocessing	RandYScale
29	16	Xception	No preprocessing	RandXShear
30	16	Xception	No preprocessing	RandYShear
31	16	Xception	No preprocessing	RandXTranslation
32	16	Xception	No preprocessing	RandYTranslation
33	16	Xception	CLAHE	No augmentation
34	16	Xception	CLAHE	RandXReflection
35	16	Xception	CLAHE	RandYReflection
36	16	Xception	CLAHE	RandRotation
37	16	Xception	CLAHE	RandXScale
38	16	Xception	CLAHE	RandYScale
39	16	Xception	CLAHE	RandXShear
40	16	Xception	CLAHE	RandYShear
41	16	Xception	CLAHE	RandXTranslation
42	16	Xception	CLAHE	RandYTranslation

CNN models used. These transformations and related parameter values are given in Table 1. After selecting preprocessing type, all images are resized into the input size of used CNN model (i.e., 299 * 299). Then, if the augmentation option is active in the related variation, the

selected data augmentation technique is applied with the same size of the training images as the dataset. So, the size of training images becomes two times the size of the training set of the dataset.

Table 3 Fine-tuning times and experimental results of the combination sets for the Montgomery, Shenzhen datasets, and mean values in terms of accuracy rate

Set	Montgomery FT Time (mm:ss)	Shenzhen FT Time (mm:ss)	Montgomery ACC (%)	Shenzhen ACC (%)	Mean datasets ACC (%)
1	03 m 15 s	11 m 51 s	67.8571	88.4956	78.1764
2	00 m 58 s	06 m 43 s	67.8571	84.9558	76.4065
3	02 m 54 s	13 m 32 s	82.1429	89.3805	85.7617
4	02 m 53 s	13 m 26 s	75.0000	86.7257	80.8629
5	02 m 53 s	13 m 22 s	82.1429	90.2655	86.2042
6	02 m 52 s	13 m 26 s	78.5714	86.7257	82.6486
7	02 m 53 s	13 m 48 s	75.0000	85.8407	80.4204
8	02 m 54 s	13 m 33 s	82.1429	88.4956	85.3193
9	02 m 54 s	13 m 32 s	78.5714	84.9558	81.7636
10	03 m 00 s	13 m 26 s	82.1429	86.7257	84.4343
11	02 m 56 s	13 m 46 s	78.5714	85.8407	82.2061
12	00 m 58 s	07 m 07 s	75.0000	87.6106	81.3053
13	02 m 56 s	13m 44s	82.1429	89.3805	85.7617
14	02 m 54 s	13 m 35 s	85.7143	88.4954	87.1049
15	02 m 56 s	13 m 25 s	85.7143	85.8407	85.7775
16	02 m 54 s	13 m 29 s	92.8571	87.6106	90.2339
17	02 m 54 s	13 m 26 s	82.1429	87.6106	84.8768
18	02 m 58 s	13 m 27 s	89.2857	89.3805	89.3331
19	02 m 54 s	13 m 49 s	82.1429	88.4956	85.3193
20	02 m 57 s	13 m 33 s	85.7143	88.4956	87.1050
21	02 m 56 s	13 m 36 s	89.2857	89.3805	89.3331
22	05 m 48 s	15 m 50 s	82.1429	89.3805	85.7617
23	02 m 17 s	10 m 35 s	85.7143	88.4956	87.1050
24	04 m 57 s	21 m 12 s	89.2857	87.6106	88.4482
25	04 m 56 s	21 m 11 s	85.7143	85.8407	85.7775
26	04 m 56 s	21 m 12 s	85.7143	87.6106	86.6625
27	04 m 58 s	21 m 12 s	82.1429	89.3805	85.7617
28	04 m 54 s	21 m 11 s	82.1429	85.8407	83.9918
29	05 m 00 s	21 m 14 s	92.8571	85.8407	89.3489
30	04 m 59 s	21 m 11 s	85.7143	89.3805	87.5474
31	04 m 56 s	21 m 27 s	71.4286	85.8407	78.6347
32	04 m 54 s	21 m 13 s	82.1429	89.3805	85.7617
33	02 m 1 6s	10 m 37 s	89.2857	88.4956	88.8907
34	04 m 59 s	21 m 17 s	85.7143	87.6106	86.6625
35	04 m 56 s	21 m 11 s	82.1429	87.6106	84.8768
36	04 m 57 s	21 m 14 s	78.5714	88.4956	83.5335
37	04 m 57 s	21 m 09 s	85.7143	89.3805	87.5474
38	05 m 00 s	21 m 09 s	89.2857	87.6106	88.4482
39	05 m 01 s	21 m 24 s	78.5714	86.7257	82.6486
40	04 m 58 s	21 m 11 s	78.5714	88.4956	83.5335
41	04 m 57 s	21 m 10 s	85.7143	88.4956	87.1050
42	04 m 56 s	21 m 20 s	89.2857	88.4956	88.8907

In the fine-tuning process, we used stochastic gradient descent with momentum (SGDM) optimizer for training the network. We determined the minibatch size as 64 and 16 for InceptionV3 and Xception, respectively, due to GPU memory limitations. The maximum number of epochs is set

to 30 for all CNN model combinations. For other training options, default values are assigned to the corresponding locations.

4.2 Image datasets

We utilized Montgomery County and Shenzhen TB CXR image datasets to evaluate the effectiveness of the proposed method [1]. Montgomery dataset CXR images have been obtained from the tuberculosis control program of the Department of Health and Human Services of Montgomery County, MD, USA [1]. This dataset consists of 138 posterior-anterior CXRs, of which 80 CXRs are normal and 58 X-rays are abnormal with signs of tuberculosis. Shenzhen dataset images have been acquired by Shenzhen No.3 Hospital in Shenzhen, China [1]. There are 326 normal CXRs and 336 abnormal CXRs showing signs of tuberculosis.

In this study, we use the predefined lung masks before the deep learning model construction and voting processes. For the Montgomery dataset, obtaining the lung masks is easy because of the fact that spanning segmentation masks in the dataset description. However, the Shenzhen dataset has no segmentation masks. Therefore, the segmented lungs masks are employed from [16, 31] for this dataset. In this case, we used 566 images of which 279 CXRs are normal and 287 CXRs are abnormal with manifestations of tuberculosis for Shenzhen dataset.

4.3 Evaluation metrics

To evaluate the predictive performance of the proposed approach, we employed two different evaluation measures including the classification accuracy rate (ACC) and Area Under ROC Curve (AUC).

Classification accuracy, shown in Eq. (2), is calculated by dividing the total of true positives and true negatives by the total number of false negatives, true negatives, true positives, and false positives (i.e., instances).

$$ACC = \frac{TN + TP}{TP + TN + FN + FP} \quad (2)$$

where FN, TN, TP, and FP, represent the number of false negatives, true negatives, true positives, and false positives, respectively.

AUC represents the area under the receiver operating characteristic (ROC) curve for the classification performance. ROC charts are two-dimensional graphics in which the TP ratio is plotted on the Y-axis and the FP ratio is plotted on the X-axis. The total area is computed as the sum of the trapezoids' areas by applying numerical integration on the ROC Curve. Trapezoids are used instead of rectangles in order to mean the effect between points [10]. The area value of 1 as max AUC value provides a perfect test [10], the area value of 0 as min AUC value shows that the learned model categorizes all instances incorrectly.

4.4 Computational results

For our approach, we produced the combination sets of preprocessing, minibatch size, CNN model, preprocessing type, and augmentation type for possible different cases of our experimental setup. The combination sets are illustrated in Table 2. The detailed experimental results of the proposed preprocessing and augmentation-based fine-tuning method for TB detection are given in Table 3 for Montgomery, Shenzhen, and mean datasets, respectively. While the columns in Table 2 represent the number of combination sets, minibatch size, CNN model, preprocessing type, and augmentation type, the columns in Table 3 represent the number of combination sets, fine-tuning time, and accuracy (%) for the related dataset. The bold values demonstrate the best values for each method utilized. For Table 3, sets 1–2 and sets 22–23 specify the results whether batch preprocessing operations are applied

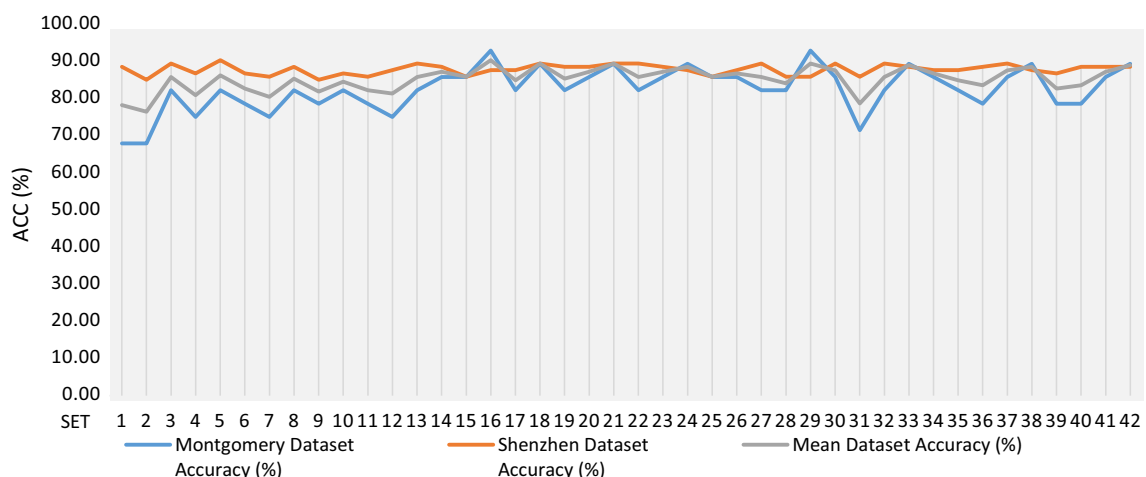


Fig. 4 A graphical representation of ACC values according to all variations in fine-tuning process for datasets

Table 4 Soft voting and Bayesian optimization-based weighted voting results according to Montgomery, Shenzhen, and both mean datasets' accuracy (%)

Montgomery ACC-based							
Soft voting	Dataset	Number of votes	Mean accuracy (%)	SD (%)	MeanAUC	SD	
10 seed	Montgomery	3	96.7857	3.5515	0.9880	0.0208	
	Montgomery	5	96.7857	3.5515	0.9880	0.0208	
Weighted voting	Dataset	Number of weights	Mean accuracy (%)	SD (%)	Mean AUC	SD	
10 seed	Montgomery	3	95.3571	4.7768	0.9875	0.0223	
Bayesian optimization-based	Montgomery	5	96.0714	3.9304	0.9865	0.0252	
Shenzhen ACC-based							
Soft voting	Dataset	Number of votes	Mean accuracy (%)	SD (%)	Mean AUC	SD	
10 seed	Shenzhen	3	97.6106	2.7688	0.9946	0.0121	
	Shenzhen	5	97.5221	2.8232	0.9939	0.0139	
Weighted voting	Dataset	Number of weights	Mean accuracy (%)	SD (%)	Mean AUC	SD	
10 seed	Shenzhen	3	97.6991	2.6119	0.9945	0.1250	
Bayesian optimization-based	Shenzhen	5	97.6106	2.5733	0.9936	0.0138	
Mean datasets ACC-based							
Soft voting	Dataset	Number of votes	Mean accuracy (%)	SD (%)	Mean AUC	SD	
10 seed	Montgomery	3	97.5000	3.3882	0.9891	0.0189	
	Montgomery	5	96.0714	4.2758	0.9859	0.0228	
	Shenzhen	3	97.2566	2.9631	0.9925	0.0157	
	Shenzhen	5	97.5221	2.8232	0.9910	0.0177	
Weighted voting	Dataset	Number of weights	Mean accuracy (%)	SD (%)	Mean AUC	SD	
10 seed	Montgomery	3	95.7143	4.0546	0.9870	0.0232	
Bayesian optimization-based	Montgomery	5	95.7143	4.0546	0.9839	0.0247	
	Shenzhen	3	97.4336	3.3491	0.9862	0.0209	
	Shenzhen	5	97.4336	3.3491	0.9889	0.0191	

collectively before the fine-tuning operation. If these operations (e.g., CLAHE and image resizing) are applied completely before the fine-tuning operation, it is observed that the frequency of GPU-CPU switch and context switch times decreases. In this case, time efficiency will improve significantly (approximately between 3 or 10 times faster concerning the used preprocessing technique and dataset size). After some preliminary tests, we utilized this efficient approach as well for the remaining combination sets on two TB image datasets.

4.4.1 Experimental results of individual classification models (with preprocessing)

In this subsection, we give comprehensive results to construct the learners (i.e., fine-tuning of CNN models) by implementing all image preprocessing variations (i.e., 40 per dataset) on TB detection. To observe the effective fine-tuning process, we also add combination sets #1 and #22 (i.e., CPU-GPU usage with batch slowly operations) for time evaluation in Table 3.

Table 3 reports the performance results including accuracy rates and fine-tuning (FT) times according to the Montgomery, Shenzhen, and mean datasets for the variations in CNN models, preprocessing types, and

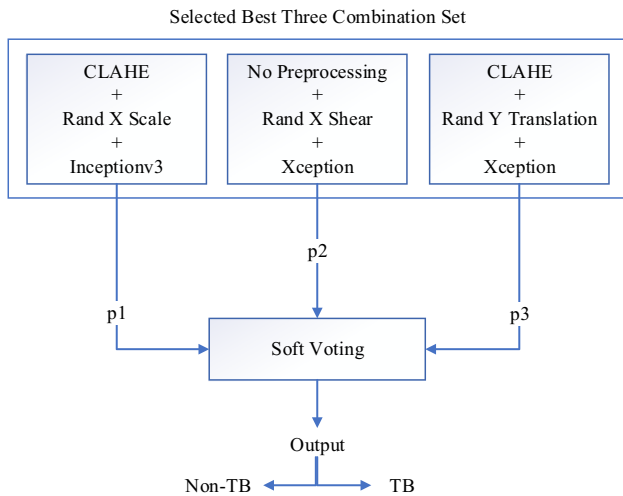


Fig. 5 The diagram of the best voting scheme for the Montgomery dataset

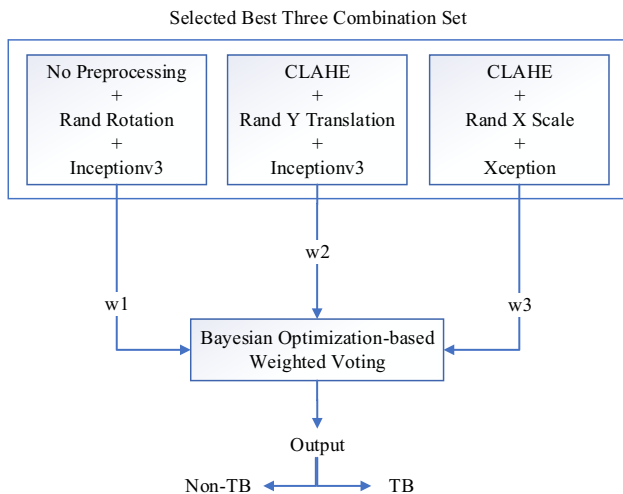


Fig. 6 The diagram of the best voting scheme for the Shenzhen dataset

augmentation methods. We obtained the best results from sets 16 and 29 as 92.8571% ACC values with InceptionV3 + CLAHE + RandXScale and Xception + No Preprocessing + RandXShear fine-tuning combinations. The fine-tuning time on this dataset takes between 3 minutes and 5 minutes per model for InceptionV3 and Xception CNNs, respectively.

We acquired the best result from fifth set as 90.2655% ACC value with InceptionV3 + No Preprocessing + RandRotation combination scheme. The fine-tuning time on this dataset takes between 13 and 21 min per model for InceptionV3 and Xception CNNs, respectively. The main reason behind this is the fact that Shenzhen has a larger size than the Montgomery dataset.

We also computed mean datasets’ accuracy to figure out the most important combination schemes on both datasets for this study. Last column of Table 3 shows the performance results (i.e., accuracy rate) according to the mean accuracy of Montgomery and Shenzhen datasets. In this case, we got the best result from a set of 16 as 90.2339% ACC value with InceptionV3 + CLAHE + RandXScale combination scheme. A graphical chart of ACC values according to all variations in the fine-tuning process for both datasets is illustrated in Fig. 4.

4.4.2 Voting-based experimental results

In this subsection, we give extensive soft voting, and the Bayesian optimization-based weighted voting results in terms of ACC and AUC values in detail. We also implemented the voting of three and five fine-tuned models. Furthermore, we handled these voting approaches according to the accuracy performance of mean datasets. As it can be observed from Table 4, the best mean ACC and AUC are obtained as values of 97.5000% and 0.9891 on the Montgomery dataset by applying soft voting of three best fine-tuned models (namely the ensemble of (InceptionV3 + CLAHE + RandXScale; Xception + No Preprocessing + RandXShear;

Table 5 Comparison of our proposed method with state-of-the-art methods on TB detection for two CXR image datasets used

Study	Method	Montgomery ACC (%)	Montgomery AUC	Shenzhen ACC (%)	Shenzhen AUC
Our study, 2021	VoPreCNNFT	97.500	0.989	97.699	0.994
Ayaz et al. [2]	HCDEL	93.470	0.970	97.590	0.990
Win et al. [39]	HDHFS	92.700	0.995	95.500	0.995
Xie et al. [40]	FRCNN	92.600	0.977	90.200	0.941
Rajaraman et al. [27]	SLMHDF	87.500	0.962	93.400	0.991
Santosh and Antani [29]	SETFV	83.000	0.900	91.000	0.960
Lopes and Valiati [20]	EDFPCNN	82.600	0.926	84.700	0.926
Pasa et al. [23]	OptCNN	79.000	0.811	84.400	0.900
Hwang et al. [13]	PreACNNF	67.400	0.884	83.700	0.926

InceptionV3 + CLAHE + RandYTranslation)) and using ten different seed values. Additionally, the best mean ACC is taken from as 97.6991% value by carrying out the Bayesian optimization-based weighted voting scheme of three best fine-tuned CNN models for the Shenzhen dataset. Furthermore, the best mean AUC is reached as 0.994 value using soft voting of three best-fine-tuned models for this dataset. The best voting schemes for related datasets are presented in Figs. 5 and 6.

4.5 Comparison with state-of-the-art methods for TB detection

We compare the performance of the proposed voting and preprocessing-based fine-tuned CNN model approach (VoPreCNNFT) with other state-of-the-art TB detection algorithms on the two image datasets used in this subsection.

State-of-the-art methods that we selected are handcrafted and deep features with ensemble learning (HCDEL) of [2]; hybrid deep and handcrafted features with feature selection (HDHFS) of [39]; faster region-based convolutional network (FRCNN) of [40]; stacked learning model with handcrafted and deep features (SLMHDF) of [27]; shape, edge, and texture-based features with a voting model (SETFV) [29]; the ensemble of deep features using pre-trained CNNs (EDFPCNN) [20]; an optimized CNN model (OptCNN) [23]; and pre-trained AlexNet CNN features (PreACNNF) [13].

Table 5 presents the accuracy rates in percentage and AUC values per algorithm/dataset pair. By considering obtained accuracy rate values, our method outperforms all its competitors with 97.500% and 97.699% for Montgomery and Shenzhen datasets, respectively. Although [39] uses only one train-test set, we used 10 different train-test sets by utilizing 10 seed values (from 1 to 10) and we also obtained final ACC and AUC results by averaging the scores. We achieved the best results according to ACC and the second-best results according to AUC values as 0.9891 and 0.9940 after [39] for Montgomery and Shenzhen datasets, respectively. Using our method including voting and preprocessing-based fine-tuned CNN models improved the TB detection performance significantly.

5 Conclusion

This study proposes a voting-based ensemble deep learning approach using diverse preprocessing/data augmentation variations on TB detection image datasets. Forty different variations are carried out to fine-tune CNN models according to preprocessing methods, types of image augmentation, and pre-trained CNNs used. In this way, we

extensively highlight the effects of various image preprocessing techniques on the fine-tuning process in this study. An effective fine-tuning process is also implemented by applying all preprocessing operations as a whole on the images before training of CNNs.

The proposed voting-based method utilizes both basic soft voting and weighted voting methods for combining the best result to achieve better performance results. Bayesian optimization, the best-known optimization algorithm for machine learning-based problems, is employed for weighted voting process. To observe the effect of the number of models in voting operations, both three and five learned models are employed.

Although the usage of CNN models is required GPU and CPU resources, fine-tuning is useful and crucial for image classification/recognition tasks without needing training from scratch. Ensemble methods provide outstanding results by using various CNN models but there is a speed-performance trade-off strategy for this purpose. The proposed voting and preprocessing-based approach can be also utilized for other image recognition (e.g., disease classification, object recognition) problems. As a future direction of this work, user-defined (i.e., expert radiologists) datasets can be obtained from the hospitals, and the proposed method can be tested on these datasets.

Acknowledgements This study is supported by Ege University Scientific Research Projects Coordination Unit. Project Number: 18-MUH-001.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Antani S (2020) Tuberculosis chest x-ray image data sets. <https://lhncbc.nlm.nih.gov/publication/pub9931>. Accessed April 2021
2. Ayaz M, Shaukat F, Raja G (2021) Ensemble learning based automatic detection of tuberculosis in chest x-ray images using hybrid feature descriptors. *Phys Eng Sci Med*. <https://doi.org/10.1007/s13246-020-00966-0>
3. Brochu E, Cora VM, De Freitas N (2010) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint [arXiv:10122599](https://arxiv.org/abs/10122599)
4. Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA (2020) Albumentations: fast and flexible image augmentations. *Information* 11(2):125. <https://doi.org/10.3390/info11020125>
5. Cayamcela MEM, Lim W (2019) Fine-tuning a pre-trained convolutional neural network model to translate American sign language in real-time. In: 2019 international conference on

- computing, networking and communications (ICNC), IEEE, pp 100–104. <https://doi.org/10.1109/ICCNC.2019.8685536>
6. CDC (2020) Tuberculosis (tb). <https://www.cdc.gov/tb/topic/basics/default.htm>. Accessed April 2021
 7. Chauhan A, Chauhan D, Rout C (2014) Role of gist and phog features in computer-aided diagnosis of tuberculosis without segmentation. *PLoS ONE* 9(11):e112980. <https://doi.org/10.1371/journal.pone.0112980>
 8. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258. <https://doi.org/10.1109/CVPR.2017.195>
 9. Dong X, Yu Z, Cao W, Shi Y, Ma Q (2020) A survey on ensemble learning. *Front Comput Sci*. <https://doi.org/10.1007/s11704-019-8208-z>
 10. Fawcett T (2006) An introduction to roc analysis. *Pattern Recognit Lett* 27(8):861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
 11. Hernández A, Panizo Á, Camacho D (2019) An ensemble algorithm based on deep learning for tuberculosis classification. In: International conference on intelligent data engineering and automated learning. Springer, pp 145–154. https://doi.org/10.1007/978-3-030-33607-3_17
 12. Hernández-García A, König P (2018) Further advantages of data augmentation on convolutional neural networks. In: International conference on artificial neural networks. Springer, pp 95–103. https://doi.org/10.1007/978-3-030-01418-6_10
 13. Hwang S, Kim HE, Jeong J, Kim HJ (2016) A novel approach for tuberculosis screening based on deep convolutional neural networks. In: Medical imaging 2016: computer-aided diagnosis, international society for optics and photonics, vol 9785, p 97852W. <https://doi.org/10.1117/12.2216198>
 14. ImageNet (2016) Imagenet database. <http://www.image-net.org>. Accessed April 2021
 15. Jaeger S, Candemir S, Antani S, Wáng YXJ, Lu PX, Thoma G (2014) Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quant Imaging Med Surg* 4(6):475. <https://doi.org/10.3978/j.issn.2223-4292.2014.11.20>
 16. Kaggle (2018) U-net lung segmentation (montgomery + shenzhen). <https://www.kaggle.com/eduardomineo/u-net-lung-segmentation-montgomery-shenzhen/data>. Accessed April 2021
 17. Khan A, Sohail A, Zahoor U, Qureshi AS (2020) A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev*. <https://doi.org/10.1007/s10462-020-09825-6>
 18. Kumar A, Kim J, Lyndon D, Fulham M, Feng D (2016) An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE J Biomed Health Inform* 21(1):31–40. <https://doi.org/10.1109/JBHI.2016.2635663>
 19. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
 20. Lopes U, Valiati JF (2017) Pre-trained convolutional neural networks as feature extractors for tuberculosis detection. *Comput Biol Med* 89:135–143. <https://doi.org/10.1016/j.combiomed.2017.08.001>
 21. MathWorks (2020) Bayesian optimization algorithm. <https://www.mathworks.com/help/stats/bayesian-optimization-algorithm.html>. Accessed April 2021
 22. Noor NM, Rijal O, Fah CY (2002) Wavelet as features for tuberculosis (mtb) using standard x-ray film images. In: 6th international conference on signal processing, 2002, vol 2. IEEE, pp 1138–1141. <https://doi.org/10.1109/ICOSP.2002.1179990>
 23. Pasa F, Golkov V, Pfeiffer F, Cremers D, Pfeiffer D (2019) Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization. *Sci Rep* 9(1):1–9. <https://doi.org/10.1038/s41598-019-42557-4>
 24. Pisano ED, Zong S, Hemminger BM, DeLuca M, Johnston RE, Muller K, Braeuning MP, Pizer SM (1998) Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *J Digit Imaging* 11(4):193. <https://doi.org/10.1007/bf03178082>
 25. Pizer SM (1986) Psychovisual issues in the display of medical images. In: Höhne KH (ed) Pictorial information systems in medicine. Springer, Berlin, pp 211–233. https://doi.org/10.1007/978-3-642-82384-8_6
 26. Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, ter Haar Romeny B, Zimmerman JB, Zuiderveld K (1987) Adaptive histogram equalization and its variations. *Comput Vis Graph Image Process* 39(3):355–368. [https://doi.org/10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X)
 27. Rajaraman S, Candemir S, Xue Z, Alderson PO, Kohli M, Abuya J, Thoma GR, Antani S (2018) A novel stacked generalization of models for improved tb detection in chest radiographs. In: 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, pp 718–721. <https://doi.org/10.1109/EMBC.2018.8512337>
 28. Sagi O, Rokach L (2018) Ensemble learning: a survey. *Wiley Interdiscip Rev Data Min Knowl Discov* 8(4):e1249. <https://doi.org/10.1002/widm.1249>
 29. Santosh K, Antani S (2017) Automated chest x-ray screening: Can lung region symmetry help detect pulmonary abnormalities? *IEEE Trans Med Imaging* 37(5):1168–1177. <https://doi.org/10.1109/tmi.2017.2775636>
 30. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6(1):60. <https://doi.org/10.1186/s40537-019-0197-0>
 31. Stirenko S, Kochura Y, Alienin O, Rokovyj O, Gordienko Y, Gang P, Zeng W (2018) Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In: 2018 IEEE 38th international conference on electronics and nanotechnology (ELNANO), IEEE, pp 422–428. <https://doi.org/10.1109/ELNANO.2018.8477564>
 32. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
 33. Tasci E (2019) Pre-processing effects of the tuberculosis chest x-ray images on pre-trained cnns: an investigation. In: The international conference on artificial intelligence and applied mathematics in engineering. Springer, pp 589–596. https://doi.org/10.1007/978-3-030-36178-5_48
 34. Tasci E (2020) Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-020-09486-1>
 35. Vajda S, Karargyris A, Jaeger S, Santosh K, Candemir S, Xue Z, Antani S, Thoma G (2018) Feature selection for automatic tuberculosis screening in frontal chest radiographs. *J Med Syst* 42(8):146. <https://doi.org/10.1007/s10916-018-0991-9>
 36. Van Ginneken B, Katsuragawa S, ter Haar Romeny BM, Doi K, Viergever MA (2002) Automatic detection of abnormalities in chest radiographs using local texture analysis. *IEEE Trans Med Imaging* 21(2):139–149. <https://doi.org/10.1109/42.993132>
 37. WHO (2020a) Infectious diseases. https://www.who.int/topics/infectious_diseases/en/. Accessed April 2021
 38. WHO (2020b) Tuberculosis. <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>. Accessed April 2021
 39. Win KY, Maneerat N, Hamamoto K, Sreng S (2020) Hybrid learning of hand-crafted and deep-activated features using particle swarm optimization and optimized support vector machine for tuberculosis screening. *Appl Sci* 10(17):5749. <https://doi.org/10.3390/app10175749>

40. Xie Y, Wu Z, Han X, Wang H, Wu Y, Cui L, Feng J, Zhu Z, Chen Z (2020) Computer-aided system for the detection of multicategory pulmonary tuberculosis in radiographs. *J Healthc Eng.* <https://doi.org/10.1155/2020/9205082>
41. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? *Adv Neural Inf Process Syst* 2:3320–3328
42. Zhang Y, Zhang H, Cai J, Yang B (2014) A weighted voting classifier based on differential evolution. *Abstr Appl Anal Hindawi.* <https://doi.org/10.1155/2014/376950>
43. Zuiderveld K (1994) Contrast limited adaptive histogram equalization. In: Heckbert PS (ed) *Graphics gems*. Academic Press, Cambridge, pp 474–485

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.