Classification of Tuberculosis Type on CT Scans of Lungs using a fusion of 2D and 3D Deep Convolutional Neural Networks

Emad Aghajanzadeh[1], Behzad Shomali[1], Diba Aminshahidi[1] and Navid Ghassemi[1]

[1]*Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran.*

Abstract

In this paper, we present a novel deep-learning-based method to deal with volumetric data like CT scans. The method ensembles a 2-dimensional convolutional neural network (2D-CNN) with a 3D-CNN followed by a recurrent neural network (RNN). We used this approach and its constituent to solve the task of categorizing tuberculosis type in the context of ImageCLEF 2021. Our best run ranked 4th based on the Kappa metric by reaching a value of 0.181 and 3rd based on the accuracy of 0.404. Also, it is worthy of mentioning that our obtained results were very similar to that of the third team with a Kappa of 0.190; and we had a big gap with the fifth team with a Kappa of 0.140.

Keywords

Deep Learning, Information Fusion, Tuberculosis, CT Scan, Diagnosis, Volumetric Data

1. Introduction

Tuberculosis (TB) is an airborne disease that usually affects the lungs and causes severe coughing, chest pains, and fever. The disease is still one of the main health concerns worldwide, being second in causing high mortality rates [1]. Approximately 10.0 million people around the world caught TB in 2019 in line with WHO[World Health Organization. Global tuberculosis report 2020. Geneva, Switzerland: World Health Organization; 2020]. A CT Scan or Computerized Tomography Scan is a versatile medical imaging modality that uses computers and rotating X-ray machines to create cross-sectional images of the patient's body. In other words, the number of detector rows in the z-axis is increased. This allows us to image the whole organ, which reduces image capturing time. It also has several advantages, including improving the quality of images, reducing radiation exposure, and illustrating the soft tissues, blood vessels, and bones of the patient's body [2, 3].

Despite all the advantages, CT scan-based diagnosing approaches have some challenges in terms of the variety of images, their corresponding size, and the complexities there exist in the diagnosing process itself. Moreover, there exist some factors, namely, eye exhaustion and the great number of visitors, which lead to human mistakes [4]. These challenges motivated researchers to use Artificial Intelligence(AI) in order to create automated diagnosis systems for

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

emad.aghajanzadeh@mail.um.ac.ir (E. Aghajanzadeh); behzad.shomali@mail.um.ac.ir (B. Shomali); d.aminshahidi@mail.um.ac.ir (D. Aminshahidi); navidghassemi@mail.um.ac.ir (N. Ghassemi)

increasing the accuracy of medical diagnosis on CT-Scan [5, 6]. In recent years, Deep Learning (DL), a sub-field of AI, has shown encouraging results in medical diagnosis [7, 8].

In this paper, we presented a strategy based on deep learning approaches to detect the type of TB disease, in the context of the ImageCLEF tuberculosis task [9, 10]. ImageCLEF 2021 is an evaluation campaign that is being organized as part of the CLEF initiative labs. The campaign offers several research tasks that welcome participation from teams around the world. In 2021, there were three medical subtasks, one of which was Tuberculosis CT analysis that we participated in. The task is to classify the CT scans into five classes based on their TB type. Besides the dataset, the organizers also provided two versions of extracted masks per each lung [11, 12]. We analyzed the effectiveness of three main approaches based on Convolutional Neural Network (CNN) [13] for this task. The first one is to use a 2 Dimensional convolutional Neural Network (2D-CNN) to learn slice-level features and then obtain the final prediction label through several strategies, such as majority voting or the most certain prediction. The second approach is to utilize a 3 Dimensional convolutional Neural Network (3D-CNN) to capture the spatial features, which are not extracted by the 2D-CNN. Finally, the last approach was to combine the 2D-CNN and 3D-CNN models to reach a model that benefits from both the slice-level and the inter-slice-level features.

The rest of the paper is organized as follows. Section 2 is devoted to describing the competition. Section 3 introduces the preprocessing steps that were used. Section 4 explains the proposed method, which obtained the results demonstrated in section 5. Finally, section 6 concludes the paper with future work directions.

2. ImageClef Tuberculosis: task, data, evaluation

The tuberculosis task of ImageCLEF 2021 Challenge was categorizing each TB case based on its type into five categories: Infiltrative, Focal, Tuberculoma, Miliary, and Fibro-cavernous. Figure 1 illustrates one example for each TB type [1]. The dataset contains 1338 CT images stored in the NIfTI (Neuroimaging Informatics Technology Initiative) format with the resolution of 512×512 pixels and around 100 slices per scan. The file format stores raw voxel intensities in Hounsfield Units (HU). The training dataset consists of 917 CTs, each of which belongs to only one of the five classes. Hence, the task is a multi-class classification (see Table 1).

Table 1
The number of training samples for each of the five TB types.

Type	# of samples
Infiltrative	419
Focal	226
Tuberculoma	101
Miliary	101
Fibro-cavernous	70
total	917

[1] https://www.imageclef.org/2021/medical/tuberculosis

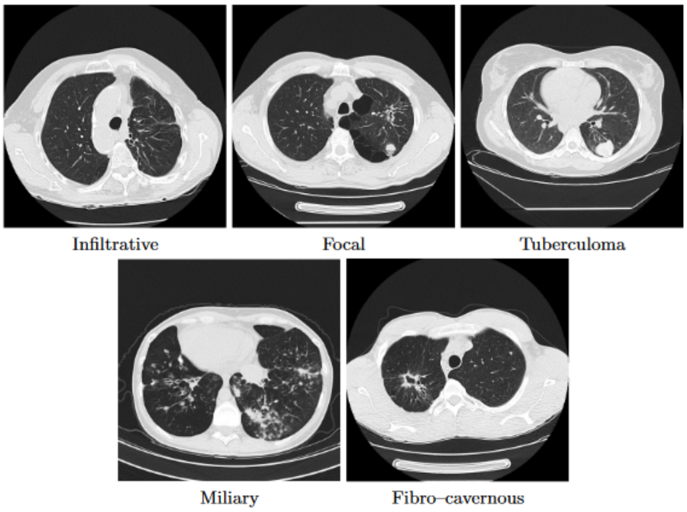


Figure 1: Examples of the five types of TB.

The results are evaluated using unweighted Cohen's Kappa [14] and accuracy metrics, but the primary ranking is done based on only the Kappa metric.

3. Preprocessing

As the dataset files were provided in the NIfTI format with the extension .nii, we used the Nibabel package[2] in Python[3] to load the dataset. Following this, a threshold between -1000 and 400 is used to normalize the CT scans, HU values are scaled to be between 0 and 1. One of the major outcomes of this normalization is reducing the existing contrast among data (See Figure 2). The volumes are then rotated by 90 degrees so that their orientation is fixed. We did not use the masks provided by the task organizers. In most cases, the first and last slices do not contain beneficial features for a model to consider [15], therefore we only selected 50 middle slices and removed the rest of them; this number was chosen empirically by testing a few alternatives. It is worth mentioning, this also saves the power of computational resources; thus helping us to search through different models and settings more efficiently. Moreover, for the same reason, we resized each slice to 100*100, so finally, we had a set of 100*100*50 CT scans. Figure 3 shows the slices of a CT after the preprocessing phase.

4. Proposed Method

In recent years, Deep Neural Networks (DNN) have shown great performance in various tasks, and medical diagnosis has not been an exception [16, 17]. More specifically, Convolutional Neural Networks (CNN), a well-known deep learning architecture inspired by the mechanism

[2] https://github.com/nipy/nibabel
[3] https://github.com/python

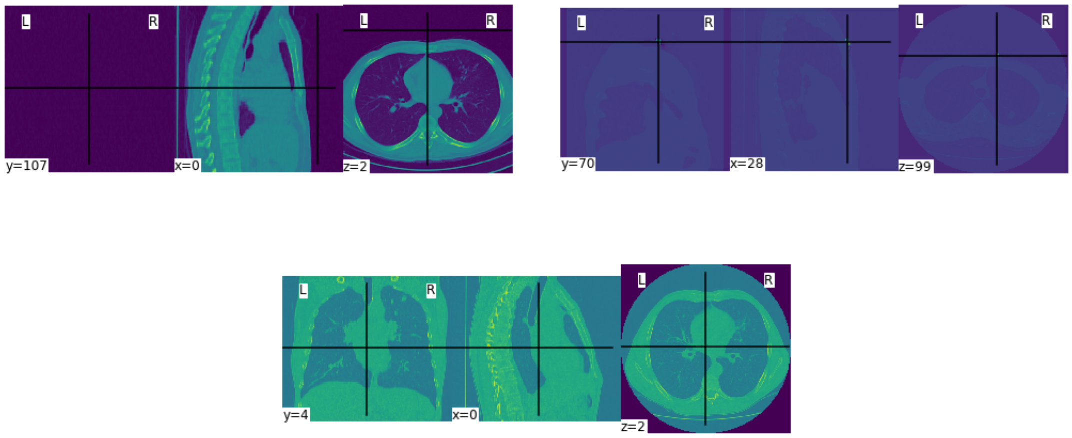


Figure 2: Examples of slices with absolutely different contrast.

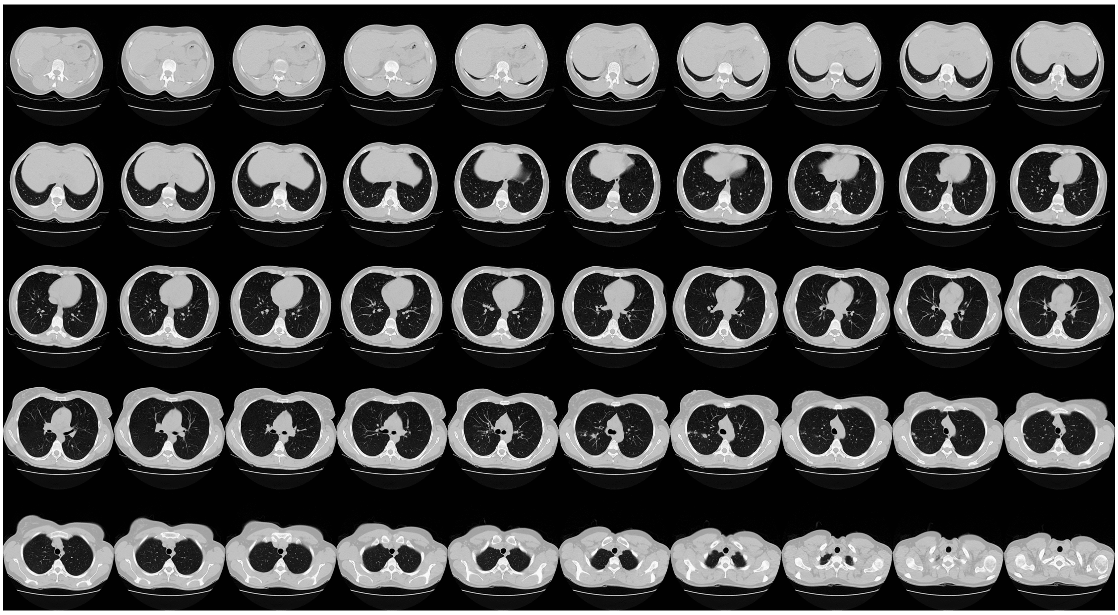


Figure 3: Illustration of 50 selected slices of a CT scan, after preprocessing.

of visual perception of creatures, have been used to solve many image processing tasks. It takes its name from Convolution, a mathematical operation, which performs mapping on input data and processes them into a new space. The main advantage of using CNN is that the kernel can automatically extract the important features from the input data such as detecting edges and distribution of colors in an image which other networks are unable to do, thus making these networks very robust in some processes like image classification. However, despite all aforementioned advantages, CNN models are data-hungry [18], making them less useful, when there are not enough data available like in medical tasks. Moreover, there are even more challenges to face to train these models properly; such as unbalanced [19].

In our attempt to use the CNN to categorize the CT images, we remedied the mentioned problems as follows:

- Small Dataset: We used data duplication and also data augmentation techniques to increase our training data. For augmenting the training data, a degree between -5 to 5 was randomly chosen to apply rotation on each data. Later, we zoomed each data by a ratio of 1.25 and then resized it to its original size; by putting resizing at the last step, we ensured that image quality is kept during data augmentation.
- Imbalanced Data: As shown in Table 1, the number of samples in different classes varies dramatically. To overcome this issue, two distinct approaches were used: the first one is to consider variable penalties for different classes, that is, the multiplier error in classes with fewer data becomes larger. The second approach is to remove the samples from the classes with more data. For this purpose, we removed plenty of samples from classes 0 and 1 from the dataset.

By doing these steps, the training dataset became larger and more balanced (see Table 2). In the proposed method, we mainly used two different approaches, 2D-CNN-based and 3D-CNN-based, where their training settings are as follows:

Table 2
The number of training samples after the preprocessing step.

Type	# of samples
Infiltrative	376
Focal	262
Tuberculoma	342
Miliary	250
Fibro-cavernous	250
total	1480

- Optimizer: Adam optimizer with default parameters (alpha=0.001, beta1=0.9, beta2=0.999, epsilon=1e-7) [20] was used for all epochs of 2D models and the first 30 epochs of 3D models. For the rest epochs of 3D models, we decayed the learning rate by a 1/2 ratio.
- Train/validation split: The dataset was split into train and validation partitions with a ratio of 0.2.
- Batch size: Due to facing some problems with memory, we had to keep the batch size small; thus we set the batch sizes 32 and 8 for 2D and 3D models, respectively.
- Loss function: A combination of cross-entropy and weighted Kappa [21] with multipliers of 0.7 and 0.3 was used for all epochs of 2D models and 30 initial epochs of 3D models. The contribution ratio of losses was then changed into 0.85 and 0.15 for the 10 last epochs of 3D models. The cross-entropy loss is defined as:

$$L_{CE} = -\sum_{i=1}^{C} t_i \log(p_i) \tag{1}$$

Where C is the number of classes, t_i is the ground truth, and p_i is the probability for the i-th class. The formula of weighted kappa with the matrix of observed scores O, the matrix of expected scores based on chance agreement E, and the weight matrix ω is defined as follows:

$$\kappa = 1 - \frac{\sum_{i,j} \omega_{i,j} O_{i,j}}{\sum_{i,j} \omega_{i,j} E_{i,j}} \forall i, j \in \{1, 2, ..., C\} \tag{2}$$

where $O_{i,j}$ is the number of observations that are predicted to be in class i, but their true classes were j. $E_{i,j}$ also denotes the outer product between the vectors of prediction and true value. Finally, $\omega_{i,j}$ represents the weight penalization for every pair i, j.

It is worthy of mentioning that all of our experiments were done using Google Colab [22].

4.1. 2D

In this method, we examined each slice of a single CT individually. To be more specific, in the training phase, we assigned the label of each CT to all of its corresponding slices and fed each slice to the 2D-CNN separately. In this case, we have a vector of 50, the number of slices, predicted labels as output for each CT. To obtain the final prediction for each CT in the testing phase, we have used two different approaches:

- Pick the label which appeared the most (majority voting)
- Pick the label whose corresponding probability was the highest, i.e., where the model is highly certain about that label.

To configure hyper-parameters, we examined various settings such as using skip connection, changing the number of neurons of the last hidden layer, using different activation functions, and differing kernel size of convolution layers and selected our final network empirically. Finally, we got the best result from the model shown in Figure 4.

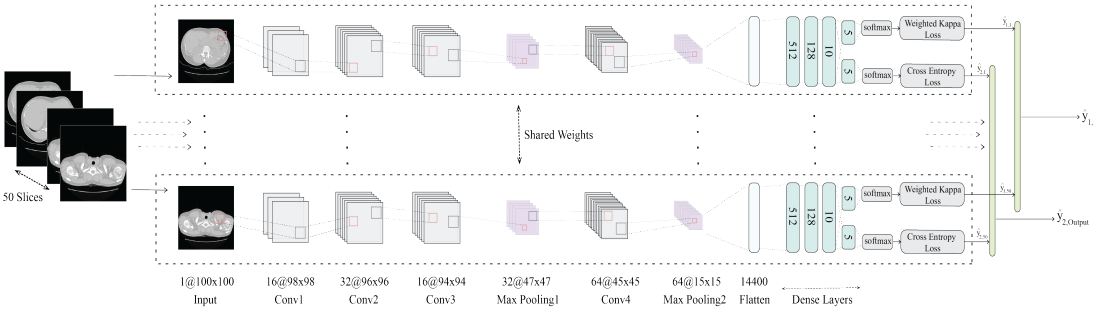


Figure 4: Illustration of 2D model

The learning curve of this model is illustrated in Figure 5. For the evaluation of the model, we used Accuracy, Kappa, and F1-score on validation data which are reported in Table 3

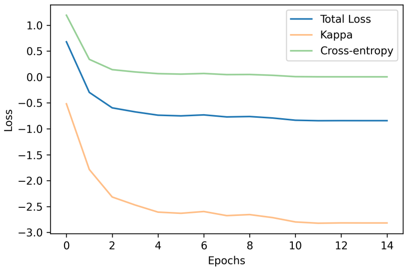


Figure 5: Learning curve of 2D-CNN model.

Table 3
Evaluation results on 2D-CNN model.

Criteria	Score
Accuracy	0.614
Kappa	0.509
F1-score	0.618

4.2. 2D + RNN

In this method, we used the best-trained 2D-CNN that we found in the previous section, but in order to have a more accurate classifier, we implemented a simple Recurrent Neural Network (RNN). As illustrated in Figure 6, we used the extracted features of the 2D-CNN as input of the RNN. To be more clear, we tried two different approaches to fulfill this:

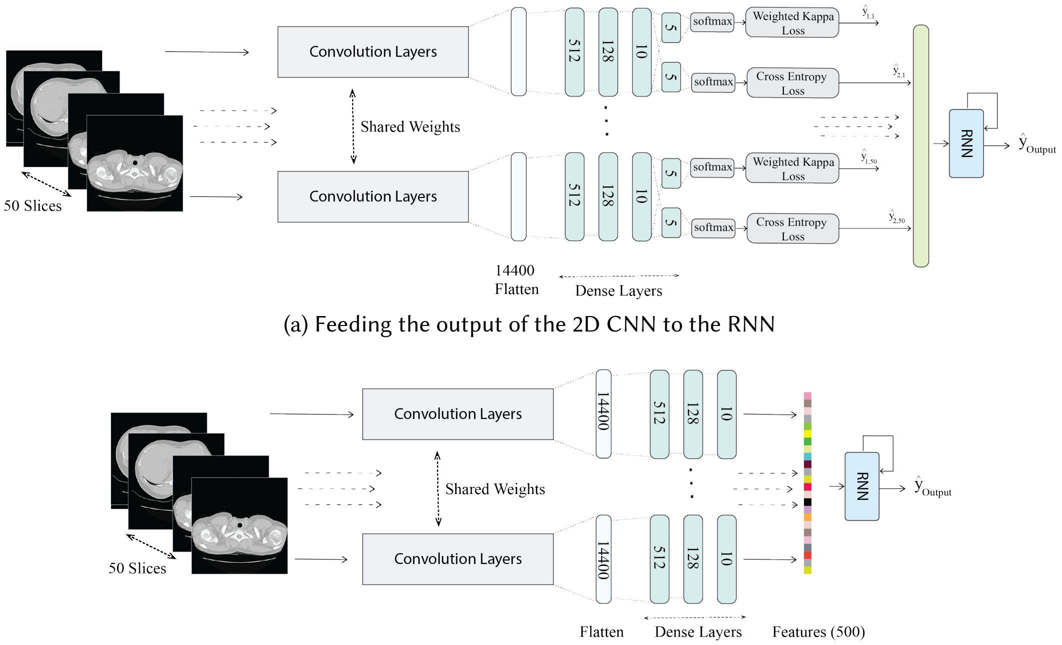
- Feeding the output of the 2D CNN (vector of 50 labels) to the RNN (see 6a)
- Feeding the features extracted by the last hidden layer of 2D CNN to the RNN (see 6b)

We tried different settings for the RNN such as trying different architectures, including long short-term memory (LSTM) [23] and gated recurrent unit (GRU) [24], besides, different number of units. Despite all of the efforts, the accuracy obtained in this method was almost in the same range of 2D-CNN and neither was superior to another one. Therefore, we decided not to submit the result of this approach.

4.3. 3D

Generally, 2D-CNN are unable to catch information that exists among slices, i.e., spatial information. This is because they take a single slice as input and the learning process is applied on each slice individually, thus some of the spatial information may be lost in the process. However, the input of 3D-CNN is a 3D matrix with dimensions of height, width, and depth and the kernel slides over these three dimensions. This property of 3D-CNNs enables them to capture the spatial information between slices. For this reason, we used a 3D model consists of 5 convolution layers as shown in Figure 7 and its corresponding learning curve is displayed in Figure 8.

The evaluation result of the model is also listed in Table 4



(a) Feeding the output of the 2D CNN to the RNN

(b) Feeding the extracted features by the 2D CNN to the RNN

Figure 6: Two types of employing RNN model after the 2D CNN.

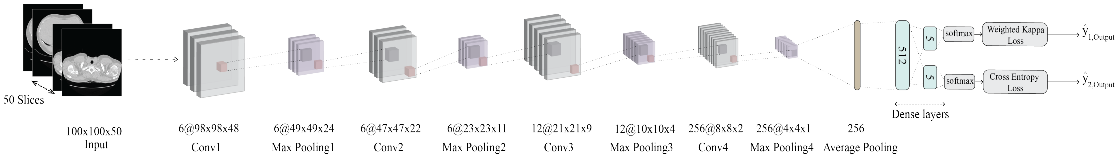


Figure 7: Illustration of proposed 3D model

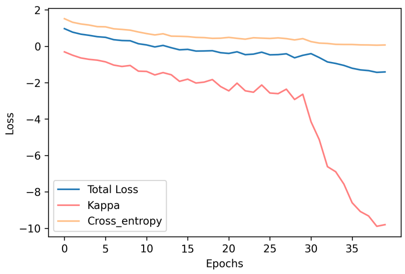


Figure 8: Learning curve of 3D-CNN model.

4.4. 3D + Transfer learning

In order to make use of pre-trained networks, we designed a model that transforms the input images into three-channel images using convolution layers and then followed by the pre-trained

Table 4
Evaluation results on 3D-CNN model.

Criteria	Score
Accuracy	0.646
Kappa	0.547
F1-score	0.656

models. In this experiment, we used ResNet [25], VGG16 [26] and EfficientNet [27], none of which obtained better result than what was obtained through the 3D model itself.

4.5. Fusion of 2D and 3D with RNN

3D CNN can capture the spatial information among CT slices, while 2D CNN can better extract 2D features in each slice. We assumed that ensembling these two models can result in a model with both advantages. Therefore, we first put the features of all slices together and then concatenate them with the features obtained by the 3D CNN model. This forms a feature vector for each CT image, which is then passed to the RNN model as Figure 9. The result was similar to that of the 3D model, which implies that the contribution of the 3D model in the learning process is more dominant than the other model.

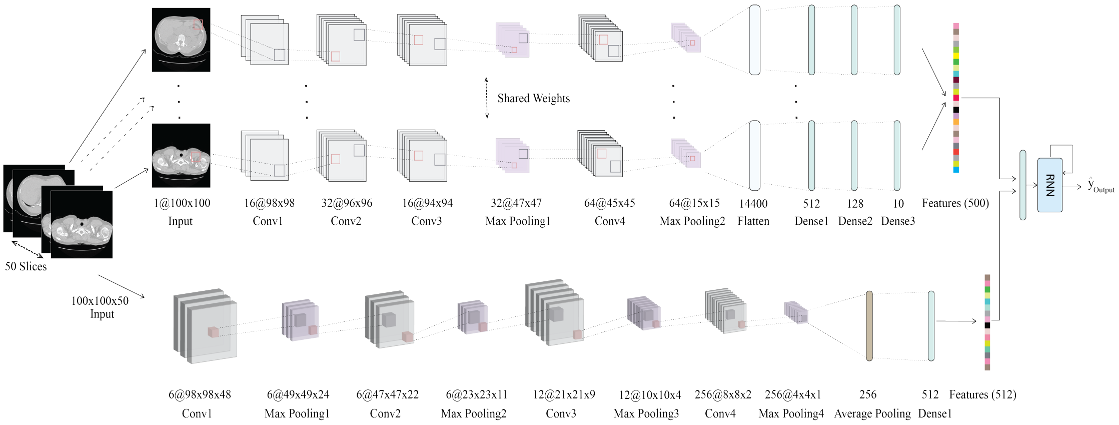


Figure 9: Illustration of fusion 2D and 3D model

4.6. Fusion of 2D and 3D to use the best of both

After investigating the confusion matrices obtained from the 3D and 2D models, we noticed that the 3D model can better separate the last 3 classes, while can not properly categorize the first two ones. However, this pattern was completely reversed for the 2D model. Therefore, we decided to select the final prediction of classes 1 and 2 on test data manually from the prediction of both models. The result was similar to the 3D model which shows that the 2D model did not help the 3D model in the prediction of classes 1 and 2.

5. Comparative results

Table 5 shows the results obtained on the test data of the competition. As it can be seen, the 2D-CNN model obtained 0.036 and 0.373 scores based on Kappa and accuracy score, which is the worst score on Kappa and second-best on accuracy metric among all our submissions. Then, the Kappa is increased by 0.02 when the RNN module is added on top of the 2D-CNN; meanwhile, the accuracy score is decreased by 0.031. Furthermore, the 3D-CNN model obtained 0.181 and 0.404 on Kappa and accuracy, respectively, which is our best score on both metrics. This result is then decreased to 0.136 and 0.371 by adding the 2D features and manually selecting the final prediction when 2 models had conflicts in the first two classes.

Table 5
Results obtained on the test data.

Method	Kappa	Acc
2D	0.036	0.373
2D + RNN	0.056	0.342
3D	0.181	0.404
3D + 2D + Manual	0.136	0.371

6. Conclusion and Future Works

In this paper, we have described our proposed method for the tuberculosis task of ImageCLEF Tuberculosis 2021. We proposed three different approaches and analyzed their corresponding results. The results demonstrate that the 2D-CNN didn't work well, while we believe that it can be significantly improved by applying a smarter voting mechanism for outputting the final label. These improvements can be such as applying Gaussian Distribution Normalization or defining a window with a fixed size, k, to move through the vector of 50 labels and pick the final label. By having a better 2D-CNN in hand, we can expect some enhancements while ensembling 2D and 3D-CNN models. Moreover, comparing the obtained results on validation data, Table 3 and Table 4, with the results on test data, Table 5, shows that there exists a considerable gap between them, which can be caused by the existence of different distributions between validation and test datasets. To resolve this issue, we suggest exchanging the order of duplicating and splitting the data. This guarantees that there is no overlap between training and validation data. We also plan to employ more complicated approaches for data augmentation; in this case, the models can learn and generalize more robustly. During our experiments, we also figured out that the main part of the incorrect predictions of models caused by predicting the first two classes interchangeably. By having this in mind, we can fix this problem by training a separate binary classification on those classes.

References

[1] N. Fogel, Tuberculosis: a disease without boundaries, Tuberculosis 95 (2015) 527–531.

[2] M. Fu, S.-L. Yi, Y. Zeng, F. Ye, Y. Li, X. Dong, Y.-D. Ren, L. Luo, J.-S. Pan, Q. Zhang, Deep learning-based recognizing covid-19 and other common infectious diseases of the lung by chest ct scan images, medRxiv (2020).

[3] P. T. Johnson, D. G. Heath, B. S. Kuszyk, E. K. Fishman, Ct angiography with volume rendering: advantages and applications in splanchnic vascular imaging., Radiology 200 (1996) 564–568.

[4] S. Hu, Y. Gao, Z. Niu, Y. Jiang, L. Li, X. Xiao, M. Wang, E. F. Fang, W. Menpes-Smith, J. Xia, et al., Weakly supervised deep learning for covid-19 infection detection and classification from ct images, IEEE Access 8 (2020) 118869–118883.

[5] D. L. Pham, C. Xu, J. L. Prince, Current methods in medical image segmentation, Annual review of biomedical engineering 2 (2000) 315–337.

[6] A. El-Baz, G. M. Beache, G. Gimel'farb, K. Suzuki, K. Okada, A. Elnakib, A. Soliman, B. Abdollahi, Computer-aided diagnosis systems for lung cancer: challenges and methodologies, International journal of biomedical imaging 2013 (2013).

[7] A. Bhandary, G. A. Prabhu, V. Rajinikanth, K. P. Thanaraj, S. C. Satapathy, D. E. Robbins, C. Shasky, Y.-D. Zhang, J. M. R. Tavares, N. S. M. Raja, Deep-learning framework to detect lung abnormality–a study with chest x-ray and lung ct scan images, Pattern Recognition Letters 129 (2020) 271–278.

[8] G. van Tulder, M. de Bruijne, Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines, IEEE transactions on medical imaging 35 (2016) 1262–1272.

[9] S. Kozlovski, V. Liauchuk, Y. Dicente Cid, V. Kovalev, H. Müller, Overview of ImageCLEFtuberculosis 2021 - CT-based tuberculosis type classification, in: CLEF2021 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Bucharest, Romania, 2021.

[10] B. Ionescu, H. Müller, R. Peteri, A. Ben Abacha, M. Sarrouti, D. Demner-Fushman, S. A. Hasan, S. Kozlovski, V. Liauchuk, Y. Dicente, V. Kovalev, O. Pelka, A. G. S. de Herrera, J. Jacutprakart, C. M. Friedrich, R. Berari, A. Tauteanu, D. Fichou, P. Brie, M. Dogariu, L. D. Ştefan, M. G. Constantin, J. Chamberlain, A. Campello, A. Clark, T. A. Oliver, H. Moustahfid, A. Popescu, J. Deshayes-Chossart, Overview of the ImageCLEF 2021: Multimedia retrieval in medical, nature, internet and social media applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 12th International Conference of the CLEF Association (CLEF 2021), LNCS Lecture Notes in Computer Science, Springer, Bucharest, Romania, 2021.

[11] V. Liauchuk, V. Kovalev, Imageclef 2017: Supervoxels and co-occurrence for tuberculosis ct image classification, in: CLEF2017 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Dublin, Ireland, 2017.

[12] Y. Dicente Cid, O. A. Jiménez del Toro, A. Depeursinge, H. Müller, Efficient and fully automatic segmentation of the lungs in ct volumes, in: O. Goksel, O. A. Jiménez del Toro, A. Foncubierta-Rodríguez, H. Müller (Eds.), Proceedings of the VISCERAL Anatomy Grand Challenge at the 2015 IEEE ISBI, CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, 2015, pp. 31–35.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[14] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1960) 37–46.

[15] M. Sohrabi, M. Parsi, S. H. Tabrizi, Statistical analysis for obtaining optimum number of ct scanners in patient dose surveys for determining national diagnostic reference levels, European radiology 29 (2019) 168–175.

[16] M. Khodatars, A. Shoeibi, N. Ghassemi, M. Jafari, A. Khadem, D. Sadeghi, P. Moridian, S. Hussain, R. Alizadehsani, A. Zare, et al., Deep learning for neuroimaging-based diagnosis and rehabilitation of autism spectrum disorder: A review, arXiv preprint arXiv:2007.01285 (2020).

[17] A. Shoeibi, M. Khodatars, R. Alizadehsani, N. Ghassemi, M. Jafari, P. Moridian, A. Khadem, D. Sadeghi, S. Hussain, A. Zare, et al., Automated detection and forecasting of covid-19 using deep learning techniques: A review, arXiv preprint arXiv:2007.10785 (2020).

[18] G. Marcus, Deep learning: A critical appraisal, arXiv preprint arXiv:1801.00631 (2018).

[19] Y. Sun, A. K. Wong, M. S. Kamel, Classification of imbalanced data: A review, International journal of pattern recognition and artificial intelligence 23 (2009) 687–719.

[20] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[21] J. de La Torre, D. Puig, A. Valls, Weighted kappa loss function for multi-class classification of ordinal data in deep learning, Pattern Recognition Letters 105 (2018) 144–154.

[22] E. Bisong, Google colaboratory, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Springer, 2019, pp. 59–64.

[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[26] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[27] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.