UAIC2021: Lung Analysis for Tuberculosis Classification

Alexandra Hanganu, Cristian Simionescu, Lucia-Georgiana Coca and Adrian Iftene

"Alexandru Ioan Cuza" University, Faculty of Computer Science, Iasi, Romania

Abstract

This article presents a methodology for chest CT scan analysis that enables the automatic categorization of pulmonary tuberculosis cases into one of the following five types: Infiltrative, Focal, Tuberculoma, Miliary, Fibro-cavernous. We showcase several deep learning methods for classifying tuberculosis in CT scans from the ImageCLEF 2021 Tuberculosis - TBT classification challenge. Furthermore, it explores the use of pre-trained models, as well as training from scratch on volumetric data and 2D projections.

Keywords

Computed Tomography, Tuberculosis, Deep Learning, 2D Projections, k-Means

1. Introduction

According to the NIH (National Institute of Allergy and Infectious Diseases), tuberculosis (TB) is the leading infectious cause of death worldwide. Despite the development of technology, in 2017, 10 million people became ill with TB, and 1.6 million people died of TB disease, including 230,000 children, according to the World Health Organization. Over the past 200 years, TB has claimed the lives of more than one billion people—more deaths than those caused by malaria, influenza, smallpox, HIV/AIDS, cholera, and plague combined. Although tuberculosis treatment exists and further research is ongoing, drug resistance poses a continued threat.

ImageCLEF [1] is an evaluation campaign that is being organized as part of the CLEF initiative labs1. The ImageCLEFmedical 2021 challenge has a task [2] which consists of automatically categorizing each TB case into one of the following five types: (1) Infiltrative, (2) Focal, (3) Tuberculoma, (4) Miliary, (5) Fibro-cavernous.

The organizers provided to participants two types of masks. The first version of segmentation provides more accurate masks [3], but it tends to miss large abnormal regions of lungs in the most severe TB cases. The second segmentation on the contrary provides more rough bounds of masks [4], but behaves more stable in terms of including lesion areas. In case the participants use the provided masks in their experiments.

This paper was organized as follows: section 2 describes the related work, section 3 presents our methods, section 4 evaluates the methods proposed and, in the end, we draw conclusions.

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

alexandra.hanganu@gmail.com (A. Hanganu)

2. Related Work

In previous editions, different approaches were proposed for the tuberculosis task. In 2017, there were two important tasks: MDR Detection which consisted of assigning each TB patient the probability of having a resistant form of tuberculosis based on the analysis of chest CT scan; Detection of TB Type task which automatically categorize each TB case into one of the following five types: Infiltrative, Focal, Tuberculoma, Miliary, Fibro-cavernous.

In 2018, the valuable result was of team UIIP-BioMed that used a Deep CNN [5], folowed by MedGift that used an SVM with and RBF kernel [6].

In 2019, UIIP-BioMed [7] was the leader again, followed by CompEle-cEngCU [8]. UIIP-BioMed used a 2D CNN whilst CompEle-cEngCU used a 2D CNN based on AlexNet [9]. The solution developed by our group uses stage-wise boosting in low-resource environments [10].

In 2020, the majority of the participants used some variations of the projection-based approach and created 2D CNNs. As a result, only four groups tried 3D CNNs for a direct analysis of the CT volumetric data [11]. SenticLab.UAIC [12] was the winner of the task using 2D and 3D CNNs. The SDVA-UCSD was ranked on second place using a 3D CNN with a convolutional block attention module (CBMA) and a customized loss function [13]. Our team was ranked on 7th place, using SVMs and CNNs for lung-wise processing [14].

3. Methods

We propose an extension to the 2D projection methods which were successfully applied in previous years by replacing the arithmetic mean aggregation function with k-means. We made this choice since it can be considered that the average is a specific case for k-means when k=1. With this, we hope to capture more relevant characteristics from the data. This gives us the advantage of working with smaller dimensionality data.

Transforming 3D CT scans into 2D images does have the downside of losing spacial proximity information along the third dimension and introduces the probability of losing important features. Due to this, we also experimented working with the 3D samples directly in order to have a reference point of how the two modalities (2D and 3D) performed.

3.1. Dataset

Based on the metadata file presented by the organizers, we concluded that the data at hand is highly imbalanced. Since this could have negatively influenced our models, we have decided to use weighted loss, in order to counteract this phenomenon.

Table 1 provides an insight for the distribution of each affection in the training set, as well as the weight used for the loss function. To calculate the weight, we have divided the lowest total number of patients affected by one of the types (in our case, this would be 70) by the number of patients affected for each type.

Table 1
Patients Affected in the Training Set and Weighted Loss Used.

Tuberculosis Type	Patients Affected (out of 915)	Weighted Loss
(1) Infiltrative	418	0.1674
(2) Focal	226	0.3097
(3) Tuberculoma	101	0.6930
(4) Miliary	100	0.6930
(5) Fibro-cavernous	70	1

3.2. Data Pre-Processing

Given that the masks provided were not able to return the best segmentation possible, a decision was made to combine the two of them and compute a new mask that would be closer to what we needed. However, this happened to be quite hard, as some of the masks did not have a proper lung segmentation, in accordance to the anatomical position given by the midsagittal line in the upper thoracic area. Furthermore, some issues were not related to the segmentations, but by the CT Scans, since some of them were not able to be opened or were considered faulty. Table 2 presents what issues we have encountered when processing the data and the proposed solution for said problems.

The most difficult case to treat was the test CT Scan for patient TST_247, because we were not able to load it properly and it needed to be processed and labeled. Since it had a similar issue to the scan of patient TRN_360, we have assumed that both files have a similar origin and that the problem was either in an underlying anatomical condition or in the acquisition of a similar device fault. Thus, we manually assigned the same TBT class that TRN_360 was assigned in the training metadata.

3.2.1. Data Normalization

When normalizing the images, we have opted for a similar approach to past high-ranking participants[7]. The erosion radius adopted was of 10 and the voxel intensity values were increased by 1024 Hounsfield Units (HU) with the threshold set to -1200, clipped to 600.

3.2.2. Segmentation Pre-Processing

The two types of lung segmentations provided by the organizers had different issues. One of them has accurate edges, but it tends to overlook or miss entirely some of the large cavities or other lesions in the lungs that would be relevant for the model. However, the other performs better in covering the entire lung, but it is generally more inaccurate, especially when referring to the edges of the lung lobes. The solution proposed for this issue was to generate a mean of the two masks. Combining them and retrieving the entire information, helped ensure that no particularity of the affections is overlooked, while also highlighting the increased importance of the sections where the segmentation methods overlap and hence agree upon. To complete this task, we looked at the first type of mask, the fully automatic multistage one, taking into account only the non-zero values, added them to the second type of mask and then computed the mean

Table 2
Dataset Issues and Solutions

Data Type	Patient ID	Issue	Solution
Train CT Scan	TRN_305	Corrupt file	Left out
Train CT Scan	TRN_360	Initial CT Scan consisted of only some white pixels on a black background and could not be used	Left out
Train Mask[1]	TRN_366 TRN_433 TRN_867 TRN_867 TRN_887 TRN_891 TRN_894 TRN_897	Wrong lung segmentation	Recolored the mask image according to the initial colour scheme and removed the trachea as it was also selected as part of the lungs
Test Mask[1]	TST_051 TST_093 TST_124 TST_269 TST_362	Wrong lung segmentation	Recolored the mask image according to the initial colour scheme and removed the trachea as it was also selected as part of the lungs
Test CT Scan	TST_247	Initial CT Scan consisted of only some white pixels on a black background and could not be used	Assumed it was similar to patient TRN_360 and assigned the same label (Type 1)

of the two, multiplying it by 255 in the end. Therefore, we computed a new image that had the opacity adjusted for the parts of the lungs where the two segmentations did not overlap.

The resulting masks were applied to the raw CT scans by multiplying the two tensors. However, when completing this step, there was a threshold set so that it could help with eliminating the non-lung pixels present in the resulting image.

3.2.3. Data Augmentation

Multiple types of data augmentation techniques were attempted to be used with a view to prevent an early over-fitting of the models. To this end, we have employed Randaugment [15], only disabling the "Invert" and "Solarize" augmentations and setting $N = 2$ and $M = 14$. Additionally, we used Random Erasing [16] which is a technique that randomly crops out a patch of size between 5% and 15% from the original image. Nonetheless, there were other approaches to augmenting the dataset that we have tried to apply, such as Mixup [17] and Cutmix [18], but these methods proved to produce images which were making it too difficult for the model to learn on.

[1] the mask referred to is the fully automatic multistage one

3.3. Volumetric Classification

When working with the 3D volumetric data, we have looked for a model that had been pre-trained, preferably on medical data. Thus, we came across MedicalNet [19], which provided some interesting models that had been trained previously. We have worked with these in order to obtain better results that could not be achieved when training the network from scratch. However, due to the large dimension of the samples and the hardware limitation imposed by using a single Nvidia RTX 2080Ti GPU, we had to resize all the images to 256x256x32. Taking these facts into consideration, we managed to fit into the GPU memory only the MedicalNet10 variant, which is essentially a pre-trained 3D-ResNet10 network.

One technique adopted when training was gradient accumulation. Therefore, since we could only use a maximum batch size of 2, we accumulated 16 gradient backpropagation steps before updating the weights, effectively allowing us to simulate a batch size of 32.

We loaded the 23-dataset pre-trained MedicalNet10 weights, replacing its segmentation head with a newly initialized classification head, where a linear layer outputs the probabilities for each of the 5 classes.

As for the hyper parameter tuning step, we attempted multiple variations of optimizers and data augmentations, but unfortunately our attempts were consistently faced with an early overfitting of the model. This means that it would happen after the first 40 epochs of training, when very little to no data augmentations were applied. One other issue was that frequently we would encounter an unstable training process, caused by exploding gradients, which would then be followed by a model collapse, usually occurring in the first 5 epochs when stronger augmentations were used.

Our best submitted run is a MedicalNet10 model trained for 44 epochs using the AdamP optimizer [20] with the default parameters, cross entropy loss and with no data augmentations. Attempts to train 3D-ResNet variants from scratch also suffered from the same issues but the overfitting would start sooner and as such the models generated poorer results.

3.4. Projection Classification

3.4.1. Projection Generation

In previous years, the computation and usage of 2D projections has proven to be one of the most successful ways of pre-processing a 3D volume. This happened mostly due to its effective reduction in dimensionality without the loss of relevant information, so, this year, we attempted to build upon this approach. We had managed to adapt the projection method to use the k-means algorithm [21][22]. As described in [7] and [12], generating 2D projections from the X, Y and Z axis, by aggregating along the a given direction with the max(), average() and std() functions and stacking these results as channels of the same image, proved to conserve enough relevant information, while simultaneously drastically reducing the size. We propose the extension of the family of aggregation functions used for this purpose with k-means. We take the centroid values from k-means as the output which will construct multiple image channels. Consequently, when looking at the issue this way, the average() function from the original projection method becomes a particular case of k-means, that is when k=1.

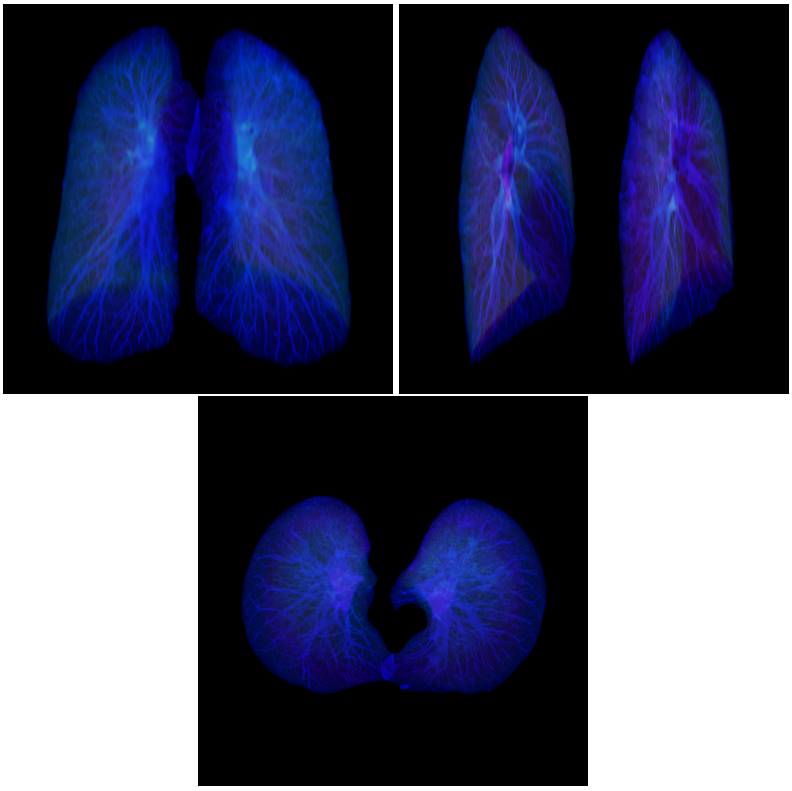

Figure 1: X, Y and Z Axis Projections Approximations.

Using k-means as an aggregation function gives us more flexibility with the number of potentially relevant features that we extract as centroids from the 3D volume. Since the function outputs a number of centroids equal to the value of k (not just 1, as in the case of average()), each of these centroids is mapped to a different channel in the resulting projection. The centroids are initialized randomly, this factor did not seem to be a big influence on the results since the algorithm always and very rapidly finds the final centroids, in very few steps, as it is a very simple 1D k-means calculation. The mapping is done by sorting the centroid values, the smallest valued centroids will be mapped to the first channel, the second smallest to the second channel and so on. Our final projection algorithm still used max() and std() as aggregation functions and replaced average() with k-means() where k=5. The resulting images consist of 7 channel projections for each of the axis directions X, Y and Z. The usage of max() could be replaced by using higher values for k since the centroid on the higher end of the interval will probably be very close in value to the max value.

An ablation study would be needed to identify an optimal value for k, as a starting point, we arbitrarily picked k=5. Since a stable training procedure was not obtained, we did not get to fine-tune this parameter.

The Y projection had to be treated separately, since projecting from that direction would overlap the two lungs. In order to prevent losing relevant information at the lung level, we

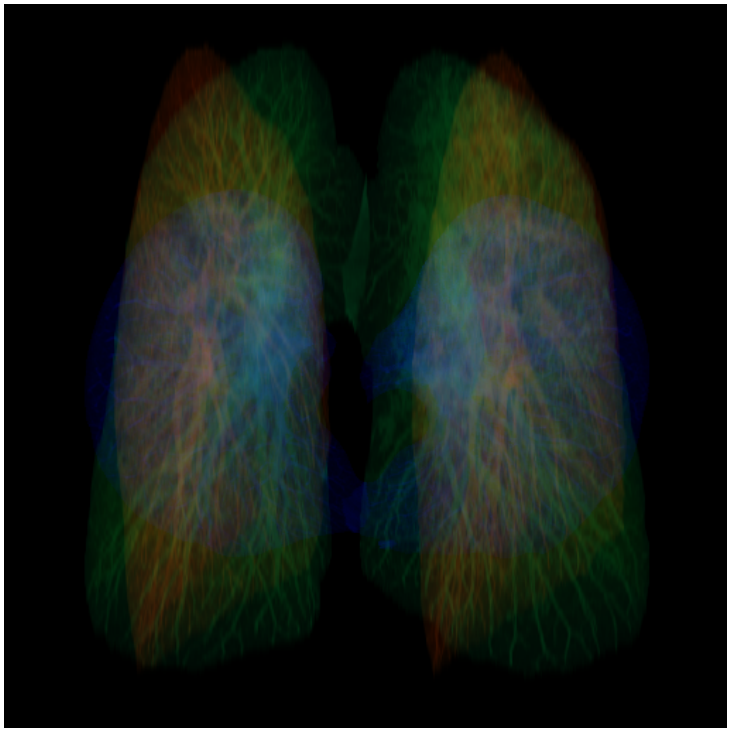

Figure 2: Stacked Projection Approximation

first separated the lungs with the help of the lung-wise segmentation and then calculated the projection separately for each one. Consequently, we concatenated the two images such that the two lungs would be positioned next to each other. The resulting X, Y and Z projections can be observed in Fig.1. Due to having more than 3 channels, the figures here were obtained by resizing the number of channels to 3, for RGB. However, there exists less information, because some was lost in the process of resizing the channels for visualization purposes.

After computing the X, Y and Z projections, each with 7 channels, we have decided to stack them along the channel dimension. Given that the anatomical positioning is the same, overlapping the lobes would not be considered an issue, especially since there would be 21 individual channels for the model to look at and learn the particularities of each tuberculosis type. An approximated look of the resulting images is presented in Fig.2. This was favorable especially because the algorithm would only use one image for the training and testing process, meaning that the data would be fed all at once. A fixed random seed was used for to generate the k-means projections.

3.4.2. Models

The considerably smaller 2D image size allowed us to use larger models and batch sizes. To do so, we tried resizing the images, while preserving the number of channels. As for the CNN models, we experimented with PreResNet56 [23] and the Swin Transformer [24] large, base and

tiny variants.

For the Swin Transformer models, we also tried to use the weights trained on ImageNet-21k [25] and to apply various transfer learning techniques in order to adapt it to the TBT classification task. One of the steps implemented to do this was to reinitialize the PatchEmbed layer and the final normalization, pooling and linear layers, so that they can match our 5 classes. The Swin Transformer model runs were not able to noticeably reduce the loss when trained from scratch or when loading the pre-trained weights, but we believe this might have been due to an implementation error on our part or from a faulty normalization technique.

The two models that delivered the best results were both based on PreResNet, with a depth of 56. Most of the experiments with this type of model were based on hyper-parameter tuning, as the base was a standard PreResNet56 that had been trained from scratch on data for the task.

For the hyper-parameters, we used the RAdam optimizer[26] with the default parameters, a cross entropy loss function and a batch size of 16 with 4 gradient accumulation steps, resulting in a simulated batch size of 64. Both models use lookahead[27]. Further information about the difference between the parameters of the two models is presented in Table.3.

Table 3
Comparison of the Two PreResNet56 Models

Parameter	#135756	#135798
Epochs	55	28
Image Size	384x384	512x512
Learning Rate	0.0001	1.0e-05
Lookahead K	8	16

4. Evaluation

When evaluating our experiments, the main reference was the kappa score, defined on the interval [-1,1]. This score is not differentiable, so we could not use it as a loss function directly for our models. Thence, the training process was also monitored by following the cross entropy loss function. As a means to track the progress of the training process, we randomly split the initial train dataset into train and validation splits, while also preserving the same class imbalanced for both splits. Hence, in order for the results to be consistent between different runs, all runs had the same data split.

The experiments were performed using only flips and rotations as data augmentations, this caused the model to overfit very early on and would not recover with more training. This phenomenon convinced us to apply stronger augmentation techniques, such as Randaugment and Random Erasing, in order to populate the latent space. However, when these stronger augmentations were being used, the models suffered from training instability since our loss values would explode and hence the gradients would too, resulting in numeric overflow. Finally, once the loss exploded, the model collapsed and did not learn anything else. This happened very fast, usually in the first 5 epochs of training.

The four submissions made by our team are listed in Table.4, detailing what model was used,

whether or not the model had pre-trained weights, the local validation kappa score and the test kappa score. Since training with strong data augmentations would result in an unstable training, which stopped the model from learning in the first few epochs, no submissions were made with them. All of the runs we submitted suffered from relatively fast overfitting, this occurring in the first approximately 50 epochs.

Table 4
Submission Runs - Official Evaluation Results

ID	Model	Pre-trained	Modality	Val. Kappa	Test Kappa
135708	MedicalNet10	Yes	3D	0.232	0.129
135756	ResNet56	No	2D	0.209	-0.003
135750	3D-ResNet20	No	3D	0.117	-0.019
135798	ResNet56	No	2D	0.134	0.016

5. Future Work

Further work needs to be invested into adapting state-of-the-art data augmentations from the general computer vision domain to medical images. Since the size of the datasets usually encountered in the field is small, stable data augmentation strategies are required, as frequently acquiring additional data is not feasible.

As a retrospective analysis of our methods, we need to carefully investigate the exact causes of training instability. We suspect that, because we did not normalize the data after the final step in the pre-processing/augmentation pipeline, the resulting images might have very different value distributions between different samples, which could even escape our intended value interval of $[0.0, 1.0]$. Another potential cause of our issues could come from the large size of the linear classification layer which could be the cause of the exploding gradients we noticed, we have to investigate if using a pooling layer before the classification one helps solve our issue.

6. Conclusion

In this paper we present ways of detecting different types of tuberculosis in lungs. The methods that have proven to work best for us are focused on pre-trained models for classifying 3D volumetric data and models trained from scratch for 2D projections.

The best result from our submissions can be improved, but in doing so, we would have to investigate different normalization techniques and types of data augmentation that would suit the task better. However, in order to work with volumetric data properly, we would also need to improve the current hardware with the aim of being able to apply larger models, as well as greater batch sizes, on medical datasets.

7. Acknowledgements

This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] V. Liauchuk, A. Tarasau, E. Snezhko, V. Kovalev, ImageCLEF 2018: Lesion-based tb-descriptor for ct image analysis, CLEF 2018 Working Notes (2018).

[6] Y. Dicente Cid, H. Muller, Texture-based graph model of the lungs for drug resistance detection, tuberculosis type classification, and severity scoring: Participation in the imageclef 2018 tuberculosis task, CLEF 2018 Working Notes (2018).

[7] V. Liauchuk, Imageclef 2019: Projection-based ct image analysis for tb severity scoring and ct report generation, CLEF2019 Working Notes 2380 (2019).

[8] A. Mossa, A. Yibre, U. Evik, Multi-view cnn with mlp for diagnosing tuberculosis patients using ct scans and clinically relevant metadata, CLEF2019 Working Notes 2380 (2019).

[9] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90.

[10] A. Tabarcea, V. Rosca, A. Iftene, ImageCLEFmed Tuberculosis 2019: Predicting ct scans severity scores using stage-wise boosting in low-resource environments, CLEF2019 Working Notes 2380 (2019).

[11] S. Kozlovski, V. Liauchuk, Y. Dicente Cid, A. Tarasau, V. Kovalev, H. Muller, Overview of imagecleftuberculosis 2020 - automatic ct–based report generation, CLEF2020 Working Notes (2020).

[12] R. Miron, C. Moisii, M. Breaban, Revealing lung affections from cts. a comparative analysis of various deep learning approaches for dealing with volumetric data, CLEF2020 Working Notes (2020).

[13] X. Lu, E. Chang, Z. Liu, C. Hsu, J. Du, A. Gentili, ImageCLEF2020: Laterality-reduction three-dimensional cbam-resnet with balanced sampler for multi-binary classification of tuberculosis and ct auto reports, CLEF2020 Working Notes (2020).

[14] L. Coca, A. Hanganu, C. Cusmuliuc, A. Iftene, UAIC2020: Lung analysis for tuberculosis detection, CLEF2020 Working Notes (2020).

[15] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.

[16] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 13001–13008.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[18] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.

[19] S. Chen, K. Ma, Y. Zheng, Med3D: Transfer learning for 3D Medical Image Analysis, arXiv preprint arXiv:1904.00625 (2019).

[20] B. Heo, S. Chun, S. J. Oh, D. Han, S. Yun, G. Kim, Y. Uh, J.-W. Ha, Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights, in: Proceedings of the International Conference on Learning Representations (ICLR), Online, 2021, pp. 3–7.

[21] S. P. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theory 28 (1982) 129–136.

[22] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, Oakland, CA, USA, 1967, pp. 281–297.

[23] 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.

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv preprint arXiv:2103.14030 (2021).

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[26] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, CoRR abs/1908.03265 (2019). URL: http://arxiv.org/abs/1908.03265. arXiv:1908.03265.

[27] M. R. Zhang, J. Lucas, G. E. Hinton, J. Ba, Lookahead optimizer: k steps forward, 1 step back, CoRR abs/1907.08610 (2019). URL: http://arxiv.org/abs/1907.08610. arXiv:1907.08610.