ImageCLEF 2018: Semantic descriptors for Tuberculosis CT Image Classification

Abdelkader HAMADI[0000−0001−9990−332X] and Djamel Eddine YAGOUB

University of Abdelhamid Ibn Badis Mostaganem
Faculty of Exact Sciences and Computer Science
Mathematics and Computer Science Department
Mostaganem, Algeria

abdelkader.hamadi@univ-mosta.dz
djamel.ed.y@gmail.com

Abstract. In this article, we present our methodologies used in our participation at the two sub-tasks of the ImageCLEF 2018 Tuberculosis Task (TBT and SVR task). We proposed to extract a single semantic descriptor of 3D CT image to describe each patient rather than using all his slices as separate samples. In TBT task, the resulting descriptors are then exploited in a second learning stage to identify the type of tuberculosis among five given classes. In SVR task, the same experimental design is used to predict the degree of severity of the disease. We reached a Kappa coefficient value of about 0.0629 in TBT sub-task, and our best run on SVR was ranked 12^{th} out of 36 submission and 5^{th} out of 7 participant teams. We believe that our approach could give better results if applied properly.

Keywords: ImageCLEF · Tuberculosis Task · Deep Learning · CT Image · Tuberculosis CT Image Classification · Tuberculosis Severity Scoring.

1 Introduction

Tuberculosis is an infectious disease caused by a bacterium called Bacillus microbacterium tuberculosis. With a high mortality rate in the world, this disease remained one of the top ten causes of death in the world in 2015. Diagnosing this sickness quickly and accurately is a vital goal that would limit its invasion and damage. One of the major problems of this disease is that traditional tests produce inaccurate or too long results. For these reasons, researchers have been interested in this disease diagnosis, particularly in the context of the international challenge ImageCLEF 2017 [3] and ImageCLEF 2018 [9] where two tasks (three tasks in ImageCLEF 2018) have been reserved for it. The first aims to detect multi-drug resistant (MDR) status of patients. The goal of the second task is to identify the type of tuberculosis. A third task has been introduced in ImageCLEF 2018 [5] which consists to predict the degree of severity of the

patient's case. In all the three tasks, the predictions are based on 3D CT scans images. Algorithms involving deep learning have been tested to diagnose the presence or the absence of tuberculosis. The results obtained were interesting. However, they must be improved for better control and effective diagnosis, helping doctors to make the decisions and to choose the necessary treatments at the right time.

We can summarize the objectives of the Tuberculosis task through the following points:

- Helping medical doctors in the diagnosis of drug-resistant TB and TB type identification through image processing techniques;
- Introducing work towards inexpensive and quick methods for early detection of the MDR status and TB types in patients;
- Predicting quickly the type of TB and its severity degree to help doctors to make quick decisions and give the effective treatments.

We present in the following our work that has been made in the context of our participation to the two sub-tasks of ImageCLEF 2018 Tuberculosis Task: TuBerculosis Types classification (TBT) and Tuberculosis Severity Scoring (SVR).

The remainder of this article is organized as follows. Section 2 describes the two tasks to which we had participated. In section 3, we present our contribution by detailing the system deployed to complete our submissions. Section 4 details our experimental protocols followed to generate our predictions. We detail and analyze in the same section the results obtained. We conclude in the last section by presenting our perspectives and future works.

2 Participation to imageCLEF 2018

2.1 Tasks description

In this paper, we focus on our participation in the TBT and the SVR sub-tasks that we describe in the following sections.

In both tasks the data is provided as 3D CT scans. For some patients several 3D CT scans are given while for some others only one is provided. All the CT images are stored in NIFTI file format with .nii.gz extension file (g-zipped .nii files). For each of the 3-dimensions of the CT image, we find a number of slices varying from about 50 to 400. Each slice has a size of about 512×512 pixels.

A training collection is provided at the beginning of the task with its ground-truth (labels of samples). Participants prepare and train their systems on this dataset. A test collection is provided at a later date. Participants interrogate their system and return their predictions to the organizers' committee. An evaluation is performed by the latter to compare the performance of the systems.

TBT task consists of the automatic categorization of TB cases in 5 target classes based on CT scans of patients. The five types considered are:

1. Infiltrative
2. Focal,
3. Tuberculoma
4. Miliary
5. Fibro-cavernous

The results will be evaluated using unweighted Cohens Kappa and accuracy.

SVR task aims to predict the degree of severity of TB cases. Given a TB patient, the main goal is to predict its severity score based on his 3D CT scan.The degree of severity is modeled according to 5 discrete values : from 1 ("critical/very bad") to 5 ("very good"). The score value is simplified so that values 1, 2 and 3 correspond to "high severity" class, and values 4 and 5 correspond to "low severity".

The classification problem are evaluated using ROC-curves (AUC) produced from the probabilities provided by the participants. For the regression problem, the root mean square error (RMSE) is used.

3 Our contribution

We proposed to extract semantic descriptors from 3D CT scans. We noticed that participants of the ImageCLEF TBT 2017 task used each extracted slice as a separate sample. Thus, hundreds of slices are considered as separate learning samples while these slices represent the same patient. This introduces a lot of noise. In addition, each slice will be assigned the label of the patient (its type) even those whose content does not present any information to identify the type of TB case. This introduces more noise. The majority of the participants [11] of ImageCLEF 2017 highlighted this problem and its impact on the results.

To overcome this problem, we believe that the simplest solution is to produce a single descriptor for each patient. This constitutes the key idea of our contribution.

Our proposed system goes through three main stages:

1. Input data pre-processing
2. Features extraction
3. Learning a classification model

We will detail each step in the following.

3.1 Input data pre-processing

We remind that in both tasks, 3D CT scans are provided in compressed Nifti format. Firstly, we decompress the files and extract the slices. At the end, we have three sets of slices corresponding to the three dimensions of the 3D image. For each dimension and for each Nifti image we obtain a number of slices ranging from 50 to 400 jpeg images.

The visual content of the images extracted from the different dimensions is not similar. Indeed, the images of each dimension are taken with from a different angle of view.We noticed from our experiments that the slices of the -Y- dimension give better results compared to the two others (X and Z). However, the following steps can be applied to slices of any of the three dimensions.

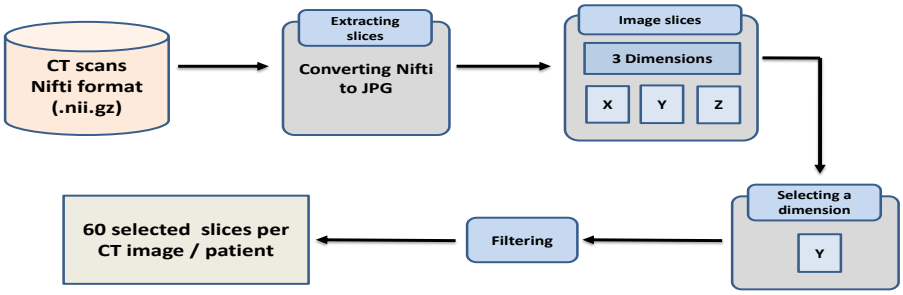

Fig. 1. Pre-processing of input data.

On the other hand, not all slices necessarily contain relevant information that can be useful to identify types of TB. This is why, it is essential to filter slices by keeping only those that can be informative and may contain relevant information. Moreover, since we want to extract a single descriptor per patient, it is essential to keep the same number of slices for each patient. We found that there is usually a maximum of 60 slices visually informative. Since the slices are ordered, the 60 most informative are usually at the center of the list. We propose then to keep the 60 middle slices. This is not optimal but we opted for this choice for a fully automatic approach. This choice can be improved by performing a manual filtering with the intervention of a human expert, preferably with medical skills on TB disease. Figure 1 summarizes the process.

3.2 Features extraction

After slices extraction and filtering, we propose to extract a single descriptor per patient. The transfer learning presents in this context an interesting track that can be exploited. The results of SGEast [11] and even other teams in the same task of ImageCLEF 2017 proved the efficiency of this approach [4, 11]. Indeed,

SGEast opted for the transfer learning where they exploited the output of a Resnet-50 [8] deep learner layer. However, this idea presents a problem of the resulting descriptor size. Indeed, for example, SGeast considered a descriptor per slice and not per patient. However, since we want to have a single descriptor, it is important that the information extracted from each slice must not be very large. Therefore, we propose to describe each slice by semantic information. This idea is inspired by the work presented in [7].

So, we choose to exploit the probabilities predicted by a deep learner trained on the set of slices. If K is the number of classes considered, this information typically corresponds to the K predicted probability values for the K classes (five probabilities of the five types for the TBT task, or the five severity degrees for the SVR task). We obtain then for each slice K values corresponding to the number of the considered classes.

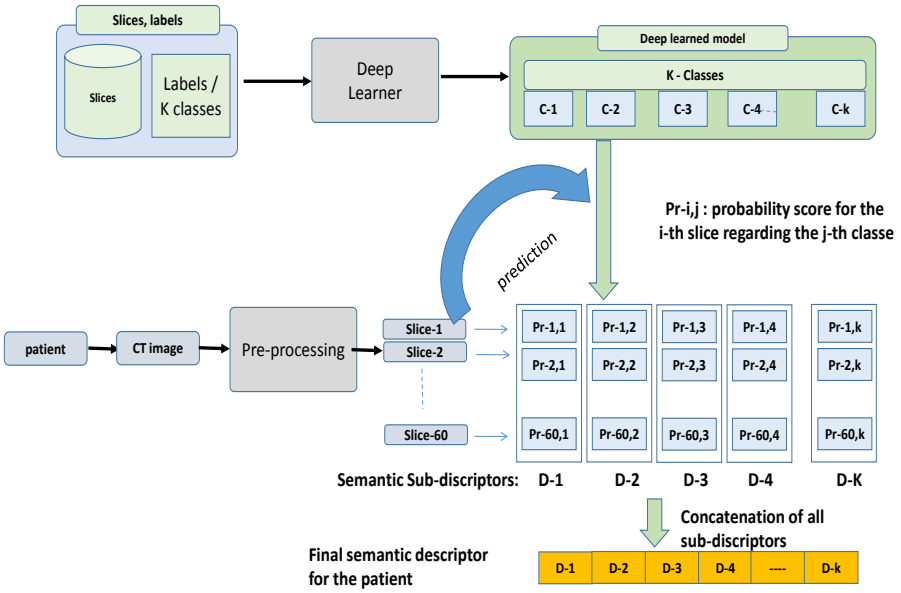

Fig. 2. Our semantic features extraction process.

Furthermore, K sub-descriptors are generated: D_1, D_2, D_3, D_4, ... D_k. Each sub-descriptor D_i contains the predicted probabilities for the class i for all the slices of the patient. A final semantic descriptor is constructed by concatenating the K sub-descriptors. Figure 2 details the process of the semantic feature extraction for one patient.

3.3 Learning a classification model

In this step, we propose to exploit the semantic descriptors of patients obtained in the previous step. Any approach of supervised classification can be applied as shown in figure 3.

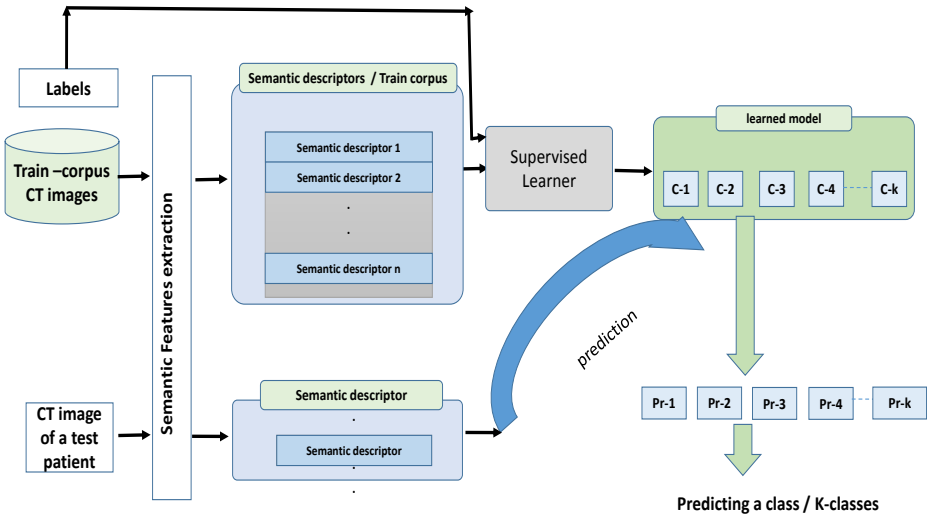

Fig. 3. Learning a classification model based on the semantic descriptors.

We recommend for this step some ideas:

- To use a deep learner having as input the semantic descriptors of patients and their labels. As an alternative, we propose to use a bagging method that collaborates several learners and sub-samples the train collection. This would lead to better results as our experiments showed.
- To apply a samples selection, especially in the TBT task where several CT images were provided for some patients. We noticed in our experiments that using all the images for each patient introduces a lot of noise and would give less good results than using only one image per patient. An alternative consists of creating multiple sub-collections where each one contains a different single CT image per patient, and generating then a learner on each sub-collection to aggregate finally their results. This would probably lead to a much more robust model.

4 Experiments and results

We describe in the following sections our runs submitted to the TBT and SVR tasks.

We implemented the semantic descriptor approach described in section 3. We used for that the following tools:

- The Caffe frawework [10] for deep learning;
- Weka [6] for testing several learning and classification algorithms;
- med2image [1] for the conversion of nifti medical images to the classic Jpeg format.

We chose to use slices of the -Y- dimension because our experiments showed that they are more suitable than those of the two others and got better results.

For descriptors extraction, our approach consists to learn a deep model to generate semantic information. Unfortunately, we had problems with our machines deployed for training our deep learner. Due to lack of time, we could not achieve the learning process. As an alternative to this step, we deployed the same model as the one proposed by the SGeast team [11] at the CLEF 2017 TBT Task. The model is accessible from the following link [2]. It is based on a Resnet-50 [8] and got the best results at the TBT task of 2017 edition. We have therefore exploited the outputs of the last layer (named *prob*) of the Resnet-50 corresponding to the probabilities of the 5 considered classes.

4.1 TBT task

Dataset: The dataset used in TBT tasks includes chest CT scans of TB patients along with the TB type. Some patients include more than one scan. All scans belonging to the same patient present the same TB type. Table 1 summarizes the distribution of CT scans according to the five types of TB considered.

Table 1. Dataset given for Tuberculosis TBT task [9].

	Train		Test	
TB types	#Patients	#CTs	#Patients	#CTs
Type 1	228	376	89	176
Type 2	210	273	80	115
Type 3	100	154	60	86
Type 4	79	106	50	71
Type 5	60	99	38	57
Total	677	1008	317	505

Experimental protocol: We used the train collection provided by the organizers and we split it into two sub-collections: 80% for training and 20% as validation set. We have exploited in all our runs the semantic descriptors generated as previously described. We tested several learners in the classification step. We finaly submitted three main runs. The other submissions are some variants or are generated through the fusion of some of these three runs:

– Run 1 (TBT_mostaganemFSEI_run1): random forest as supervised classifier. We tuned the two parameters referring to the number of iterations performed and the number of features selected randomly;
– Run 2 (TBT_mostaganemFSEI_run2): bagging of a set of random forest learners. We tuned the number of learners for the bagging and the same two parameters as Run1 for random forest;
– Run 4 (TBT_mostaganemFSEI_run4): A hierarchical classification. We organized the five 5 classes into a hierarchical structure as described in figure 4. We have created two new virtual classes $V-1$ and $V-2$. $V-2$ regroups the three classes Type 1, Type 2, and $V-2$ contains the classes Type 3, Type 4 and Type 5. We have reorganized our collections in order to achieve a classification on two different levels. In the first stage, we classify the samples into two virtual classes $V-1$ and $V-2$. In the second level of classification, we performed a classification of the samples regarding the set of classes of the predicted class in the previous stage. In two classification process we used a random forest learner by tuning its two parameters as described for Run1.

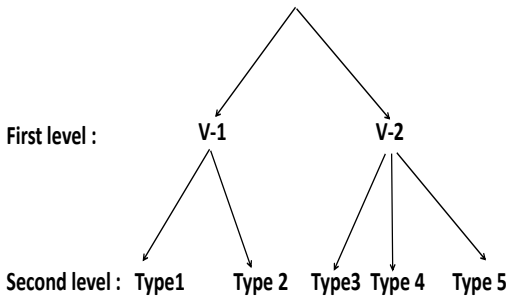

Fig. 4. Hierarchical re-organization of TBT types.

Results: Table 2 shows the results obtained by our runs on validation collection.

Table 2. Results on validation set for TBT task.

Runs	Kappa	Accuracy
Run 1 (TBT_mostaganemFSEI_run1)	0.21	0.38
Run 2 (TBT_mostaganemFSEI_run2)	0.25	0.41
Run 4 (TBT_mostaganemFSEI_run4)	0.26	0.52

Table 3 shows the results obtained by our runs on the evaluation performed by the ImageCLEFcommittee.

Table 3. Results on test set for TBT task.

Runs	Kappa	Rank	Accuracy	Rank
Run 1 (TBT_mostaganemFSEI_run1)	0.0412	28	0.2650	29
Run 2 (TBT_mostaganemFSEI_run2)	0.0275	29	0.2555	32
Run 4 (TBT_mostaganemFSEI_run4)	0.0629	25	0.2744	27

As shown on validation results, Run 4 has been our best submission and got also the best results on test collection compared to run 1 and run 2.

Figures 5 and 6 describes the results and ranking of all submissions on TBT task in terms of kappa coefficient and accuracy, respectively.

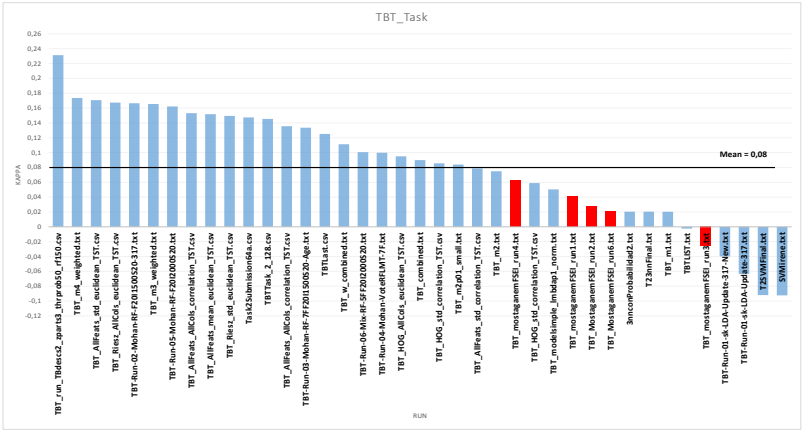

Fig. 5. Results and ranking in terms of Kappa coefficient on test data for TBT Task.

Although the results achieved by our submissions are not well ranked compared to those of the top of the list, we can notice that several runs belong to the same teams that had good results, and they probably do not differ too much. On the other hand, we recall that our semantic descriptors were extracted using a model that was not very well trained. In fact, we met problems with our machines during the training of our deep learner. Indeed, although SGEast's deployed model got the best results at ImageCLEF 2017 Tuberculosis TBT task, we did not have the ability to perform exactly the same pre-processing performed by this team as described in [11]. We believe that our semantic descriptors could give better results if they are extracted from a more adapted and well-developed deeper model.

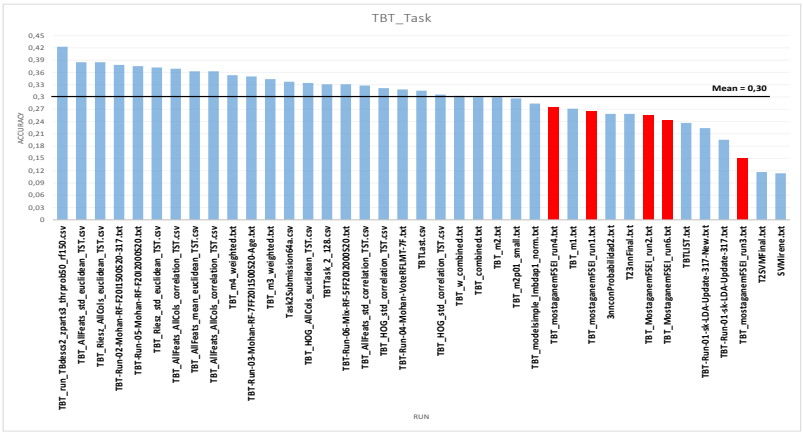

Fig. 6. Results and ranking in terms of accuracy on test data for TBT Task.

4.2 SVR task

Dataset: The dataset for SVR task includes chest CT scans of TB patients along with the corresponding severity score (1 to 5). Scores from 1 to 3 correspond to the "High" severity whereas the two scores 4 and 5 refer to the "Low" degree of severity. Table 4 summarizes the distribution of CT scans according to two severity classes.

Table 4. Dataset given for Tuberculosis SVR task [9].

	Train	Test
Low severity	90	62
High severity	80	47
Total	170	109

Experimental protocol: We generated in a first step the semantic descriptors following the approach described in the section 3. For the prediction of TB severity scores, we treated the problem as a classification problem. We used for this two approaches :

1. Multi-class classification problem: we considered the five scores as separate classes. We then tested several classifiers. We selected two that have been most effective compared to those tested: Random forest, bagging of a set of random forest learners.
2. Hierarchical classification: We organized our data in order to carry out a hierarchical classification. We considered the hierarchy described in figure 7. Then, a two-level hierarchical classification is carried out. In the first level

the samples are classified into “High” or “Low” classes. In the second level, the samples are reclassified into the descending classes of the one predicted in the first level.

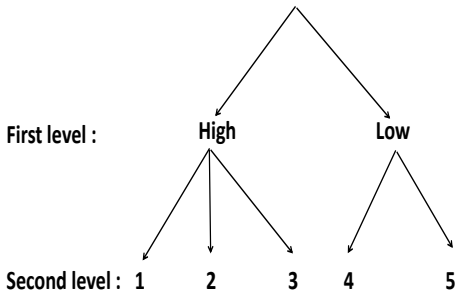

Fig. 7. The hierarchy of classes considered for SVR Task.

We submitted five runs:

1. Run 1 (SVR_mostaganemFSEI_run1): Multi-class model using Random forest as classifier. We tuned the two parameters : the number of iterations performed and the number of features randomly chosen;
2. Run 2 (SVR_mostaganemFSEI_run2) : Multi-class model using a bagging of a set of random forest learners with sub-sampling of the main train collection. We created two sub-collections by balancing the number of samples for the 5 classes. We then merged the results obtained by the two sub-collections;
3. Run 3 (SVR_mostaganemFSEI_run3): Hierarchical classification using a Bagging of a set of Random forest learners in each level of the hierarchical classification process.
4. Run 4 (SVR_mostaganemFSEI_run4): fusion of Run 1 and Run 2
5. Run 6 (SVR_mostaganemFSEI_run6): fusion of Run 3 and Run 1

Results: Table 5 shows the results obtained by our runs on validation collection.

Table 5. Results on validation set for SVR task in terms of Accuracy and Root Mean Square Error (RMSE).

Runs	Accuracy	RMSE
Run 1 (SVR_mostaganemFSEI_run1)	0.41	0.37
Run 2 (SVR_mostaganemFSEI_run2)	0.36	0.45
Run 3 (SVR_mostaganemFSEI_run3)	0.56	0.3
Run 4 (SVR_mostaganemFSEI_run4)	0.42	0.36
Run 6 (SVR_mostaganemFSEI_run6)	0.48	0.34

Table 6 shows the results obtained by our runs on the evaluation performed by the ImageCLEF committee on test collection.

Table 6. Results on test set for SVR task.

Runs	RMSE	Rank	AUC	Rank
Run 1 (SVR_mostaganemFSEI_run1)	1.0227	19	0.5971	26
Run 2 (SVR_mostaganemFSEI_run2)	1.0837	22	0.6127	22
Run 3 (SVR_mostaganemFSEI_run3)	0.9721	12	0.5987	25
Run 4 (SVR_mostaganemFSEI_run4)	1.0137	18	0.6107	24
Run 6 (SVR_mostaganemFSEI_run6)	1.0046	16	0.6119	23

We can see that our Run 3 got best results in terms of RMSE compared to our other runs on validation collection and even on test data. However, in terms of AUC, Run 2 seems to be more efficient.

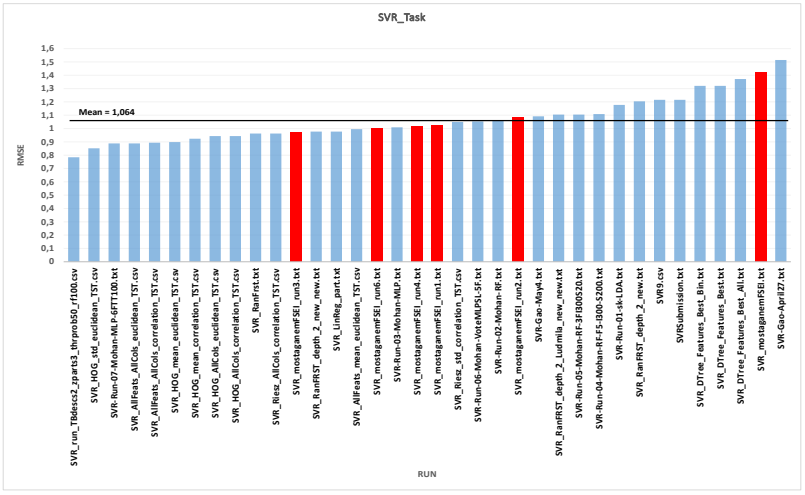

Fig. 8. Results and ranking in terms of Root Mean Square Error on test collection.

Figures 8 and 9 describes the results and ranking of all submissions on SVR task in terms of RMSE and AUC values, respectively.

We can see that our best run is ranked 12^{th} out of 36 submissions. However, the difference between the performances of the 12 best runs is not very significant. We recall that our best result is achieved by a hierarchical classification approach using a bagging of random forest learners at each level of the hierarchy. We believe that our approach could give better results using a well-trained deep model in the semantic features extraction step.

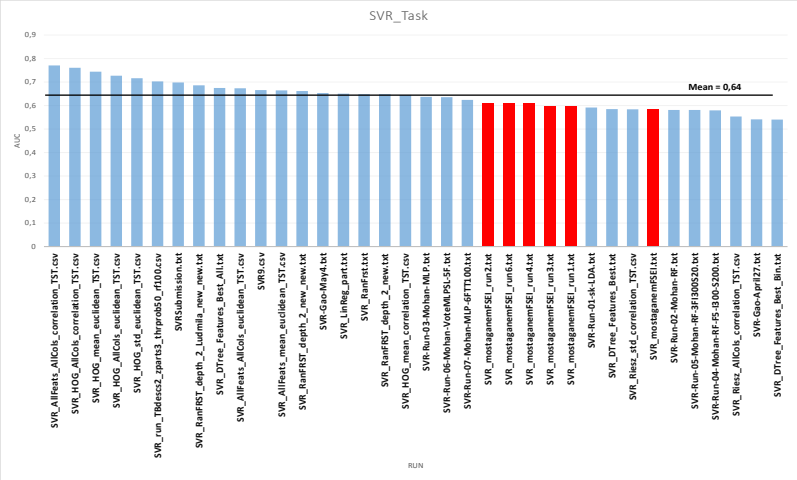

Fig. 9. Results and ranking in terms of Area Under ROC curve on test collection.

5 Conclusion and future works

We have described in this article our contributions to the TBT and SVR tasks of ImageCLEF Tuberculosis 2018. We proposed an approach that consists in extracting a single semantic descriptor for each CT image / patient instead of considering all the slices as separate samples. Unfortunately, we could not achieve the training of our deep learner. However, the results obtained show that this approach could be much more efficient and give more interesting results if it is applied properly.

As perspectives, we plan to adopt enrichment strategies and learning samples selection. Indeed, one of the characteristics of the problematic addressed in the SVR and TBT tasks is the nature of the provided data collections, which are of a small size and are noisy because of the presence of many slices that do not contain useful information. Our bagging and sub-sampling strategies adopted in our experiments confirmed this. In addition, we noticed during the sub-sampling of our data that the deletion or addition of some samples had an impact on the results. On the other hand, filtering slices effectively to keep only those that are truly informative is a key idea that could further improve system performance as reported by several participating teams [11]. Furthermore, we noticed in our experiments that there is a difference in terms of precision achieved for each studied class. Indeed, some classes are more difficult to identify than others. This is also an interesting track to study.

References

1. med2image: https://github.com/fnndsc/med2image. Last check: 30/05/2018.
2. Sgeast model for imageclef 2017 tubeculosis task : https://github.com/maizesix92/imageclef2017_tb_sgeast. Last check: 30/05/2018.

3. Cid, Y.D., Kalinovsky, A., Liauchuk, V., Kovalev, V., Müller, H.: Overview of the imageclef 2017 tuberculosis task - predicting tuberculosis type and drug resistances. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. (2017), http://ceur-ws.org/Vol-1866/invited_paper_1.pdf
4. Dicente Cid, Y., Kalinovsky, A., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEFtuberculosis 2017 - predicting tuberculosis type and drug resistances. In: CLEF2017 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Dublin, Ireland (September 11-14 2017)
5. Dicente Cid, Y., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEFtuberculosis 2018 - detecting multi-drug resistance, classifying tuberculosis type, and assessing severity score. In: CLEF2018 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Avignon, France (September 10-14 2018)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations **11**(1), 10–18 (2009)
7. Hamadi, A., Mulhem, P., Quénot, G.: Extended conceptual feedback for semantic multimedia indexing. Multimedia Tools Appl. **74**(4), 1225–1248 (2015). https://doi.org/10.1007/s11042-014-1937-y, https://doi.org/10.1007/s11042-014-1937-y
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
9. Ionescu, B., Müller, H., Villegas, M., de Herrera, A.G.S., Eickhoff, C., Andrearczyk, V., Cid, Y.D., Liauchuk, V., Kovalev, V., Hasan, S.A., Ling, Y., Farri, O., Liu, J., Lungren, M., Dang-Nguyen, D.T., Piras, L., Riegler, M., Zhou, L., Lux, M., Gurrin, C.: Overview of ImageCLEF 2018: Challenges, datasets and evaluation. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018), LNCS Lecture Notes in Computer Science, Springer, Avignon, France (September 10-14 2018)
10. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
11. Sun, J., Chong, P., Tan, Y.X.M., Binder, A.: Imageclef 2017: Imageclef tuberculosis task - the sgeast submission. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. (2017), http://ceur-ws.org/Vol-1866/paper_130.pdf