ImageCLEF 2019: Deep Learning for Tuberculosis CT Image Analysis

Abdelkader Hamadi[0000−0001−9990−332X], Noreddine Belhadj Cheikh, Yamina Zouatine, Si Mohamed Bekkai Menad, and Mohamed Redha Djebbara

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
noreddine.belhadjcheikh@univ-mosta.dz
zouatineyamina@gmail.com
bekkai.menad@univ-mosta.dz
redha.djebbara@univ-mosta.dz

Abstract. In this article, we present our methodologies used in our participation to the two subtasks of the ImageCLEF 2019 Tuberculosis Task (SVR and CTR). Our contributions are essentially based on deep learning and other machine learning techniques. In addition to the use of deep learners, semantic descriptors are tested to represent patients CT scans. These features are extracted after a first learning step. Our submissions on the test corpus reached AUC value of about 65% in the SVR task and 63% in CTR. These results offered us the seventh and the eighth places in SVR and CTR, respectively. We believe that our contributions could be further improved and might give better results if they applied properly and in an optimized way.

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

1 Introduction

Tuberculosis (TB) is a deadly disease. Its early diagnosis can give the necessary treatment and prevent the death of patients. The technological advancement especially in the field of artificial intelligence and precisely supervised learning opens the door for researchers to study the possibility of an automatic diagnosis. This would speed up the process and lower its cost. Several researchers have invested their efforts in recent years, especially within the medical image analysis community. In fact, a task dedicated to this disease had been adopted as part of the ImageCLEF evaluation campaign in its editions of the three last years [5, 7, 6]. In this task, the objective is to automatically analyze the 3D CT images of

TB patients to detect semantic information: the type of Tuberculosis, the degree of severity of the disease, information related to the state of the lungs, etc. In ImageCLEF 2019 two sub-tasks of the main task “ImageCLEFmed Tuberculosis” are considered: Severity Scoring (SVR) and CT Report (CTR). In the first task, the goal is to deduce automatically from a CT image whether a TB case is severe or not. In the second one, the problematic consists to deduce automatically from the same type of image information about patient lungs state: which of the two lungs is affected, the presence of caverns, etc.

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

- Helping medical doctors in the diagnosis and determining the state of the patient through image processing techniques;
- Predicting quickly the TB severity degree to make quick decisions and give effective treatments;
- Assist doctors and medical officers to have accurate details about the patient’s lung condition by providing a report summarizing information describing the state of the lungs.

We present in the following our work that had been made in the context of our participation in the two sub-tasks of ImageCLEF 2019 Tuberculosis task: Tuberculosis Severity Scoring (SVR) and Tuberculosis CT Report (CTR) [6].

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 used to generate our predictions. We present and analyze in the same section the results obtained. We conclude in the last section by presenting some perspectives and future works.

2 Participation to ImageCLEF 2019

ImageCLEF 2019 [10] is an evaluation campaign that is being organized as part of the CLEF initiative labs. This campaign offers several research tasks that welcome participation from teams around the world. For the 2019 edition, ImageCLEF organises four main tasks: ImageCLEFcoral, ImageCLEFlifelog, ImageCLEFmedical and ImageCLEFsecurity. In this work, we focus on the Tuberculosis task that takes part in the ImageCLEFmedical challenge. ImageCLEFmed Tuberculosis task includes two sub-tasks: Severity Scoring (SVR) and CT Report (CTR) that we describe in the following.

2.1 SVR and CTR Tasks description

In this paper, we focus on our participation in the SVR and the CTR sub-tasks. The main objective of these two challenges is the automatic analysis of Tuberculosis CT scans. In both tasks, the same data collections are used, one corpus for training and another set for test. The data is provided as 3D CT scans. All the CT images are stored in NIFTI file format with "nii.gz" extension file (gzipped .nii files). For each of the three dimensions of the CT image, we find a number of slices varying according to the dimension considered (512 images for the Y and X dimensions, from 40 to 250 images for the Z dimension). Each slice has a size of about 512×128 pixels for the X and Y dimensions and 512×512 pixels for the Z dimension.

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 submitted systems.

SVR task aims to predict the degree of severity of TB cases. Given a CT scan of TB patient, the main goal is to predict the severity of his illness 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 is evaluated using two measures: 1) Area Under ROC-curve (AUC) and 2) Accuracy.

CT Report task has as objective to automatically generate a report based on the patient's CT scan. This report should include the following six pieces of information in the binary form (0 or 1):

1. is the left lung affected ?
2. is the right lung affected ?
3. the presence of calcifications;
4. the presence of caverns;
5. the presence of pleurisy;
6. the lung capacity decrease.

This task is considered as a multi-binary classification problem (6 binary findings). The ranking of this task is done first by average AUC and then by min AUC (both over the 6 CT findings).

3 Our contributions

We proposed to use the system presented in Figure 1. The latter goes through two essential steps: input data pre-processing and learning a classification model. A third optional step is added in order to improve the performance of the first learning step. The latter includes a second learning stage by using a recurrent neural network (LSTM) or by generating semantic features and exploiting them through a learner or a deep learner. We will detail our proposed system in the following.

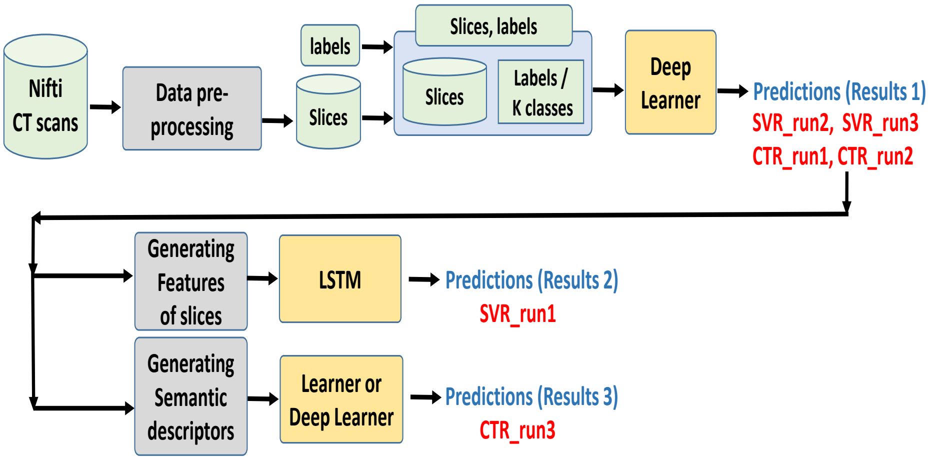

Fig. 1. The architecture of the overall proposed system

3.1 Input data pre-processing

We remind that in both tasks, 3D CT scans are provided in compressed Nifti format. Firstly, we decompressed the files and extracted the slices. At the end, we got three sets of slices corresponding to the three dimensions of the 3D image. For each dimension and for each Nifti image we obtained a number of slices ranging according to the dimension considered (512 images for the Y and X dimensions, from 40 to 250 images for the Z dimension).

The visual content of the images extracted from the different dimensions is not similar. Indeed, the images of each dimension are taken from a different angle of view. We noticed from our experiments that the slices of the -Z- dimension give better results compared to the two others (X and Y). This remark concerns our proposed approaches. This is why we used in our work the Z-dimension. However, all steps can be applied to slices of any of the three dimensions.

After choosing the dimension to consider, we propose to filter the slices of each patient. Indeed, we can notice that many slices do not necessarily contain relevant information that could help to classify the samples. This is why we added a step to filter and select a number of slices per patient. For this, we propose two filtering approaches:

Automatic supervised filtering: In this approach, we select a set of patients from each of the considered classes (the five degrees of severity for the SVR task). Then, a professional radiologist selects for each patient, the slices likely to contain relevant information indicating the presence of Tuberculosis. The resulting set of slices constitutes a filtering group. Given a new patient, we compare each of its slices to the filtering group by calculating a distance measure: a weighted sum of distances between the slice and those of the filtering group. This comparison was made through a direct pixel-wise comparison. We selected at the end N slices that are judged to be the most similar to the filtering group. So, at the end, each patient is represented by the N filtered slices instead of all its extracted images. We think that this would reduce the noise introduced by the consideration of all slices. We tested in our contributions the value N=10.

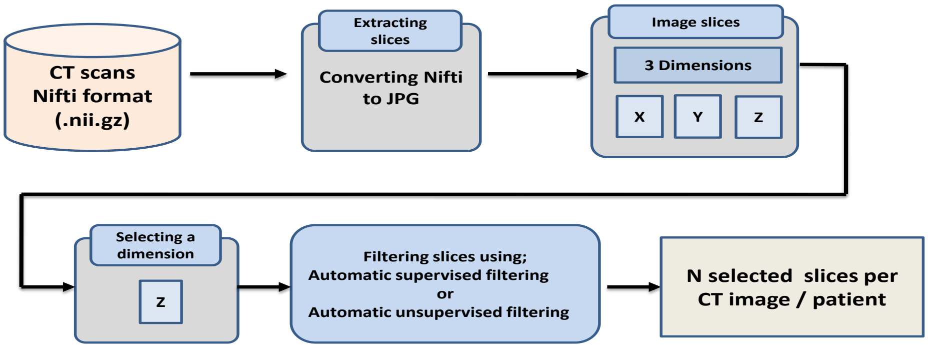

Fig. 2. Pre-processing of input data.

Automatic unsupervised filtering: We noticed that there is usually a maximum of 50/60 slices visually informative. Since the slices are ordered, the most informative slices are usually at the center of the list. We propose then to keep only the N middle ones. This is not optimal but we opted for this choice for a fully automatic and unsupervised 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 2 summarizes the pre-processing steps.

3.2 Deep learning model for CT image classification (first learning step)

As a deep learner, we chose to use Resnet-50 architecture because of its good results in the context of the same problematic in last Tuberculosis task editions [12]. On the other hand, we developed our model that we called “LungNet”. We present more details about this deep learner in the following section. The outputs of the deep learners deployed are considered as initial results. We exploited then these outputs to generate : 1) semantic features of patient that are used to reclassify the samples, and 2) features of slices organized in a sequence format that are pushed to LSTM input as it is described in section 3.5.

LungNet Deep Learner: We proposed and developed our deep learner architecture for CT Image Analysis that we called “LungNet”. The input to the latter is an RGB image of size 119x119, followed by five convolutional layers and two fully connected layers. Initially input data were in nifty format. Slices of the CT scans are 1-channel gray-level images. However, we extracted the slices using med2image tool [1]. This software converts the slices to jpeg format. To avoid introducing noise by using this extraction method, we can do better by reading directly image pixels values using Niftilib library for python that was suggested by the task organizers. The idea behind using med2image to extract slices is that we planned to filter the slices by a medical expert intervention. This process needs to sees the content of slices.

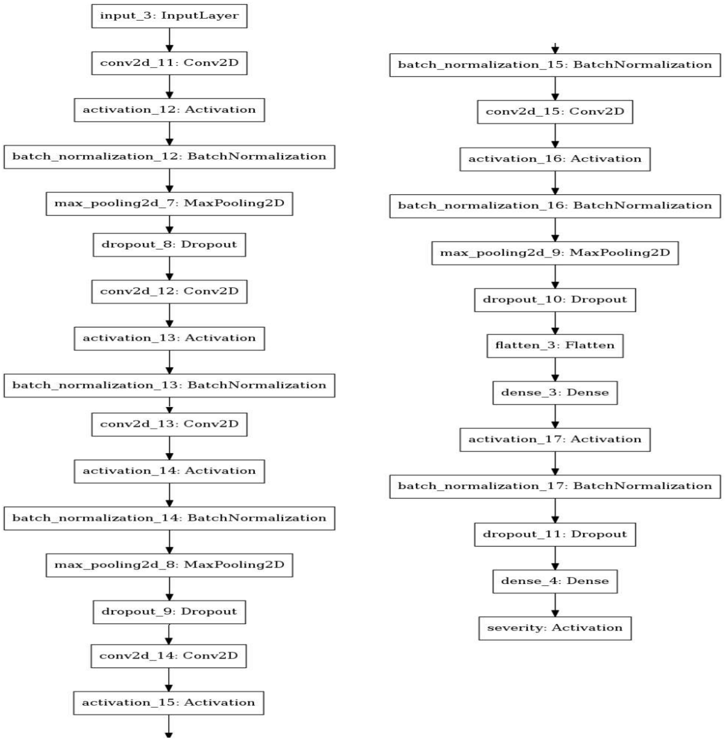
input_3: InputLayer
conv2d_11: Conv2D
activation_12: Activation
batch_normalization_12: BatchNormalization
max_pooling2d_7: MaxPooling2D
dropout_8: Dropout
conv2d_12: Conv2D
activation_13: Activation
batch_normalization_13: BatchNormalization
conv2d_13: Conv2D
activation_14: Activation
batch_normalization_14: BatchNormalization
max_pooling2d_8: MaxPooling2D
dropout_9: Dropout
conv2d_14: Conv2D
activation_15: Activation
batch_normalization_15: BatchNormalization
conv2d_15: Conv2D
activation_16: Activation
batch_normalization_16: BatchNormalization
max_pooling2d_9: MaxPooling2D
dropout_10: Dropout
flatten_3: Flatten
dense_3: Dense
activation_17: Activation
batch_normalization_17: BatchNormalization
dropout_11: Dropout
dense_4: Dense
severity: Activation

Fig. 3. The architecture of the LungNet Deep Learner

After each convolutional layer "relu" activation is applied followed by a local normalization and MaxPooling. The first, the second and the third convolution blocks have dropout layers to reduce overfitting. The sigmoid activation function is applied to the output layer in order to predict values in the range of 0 to 1.

Figure 3 illustrates the architecture of the Lungnet model.

3.3 Semantic Features extraction

We implemented the method of semantic descriptors extraction described in [8] with slight differences. After slices extraction and filtering, we generated a single descriptor per patient to exploit it through a transfer learning process. The results of SGEast [12] and even other teams in the same task of ImageCLEF 2017 proved the efficiency of this approach [12, 8].

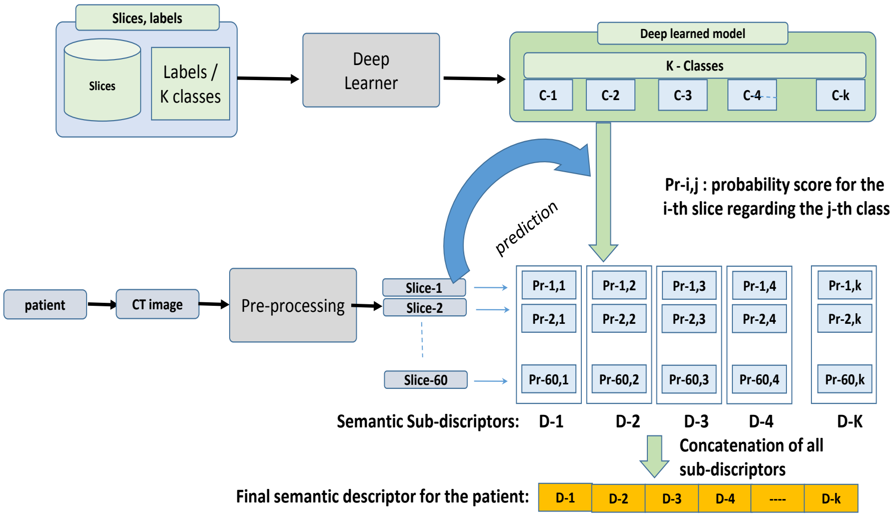

Fig. 4. Semantic features extraction process [8].

So, we chose to exploit the probabilities predicted by a deep learner trained on a set of slices. If K is the number of classes considered, these predictions typically correspond to the K predicted probability values for the K classes (For SVR Task, $K = 5$: the five severity degrees). We obtain then for each slice K values corresponding to the probabilities of the K considered classes.

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 4 details the whole process of the semantic features extraction for one patient.

3.4 Learning a classification model based on semantic features (second learning step)

We propose to exploit the semantic descriptors of patients described previously. Any approach of supervised classification can be applied as shown in figure 5. We tested in our experiments SVM as supervised classifier. However, Random Forests and bagging of Random Forests have shown good results in the context of the same problematic [8].

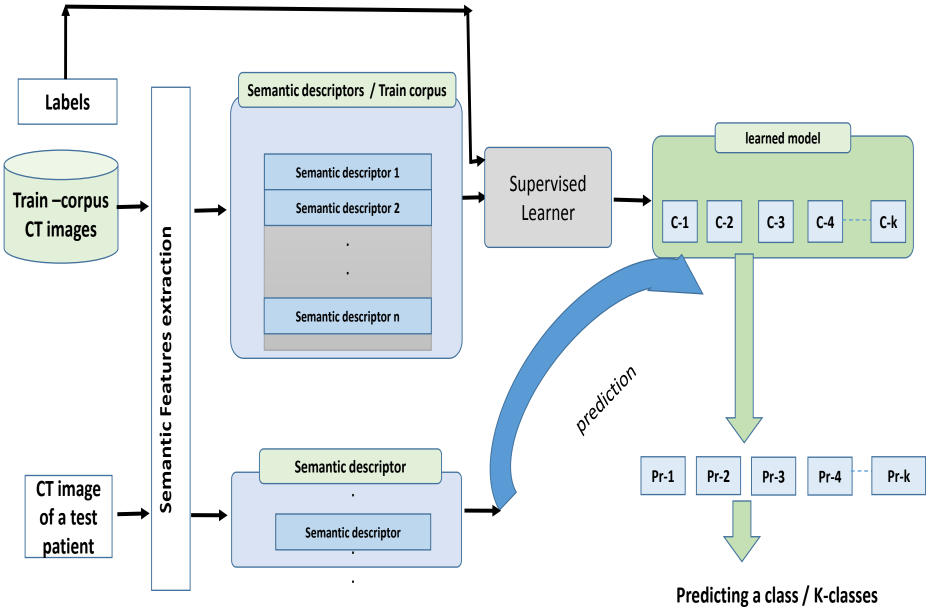

Fig. 5. Learning a classification model based on the semantic descriptors [8].

We recommend some ideas for this step:

- To use a deep learner having as input the semantic descriptors of patients and their labels. As an alternative, it would be interesting to use a bagging method that collaborates several learners and sub-samples the train collection. This would lead to better results as presented in [8];
- To apply samples selection and data augmentation.

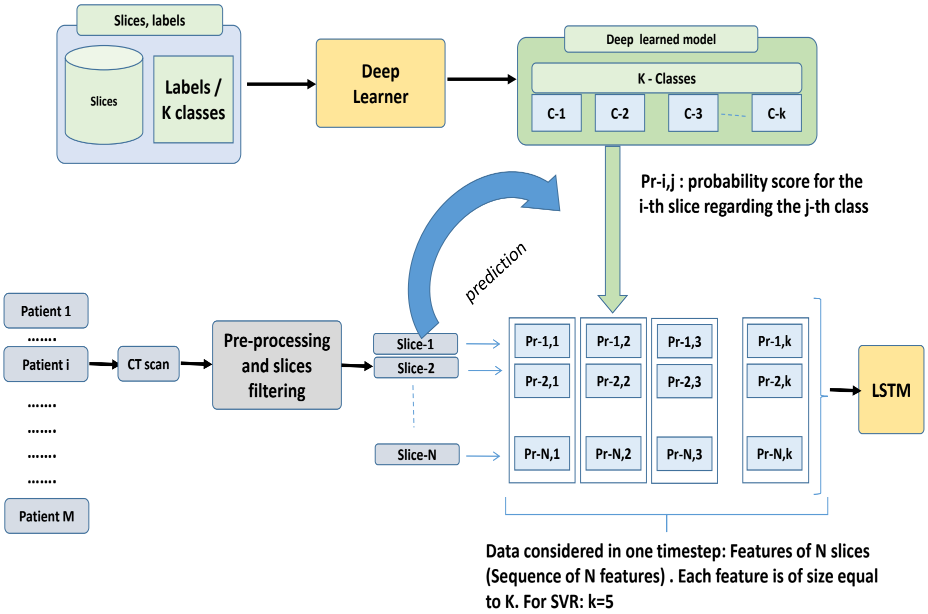
Slices, labels
Slices
Labels / K classes
Deep Learner
Deep learned model
K - Classes
C-1
C-2
C-3
C-k
Pr-i,j : probability score for the i-th slice regarding the j-th class
prediction
Patient 1
Patient i
Patient M
CT scan
Pre-processing and slices filtering
Slice-1
Slice-2
Slice-N
Pr-1,1
Pr-1,2
Pr-1,3
Pr-1,k
Pr-2,1
Pr-2,2
Pr-2,3
Pr-2,k
Pr-N,1
Pr-N,2
Pr-N,3
Pr-N,k
LSTM
Data considered in one timestep: Features of N slices (Sequence of N features) . Each feature is of size equal to K. For SVR: k=5

Fig. 6. Exploiting slices by LSTM as time series.

3.5 LSTM as classification model

As each patient is described by a sequence of slices, it is interesting to test the LSTM (Long Short-Term Memory) [9] recurrent neural network that is suitable for such data type. However, it is not recommended to apply LSTM on slices as input. Extracting features from slices using deep learner and pushing them to LSTM seems to be a good alternative. We propose to describe each slice by a feature of size equal to the number of the considered classes (five classes for SVR task). This feature is composed of the five values corresponding to the probabilities of the considered classes. These values are obtained through a deep learning stage. After generating these features, they are pushed to LSTM neural network by considering the ordered set of slices of each patient as a sequence. Figure 6 describes the whole process.

4 Experiments and results

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

We used in our experimental work the following tools:

- med2image [1] for the conversion of nifti medical images to the classic Jpeg format;

- Tenserflow frawework [3] and Keras library [4] for deep learning;
- scikit-learn [11] library for testing several machine learning techniques.

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

Dataset: The dataset used in SVR task includes chest CT scans of TB patients along with some metadata regarding a set of 19 classes. 2 classes concern the SVR task, six other classes concern CTR task. The other values are considered as additional information regarding the patients that could be used as contextual information. Table 1 summarizes the number of CT scans for train and test collections.

Table 1. Dataset given for Tuberculosis SVR and CTR tasks [10].

	Train Collection	Test Collection
Number of patients	218	117

The same dataset is given for the CTR task. The samples are labeled regarding seven main target classes:

1. Target classes for SVR Task:
   (a) SVR_severity (binary class: HIGH and LOW). Another label called md_Severity is given (Five discrete values ranging from 1 to 5). We remind that values of md_Severity (1, 2 and 3) belong to the “HIGH” Severity case. The other two values (4 and 5) correspond to the “LOW” Severity.
2. Target classes for CTR Task (binary classes):
   (a) Left lung affected;
   (b) right lung affected;
   (c) presence of calcifications;
   (d) presence of caverns;
   (e) presence of pleurisy;
   (f) lung capacity decrease.

4.1 SVR task

Experimental protocol: We used the train collection provided by the organizers and we split it into two sub-collections: training and validation sets. We finally submitted three main runs. The other submissions concern some tested approaches that we could not optimize and finalize correctly because of luck of time:

- **SVR_FSEI_resnet50_run3**: results of ResNet-50 trained on 50% of training data. Each patient was represented by 50 slices filtered using the automatic unsupervised filtering approach that was described in section 3.1. The slices were adapted by resizing them directly using the Python Imaging Library (PIL). The input images of Resnet50 are of size 199 × 199;
- **SVR_FSEI_lungnet_run2**: results of LungNet deep learner trained on 80% of data. Each patient was represented by 10 slices filtered using the automatic supervised filtering approach that was described in section 3.1;
- **SVR_FSEI_lstm_run8**: results of LSTM exploiting outputs of Lungnet deep learner. Each patient was represented by 50 slices filtered using the automatic unsupervised filtering approach. So, a sequence for the LSTM learner is composed of the 50 features representing the 50 slices of the patient.

We considered for each run a hierarchical classification problem. Firstly, we classified the samples in the 5 classes corresponding to the five degrees of severity. Secondly, We deduced for each patient its predicted class using a majority vote on the predicted labels of all slices. Finally, the class predicted in the previous step is transformed to a binary value corresponding to the SVR_Severity class (HIGH if predicted_class $\in \{1, 2, 3\}$ and LOW if not).

Our tools and scripts used in our experiments are accessible in [2].

Results: Table 2 shows the results in terms of AUC and accuracy obtained by our runs on the evaluation performed by the ImageCLEF committee on test collection.

Table 2. Results on test set for SVR task.

Runs	AUC	Accuracy	Rank
SVR_FSEI_resnet50_run3	0.6510	0.6154	22
SVR_FSEI_lungnet_run2	0.6103	0.5983	33
SVR_FSEI_lstm_run8	0.6475	0.6068	25

We can see that **SVR_FSEI_resnet50_run3** got the best performance followed by **SVR_FSEI_lstm_run8**. These two runs were ranked 22th and 25th out of 54 submissions.

We note that for **SVR_FSEI_lungnet_run2** patients were represented by 10 slices (50 slices for the two other runs), it would be interesting to see the performance of the Lungnet model within 50 slices in order to make a detailed comparison regarding the other two runs.

Figures 7 and 8 describe the results and ranking of all submissions of SVR task in terms of AUC and accuracy, respectively.

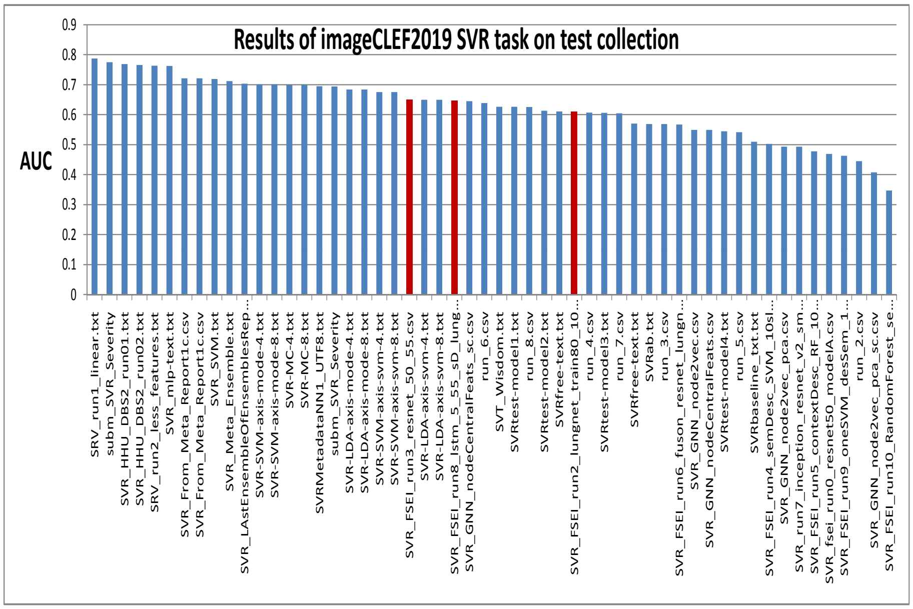

Fig. 7. Results and ranking in terms of AUC on test data for SVR Task.

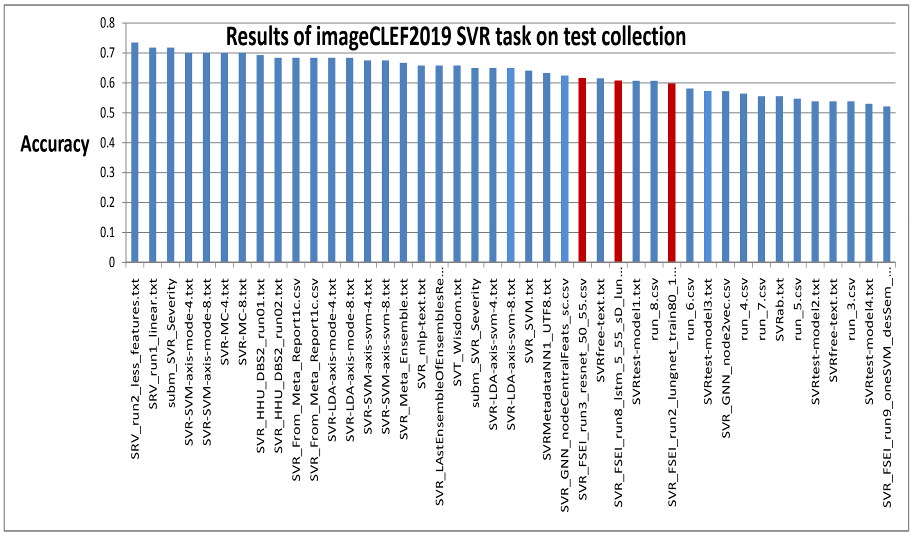

Fig. 8. Results and ranking in terms of Accuracy on test data for SVR Task

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 believe that our models could give better results after a more advanced data preprocessing including the use of masks, samples selection and data augmentation.

4.2 CTR task

Experimental protocol: We trained in a first step our deep models (Resnet and Lungnet). Secondly, we generated the semantic descriptors following the approach described in section 3. We treated the problematic as a multi-class problem in the first learning stage and as a binary classification problem in the second learning step. We used in the latter SVM as a binary classifier. We optimized its parameters independently for each target class.

We submitted three main runs:

1. **CTR_FSEI_run1**: results of LungNet trained on 50% of training data. Each patient was represented by 10 slices filtered using the automatic supervised filtering approach that was described in section 3.1;
2. **CTR_FSEI_run2** : results of LungNet trained on 70% of training data. Each patient was represented by 50 slices filtered using the automatic unsupervised filtering approach that was described in section 3.1;
3. **CTR_FSEI_run5**: SVM using semantic features. The latter are extracted using Resnet-50 as deep learner. Each patient was represented by 10 slices filtered using the automatic supervised filtering approach that was described in section 3.1.

Our tools and scripts used in our experiments are accessible in [2].

Results: Table 3 shows the results (in terms of Average-AUC and Min-AUC) and ranking obtained by our runs on the evaluation performed by the ImageCLEF committee on test collection.

Table 3. Results on test set for CTR task.

Runs	Mean AUC	Min AUC	Rank
CTR_FSEI_run1	0.6273	0.4877	14
CTR_FSEI_run2	0.6061	0.4471	17
CTR_FSEI_run5	0.5064	0.4134	32

We can see that our best results were obtained by **CTR_FSEI_run1** followed by **CTR_FSEI_run2**. However, we should mention here that we used the same sub-division of the corpus in two sub-parts (train and validation) for all CTR target classes, which is not optimal since the distribution of class values is not the same for the six target classes. This explains the disadvantage of the run **CTR_FSEI_run5** compared to the other two and also the low value of Min-AUC for the three runs. We believe that the semantic descriptors approach might perform better by making more efforts to optimize parameters or by testing another learner like the Bagging of Random Forests as presented in [8]. Considering a multi-label classifier constitutes also an interesting idea to test.

Figure 9 describes the results (in terms of Average-AUC) and ranking of all submissions of CTR task. Our best two runs were ranked 14th and 17th out of 35 submissions.

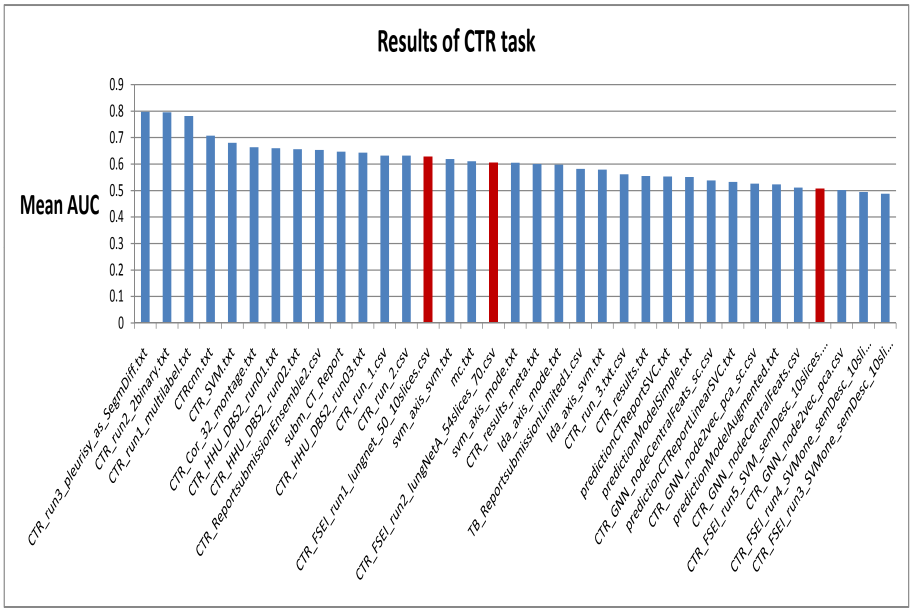

Fig. 9. Results and ranking in terms of Mean-AUC on test collection for CTR Task.

5 Conclusion and future works

We have described in this article our contributions to the SVR and CTR subtasks of ImageCLEFmed 2019 Tuberculosis task. We proposed to use after a data preprocessing step, a deep learner to classify samples in the target classes. We used for that ResNet-50 and proposed our LungNet architecture. Moreover, we proposed to extract a single semantic descriptor for each CT image / patient instead of considering all the slices as separate samples. We tested also LSTM as another alternative. Although our proposals had not been the best, the results obtained showed that these approaches could be much more efficient and might give more interesting results if they are applied in an optimized way.

As perspectives, we plan to adopt enrichment strategies and learning samples selection. 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 in an optimized way is a key idea that could further improve system performance. 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. The use of masks and tracking the visual contents of lungs could also be a good trail to explore.

References

1. med2image: https://github.com/fnndsc/med2image. Last check: 30/05/2018.
2. Resnet-50 and lungnet for tuberculosis severity scoring. mostaganem university at imageclefmed 2019 : tools to run experiments. https://github.com/anouar1991/imageCLEFfsei/tree/master/tools/application
3. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org
4. Chollet, F., et al.: Keras. https://github.com/fchollet/keras (2015)
5. 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)
6. Dicente Cid, Y., Liauchuk, V., Klimuk, D., Tarasau, A., Kovalev, V., Müller, H.: Overview of ImageCLEFtuberculosis 2019 - automatic ct-based report generation and tuberculosis severity assessment. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Lugano, Switzerland (September 9-12 2019)
7. 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)
8. Hamadi, A., Yagoub, D.E.: Imageclef 2018: Semantic descriptors for tuberculosis CT image classification. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. (2018), http://ceur-ws.org/Vol-2125/paper_82.pdf

9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Ionescu, B., Müller, H., Péteri, R., Cid, Y.D., Liauchuk, V., Kovalev, V., Klimuk, D., Tarasau, A., Abacha, A.B., Hasan, S.A., Datla, V., Liu, J., Demner-Fushman, D., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Garcia, N., Kavallieratou, E., del Blanco, C.R., Rodríguez, C.C., Vasillopoulos, N., Karampidis, K., Chamberlain, J., Clark, A., Campello, A.: ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), LNCS Lecture Notes in Computer Science, Springer, Lugano, Switzerland (September 9-12 2019)
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. 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