

UAIC2020: Lung Analysis for Tuberculosis Detection

Lucia-Georgiana Coca, Alexandra Hanganu, Ciprian-Gabriel Cusmuluc, Adrian Iftene

“Alexandru Ioan Cuza” University, Faculty of Computer Science, Iasi, Romania

{coca.lucia.georgiana, alexandra.hanganu,
cusmuluc.ciprian.gabriel, adiftene}@info.uaic.ro

Abstract. Tuberculosis is a bacterial infection that is transmitted through the air and affects the lungs. According to a WHO report, in Europe, in 2016 there were almost 228,000 people infected with tuberculosis. This disease can be treated if diagnosed in time and followed by a CT scan. For the 4th edition of the ImageCLEFmed Tuberculosis task, organizers proposed a new approach with a major impact on the real-world clinical routines, regarding generation lung-based rather than CT-based. Our solution proposed for generating an automated CT report is based on Support Vector Machine and Convolutional Neural Networks. The first algorithm we used is SVM, which provided the best results for the test images. Furthermore, CNN was also used and provided results that were almost as accurate as SVM.

Keywords: Support Vector Machine, Tuberculosis, CNN.

1 Introduction

Tuberculosis (TB) is caused by bacteria (*Mycobacterium tuberculosis*) that most often affects the lungs. Tuberculosis is curable and preventable if it is discovered and diagnosed in time. TB is spread from person to person through the air. When people with lung TB cough, sneeze or spit, they propel the TB germs into the air. A person needs to inhale only a few of these germs to become infected, due to the large number of people infected with TB, many researchers are involved in finding better solutions in treating and maintaining TB patients.

ImageCLEF 2020 [1] is an evaluation campaign that is being organized as part of the CLEF initiative labs¹. In this year's edition, the organizers decided to focus on the task of automatically generating a CT report, as it has important results that can have a major impact on real-world clinical routines. To make the task as attractive as possible, this year the generation of reports is based on the lungs and not on CTs. Labels for the left and right lungs were provided independently. The set of target labels in the CT Report have been updated in accordance with the opinion of the medical experts. This year, 3 labels were offered for each lung: the presence of tuberculosis lesions in

¹ <http://www.clef-initiative.eu/>

general, the presence of pleurisy and especially of the caverns. The size of the data set was also increased compared to the previous year.

This paper describes the participation of team UAIC2020, from the Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, in ImageCLEFtuberculosis 2020 task [2], at ImageCLEF 2020. The remaining of this paper was organized as follows: Section 2 describes the state of the art, Section 3 presents our methods, Section 4 evaluates the methods proposed and, in the end, we draw a conclusion and present future work.

2 State of the art

In previous editions, multiple different approaches were proposed for this task, mainly based on SVMs and CNNs.

In 2018 the most notable result was of team UIIP_BioMed [3] that managed to obtain the highest kappa score, 0.2312, with an accuracy of 0.4227 and root mean squared error of 0.7840. Furthermore, the team MedGift [4] scored 0.7708 in terms of ROC AUC, the highest in that year. Both teams had very different approaches, UIIP_BioMed used a Deep CNN and MedGift had an SVM with and RBF kernel.

In 2019 the leaderboard has not changed much, UIIP_BioMed [5] is still the leader with a mean AUC of 0.7968 and a min AUC of 0.6860 followed by CompElecEngCU [6] with an AUC of 0.7629. UIIP_BioMed used a 2D CNN whilst CompElecEngCU created 2D derived images by concatenating sagittal and coronal CT slices that are classified with a hybrid of a 2D CNN based on AlexNet [7]. Another approach, FI-IAugt [8], uses group performed random sampling of pixels of the CT volumes and used a combination of decision trees and weak classifiers in order to participate in the SVR task, this team is from our University and ranked 38 out of 54.

Our experience with classification and predictions stems from other similar works such as predicting cracks in images using CNN [9] but also fake news identification [10], anorexia detection [11] and claim identification [12].

3 Methods

The solution proposed for generating an automated CT report is based on Support Vector Machine and Convolutional Neural Networks.

Regarding the data, we converted the 3D NIFTI² file format to 2D PNGs representing specific sides of the lung. This transformation also had to be done to the masks provided by the organizers [13] [14] as they needed to match the previously resulted CT scans. In terms of pre-processing, the mask images were divided by lung lobes and the background was changed to black whilst the lung color was set to white. By doing so, new masks were generated for each lung, enabling us to work with each of them.

² <https://radiopaedia.org/articles/nifti-file-format>

The first algorithm we used is SVM [15], which provided the best results for the test images. Furthermore, CNN [16] was also used and provided results that were almost as accurate as SVM. The work was done in parallel and all the findings from one test run were tested on the other, leading both to a final form. The two algorithms dealt with each area and lung separately (each lung had a correspondent in the code for the Affected column, the Caverns column and the Pleurisy column), leading to a more complex solution that was catered to fit the needs of the given task. The final CT report was computed individually for both algorithms and the results were compared after each hyperparameter tuning, ensuring the best result.

3.1 Dataset

The given NIfTI files for the train images also had a CT Report, which was extremely helpful when training an algorithm as one could label the data accordingly. The dataset was highly imbalanced and that could have had a negative impact on the training of an algorithm.

Table 1 provides an insight in the patient affected in the training set.

Table 1. Patients Affected in the Training Set

Columns	Patients Affected in the Training Set (out of 283)
LeftLungAffected	211
RightLungAffected	233
CavernsLeft	66
CavernsRight	79
PleurisyLeft	7
PleurisyRight	14

3.2 Architecture

This section contains a description of the methods used for obtaining the results. The overall architecture of the project used to generate the CT Report is depicted from Figure 1. It can be seen that the generated CT is fed to each algorithm individually. The algorithms use the masks applied on the CT scans in order to issue a response.

3.3 Data Preprocessing

The solutions proposed required the images in NIfTI file format to be transformed into a 2D format. To achieve this, the file has been loaded using the NiBabel library³ and then transformed into a NumPy⁴ array. This array was then transformed to respect

³ <https://nipy.org/nibabel/>

⁴ <https://numpy.org/>

the type `numpy.int165`, having the shape equal to 3. This meant that there would be three available ways for the slices to be made: from left to right, from front to back and from top to bottom. Initially, 512 slices were made for each region.

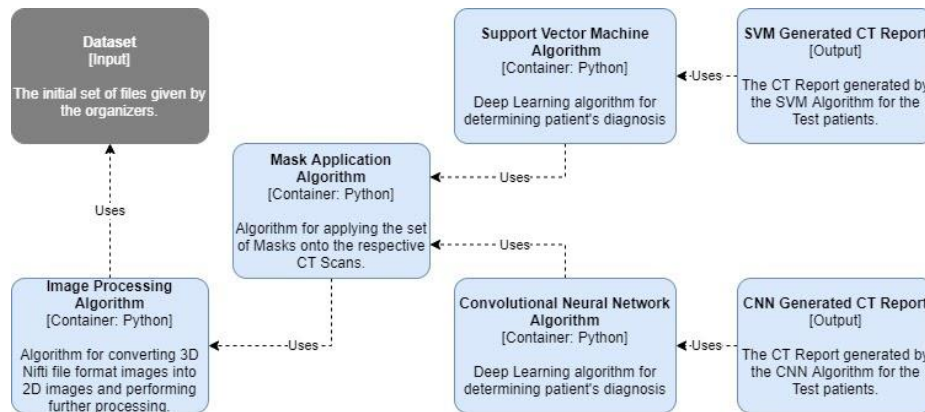


Fig. 1. Architecture of the model.

Working with the initial results of such a slicing led to a different approach for each of the regions. For the slices made from left to right, only a quarter of them remained, as they tended to become repetitive and the data was ruled redundant; moreover, the first 5 and last 10 pictures were removed as they oftentimes contained little to no information for our algorithms. The entirely empty pictures were eliminated based on the fact that the matrix was filled with the value 255 and offered no relevant information for further use. However, for slices from the front of the lungs to the back, the delete process was done in batches. First one was from 0 to 241, where the black images were removed and only a quarter of the remaining slices were used for working further with them. The second batch was from 241 to 272. Here, all slices were kept as these have proven to be the ones with the best resolution, highest contrast and most information for our algorithms. The last batch, was the one which contained images from 272 to 512, which is the total number of slices and its' processing consisted in the removal of the black images, while keeping only a quarter of the number of slices for further work. Finally, for the top to bottom resulting slices, only a quarter remained, the first 4 and last 8 pictures were removed and the entirely empty pictures were eliminated also. After selecting the required slices from both the CT scans and masks, a decision was taken to separate the lungs. Using the first set of masks, as the color-coded lungs were easier to work with, the images have been transformed into NumPy arrays and the work was done on the resulting array. For the anatomically left lung, the 127 values were set to 0 and for the right lung, 255 values were transformed into 0 and 127 values to 255. The masks that have resulted from this preprocessing

⁵ <https://numpy.org/doc/stable/reference/generated/numpy.dtype.html>

were then separated so work could be done on each lung lobe, individually for the three types of affections.

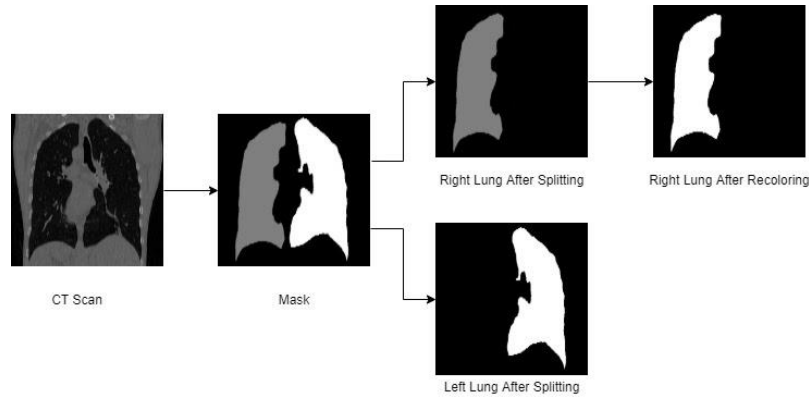


Fig. 2. Image preprocessing example.

Table 2. Dataset before and after preprocessing.

Images	Train		Test	
	Before	After	Before	After
CT Scan Images	455117	47892	184320	21951
Mask Images	477117	73762	184320	31552

3.4 Mask Application

After obtaining the images for both CT Scans and Masks, the next step was to apply the masks onto the scans⁶. To do this, the `cv2`⁷ library was used.

The method was based on taking a mask image and finding its correspondent in the CT Scan images. Our approach was to apply the images on the masks and not the other way around as to ensure that there would be no errors. After using `cv2.imread`⁸ to load the images, the masks had to be resized and for that we used as arguments the loaded mask image and for shape the scan image's `shape[1::-1]`. The last step was to use the `bitwise_and`⁹ function to apply the mask onto the image and then `cv2.imwrite`¹⁰ to save the resulting file.

⁶ <https://note.nkmk.me/en/python-opencv-numpy-alpha-blend-mask/>

⁷ <https://opencv.org/>

⁸ https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html?highlight=imread

⁹ https://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#bitwise-and

¹⁰ https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html?highlight=imwrite

3.5 Implementation

Our team proposed two different types of algorithms for the generation of an automatic CT Report: Support Vector Machine and Convolutional Neural Network. The two algorithms are different at core and required data to be looked at individually from this point on.

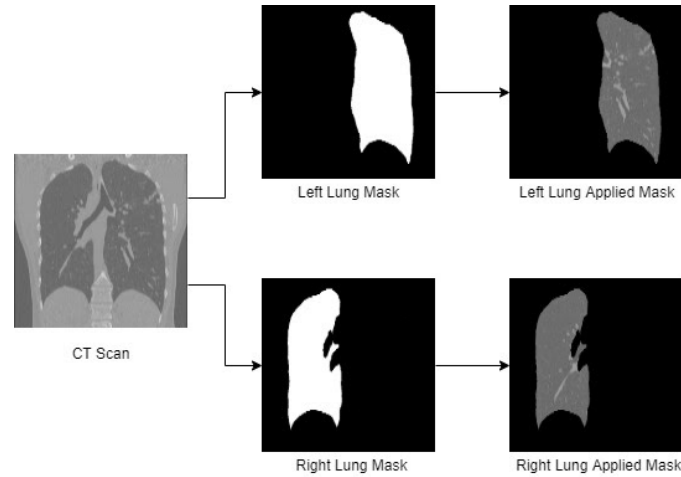


Fig. 3. Before and after mask application

To fully understand the implications of such an impactful disease and how to implement these algorithms, several professionals and students working in the medical field have been asked for insight. They have provided important information about how an expert looks at a CT scan of a patient with tuberculosis. Moreover, pointers have been given for important details to keep in mind when examining such an image. This initial research into the medical field has proved to be of great help. Being told that looking exclusively at the inferior part of a lung to find signs of pleurisy or for missing areas inside lungs to find caverns, has been a considerable guidance. Hence, this part of the research has been insightful and of tremendous value when going forward with the implementation of the two machine learning algorithms. The troubleshooting and the interpretation of the results has been easier as the problem was understood from both a technical and medical point of view.

Since SVM is a classification problem at core, the idea behind this implementation is that, taking all individual pictures from one patient, they were labeled accordingly, with 1 or 0, and then the final result represented the mean of all values returned for that patient, for that column. For the training of this algorithm, there were multiple libraries used, such as NumPy, Joblib¹¹, Keras¹² and sklearn¹³. The data needed to be transformed into a NumPy array first. For that, it was saved into a three dimensional

¹¹ <https://joblib.readthedocs.io/en/latest/>

¹² <https://keras.io/>

¹³ <https://scikit-learn.org/stable/>

array, that held the side (left or right), the region of the lung (front to back, left to right or top to bottom) and the type of the illness (affected, caverns, pleurisy) and only after it was saved into a `np.array`¹⁴. What is to be noted is that the pictures have been resized to 32×32 pixels, as they were initially 512×512. When it comes to the actual training, `svm.SVC`¹⁵ was used with the ‘rbf’ kernel and other different parameters that will be discussed during evaluation. The resulting output was saved using `joblib`, so that when conducting further tests, it will be easier to load the data. Computing was done also onto a three-dimensional array, as it was the best way to hold information individually. Our training function had an input parameter so that the state of the models would be available to the algorithm (saved was `True` if there existed saved models and `False` otherwise). The next step was to compute the final values for each patient and then use them to write the final CT Report.

The CNN algorithm had a different approach to the issue. Each of the columns was inspected individually, one script for each one, as there was a need to control the data better and to adapt the parameters for the best possible results. In each one, the approach is similar. The training dataset provided by the organizers was split into training and validation subsets for computing the final trained model. The images have been resized, different values for different outputs and reshaped accordingly. For this to happen, many libraries needed to be used for the backend: `OpenCV` (`cv2`), `Numpy`, `Pillow`¹⁶, `Keras` and `Tensorflow`¹⁷. Our approach was based on individual picture labeling and achieving a result for the whole set of pictures for a test patient, by computing, once again, a mean of the resulted values that, this time were not 0 or 1, but rather values belonging to this interval. Before the actual training, the images had to be assigned a label. This was done making use of the CT Report given by the organizers, and thus each picture from a patient received the corresponding label. Then, the images were resized and transformed into `NumPy` arrays, by going through all the pictures from all of the regions from the same patient. After saving the training data so it can be used further for testing, the data was shuffled. The test data had a similar process, only there was no shuffling involved at the end.

For training the model, there were 500 pictures set aside for the validation set. Then, the arrays were reshaped to accommodate our image size. Moving further, `Conv2D`¹⁸ was used for adding layers to the model. The number of said layers was not constant throughout all of the submissions, but they had the activation set to ‘`ReLU`’ and the kernel size to 3. After, a flatten layer was added and after this one, a new fully connected layer was the one to end this cycle, the activation used for it being ‘`softmax`’. Following this, the model was compiled using the ‘`Adam optimizer [17]`’, the loss set to ‘`categorical_crossentropy`’ and the metrics equal to ‘`accuracy`’.

The next and final step from this set was the fitting of the model and, for this, there were several parameters used, as they vary from one submission to another, but the batch size is consistently equal to 200. The last step in this solution was writing the

¹⁴ <https://numpy.org/doc/stable/reference/generated/numpy.array.html>

¹⁵ <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

¹⁶ <https://pillow.readthedocs.io/en/stable/>

¹⁷ <https://www.tensorflow.org/>

¹⁸ https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D

findings in individual comma separated values files and then combining all 6 resulted files to obtain the final result.

4 Evaluation

In this section we will discuss the evaluation of the model and the differences between each submission.

In official results we ranked 7 out of 9, with submission ID 68081, and total of 67 teams. Table 3 presents each submission results, we made a total of 8 submissions.

Table 3. Official evaluation results

ID	mean_aucc	min_aucc
67548	0.617	0.537
67549	0.591	0.503
67573	0.609	0.598
67977	0.505	0.402
68081	0.659	0.562
68121	0.626	0.513
68134	0.608	0.513
68135	0.491	0.382

Submissions information:

- ID 67548 – This submission was computed using SVM algorithm, where the gamma parameter was equal to 0.005 and the C parameter was equal to 1000.0.
- ID 67549 – This submission was computed using SVM algorithm, where there was no gamma parameter preset and the C parameter is equal to 1.0.
- ID 67573 – This submission was computed using CNN algorithm, where the image size was set to (64, 64).
- ID 67977 – This submission was computed using CNN algorithm, where the image size was set to (128, 128).
- ID 68081 – This submission is based on the CNN approach, we tried to improve that algorithm. We used Pre-Activation ResNet with Identity Mapping [18] with standard parameters; unfortunately, we got marginal result.
- ID 68121 – This submission is based on submission with ID 67548 in an effort to improve that result. We tried using a different algorithm than SVC, we used LinearSVC and the parameters were the following: C=15, maximum iterations were 1,000 and dual was False.
- ID 68134 – This submission is based on submission with ID 68121. We noticed that the pleurisy for the left lung had very low accuracy so we tried us-

ing a special SVM for that one so we used SVC with ‘rbf’ kernel, C 1000, gamma 0.005 and cache size 2048; we left the other settings the same as in ID 68121. We did not get the expected result.

- ID 68135 – This submission was computed using CNN algorithm, where there were 3 layers in the network (one more than the others before it) and the image size was set to (64, 64). In this submission the code was supposed to be adapted to work best on each column of the dataset, so there would be some of the columns with just 2 layers, or some with only 2 epochs.

We implemented an accuracy calculation mechanism using `sklearn.metrics19 roc_curve20`, we calculated the accuracy for each side of the lung then we averaged it, this allowed us to predict the accuracy of our algorithms and analyze the errors without submitting a run.

4.1 Error analysis

The SVM algorithm was the first one to prove that the difference between affected and unaffected left lobes in patients was a real issue we would have to deal with. This has made it easier to understand where the issue is and to try and find solutions. The hyperparameter tuning in the SVM algorithm was done using grid search [19] on 5 values for gamma and C, as follows: ‘gamma’: [1e-3, 1e-4, 10, 100, 1e-5], ‘C’: [1, 10, 100, 1,000, 10,000, 100,000]. Unfortunately, these 30 tests were cut short by the lack of proper equipment and only provided the first submission parameters. Considering in [20] SVM had an average accuracy of 0.78 by using a combination of “Gaussian kernel function with a width parameter of 2.0, while the value of C was set to 10” it is safe to say there is still room for improvement in future work.

When training the CNN algorithm, oftentimes there was an overfitting problem for some of the columns due to unbalanced data, also a problem was the fact that long training times resulted in a lack of hyperparameter tuning possibility. Data imbalance could not be solved by removing unnecessary data as the result suffered when this was done. Error reduction can be achieved by creating a more complex architecture as described in [5] which obtained an accuracy of 0.7968 proving the superiority of this algorithm over SVM and the potential of neural networks.

5 Conclusion

In this paper we present different ways of detecting tuberculosis and thanks to competitions such as ImageCLEFmed Tuberculosis every year we are a step closer to perfecting these techniques.

Our methods are mainly focused around SVM and Neural Networks. We got a good result that can be improved; thus, we plan on further building on the ResNet

¹⁹ <https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics>

²⁰ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

architecture, improving the current SVM approach and improving our current hardware in order to process the data on GPU for a more streamlines hyperparameter tuning.

Acknowledgements

Special thanks go to our colleagues from second year group E1 without whom this work would probably have not been possible.

References

1. Bogdan Ionescu, Henning Müller, Renaud Péteri, Asma Ben Abacha, Vivek Datla, Sadid A. Hasan, Dina Demner-Fushman, Serge Kozlovski, Vitali Liauchuk, Yashin Dicente Cid, Vassili Kovalev, Obioma Pelka, Christoph M. Friedrich, Alba García Seco de Herrera, Van-Tu Ninh, Tu-Khiem Le, Liting Zhou, Luca Piras, Michael Riegler, Pål I. Halvorsen, Minh-Triet Tran, Mathias Lux, Cathal Gurrin, Duc-Tien Dang-Nguyen, Jon Chamberlain, Adrian Clark, Antonio Campello, Dimitri Fichou, Raul Berari, Paul Brie, Mihai Dogariu, Liviu Daniel Ștefan, Mihai Gabriel Constantin, Overview of the ImageCLEF 2020: Multimedia Retrieval in Medical, Lifelogging, Nature, and Internet Applications In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), Thessaloniki, Greece, LNCS Lecture Notes in Computer Science, Springer (September 22-25 2020).
2. Serge Kozlovski, Vitali Liauchuk, Yashin Dicente Cid, Aleh Tarasau, Vassili Kovalev, Henning Müller, Overview of ImageCLEFtuberculosis 2020 - Automatic CT-based Report Generation and Tuberculosis Severity Assessment, CLEF working notes, CEUR, 2020.
3. Liauchuk, V., Tarasau, A., Snezhko, E., Kovalev, V. (2018) Imageclef 2018: Lesion-based tb-descriptor for ct image analysis. In *CLEF 2018 Working Notes*.
4. Dicente Cid, Y., Müller, H. (2018) Texture-based graph model of the lungs for drug resistance detection, tuberculosis type classification, and severity scoring: Participation in the imageclef 2018 tuberculosis task. In *CLEF 2018 Working Notes*.
5. Liauchuk, V. (2019) ImageCLEF 2019: Projection-based CT Image Analysis for TB Severity Scoring and CT Report Generation. In *CLEF2019 Working Notes*. 2380, Lugano, Switzerland, CEUR-WS.org <http://ceur-ws.org/Vol-2380> (September 9-12 2019)
6. Mossa, A.A., Yibre, A.M., C, Evik, U. (2019) Multi-View CNN with MLP for Diagnosing Tuberculosis Patients Using CT Scans and Clinically Relevant Metadata. In *CLEF2019 Working Notes*. 2380 of CEUR Workshop Proceedings., Lugano, Switzerland, CEUR-WS.org
7. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017) ImageNet classification with deep convolutional neural networks. In *Communications of the ACM*. 60 (6): 84-90. doi:10.1145/3065386. ISSN 0001-0782. <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
8. Tabarcea, A., Rosca, V., Iftene, A. (2019) ImageCLEFmed Tuberculosis 2019: Predicting CT Scans Severity Scores using Stage-Wise Boosting in Low-Resource Environments. In: *CLEF2019 Working Notes*. 2380, Lugano, Switzerland, CEUR-WS.org
9. Coca, G. L., Romanescu, S. C., Botez, S. M., Iftene, A. (2020) Crack detection system in AWS Cloud using Convolutional neural networks. In *24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2020)*.

10. Cuşmuliuc, C. G., Coca, L. G., Iftene, A. (2018) Identifying Fake News on Twitter using Naive Bayes, SVM and Random Forest Distributed Algorithms. In *Proceedings of The 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018)*. ISSN: 1843-911X, 177-188
11. Cuşmuliuc, C. G., Coca, L. G., Iftene, A. (2019) Early Detection of Signs of Anorexia in Social Media. In *5th Proceedings of the Conference on Mathematical Foundations of Informatics*. 3-6 July 2019, Iasi, Romania, 245-260.
12. Coca, L. G., Cuşmuliuc, C. G., Iftene, A. (2019) CheckThat! 2019 UAICS. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, Lugano, Switzerland, September 9-12
13. Yashin Dicente Cid, Oscar A. Jiménez-del-Toro, Adrien Depeursinge, and Henning Müller. Efficient and fully automatic segmentation of the lungs in CT volumes. In: Goksel, O., et al. (eds.) Proceedings of the VISCERAL Challenge at ISBI. No. 1390 in CEUR Workshop Proceedings (Apr 2015)
14. Liauchuk, V., Kovalev, V. ImageCLEF 2017: Supervoxels and co-occurrence for tuberculosis CT image classification. In: CLEF2017 Working Notes. CEUR Workshop Proceedings, Dublin, Ireland, CEUR-WS.org <http://ceur-ws.org> (September 11-14 2017) (URL: http://ceur-ws.org/Vol-1866/paper_146.pdf)
15. Cortes, C., Vapnik, V. N. (1995) Support-vector networks. In *Machine Learning*. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018
16. Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., Chervyakov, N. I. (2020) Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. In *Mathematics and Computers in Simulation*. Elsevier BV. 177: 232–243. doi:10.1016/j.matcom.2020.04.031. ISSN 0378-4754. Convolutional neural networks are a promising tool for solving the problem of pattern recognition.
17. Kingma, D. P. & Ba, J. (2014), 'Adam: A Method for Stochastic Optimization' , cite arxiv:1412.6980 Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 .
18. He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Identity Mappings in Deep Residual Networks. 9908. 630-645. 10.1007/978-3-319-46493-0_38.
19. Bergstra, J., Bengio, Y. (2012) Random search for hyper-parameter optimization. In *Journal of Machine Learning Research*, 13: 281-305 <http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>
20. Niházy L., Horváth G., Horváth Á., Müller V. (2010) Computer-Aided Detection of COPD Using Digital Chest Radiographs. In: Bamidis P.D., Pallikarakis N. (eds) XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010. IFMBE Proceedings, vol 29. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13039-7_63