3D-CNN in Drug Resistance Detection and Tuberculosis Classification

João Figueira Silva, Jorge Miguel Silva, Eduardo Pinho, and Carlos Costa

DETI - Institute of Electronics and Informatics Engineering of Aveiro
University of Aveiro, Portugal
{joaofsilva,jorge.miguel.ferreira.silva,eduardopinho,carlos.costa}@ua.pt

Abstract. Object classification is a very demanding field in computer vision, especially when dealing with medical imaging datasets, which are often small and have unbalanced distributions. Deep learning (DL) methods have proven to be effective in dealing with such problems and have established themselves as the state-of-the-art. ImageCLEFtuberculosis is a challenge that encompasses the classification problem on medical images, and is divided into two subtasks: Drug Resistance Detection and Tuberculosis classification. For both subtasks, provided images were pre-processed to segment the lungs from the CT volumes. Afterwards, pre-processed CT volumes were fed in batches to a 3D convolutional neural network. Test results for the Drug Resistance detection task scored an accuracy of 46.5% and AUC of 0.46, while in the Tuberculosis classification task an accuracy of 24% and Cohen's Kappa value of 0.022 were obtained. Using data augmentation and weight normalization, the overfitting problem could be reduced, and submitted models' performance improved.

Keywords: 3D-CNN, Neural Networks, Deep Learning, Medical Imaging, CT, Tuberculosis, ImageCLEF

1 Introduction

The ImageCLEFtuberculosis task [1] from ImageCLEF 2017 [2] is a challenge centered on medical imaging, that has the motivation of improving tuberculosis treatment and reducing its impact on patients through the development of systems capable of extracting the tuberculosis type and drug resistances from image data alone. Usually, working with medical imaging datasets encompasses distinct challenges such as the limited access to data, its reduced size and the unbalanced distributions. Deep learning (DL) methods have been increasingly explored in the field of image analysis, with neural networks leading to major breakthroughs in renown challenges, such as the MNIST Digit Image Classification Problem and the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [3], where they are considered the state-of-the-art [4]. These networks present great interest since they automatically learn high-level representations from the data, and can be used to reduce the data dimensionality [5].

In recent years, deep learning has started to make a significant appearance in the field of medical imaging with promising results [6]. Following this trend, this article assesses the viability of this technology to solve ImageCLEF challenges through the development of a 3D Convolutional Neural Network (CNN) model.

The ImageCLEFtuberculosis task is divided into two separate and independent subtasks: drug resistance detection and tuberculosis classification. The goal of the first task was to assess the probability of a tuberculosis (TB) patient having a resistant form of tuberculosis based on the analysis of a chest CT scan, whereas the second one focused on classifying the TB type from five possible types of TB.

This article describes the proposed solution and runs submitted by the Bioinformatics team for both subtasks. The developed methodology is presented in Section 2, results are presented and discussed in Section 3, and finally Section 4 draws some conclusions and future work.

2 Methodology

To address the MDR detection and TB type subtasks from ImageCLEFtuberculosis, we propose a two-stage pipeline: Data pre-processing and a DL model (Figure 1). The pre-processing stage was applied to both subtasks whereas the DL model was fine-tuned for each subtask.

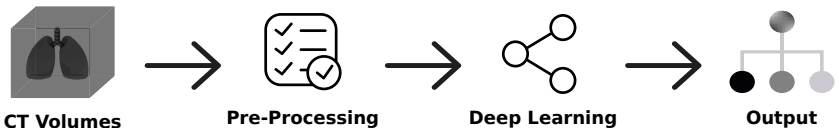

Fig. 1. Pipeline Overview.

Pre-processing stage used the Computed Tomography (CT) images, segmented the lungs, and resized data to be ready to feed the DL model. On the other hand, the DL model used batches of pre-processed data and classified it. Each of the stages is explained in more detail in the next subsections.

An important aspect of proposed approach is related with the fact that a CT volume is composed by several images and the observation that a single slice might provide poor classification results. So, we decided to feed the DL models with volumes composed of stacks of CT slices, option that conducted us to use of a 3D-CNN model instead of a conventional CNN model. This option brought also implications concerning the shape of the models input tensors, which were solved in the data pre-processing step.

Data Pre-processing

Pre-processing stage has the responsibility of preparing data for posterior processes, namely feeding the DL model. For the drug resistance detection subtask,

a train dataset with 230 CT volumes and a test dataset with 214 CT volumes were provided. In this subtask data had two possible classes. Regarding the tuberculosis classification subtask, a train dataset with 500 CT volumes and a test dataset with 300 CT volumes were provided, with data having five possible classes. CT volumes had a variable number of slices, with slice size being 512x512 pixels [1].

In the training datasets, CT volumes had the lungs segmented using masks created with a developed algorithm. To create the masks, the following method was used: a thresholded was applied to the images where intensities below -300 Hounsfield units were set as background, the pixel values were normalized to have an intensity range from 0 to 255, and resulting images were passed through a binary thresholding process with a threshold value of 20. Using scikit-image[1], small holes and small objects were removed, using methods with the same name and parameterized with minimum size of 100 and connectivity of 4. Next, the two methods were reapplied but with a minimum size of 1000. The result is the desired masks.

Obtained masks were highly similar compared to those provided to the participants [7]. Dice's coefficient, which is scaled from 0 to 1 with 1 corresponding to image equality, was computed to assess the similarity between created masks and the original provided masks, with a global average value of 0.9755 being obtained. Regarding the test dataset, provided masks were used to ensure that test data to feed the 3D-CNN was not tampered.

After this, the resulting masked volumes were reshaped to comply with the NHWC channel ordering (number of samples x height x width x channels) used in CNNs. In our case, the number of samples corresponds to the number of CT slices. Next, each CT slice was resized to dimensions of 256x256 pixels.

The resulting volumes were resized, regarding the number of slices, so that all volumes had the same number of slices. This was achieved by padding the top and bottom of each volume, resulting in a final volume with fixed size (real data in the center, and padding in the extremities). Finally, data was normalized to have zero mean.

For each subtask, processed datasets were saved in HDF5 files resulting in two HDF5 files per task with the train and test sets.

Deep Learning

As expressed, we opted by a 3D-CNN model for the DL model stage. The model was implemented with TensorFlow [8] version 1.0.0 with support for GPU, which massively increases the speed and efficiency of training and developing models such as neural networks. Moreover, TensorBoard was used during the development of the 3D-CNN model for debugging and optimization purposes. Some additional functions needed for the models' development were imported from TFLearn[2] (v0.3), a DL library that provides a higher-level API to TensorFlow.

[1] http://scikit-image.org
[2] TFLearn: http://tflearn.org

The 3D-CNN model training ran on an Ubuntu server machine equipped with an NVIDIA Tesla K80 GPU accelerator.

Regarding the DL model itself, Figure 2 presents an overview of the built model with the respective composition of each layer. The decision to use a 3D-CNN model with seven convolutional layers and two fully connected layers was empirical.

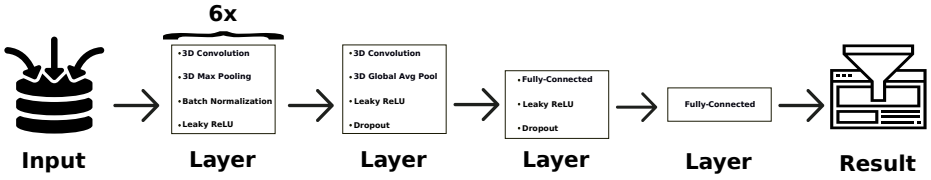

Fig. 2. Diagram of the neural network model used.

Literature supports that deeper models can be more powerful than shallow ones, as the former can learn how to represent high-level abstractions, presenting particular interest for the fields of vision, language and other AI-level tasks [9]. However, it is also known that deeper models are more difficult to train due to problems such as the vanishing gradient problem, where initial layers learn at slower speeds than final layers. Naturally, the deeper the network, the more prone it is to the vanishing gradient problem [10]. Moreover, deeper models demand bigger compute power, which is a very significant overhead. Thus, bearing in mind the associated implications of creating a deep neural network, and the existing limited compute power, it was decided to build a network with a small number of layers.

As it is possible to observe in Figure 2, the network's first six layers share the same structure (but not the hyperparameters). In these six layers, the incoming tensor is passed through a sequence of 3D convolution, 3D max pooling, batch normalization and non-linear activation function.

Batch normalization is very important as it addresses a phenomenon called internal covariate shift, which slows down the training of neural networks [11]. Concerning the activation function, since the sigmoid activation function can cause problems when training deep neural networks [12], a variation of the rectified linear unit (ReLU) – the leaky ReLU – which can lead to better performances was used in this neural network [13].

Overfitting is other serious concern in neural networks, specially when dealing with medical imaging datasets which frequently consist of reduced amounts of data, with unbalanced distributions. For that reason, dropout [14], a regularizer used to reduce overfitting in neural networks, was used in our model. However, it was only applied to the fully-connected part of the network as convolutional layers have considerable inbuilt resistance to overfitting [15]. Also, L2 regularization was used in each convolutional layer to reduce model overfitting, and the last Fully-Connected layer has a softmax activation function.

The described 3D-CNN was used for both subtasks, though with different hyperparameters due to the fine-tuning procedure performed for each subtask. All Leaky ReLUs were used with the leaking coefficient $\alpha = 0.1$ and Dropout with a drop probability of 0.5 for both subtasks. Table 1 summarizes the remaining hyperparameters for the models' layers. It should be noted that the hyperparameters were defined with compute power constraints in mind. All weights were initialized as described in [16].

Table 1. List of layer hyperparameters for the MDR detection and TB type models. The following parameters are displayed: number of units/filters, kernel size, stride and L2 weight decay.

Layer	**MDR Detection**				**TB Type**			
	Units	**Kernel**	**Stride**	**L2**	**Units**	**Kernel**	**Stride**	**L2**
Conv1	35	11	2	0.01	20	7	3	0.001
Max1	_	5	5	_	_	5	2	_
Conv2	60	7	7	0.001	15	11	3	0.001
Max2	_	5	5	_	_	5	2	_
Conv3	60	5	3	0.002	15	9	3	0.001
Max3	_	3	2	_	_	5	2	_
Conv4	60	5	3	0.002	15	7	7	0.001
Max4	_	3	2	_	_	5	2	_
Conv5	92	5	3	0.002	32	5	3	0.001
Max5	_	2	2	_	_	2	2	_
Conv6	92	3	2	0.003	64	3	2	0.001
Max6	_	2	2	_	_	2	2	_
Conv7	128	1	1	_	128	1	1	_
FC1	128	_	_	_	128	_	_	_
FC2	Num Classes	_	_	_	Num Classes	_	_	_

Concerning data handling, the training dataset was split into 80/20 parcels, for training and validation splits respectively. Data distribution had moderately balanced classes for the MDR detection subtask, whereas for the TB type subtask a less balanced dataset was provided. For each subtask, data was split taking into account class distributions, in order to ensure the same class distribution in training and validation splits. Even though the network was prepared to work with K-fold cross validation, due to time constraints and the inherent nature of the training process of a neural network, the network was validated offline using a single combination of the 80/20 split.

Furthermore, the model was fed with mini-batches of data containing complete CT scans, where each sample is one of the CT volumes being forwarded through the net. Since this type of network is demanding in terms of memory and computational cost, and aiming to enable the use of bigger batch sizes, each pre-processed volume was cropped into a fixed smaller number of slices, corresponding to the size of the smallest volume in the original dataset. As expressed, this cropping method extracts data from the center of each CT volume.

In order to prevent the network from learning a given data sequence/order, data splits were shuffled in each epoch, prior to being fed to the model. Finally, a group of four metrics was used to assess model performance, consisting of: cross entropy, accuracy, precision, and recall.

3 Submitted Runs and Results

A single run was submitted for each subtask of the ImageCLEFtuberculosis task, with the results and respective neural network configurations being discussed in this section.

In both subtasks the neural network models were trained using an Adam optimizer [17] for stochastic optimization, with the following settings being used: $\alpha = LearningRate$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Table 2 summarizes some of the hyperparameters used in order to train the neural network for each subtask. All hyperparameters were defined empirically.

Table 2. Hyperparameters used in the neural networks for the submitted runs.

Hyperparameter	**MDR detection (Subtask 1)**	**TB type (Subtask 2)**
Batch Size	18	30
Learning Rate (LR)	7 x 10^{-5}	9 x 10^{-6}
LR Decay	5%	5%
Decay Interval	10 *epochs*	15 *epochs*

As previously mentioned, the graphics card used to accelerate the training of the neural network was an NVIDIA Tesla K80, which possesses two separate GPUs. Due to the use of the graphics card for other tasks, the MDR detection model was trained using a single GPU whereas the TB type model was trained using both GPUs. Therefore, and as shown in Table 2, it was possible to significantly increase the batch size for the TB type network.

Learning rate was reduced by a fraction of 5 percent of its value after 10 and 15 epochs for subtask 1 and subtask 2, respectively. Validation was performed in intervals of 3 epochs for the MDR detection's model, and in intervals of 2 epochs for the TB type's model.

Table 3. Performance metrics used in the validation of submitted models.

Metric	MDR detection (Subtask 1)	TB type (Subtask 2)
Accuracy	0.5501	0.1744
Precision	0.5470	0.5223
Recall	0.3440	0.4413

The best results obtained for each subtask during the validation phase are shown in Table 3. In MDR detection, which is a two class problem, the trained model favors the retrieval of the most frequent class but struggles to detect the less frequent and more relevant class, leading to a substantially lower recall comparatively to obtained accuracy and precision.

In the TB type task, a multi-class problem (five classes), it is possible to see that the tuned model attained a lower accuracy, while keeping slightly similar precision and improved recall. The impact of having a higher number of classes, combined with a less balanced dataset for this task had a repercussion on the validation accuracy which was significantly lower than in the MDR detection task. Such accuracy value demonstrates that the neural network had difficulties in identifying the classes in data, which explains why some classes had no occurrence registered in the validation dataset.

Regarding the testing phase, in the MDR detection subtask contestants had to submit the probability of each patient having MDR, whereas for the TB type task submissions had to contain the expected TB type for each TB patient. Model performance was assessed with different metrics for each subtask: in MDR detection, performance was measured with Accuracy and Area Under the Curve (AUC) obtained from the ROC-curves produced with the submitted probabilities; in TB type classification Accuracy and Cohen's Kappa were the selected metrics. Table 4 presents test results both for the submitted run and for the best run in each subtask.

The list of test results for the two subtasks comprised in ImageCLEF's tuberculosis task [1] clearly demonstrates the high difficulty associated with this challenge's proposition. On the one hand, submitted runs performed worse in each subtask than the remaining entries. On the other hand, for the MDR de-

Table 4. Test results obtained for the submitted models, and best submissions for each subtask.

Metrics	MDR Detection		TB Type	
	Our Run	Best Run	Our Run	Best Run
Test Accuracy	0.4648	0.5164	0.2400	0.4033
AUC	0.4596	0.5825	_	_
Cohen's Kappa	_	_	0.0222	0.2438

tection subtask, the top ranking model had an accuracy just slightly over 50% whilst our model's test accuracy was nearly 47% (the best overall accuracy was 56.8%). Concerning AUC, our model scored lower than the top ranking entry by a bigger margin.

For the TB type subtask, test results were in general worse comparatively to results of the MDR detection subtask. In this subtask, the top ranking entry had an accuracy of 40% compared to our model's 24.3%, and a Cohen's Kappa of 0.24 compared to the marginal value of 0.02 obtained by our model.

Aside from the comparison with other models' performance, it is noticeable that for subtask 1 our model had lower accuracy in the test phase (46.5%) than in the validation phase (55%), whereas for subtask 2 the opposite occurred with test accuracy (24%) being higher than validation accuracy (17.4%). For the first part, it is very likely that the model suffered overfitting to the training dataset (a common issue when dealing with medical imaging datasets), and testing performance suffered a significant impact from that. For the second part, there exists the possibility of having a test dataset less balanced, regarding class distribution, than the training dataset. By having a class distribution more skewed towards the more frequent classes, our model can attain higher accuracy scores than during the validation process.

In spite of our models' poor performance in general, the final ImageCLEFtuberculosis result list [1] shows that there are other entries with comparable performance. Fine-tuning a model is a slow, thorough process that should be methodical. In our approach, the search for the best hyperparameters was empirical and not extensive enough due to limitations in terms of available time. There is much confidence that there exists a big margin for progress and improvement in our work, provided there is more time to better train the models, and correctly fine-tune them.

4 Conclusion and Future Work

The ImageCLEFtuberculosis task is a challenge that encompasses the classification problem on medical images. This task was divided into two subtasks: Drug Resistance Detection and TB classification. In the first subtask the objective was to assess the probability of a TB patient having a resistant form of tuberculosis, whereas on the second one the goal was to classify the TB type from a pool of five possible types.

In this paper we presented two separate runs that were submitted for the two subtasks. In both subtasks, provided images were pre-processed for this challenge. Although the test results of our submitted runs for both subtasks were low (46.5% accuracy, AUC of 0.46 and 24% accuracy, Cohen's Kappa of 0.022, respectively), the majority of the submitted runs behaved in a similar way, since the differences in terms of accuracy between the best submitted run and our own were of 5% and 15% for the MDR detection and TB type subtask, respectively. As a side note, it is interesting to notice in the list of submissions

that various entries used DL approaches to tackle this challenge, which shows that DL is an area that holds great promise.

Since overfitting was an effective reality during the development of the neural network models, in the future we hope to evaluate the impact of techniques such as data augmentation and weight normalization on our models' results. Furthermore, running the model with K-fold cross-validation and performing an ensemble of the resulting networks could further improve our results.

Acknowledgments

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the FCT Fundação para a Ciência e a Tecnologia. João Figueira Silva is funded by the research grant of PTDC/EEI-ESS/6815/2014 project and Jorge Miguel Silva is funded by the research grant of CMUP-ERI/ICT/0028/2014-SCREEN-DR project. Eduardo Pinho also was funded by the FCT under the grant PD/BD/105806/2014.

References

1. Dicente Cid, Y., Kalinovsky, A., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEFtuberculosis 2017 - Predicting Tuberculosis Type and Drug Resistances. CLEF working notes, CEUR (2017)
2. Ionescu, B., Müller, H., Villegas, M., Arenas, H., Boato, G., Dang-Nguyen, D.T., Dicente Cid, Y., Eickhoff, C., Garcia Seco de Herrera, A., Gurrin, C., Islam, B., Kovalev, V., Liauchuk, V., Mothe, J., Piras, L., Riegler, M., Schwall, I.: Overview of ImageCLEF 2017: Information extraction from images. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF 2017. Volume 10456 of Lecture Notes in Computer Science., Dublin, Ireland, Springer (September 11-14 2017)
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3) (2015) 211–252
4. Bengio, Y., Courville, A.C., Vincent, P.: Representation Learning: A Review and New Perspectives. CoRR (2012)
5. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. Science **313**(5786) (2006) 504–507
6. Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.Z.: Deep Learning for Health Informatics. Biomedical and Health Informatics, IEEE Journal of **21**(1) (jan 2017) 4–21
7. Dicente Cid, Y., del Toro, O.A., Depeursinge, A., Müller, H.: Efficient and fully automatic segmentation of the lungs in CT volumes. In: Proceedings of the VISCERAL Anatomy Grand Challenge at the 2015 IEEE ISBI. CEUR Workshop Proceedings, CEUR-WS (2015) 31–35

8. 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., Mane, 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., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous distributed systems. (2016)
9. Bengio, Y., Yoshua: Learning Deep Architectures for AI. Foundations and Trends® in Machine Learning **2**(1) (2009) 1–127
10. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. In: Field Guide to Dynamical Recurrent Networks. IEEE Press (2001)
11. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. CoRR **abs/1502.03167** (2015)
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics. (2010)
13. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: ICML Workshop on Deep Learning for Audio, Speech and Language Processing. Volume 30. (2013)
14. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research **15** (2014) 1929–1958
15. Nielsen, M.A.: Neural Networks and Deep Learning (2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1026–1034
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)