2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) | 978-1-6654-0191-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCVW54120.2021.00065

Evaluating volumetric and slice-based approaches for COVID-19 detection in chest CTs

Radu Miron, Cosmin Moisii, Sergiu Dinu and Mihaela Elena Breaban*

Faculty of Computer Science, Alexandru Ioan Cuza University of Iaşi, Romania
SenticLab, Iaşi, Romania
Email: *pmihaela@info.uaic.ro,

Abstract

The paper presents a comparative analysis of several distinct approaches based on deep learning for identifying COVID-19 cases in chest CTs. A first approach is a volumetric one, involving 3D convolutions, while other two approaches perform at first slice-wise classification and then aggregate the results at the volume level. The experiments are carried on the COV19-CT-DB dataset, with the aim of addressing the challenge raised by the MIA-COV19D Competition within ICCV 2021. Our best results reach a macro F1 score of 92.34% *on the validation subset and 90.06% on the test set, obtained with the volumetric approach which was ranked second in the competition. Its performance can be further improved by a simple trick, using semi-supervised training in the form of self-training, technique which proved to bring a consistent increase over the reported F1-score on the validation subset.*

I. Introduction

There is a high effervescence in the Artificial Intelligence (AI) community trying to provide tools to assist medical diagnosis, with a special focus on medical imaging, exploiting but also triggering important advancements in the area of computer vision and deep learning. The pandemic scenario we face today clearly calls for such approaches in order to be able to address the high incidence rate which overwhelms the medical system. Medical imaging, mostly in the form of x-Rays and CTs, is used to assess the lung involvement. In this context, COVID-19 datasets are publicly released and competitions are organized, aiming at involving and stimulating the AI community to produce models that can accurately detect Covid-19 affections in the lungs.

The current work addresses the challenge raised by the MIA-COV19D competition[1] organized within the ICCV 2021 conference. The challenge consists in tackling a two-class classification problem on the COV19-CT-DB dataset consisting of CTs classified in two groups: COVID patients and non-COVID patients, with the second class containing both healthy patients or patients presenting lung lesions due to other causes. The dataset was split by the organizers into 3 subsets: training, validation and test, with the first two subsets also exposing the class label. The baseline set for this classification task by the organizers, for comparison purposes, is a 0.70 macro-F1 score on the validation set, obtained with a CNN-RNN network described in [12], which is the result of authors' previous work reported in [14], [13], [15]. The final evaluation of the models built by the participants in the competition is made on the test set which was released with no class/label information.

Given the nature of the data, we address the challenge in two different ways: 1) treating the CT as a volume and thus using 3D convolutions, and 2) treating the CT as a set of 2d images (slices), the second approach calling for 2D convolutions and predictions at slice level, followed by an aggregation step where the outputs at slice level are aggregated into a response at CT level.

For the first (volumetric) approach we use pretrained networks and several tricks as data augmentation, regularization, simultaneous minimization of the loss value and loss sharpness by involving Sharpness-Aware Minimization [6], training an ensemble of NNs and finally exploiting the unlabeled data by self-training.

Our second approach is motivated by the results reported in the 2020 and 2021 ImageClefMed competitions on tuberculosis tasks, where slice-based methods were

[1] https://mlearn.lincoln.ac.uk/mia-cov19d/

DOI 10.1109/ICCVW54120.2021.00065

ranked first [19], [20], outperforming volumetric methods. This approach needs in the training phase more refined information: it needs CT slices split in two categories: slices that present COVID lesions and slices that do not present COVID lesions, the last category coming from patients having no affection or presenting various other affection types. Because the dataset exposes class information only at CT level and a COVID CT presents both healthy and affected lung slices, we needed to come up with a procedure to extract COVID-affected slices from the COVID CTs in the training set. We use at first a naive method to label slices, where all the slices belonging to a COVID CT are considered as slices presenting COVID lesions. As a second, more refined method, we use a model we had trained for the tuberculosis classification task on data published in the ImageClefMed 2021 competition; this method is able to detect slices presenting different types of tuberculosis lesions and differentiate them from clean slices (slices presenting no lesion). This model is basically used to identify healthy slices from the COVID CTs.

After slice labeling, we use two distinct approaches for COVID classification: one that makes use of a manually designed NN architecture working with "mini-volumes" made of 3 slices and one that performs NN architecture search with sharpDARTS[9].

The paper is structured as follows. Section II describes the COV19-CT-DB dataset. SectionIII describes our first approach that makes inference directly at the CT level. Sections IV-B and IV-A describe the two approaches that learn to classify the slices as COVID/non-COVID and then aggregate the results at CT level. Section V presents the results obtained on the training and validation sets and section VI concludes the paper.

II. The dataset

Excepting pre-trained models, the only dataset used in the experimental analysis is COV19-CT-DB [12], provided in the MIA-COV19D competition.

Our slice-based approaches described in sections IV-B and IV-A do not involve directly an external dataset, but only a pretrained model built by us in the ImageClef2021 competition on CT data consisting of non-COVID patients that present tuberculosis lesions; this pretrained model serves only to select a subset of slices from the COVID class in the COV19-CT-DB training dataset - slices likely to presents COVID lesions.

The provided COV19-CT-DB training set has a total of 1560 CT scans. The class distribution is 690 COVID-19 cases versus 870 Non-COVID19 cases. Out of these, we found 8 volumes with less than 20 slices and 5 volumes with more than 700 slices in their corresponding directories.

The provided validation set has a total of 374 CT scans. The class distribution is 165 COVID-19 cases versus 209 Non-COVID-19 cases. Out of these, we found no volume with less than 20 slices and 1 volume with more than 700 slices in their corresponding directories.

As explained in [12], the labeling of the dataset was made by a consensus of two radiologists and two pneumologists. The difference with other publicly available datasets is the annotations based on expert opinion rather than just positive RT-PCR testing.

The whole COV19-CT-DB database consists of about 5000 CT scans, corresponding to more than 1000 patients and 2000 subjects, leading to 3455 cases for the test set.

The images are provided in the jpg format and were obtained by clipping the voxel intensities using a window/level of 350 Hounsfield units (HU) and 1150 HU and normalization to the range of [0,1] of each slice from a whole volume rather than the dicom files.

Due to the large number of slices per volume and large dimensions of an individual slice, in our approaches we usually rescale the given images. Each classification approach described in the sections below will also list the modifications done to the input data.

III. A volumetric approach

Convolutional neural networks have emerged as a successful tool in tackling a wide range of image vision tasks. These architectures have obtained, ever since the appearance of Alex-Net [16], state of the art results on tasks like image classification, object detection and segmentation.

Even though applications in 2D image tasks have shown great results, architectures for 3D image tasks still have to prove themselves as good contenders for state of the art results.

Using transfer learning in the form of exploiting pretrained models is a popular technique for achieving good performance on the dataset at hand. Since large 3D datasets are not so numerous, we turn our attention to a specific kind of model to fully benefit of the advantages of pretrained models.

A. The NN architecture

We use an inflated convolutional neuronal network pretrained on Kinetics dataset [5]. Inflated convolutions are obtained by expanding filters and pooling kernels of 2D ConvNets into 3D, resulting in the possibility to learn spatio-temporal feature extractors from 3D images while using successful ImageNet architectures. Due to gpu limitations we only used an Inflated ResNet50 model [8].

For capturing long range dependencies within the slices of a same volume, we use non-local features which have been proven to increase the results of basic architectures. [28]

The final model is an Inflated 3D ResNet50 with non local operations on the second and third layers (based on the official implementation from PyTorch) [2]. The pretrained model we started the training from is the one from the official caffe2 implementation [3] .

Due to the large number of parameters, 3D conv nets are easy to overfit. In order to mitigate the overfitting, we use label smoothing [21] with cross entropy and Sharpness Aware Minimization [6] on top of Stochastic Gradient Descent as loss function and optimizer, respectively.

B.Data augmentation and training settings

We use augmentations like Random Horizontal and Vertical Flip, Gaussian Blur and Contrast and Color changes. We also experimented with Cut-Out and Affine transformations like ShearX and ShearY, but the results were not improved compared to the basic augmentations mentioned above. Thus, in an effort to reduce training time, we stayed with the basic augmentations. We also use volume flipping on the depth axis. All these augmentations are done with 0.5 chance.

We train the above architecture several times, on different samples of the dataset, for 150 epochs or 100 epochs, in order to come up with an ensemble of models for final predictions on the competition test set. Thus, one model is trained on the official train-validation split for 150 epochs whereas other 4 models are trained on 4 in-house generated folds for 100 epochs. Initial learning rate is set to 1e-3 and the learning rate scheduler is cosine scheduler with 5 epochs of warm-up for each model. Batch size is 2.

The input consists of volumes of size $128 \times 224 \times 224$. First, we resize all the images to 224×224 and take only 128 slices from each volume (with zero-padding if necessary to reach the depth 128). Since there are large volumes in our dataset, we solve this issue using the following rule: if a volume has between $128k$ and $128(k+1)$ slices, we choose as starting slice a random number between 0 and k and then sample every k^{th} slice, discarding the others, generating a volume of 128 slices. This way, we make sure that most of the volume is preserved. Our procedure is based on the fact that volumes with large number of slices (corresponding to small slice thickness) admit a transversing with a small sliding window without great loss of information for an accurate Covid diagnosis.

During the inference process, parts of a single volume will be several times input for the model. If a volume has between $256k$ and $256(k+1)$ slices, we choose as starting slice each number between 0 and k and take each k^{th} slice discarding the others. This procedure generates $k+1$ sub-volumes of 256 slices at inference time. For each sub-volume we apply horizontal or vertical flip or depth axis flip (meaning we could apply more than one of these test-time-augmentations). This means $8(k+1)$ inferences for a certain volume. For inference we define two important thresholds: the confidence threshold for non-covid labels and confidence threshold for covid labels. Every non-covid prediction which is below the first threshold is ignored. The same happens for the second threshold but with the covid labels. After the eliminations based on the thresholding, the most frequent diagnosis is given to the specific volume.

Some of the results obtained by the models trained on our custom samples/folds $2-5$ are illustrated in table I, together with the results obtained for training on the official train-validation split. For the official train-validation split the results are obtained by storing 4 models at different epochs during the training process and taking the majority vote (favoring the covid class in case of equal votes) in order to guarantee more stable predictions. For the other entries, a single model is used and the epoch number corresponding to the stored model is reported. The results indicate similar performances, thus rejecting possible hypotheses related to the existence of some biases in the official train-validation split.

Fold/Epoch	F1 score
$2/93$	0.936
$3/99$	0.923
$4/99$	0.936
$5/96$	0.925
official / ensemble	0.9234

TABLE I. Results of the volumetric approach when trained on different train-validation splits. For the official train-test split an ensemble of 4 models is used to provide final predictions these being stored at different epochs during training

For the final prediction pipeline designed for the test set, we constructed an ensemble with 2 models from each fold we trained on, stored at the epochs we obtained the highest validation accuracy, resulting in a total of 10 models used in the ensemble.

C.Self-training

Recently, Semi-Supervised learning has shown to bring improvements over the baselines with minimum human effort.

Some branches of the semi-supervised learning technique are consistency training [[3], [23], [27], [4]], pseudo-labeling [[17], [10], [1], [22]], graph-based methods [[30], [29]], methods that make use of latent variables as target variables [[11], [18]] and low-density separation methods [[7], [24]].

Recently [25] have used consistency training to improve COVID-19 prognosis prediction on chest X-rays. Also recently, [2] have used pseudo-labeling in combination with a graph-based method for identifying COVID-19 on Chest X-rays.

[2]https://pytorch.org/vision/0.8/_modules/torchvision/models/resnet.html
[3]https://github.com/facebookresearch/video-nonlocal-netmain-results

Consistency training methods constrain the model predictions to not change when noise is added to the input, hidden states or model parameters. Pseudo labeling methods use a model to predict labels on unlabeled data either during training or after a model converges. We tried to apply psuedo labeling with a student-teacher approach. We use the same neural network for both the student and teacher. We did not increase the noise in the student training phase, keeping the same augmentations as the ones for the teacher network. We also continued the training for the student from the weights of the last epoch of the teacher network.

Thus, we went further with the experiments evaluating the improvements the Self-Training method could bring on the train-validation split provided by the organisers. We used the best model (a single NN, not an ensemble) trained on the official split to predict on the validation dataset. Out of all these predictions on the validation set we only kept the ones with a confidence score of at least 0.85. This threshold was empirically chosen in order to keep a big part of the validation dataset while also assuring more accurate predicted labels. The classification report on this subset of the validation dataset is shown in Table II

class	precision	recall	f1-score	support
non-covid	0.96	1.0	0.98	171/209
covid	1.0	0.93	0.97	123/165
macro			**0.971**	

TABLE II. Results on the subset with threshold 0.85

The table above shows that the predictions with at least 0.85 confidence score are highly qualitative and quantitative (consisting of at least 80% of the validation dataset).

We enlarged the training set with this subset of the validation dataset (together with the labels predicted by the neural network). We finetuned the model trained for 150 epochs on this new extended training set for another 60 epochs. We also lowered the base learning rate to $1e-4$ for smaller modifications of the gradients. We kept the other configurations the same.

Testing again with the model with the highest accuracy on the validation dataset we obtained the results shown in Table III on the official validation dataset:

Fold/Epoch	F1 score
$1/43(+93)$	**0.9399**

TABLE III. Results on the validation dataset after Self Supervised Training

The increase in the F1 score is of about 1.65% which is a consistent one. We believe that such approach could improve our result on the test set.

IV. Slice-based approaches

Slice-based classification approaches for the CTs would bring two important advantages over a volumetric approach: 1)lower needs in terms of computational resources and 2) more informative output/decision, since they identify the slices in the CT that conduct to a certain diagnostic.

We further present two slice-based approaches in an attempt to compete the volumetric approach described previously.

A. A slice based approach tackled with neural network architecture search

This approach is intended to act as our own baseline, as it consists of very simple steps that could be taken to build a classifier for CT data, requiring no advanced knowledge into the domain nor fine tuning, but resorting to existing frameworks. These steps are:

- label each CT slice in the training set with its corresponding CT label;
- use an existing implementation that performs neural architecture search for image classification to generate a network architecture that performs well on our problem at hand - classifying the CT slices;
- aggregate the results at slice level using simple statistics in order to classify a CT.

In the last few years there have been great improvements in the field of Neural Architecture Search (NAS). The state-of-the-art models are able to find neural network architectures in a small number of GPU days, much less than the long search times that were required just a few years ago. Since we are dealing with a classification problem, employing a neural architecture search algorithm that has a very good result on a benchmark classification dataset is an idea worth exploring.

The current state-of-the-art neural architecture search algorithm for the classification of the CIFAR-10[4] dataset is sharpDARTS, introduced in [9]. The sharpDARTS algorithm operates in such a way that it searches for a primitive that will appear multiple times in a larger neural network architecture.

The authors of the sharpDARTS algorithm are modelling the space of the neural network architectures in such a way that it is differentiable. In this differentiable search space, gradient based search methods are proven to be very effective. The authors have also made the implementation of sharpDARTS available [5]. Starting from the authors' implementation, we have made the necessary adjustments to the implementation in order to run the search on the present dataset. In this approach, the classification takes place at the level of each CT slice. We have downsampled

[4]https://www.cs.toronto.edu/ kriz/cifar.html
[5]https://github.com/ahundt/sharpDARTS

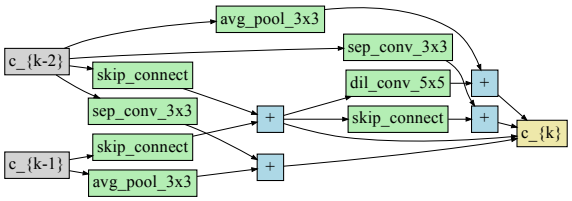

Fig. 1. The primitive found by the sharpDARTS search algorithm.

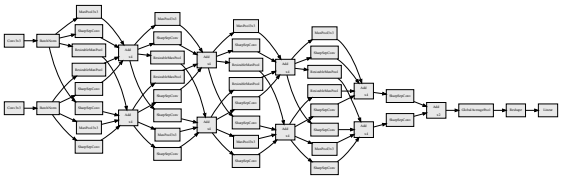

Fig. 2. The final architecture found by the sharpDARTS search algorithm.

the resolution of the images from 512x512 to 192x192 in order to speed up the training process by avoiding the loading in memory of the entire dataset multiple times.

We build the slice-based dataset in a very simple manner, being aware that it has some important flaws: every label in a COVID CT is marked as COVID, while every label in a non-COVID CT is marked as non-COVID. No filtering is performed, meaning that we expect to have many similar slices in the dataset with contradictory labels, coming from both COVID patients and healthy patients, since not all the slices in a COVID CT present COVID specific lesions.

SharpDARTS' authors have found a state-of-the-art model on the CIFAR-10 dataset in 0.8 days. We have run the search algorithm for 1 day, finding a primitive presented in Figure 1.

With this primitive, we have constructed the final architecture as shown in Figure 2. We continued to train the final neural network architecture for 300 epochs.

We present the results of this per-slice approach in Table IV - scores are computed here based on slice labels and not CT labels (basically we measure the performance of the model for labeling slices as coming from COVID or non-COVID CTs).

We compute the label for each volume by utilizing the majority vote rule, based on the labels predicted for all the slices in the volume. The results for CTs classification are presented in Tables V and VI. Despite it being a relatively small architecture of only 61.3 MMAC (million multiply-accumulate operations; 122.6 MFLOPS) and being trained on noisy data, it obtained an unexpectedly good F1 score on the validation dataset of 0.74.

It is worth improving this approach by utilizing the slice labeling procedure described in Section IV-B. This would help eliminate the issue of slices that are not informative relative to the covid or non-covid nature of the CT. We would also like to remind that this approach is at a disadvantage since it only utilizes a single slice at a time, losing context information.

Data	precision	recall	macro F1
Train	0.724	0.731	0.727
Validation	0.729	0.777	0.752

TABLE IV. Per slice results of the sharpDARTS algorithm

B.A slice based approach with mini-volumes

This method is a slice based approach with the addition that it incorporates the immediate previous and next slice creating a "minivol". It uses both slice and lung (side) level labels which we obtained in two steps:

- a neural network trained on an external dataset but with slice level labels (the dataset in question is a small subset of the training set for the ImageCLEF 2021 - Tuberculosis detection competition [20], which we manually labeled)
- manual correction of the predictions and further labeling

As the COV19-CT-DB dataset it too big to manually address slice labeling, we applied this process to only a small subset of the data. The final dataset we construct will have 3 labels: COVID, non-COVID pneumonia, healthy.

To use the aforementioned neural network trained at ImageClef, we need to apply on the COV19-CT-DB data the same pre-processings used for its training. This turned out to be a complex and imprecise task as the original ImageClef training set used CT volumes in DICOM format and some of the pre-processing steps were applied on the Hounsfield values of the input, while the COV19-CT-DB is in jpg format.

Next, we shortly describe the process involved in building the model for tuberculosis classification in the ImageClef competition, in order to understand its use for building the COVID refined dataset here. Given a selected slice in a CT, we grouped it together with the previous and the next slice in the volume; these mini volumes of 3 consecutive slices, we thought, could better highlight the lesions present in the tuberculosis dataset, emphasising the difference between an infiltration and an artery, or a cavern and a lumen as these can be very similar at a certain point in space but continue in a different manner. We changed its window and level values to highlight the lung features. The selected slices were split into half, corresponding to each lung, and we kept only the side that was labeled. We cropped the images, using a simple threshold method to remove the padding and keep only the body. The resulting images were resized to 256×256 pixels. These were then normalized with values in the range $[0, 255]$ corresponding

to 3 black and white images which were concatenated at channel level. As augmentations we used a random crop of size 224×224, a random horizontal flip with a probability of 0.5 and normalized the image. We trained an EfficientNet-B3 [26], with batches of 32 for 90 epochs.

We used this network to predict on each slice of a volume for the COVID patients in the COV19-CT-DB training set the presence of an affection, and obtain the probabilities of each affection type that was present in the ImageClef data (a total of five lesion types and one healthy class).

In order to be able to apply this pre-trained model on the COVID dataset we had to address several problems. The steps we followed to address the pre-processing are:

- We aggregated the slices into volumes
- Constructed an equivalent cropping functions based on pixel thresholds
- Made a prediction with the pre-trained neural network and filtered the results based on one specific label - "healthy". Using an upper and a lower threshold we filtered out the predictions with a score for the "healthy" label between those values, thus keeping only high confidence predictions. We also filtered based on slice location, preferring slices at the middle at the series so as not to saturate the "healthy" class with slices at the ends of the volume that were by default healthy as they did not contain yet the lungs.
- We retrained and made the predictions on the entire original training set. In order to obtain the same number of predictions per patients for the last training we resized the reconstructed volumetric image to a depth of size 96. As some of the patients' folder have very few slices this means the resize would just multiply the same images along the volume.
- For each slice we predicted the probabilities of the 3 classes enumerated above. We grouped the slices predictions into an array of size $(96, 3)$. We did this for each patient, thus creating a data set that was used as training data for a linear classifier that learns to predict the class at CT level.

To predict on the validation set we would first have to predict with the first neural network on the reconstructed slices at slice level and then predict on the results with the second classifier at patient level. We trained a logistic regression classifier and a multilayer perceptron with 100 neurons and a single layer. The results on the training and validation sets are shown in tables 2 and 3.

V. Experimental results

Tables V and VI present the results obtained on the training and validation sets, in terms of precision and recall for the COVID class, and the macro F1 score, as required in the competition. Table VII presents the results obtained on the test set, according to the evaluation made by the organizers based on the labels we submitted. We report the results for all our methods, excepting for the method that uses self-training, situation which is due to our impossibility of providing the labels on the test set in time to be evaluated in the competition.

All the approaches taken outperform the baseline score set by the competition organizers [12] on the validation set.

The volumetric approach achieves the best results, bringing a consistent improvement over the slice-wise approaches, with a 90.06% F1 score on the test data.

The best slice-wise approach is the one building the training set by labeling the slices using a network pre-trained on a Tuberculosis task, then taking a mini-volume approach for slice classification and aggregating the results at CT level using logistic regression. This one achieved a 81.85% F1 score on the test data.

The simplest approach, in terms of the effort taken to build the classifier, achieved an F1 score of 76.73% on the test data.

Although with lower accuracy, we consider the slice-based approaches to be more relevant in practice, providing more information related to lesion localization in the lungs and implicitly its extension/size. Given previous experience on similar CT classification tasks [19], [20], we argue that the poorer results are due to the presence of noisy labels at slice level in the training data (because of the automated slice labeling procedures we use) and filtering manually COVID slices is necessary to build a good training set for increasing the performance of this approach.

We still think, in the light of the results obtained on the validation data, that semi-supervised training could improve our best results obtained on the testing set, but this is for the moment just a hypothesis to be checked.

VI. Conclusions

The paper presents and evaluates distinct approaches for the problem of COVID19 detection in CTs. Given the nature of the data, consisting of volumetric (3D) images, two distinct ways to perform CT classification are used: one that treats the CT as a whole, and one that performs classification at slice level and then aggregates the results at CT level. The experiments show best results for the volumetric approach surpassing a 0.90 F1 score on the validation and test data. However, with a better strategy to build a clean slice-labeled data set, the slice based approach could definitely achieve higher performance compared to its current reported results (0.81 F1 score), at the advantage of less computational resources used and a more informative diagnostic when compared to the volumetric approach.

Method	precision (Covid class)	recall (Covid class)	macro F1 score
volumetric approach	0.97	0.96	0.96
slice-wise approach based on sharpDARTS	0.93	0.94	0.93
slice-wise approach based on minivolumes & log reg	0.95	0.89	0.93
slice-wise approach based on minivolumes & mlp	0.98	0.93	0.97

TABLE V. Results on the train set for: the volumetric approach described in Section III, the slice based approach using neural architecture search described in Section IV-A and the two slice based approaches based on mini-volumes described in Section IV-B

Method	precision (Covid class)	recall (Covid class)	macro F1 score
volumetric approach	0.95	0.88	0.92
slice-wise approach based on sharpDARTS	0.67	0.82	0.74
slice-wise approach based on minivolumes & log reg	0.83	0.71	0.82
slice-wise approach based on minivolumes & mlp	0.87	0.76	0.84

TABLE VI. Results on the validation set for: the volumetric approach described in Section III, the slice based approach using neural architecture search described in Section IV-A and the two slice based approaches based on mini-volumes described in Section IV-B

Method	macro F1 score
volumetric approach	0.9006
slice-wise approach based on sharpDARTS	0.7673
slice-wise approach based on minivolumes & log reg	0.8185
slice-wise approach based on minivolumes & mlp	0.6435

TABLE VII. Results on the test set for: the volumetric approach described in Section III, the slice based approach using neural architecture search described in Section IV-A and the two slice based approaches based on mini-volumes described in Section IV-B

VII. Acknowledgements

This research was partially supported by the Competitiveness Operational Programme Romaniaunder project number SMIS 124759 - RaaS-IS (Research as a Service Iasi)

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2020.

[2] Angelica I Aviles-Rivero, Philip Sellars, Carola-Bibiane Schönlieb, and Nicolas Papadakis. Graphxcovid: Explainable deep graph diffusion pseudo-labelling for identifying covid-19 on chest x-rays, 2021.

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles, 2014.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning, 2019.

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[6] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *CoRR*, abs/2010.01412, 2020.

[7] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 529–536, Cambridge, MA, USA, 2004. MIT Press.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Andrew Hundt, Varun Jain, and Gregory D. Hager. sharpdarts: Faster and more accurate differentiable architecture search. *CoRR*, abs/1903.09900, 2019.

[10] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning, 2019.

[11] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models, 2014.

[12] Dimitrios Kollias, Anastasios Arsenos, Levon Soukissian, and Stefanos Kollias. Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis. *arXiv preprint arXiv:2106.07524*, 2021.

[13] Dimitrios Kollias, N Bouas, Y Vlaxos, V Brillakis, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and S Kollias. Deep transparent prediction through latent representation analysis. *arXiv preprint arXiv:2009.07044*, 2020.

[14] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias, and Georgios Tagaris. Deep neural architectures for prediction in healthcare. *Complex & Intelligent Systems*, 4(2):119–131, 2018.

[15] Dimitris Kollias, Y Vlaxos, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and Stefanos D Kollias. Transparent adaptation in deep medical image diagnosis. In *TAILOR*, pages 251–267, 2020.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[18] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models, 2016.

[19] Radu Miron, Cosmin Moisii, and Mihaela Breaban. Revealing lung affections from cts. A comparative analysis of various deep learning approaches for dealing with volumetric data. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, editors, *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[20] Cosmin Moisii, Radu Miron, and Mihaela Breaban. Identifying tuberculosis type in cts. In Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, and Florina Piroi, editors, *Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021*, CEUR Workshop Proceedings. CEUR-WS.org, 2021.

[21] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? *CoRR*, abs/1906.02629, 2019.

[22] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, and Quoc V. Le. Meta pseudo labels, 2021.

[23] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks,

2015.
[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.
[25] Anuroop Sriram, Matthew Muckley, Koustuv Sinha, Farah Shamout, Joelle Pineau, Krzysztof J. Geras, Lea Azour, Yindalon Aphinyanaphongs, Nafissa Yakubova, and William Moore. Covid-19 prognosis via self-supervised representation learning and multi-image prediction, 2021.
[26] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.
[27] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2018.
[28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks, 2018.
[29] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1168–1175, New York, NY, USA, 2008. Association for Computing Machinery.
[30] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 912–919. AAAI Press, 2003.