COVID Detection and Severity Prediction with 3D-ConvNeXt and Custom Pretrainings

Daniel Kienzle[(✉)], Julian Lorenz, Robin Schön, Katja Ludwig, and Rainer Lienhart

Augsburg University, 86159 Augsburg, Germany
{daniel.kienzle,julian.lorenz,robin.schoen,katja.ludwig,
rainer.lienhart}@uni-a.de

Abstract. Since COVID strongly affects the respiratory system, lung CT-scans can be used for the analysis of a patients health. We introduce a neural network for the prediction of the severity of lung damage and the detection of a COVID-infection using three-dimensional CT-data. Therefore, we adapt the recent ConvNeXt model to process three-dimensional data. Furthermore, we design and analyze different pretraining methods specifically designed to improve the models ability to handle three-dimensional CT-data. We rank 2nd in the *1st COVID19 Severity Detection Challenge* and 3rd in the *2nd COVID19 Detection Challenge*.

Keywords: Machine learning · COVID detection · Severity prediction · Medical image analysis · CT scans · 3D data

1 Introduction

The last few years have been strongly shaped by the COVID-19 pandemic, with a considerable amount of cases ending deadly. For the treatment of patients it is crucial to predict the severity of lung damage caused by a SARS-CoV-2 infection accurately. The lung damage is visually detectable by visible ground-glass opacities and mucoid impactions on the slices of a patients CT-scan [34]. Thus, it might be beneficial to automatically process CT-scans for the diagnosis of the patients.

In this paper, we introduce a neural network to automatically analyze CT-scans. We train our model to classify the severity of lung damage caused by SARS-CoV-2 into four different categories. The model is trained and evaluated using the COV19-CT-DB database [19]. Additionally, we transfer our architecture and training pipeline to the detection of SARS-CoV-2 infections in CT-scans and train a separate model for this task. Consequently, we show that our method can easily be transferred to multiple COVID-related analyses of CT-scans. We rank 2nd in the *1st COVID19 Severity Detection Challenge* and 3rd in the *2nd*

D. Kienzle, J. Lorenz and R. Schön—Authors contributed equally.

COVID19 Detection Challenge. Moreover, our model is especially good at identifying the most severe cases that are most important to detect in a clinical setting.

As medical datasets are small in comparison to common computer-vision datasets, the application of large computer-vision architectures is not straight forward as they tend to overfit very quickly. As a result, the development of a good pretraining pipeline as well as the utilization of additional data is essential in order to get adequate results.

Since medical datasets are comparably small, the validation split is as well in most cases very small. However, evaluating the models performance on a single small validation set leads to non-representative results as the validation set is not representative for the overall data distribution. Furthermore, the evaluation on a single small dataset could cause overfitting of the hyperparameters to the validation set characteristics and, therefore, reduces the models test-set performance. As a result, it is very important to use strategies like cross-validation in order to get a better estimate of the models performance.

Goal of this paper is to develop a neural network that is capable of automatically predicting four degrees of severity of lung damage from a patents lung CT-scan. In addition, we also adapt our architecture to predict infections with the SARS-CoV-2 virus using CT-scans. In order to improve the performance on these two tasks, our main contributions are:

1. We adapt the recent ConvNeXt architecture [27] to process three dimensional input-data.
2. We introduce multiple techniques for pretraining of our architecture in order to increase the ability of our network to handle three-dimensional CT-scans.

2   Related Work

The idea of using neural networks for the prediction of certain properties visible in medical data has developed to increasing levels of importance in the last few years (examples can be found in [1,40,43]). The authors of [4,20,21] and [22] have used CNNs and Recurrent Neural Networks (RNNs) for the prediction of Parkinson's disease on brain MRI and DaT scans. In [29,33,39] NN-based methods for the detection of lung cancer are developed.

Since its occurrence in late 2019, a considerable number of articles have concerned themselves with using neural networks for the purpose of predicting a potential SARS-CoV-2 infection from visual data. The authors of [31] propose neural networks for the usage of CT scans as well as chest x-rays, whereas [32] puts a lot of focus on computational efficiency and design a lightweight network, in order to be able to also run on CPU hardware. In the wake of this development, there have also been methods which combine neural and non-neural components. In [5] the authors first extract the features by means of neural network backbone, and the utilize an optimization algorithm in combination with a local search method before feeding the resulting features into a classifier. The method of [24] applies a fusion based ranking model, based on reparameterized Gompertz

function, after the neural network has already produced its output probabilities. In [22] and [20] the authors also make use of clustering in order to carry out a further analysis of the produced features vectors, and classify the CT scans according to their proximity to the cluster centers.

In the context of last years ICCV there has been a challenge with the aim of detecting COVID-19 from CT images [19]. The winners of this contest [16] were using contrastive learning techniques in order to improve their networks performance. [30] and [25] use 3D-CNNs for the detection of the disease, whereas the teams in [45] and [41] happen to use transformer-based architectures. In addition to that, [3] proposes the usage of AutoGluon [12] as an AutoML based approach.

Our architectures is, similar to other previously existing approaches for the processing of 3D-data, based on the idea, that 2D architectures can be directly extended to 3D architectures [7,23,36]. In our case we use 3D modification of the ConvNeXt architecture [27]. This particular approach is characterized by architectural similarities to MobileNets [17,37] and Vision Transformers (ViT, [11,26]).

In the medical field, due to privacy restrictions that protect the patients' data, the potential amount of training data is rather sparse. This especially holds, when it comes to datasets that accompany particular benchmarks. However, datasets for similar tasks may be exploited for pretraining and multitask learning, if one can assume that insights from one datasets might benefit the main objective. This inspired the authors of [42] and [13] to pretrain the model on pretext tasks, which are carried out on medical data. The authors of [28] show that pretraining on ImageNet is useful, by the virtue of the sheer amount of data. In some particular cases, we might have access to a larger amount of data while lacking labels. The publications [38] and [8] use semi-supervised learning techniques to overcome this particular situation.

3   Methods

Goal of this work is to develop a neural network architecture capable of predicting the severity of a SARS-CoV-2 infection and to transfer the method to the task of infection detection. We apply our models to the COV19-CT-DB database [19]. The train and test set of this database consist of 2476 scans for the task of infection detection and 319 images for severity prediction. Each scan is composed of multiple two-dimensional image slices (166 slices on average). We concatenate these slices into a three-dimensional tensor and apply cubic spline interpolation to get tensors of the desired spatial dimension. In this section we introduce the key methods for improving the automatic analysis of this data.

3.1   ConvNeXt 3D

The architecture we utilize is a three-dimensional version of the recent ConvNeXt architecture [27]. This architecture type is especially characterized by multiple alterations, which have already proven themselves to be useful in the context of Vision Transformers, and were applied to the standard ResNet [14].

For example, ConvNeXt has a network stem that patchifies the image using non-overlapping convolutions followed by a number of blocks with a compute ratio of $(3:3:9:3)$ that make up the stages of the network. The influence which MobileNetV2 [37] had on this type of architecture is expressed by the introduction of inverted bottleneck blocks and the usage of depthwise convolutions, which, due to their computational efficiency, allow for an unusually large kernel size of 7×7. Additional distinguishing properties of this architecture are the replacement of Batch Normalization by Layer Normalization, the usage of less activation functions and the replacement of the ReLU activation function by the GELU activation [15].

The standard ConvNeXt architecture in its initial form was conceptualized for the purpose of processing 2D images with 3 color channels, whereas we want to process 3D computational tomography scans that only have one color channel (initially expressed in Hounsfield Units [6]). We adapted the ConvNeXt architecture to our objective by using 3D instead of 2D convolutions. In order to be able to make use of potentially pre-existing network weights, we apply kernel weight inflation techniques to the 2D networks parameters as described in Sect. 3.2.

3.2   Pretraining

In contrast to ordinary computer vision datasets like e.g. ImageNet [10], medical datasets are usually considerably smaller. In order to still be able to train large neural networks with these datasets we utilize various pretraining techniques. As our data consists of three-dimensional gray-scale tensors as input data instead of two-dimensional RGB-images and we use 3D-convolutions instead of 2D-convolutions, it is not possible to directly use the publicly-available pretrained ConvNeXt weights. In this section we present various possibilities for the initialization of our network with pretrained weights.

For 2D models, it is common to pretrain a model on the task of ImageNet classification. As our data consists of gray-scale tensors, we implemented a ConvNeXt pretraining with gray-scale ImageNet images to obtain weights for a two-dimensional ConvNeXt model. To use those weights for our 3D model, we propose three different inflation techniques for the two-dimensional weights of the pretrained 2D model. We will refer to these as *full inflation*, *1G inflation* and *2G inflation*.

Let $K \in \mathbb{R}^{I \times O \times H \times W}$ be the 2D kernel weight tensor, and $K^{\uparrow} \in \mathbb{R}^{I \times O \times H \times W \times D}$ be the 3D kernel weights after inflation. For these kernel weights we denote by I the input channels, O the output channels, H the height, W the width and D the additional dimension of the 3D kernel. Also, let i, o, h, w, d denote all possible positions along the aforementioned dimensions. γ is a normalization factor that normalizes the inflated tensor $K^{\uparrow}$ to have the L2 norm of the 2D kernel K.

The first way, called *full inflation*, is the commonly used option of simply copying the weights along the new tensor axis [7]. This can be described as an equation of the form

$$\forall i, o, h, w, d \ : \ K^{\uparrow}_{i,o,h,w,d} = K_{i,o,h,w} \cdot \gamma. \tag{1}$$
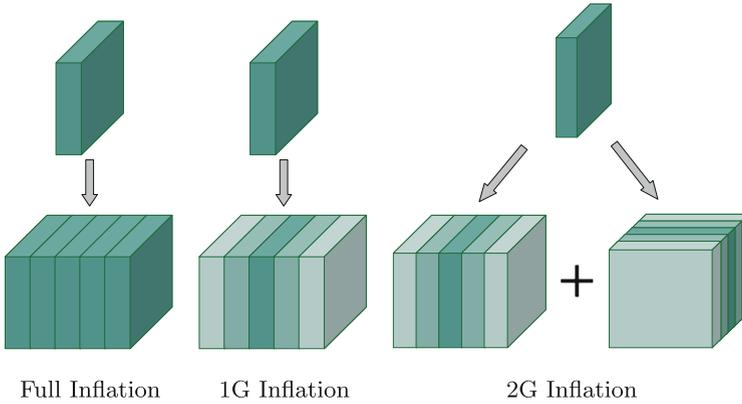
In *1G inflation*, we use a Gaussian weight $\mathcal{N}(\cdot, \mu, \sigma)$ in order to create different weights, that are the largest in the kernel center:

$$\forall i, o, h, w, d \ : \ \mathrm{K}^{\uparrow}_{i,o,h,w,d} = \left(\mathrm{K}_{i,o,h,w} \cdot \mathcal{N}\left(d, \frac{D}{2}, \frac{D}{8} \right) \right) \cdot \gamma \tag{2}$$

The third, that is referred to *2G inflation* is based on multiplying the 2D weights along 2 axes:

$$\forall i, o, h, w, d \ :$$

$$\mathrm{K}^{\uparrow}_{i,o,h,w,d} = \left(\mathrm{K}_{i,o,h,w} \cdot \mathcal{N}\left(d, \frac{D}{2}, \frac{D}{8} \right) + \mathrm{K}_{i,o,h,w} \cdot \mathcal{N}\left(w, \frac{W}{2}, \frac{W}{8} \right) \right) \cdot \gamma \tag{3}$$

The different inflation approaches to create 3D kernels are visualized in Fig. 1.

Fig. 1. A visualization of the different inflation approaches to generate 3D kernels. *Full inflation* simply copies the weights along the new axis. *1G inflation* also copies the weights along the new axis but multiplies them with Gaussian weights. *2G inflation* acts similar to 1G inflation but add the weights after going over two dimensions.

Since the images in the ImageNet database are very distinct from CT-images as used in this paper, we introduce various further ways to adjust the model to three-dimensional CT-scans. For instance, we use an additional dataset designed for lung-lesion segmentation in CT scans [2,9,35] and the STOIC dataset created for SARS-CoV-2 severity prediction [34]. As those datasets consist of CT-scans of SARS-CoV-19 infected patients similar to the COV19-CT-DB database, we assume that pretraining with these additional datasets will be beneficial for our model performance and will increase in robustness as it is able to deal with a greater variety of data.

Since the lung damage caused by SARS-CoV-2 is visually detectable in lung CT-scans, we use the segmentation dataset to pretrain our model to segment lung lesions. By directly showing the damaged lung regions to the network we hope

to provide a reasonable bias for learning to predict the severity. Furthermore, a segmentation pretraining is beneficial as the segmentation task is more robust to overfitting in contrast to a classification task. This is important as it enables us to apply large-scale architectures to the small medical datasets.

The STOIC dataset provides two categories of severity for each patients CT-scans. Even though the categories in the STOIC dataset are different to the categories in the COV19-CT-DB database we assume, nevertheless, that pretraining with the STOIC dataset teaches the network a general understanding of severity.

In order to be able to generate segmentation masks with our model additionally to severity classification outputs, we extend our architecture similarly to the Upernet architecture [44]. Our architecture is explained in Fig. 2
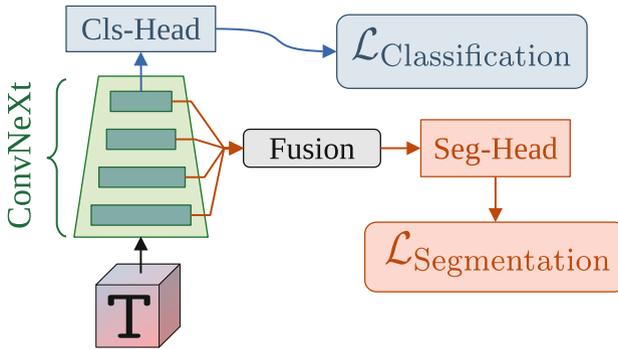
Fig. 2. Visualization of the pretraining architecture. An *input-tensor* T is processed by the *ConvNeXt* model. To generate a classification label (e.g. for severity prediction or infection detection) the output-features of the last block are processed by a task-specific *classification head* in order to generate the class probabilities. In order to compute a segmentation mask, the output-features of every block of the ConvNeXt architecture are upsampled and concatenated similar to [44]. This is further processed by a *segmentation-head* in order to produce the segmentation output.

In this work we compare 4 different pretraining methods:

1. We directly use our inflated grayscale ImageNet pretraining weights. This approach is referred to as *ImageNet model.*
2. We train our network for the task of segmentation on the segmentation dataset starting from inflated ImageNet weights (1.). This approach is referred to as *segmentation model*.
3. Pseudo-labels are generated with the segmentation model (2.) in order to get segmentation masks for the COV19-CT-DB database. We train a new model for the task of segmentation on the COV19-CT-DB database using the pseudo-labels as ground-truth. The model is initialized with our inflated ImageNet weights (1.). This is referred to as *segmia model.*

4. Pseudo-labels are generated with the segmentation model (2.) to get segmentation masks for the STOIC dataset. We train a new model to jointly optimize severity classification using real labels and segmentation using pseudo-labels for the STOIC dataset. This model is initialized with our inflated ImageNet weights (1.). This is referred to as *multitask model*

After each pretraining method, we finetune our model for either severity prediction or infection detection on the COV19-CT-DB database.

3.3   Approaches for Increased Robustness

When automatically analyzing CT images, we have to account for multiple potential forms of irregularities. Not only might different CT scanners result in varying image details, but the person in the scanner might also lie differently on every image. We thus have to increase the models robustness.

We use the following classical augmentations: Random flips along all three axes, Gaussian noise with random standard deviation between 0.6 and 0.8, and Gaussian blur. Since some patients lie in the CT scanner with a slight inclination to the side of their body, we rotate the tensors along the transversal axis by a randomly picked angle from the interval $(-30°, 30°)$.

As is common practice in the field of medical computer-vision, we also use elastic deformations (with a chance of 50%). For the generation of those deformations the vector field is scaled by a randomly drawn $\alpha \in (1, 7)$ and then smoothed with a Gaussian kernel with $\sigma = 35$. We create our own GPU compatible implementation, and decompose the used Gaussian filter along its axes. This results in an augmented computational speed, rendering the deformations viable for fast online computation.

During manual inspection, we came across some tensors that were oriented in a different way (for example vertically instead of horizontally) or with patients facing a different direction. In order to make our network robust to these variations, we add an augmentation that simulates different orientations: For each of the three axes (x-, y- and z-axis) we randomly pick a multiple of $90°$, and rotate the tensor by this angle. Since the misoriented tensors still constitute a considerable minority of all cases we do not apply this augmentation to every tensor during training, but only with a probability of 25%.

In addition, we apply random crops with a probability of 50%. Therefore, we first rescale the scan to a resolution of $(256 \times 256 \times 256)$ and take a random crop of $(224 \times 224 \times 224)$. When no random crop is applied, we directly rescale to the latter resolution. Rescaling of the tensors is performed with cubic spline interpolation and the rescaled tensors are precomputed prior to the training to decrease computation time. When inspecting the data, we discovered some tensors where the slice resolution happened to be internally inconsistent between different slices of the same CT. In those cases, we discarded the inconsistent slices. In order to stabilize our performance, we kept a second copy of our model whose weights were not learned directly, but which is an exponential moving

average (EMA) of the trained models weights. This copy of the network is used for the evaluations and final predictions.

Besides data augmentation, we also use 5-fold cross validation to improve model robustness. We split the public training set into five folds with almost equal size and make sure that each class is evenly distributed across all folds. For example, each fold for severity prediction contains 12 or 13 moderate cases. Each fold forms the validation set for one model, the remaining folds serve as the training set. This way we get 5 models that are trained and evaluated on different datasets. To get the final predictions for the official test set, we predict every case from the test set using all of the 5 models and take the mean of the model outputs. Before averaging the outputs, we apply Softmax to bring all values to the same scale.

4 Experiments

Most experiments were performed for the task of severity prediction because of two reasons. First, it is supposedly the more challenging task due to the classification into four classes and, second, it is computationally less expensive due to a smaller dataset allowing for a greater number of experiments.

We evaluate our models using the COV19-CT-DB database. As the validation set is comparatively small, evaluating the models' performance on its validation set leads to non-representative results as the validation set is not representative for the data. Furthermore, evaluation on a single validation set easily causes overfitting of the hyperparameters. As a result, we do not use the validation set to analyse the models performance and instead perform 5-fold cross-validation only on the training set. The average performance of the 5 training runs is used as an estimate for the performance. In order to generate predictions for the validation set and test set we use an ensemble consisting of the 5 models from the cross-validation runs. The predictions of the models are averaged after application of the softmax to produce the final prediction. In addition, we also add the results obtained only with the 5th model.

4.1 Preliminary Experiments

As the frequency of the classes in the COV19-CT-DB database is not evenly distributed, we recognized that our neural network trained with ordinary cross-entropy as loss function has problems detecting the less frequent classes. Especially patients with critical severity are not correctly classified. As it is especially important to recognize the critical cases, we introduced a weight in order to balance the cross entropy. Thus, the loss for every class is multiplied with the normalized class-frequency. The severity-prediction results for the ordinary cross entropy (*CE*) and balanced cross entropy (*balanced CE*) are given in Table 1. The cross-validation performance of both loss functions is very similar. Even though the ordinary cross-entropy performs a little bit better, we chose to use the balanced cross entropy for our further experiment since we think that the

balanced cross entropy improves the performance for the underrepresented critical cases. We assume that the good performance of our challenge submission (see Table 4) for critical cases is partly because of the balanced cross-entropy. Consequently, we would advise to use balanced cross-entropy instead of ordinary cross-entropy in a clinical setting.

Table 1. Comparison of balanced cross entropy with ordinary cross entropy. The models are initialized with full ImageNet initialization. The cross validation results and the results for the official validation set are reported. The ensemble predictions are marked with a †. F1 scores are macro F1 scores

Loss	F1 cross val	F1 val	F1 val mild	F1 val moderate	F1 val severe	F1 val critical
balanced CE	64.99	$62.82^{\dagger}$	$82.93^{\dagger}$	$57.14^{\dagger}$	$57.89^{\dagger}$	$53.33^{\dagger}$
CE	**65.37**	$60.10^{\dagger}$	$82.05^{\dagger}$	$50.00^{\dagger}$	$55.00^{\dagger}$	$53.33^{\dagger}$

4.2 Comparison of Pretrainings

One main goal of this paper is to enhance the severity-prediction performance by introducing several pretrainings as explained in Sect. 3.2. Results for the various ImageNet inflation methods are added in Table 2.

Table 2. Comparison of ImageNet initialization. The cross validation results and the results for the official validation set are reported. The ensemble predictions are marked with a †. F1 scores are macro F1 scores

Initialization	F1 cross validation	F1 validation
Full	65.67	$\mathbf{61.28}^{\dagger}$
1G	65.81	$60.71\ 59.93^{\dagger}$
2G	**69.61**	$56.92^{\dagger}$

According to these results, the performance is best for 2G inflation in terms of the cross-validation metrics and, consequently, we assume that using multiple geometrical-oriented planes is beneficial as it better utilizes all three dimensions.

Results for the various pretraining methods can be seen in Table 3. For some of those experiments, the scores for the official test set of the COV19-CT-DB database are available. As this test set is substantially larger than the official validation set, the performance estimate is considered as more accurate. Thus, we use this metric in addition to our cross-validation results to interpret our performance. The results clearly indicate that the cross-validation metrics is a much better estimate of the models performance than the validation-set metrics since the *segmentation*, *segmia* and *multitask model* perform better than the

Table 3. Comparison of models initialized with different pretrainings. Performance is evaluated for the severity-prediction task. Random-initialization is denoted as *Random*. The ensemble predictions are marked with a †. F1 scores are macro F1 scores.

Pretraining	F1 cross validation	F1 test	F1 validation
Random	62.71	–	$56.46^{\dagger}$
ImageNet (full)	65.67	$45.73^{\dagger}$	$61.28^{\dagger}$
Segmentation	67.25	$46.21^{\dagger}$	$61.28^{\dagger}$
Segmia	66.48	$48.85^{\dagger}$	$63.05^{\dagger}$
Multitask	**68.18**	$\mathbf{48.95}^{\dagger}$	$58.77^{\dagger}$

ImageNet model on both cross-validation metrics and test-set metrics. Moreover, the best model in terms of cross-validation metrics performs also best on the test-set. As a result, we strongly advise to evaluate models intended for clinical usage based on cross-validation metrics.

Even though cross validation gives well reasoned clues about the models performance, a large gap between the cross-validation score and the test-set metrics can be observed. Because the test-set performance is worse by a large margin, we suppose that the test-set statistics do not fully match the train-set characteristics and, thus, there could be a small domain shift in the test-set data. As a result, good test-set results can only be achieved with a robust model and it seems that the utilization of additional datasets and the use of pseudo-labels both increase the robustness of the model significantly.

In Table 3 it is clearly visible that all pretrainings yield significantly better results than a randomly initialized model in terms of the cross-validation metrics and the *segmentation*, *segmia* and *multitask* models score is higher than the score of the *ImageNet model*. Consequently, it can be concluded that a pretraining utilizing segmentation labels is highly favorable. As the *multitask model* outperforms the other variants on the test-set as well as on the cross-validation metrics, we think that a pretraining with a task similar to the final task as well as the utilization of segmentation pseudo-labels is very beneficial. We suppose that the models gain a greater robustness due to the usage of the additional datasets in the pretraining pipeline. As the STOIC dataset used for the multitask pretraining is comparably large, the *multitask model* seems to be especially robust, thus performing best on the test-set. Subsequently, we recommend combining a task similar to the final task with a segmentation task for superior pretraining results.

4.3   Challenge Submission Results

We participate in two challenges hosted in the context of the Medical Image Analysis (MIA) workshop at ECCV 2022 [18]. The two tasks were the detection of COVID infections and the prediction of the severity the patient is experiencing. We rank 2nd in the *1st COVID19 Severity Detection Challenge* and 3rd

in the *2nd COVID19 Detection Challenge*. In this section, we present our submissions and further discuss the results. Our submission code is published at https://github.com/KieDani/Submission_2nd_Covid19_Competition.

Since we apply 5-fold cross-validation, we suggest to use an ensemble of the 5 trained models to generate the validation-set as well as test-set predictions. However, due to a coding mistake, we only use the fifth model instead of the full ensemble during the challenge. As a result, this causes deviations from the cross-validation estimate as our model is only trained with 80% of the training data. Nevertheless, it is even more impressive that we still achieved such good results and, thus, our architecture and pretraining pipeline are very well suited for COVID-related tasks.

In contrast to using the fifth model for predictions, we advise to use either an ensemble of the 5 models or to train a single model with all available data based on the settings found with cross validation.

1st COVID19 Severity Detection Challenge: The ranking for the winning teams is shown in Table 4. It can be seen that our method is by a large margin the best in predicting *Severe* and *Critical* cases. We suppose this is achieved through the utilization of the balanced cross entropy and we emphasize that this property is exceptionally important for clinical use cases.

Table 4. Comparison of best submissions of the winning teams in the 1st COVID19 Severity Detection Challenge. Performance is evaluated for the severity-prediction task. Our prediction is calculated only with the 5th model of the 5-fold cross-validation. F1 scores are macro F1 scores.

Team	F1 test	F1 test mild	F1 test moderate	F1 test severe	F1 test critical
1st: FDVTS	**51.76**	58.97	**44.53**	58.89	44.64
2nd: Ours	51.48	**61.14**	34.06	**61.91**	**48.83**
3rd: CNR-IEMN	47.11	55.67	37.88	55.46	39.46

Since it was possible to submit up to 5 different solutions to the challenge, we list our submissions in Table 5. The *segmia model* performs best. Furthermore, the comparison of submission 2 and 5 indicates that the random-orientation augmentation increases the performance. However, as the test-set scores are very similar the augmentation does not have a great effect on the test set and, thus, is negligible for this challenge. We suppose that this is due to fewer CT-scans with deviating orientation in the test set compared to the train and validation sets.

Table 5. Our submissions to the 1st COVID19 Severity Detection Challenge. The ensemble predictions are marked with a †. Predictions marked with 五 are calculated only with the 5th model of the 5-fold cross-validation. Usage of the random-orientation augmentation is denoted with *ROr*. F1 scores are macro F1 scores.

Submission #	Pretraining	F1 cross validation	F1 test	F1 validation
1	ImageNet (Full)	65.67	$46.67^{五}$	$67.21^{五}$ $61.28^{†}$
2	Segmentation	67.25	$49.36^{五}$	$63.43^{五}$ $61.28^{†}$
3	Segmia	66.48	$\mathbf{51.48^{五}}$	$60.89^{五}$ $63.05^{†}$
4	Multitask	68.18	$46.01^{五}$	$55.51^{五}$ $58.77^{†}$
5	Segmentation ROr	71.74	$49.90^{五}$	$60.02^{五}$ $62.68^{†}$

We added an example for a correctly classified and an incorrectly classified CT-scan in Fig. 3.

2nd COVID19 Detection Challenge: In addition to severity-detection, we used our architecture and training-pipeline also to train our model for the task of infection detection and participated in the *2nd COVID19 Detection challenge*. The ranking for the winning teams is depicted in Table 6. Because it was also possible to submit up to 5 solutions, our submissions can be seen in Table 7. We achieve the best results without cross validation using the *segmentation model*. This is probably due to the coding mistake mentioned above as this model is trained with 100% of the training data in contrast to 80%. Nearly the same performance is achieved using the (fifth) *multitask model*, thus indicating that the multitask pretraining is a good choice for infection detection as well. As submission 2 to 5 are considerably better than submission 1, we conclude that our custom pretrainings improve the results in contrast to the ImageNet model for the infection-detection task, too.

Table 6. Comparison of the best submissions of the winning teams in the 2nd COVID19 Detection Challenge. Our prediction is calculated only with the 5th model of the 5-fold cross-validation. F1 scores are macro F1 scores.

Team	F1 test	F1 test non-COVID	F1 test COVID
1st: ACVLab	89.11	97.45	80.78
1st: FDVTS	89.11	97.31	80.92
2nd: MDAP	87.87	96.95	78.80
3rd: Ours	86.18	96.37	76.00

Correctly Classified   **Underestimated**
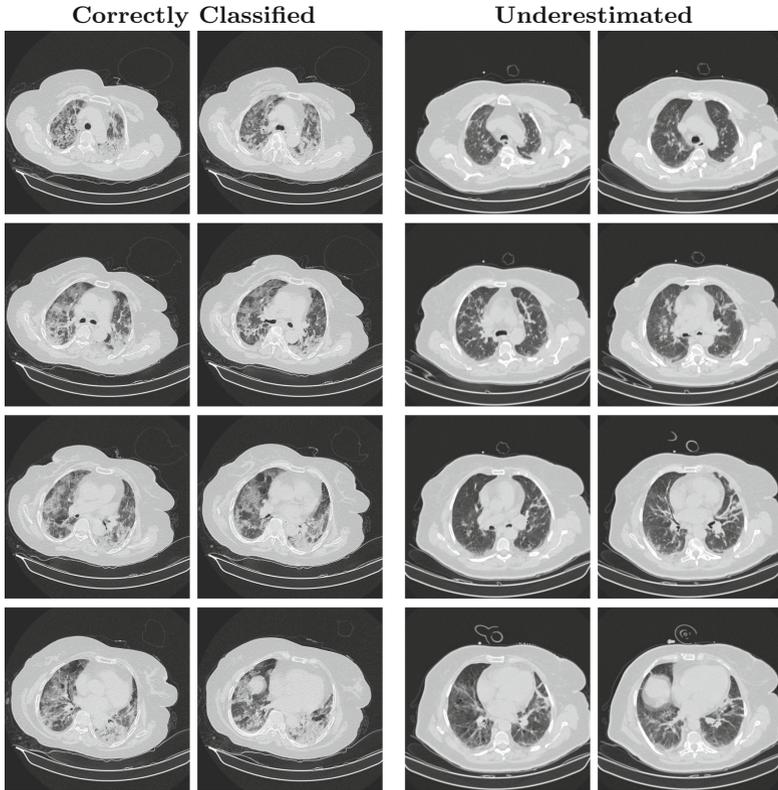
Fig. 3. Qualitative example slices for the severity prediction. The CT scan on the *left* has been correctly identified to display a patient that is in a critical state. The CT scan on the *right* has been predicted to be in a moderate state although the correct prediction would have been a severe state.

Furthermore, by analyzing submission 1 to 3, we deduce that the cross-validation results are good estimates for the models performance as the order of the scores matches the test-set scores. Moreover, since the gap between cross-validation metrics and test-set metrics is considerably smaller than for the severity prediction task, we reason that the dataset statistics of the train-set and the test-set are much more similar for the infection-detection task. We guess that the statistics are more similar in this challenge because the dataset size is substantially larger and, consequently, we emphasize the need to use larger datasets in order to get valid performance estimates for clinical usage.

Table 7. Our submissions to the 2nd COVID19 Detection Challenge. F1 scores are macro F1 scores. The * denotes that no cross validation was used. The ensemble predictions are marked with a †. Predictions marked with 五 are calculated only with the 5th model of the 5-fold cross-validation. Usage of the random-orientation augmentation is denoted with *ROr*.

Submission #	Pretraining	F1 cross validation	F1 test	F1 validation
1	ImageNet (full)	91.73	$82.13^{五}$	$86.60^{五}$ $89.71^{†}$
2	Multitask	93.53	$86.02^{五}$	$87.80^{五}$ $88.79^{†}$
3	Segmia	93.33	$83.63^{五}$	$88.31^{五}$ $89.22^{†}$
4	Segmia ROr*	–	83.93	92.03
5	Segmentation*	–	**86.18**	93.48

5   Conclusion

In this paper, we analyzed various pretraining techniques designed to enhance SARS-CoV-2 severity-prediction performance of our neural network and show that the performance can be significantly increased utilizing segmentation labels and additional datasets. Additionally, we show that our architecture and pretraining pipeline can easily be transferred to the task of infection detection and, thus, our method can be regarded as a general method to enhance COVID-related CT-scan analysis.

The pretraining methods were applied to a three-dimensional ConvNeXt architecture and a finetuning for the COV19-CT-DB dataset was performed. We achieved 2nd rank in the *1st COVID19 Severity Detection Challenge* and 3rd rank in the *2nd COVID19 Detection Challenge*, consequently proving that our method yields competitive results.

In addition to that, we introduced the balanced cross-entropy and argued that this loss-function is important for clinical use cases. We emphasize that our model achieved best results in detecting the most-severe cases.

Altogether, we presented a framework for severity prediction as well as infection detection and achieved good performance by applying this framework to the ConvNeXt architecture. We encourage further research based upon our framework to enhance the diagnosis options in clinical use cases.

References

1. Abdou, M.A.: Literature review: efficient deep neural networks techniques for medical image analysis. Neural Comput. Appl. 1–22 (2022)
2. An, P., et al.: CT images in Covid-19 [data set]. The Cancer Imaging Archive (2020)
3. Anwar, T.: Covid19 diagnosis using autoML from 3D CT scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 503–507, October 2021

4. Arsenos, A., Kollias, D., Kollias, S.: A large imaging database and novel deep neural architecture for Covid-19 diagnosis. In: 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), pp. 1–5. IEEE (2022)

5. Basu, A., Sheikh, K.H., Cuevas, E., Sarkar, R.: Covid-19 detection from CT scans using a two-stage framework. Expert Syst. Appl. **193**, 116377 (2022)

6. Buzug, T.M.: Einführung in die Computertomographie: mathematisch-physikalische Grundlagen der Bildrekonstruktion. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18593-9

7. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733 (2017)

8. Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E.: Local contrastive loss with pseudo-label based self-training for semi-supervised medical image segmentation, December 2021

9. Clark, K., et al.: The cancer imaging archive (TCIA): maintaining and operating a public information repository. J. Digit. Imaging **26**(6), 1045–1057 (2013)

10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)

11. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)

12. Guo, J., et al.: Gluoncv and gluonnlp: deep learning in computer vision and natural language processing. J. Mach. Learn. Res. **21**, 23:1–23:7 (2020)

13. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., Xu, D.: Swin unetr: swin transformers for semantic segmentation of brain tumors in MRI images. arXiv preprint arXiv:2201.01266 (2022)

14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016

15. Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR abs/1606.08415 (2016). https://arxiv.org/abs/1606.08415

16. Hou, J., Xu, J., Feng, R., Zhang, Y., Shan, F., Shi, W.: CMC-Cov19d: contrastive mixup classification for Covid-19 diagnosis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 454–461, October 2021

17. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

18. Kollias, D., Arsenos, A., Kollias, S.: AI-MIA: Covid-19 detection & severity analysis through medical imaging. arXiv preprint arXiv:2206.04732 (2022)

19. Kollias, D., Arsenos, A., Soukissian, L., Kollias, S.: MIA-Cov19d: Covid-19 detection through 3-D chest CT image analysis. arXiv preprint arXiv:2106.07524 (2021)

20. Kollias, D., et al.: Deep transparent prediction through latent representation analysis. arXiv preprint arXiv:2009.07044 (2020)

21. Kollias, D., Tagaris, A., Stafylopatis, A., Kollias, S., Tagaris, G.: Deep neural architectures for prediction in healthcare. Complex Intell. Syst. **4**(2), 119–131 (2018)

22. Kollias, D., et al.: Transparent adaptation in deep medical image diagnosis. In: TAILOR, pp. 251–267 (2020)

23. Kopuklu, O., Kose, N., Gunduz, A., Rigoll, G.: Resource efficient 3D convolutional neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
24. Kundu, R., Basak, H., Singh, P.K., Ahmadian, A., Ferrara, M., Sarkar, R.: Fuzzy rank-based fusion of CNN models using Gompertz function for screening Covid-19 CT-scans. Sci. Rep. **11**(1), 1–12 (2021)
25. Liang, S., Zhang, W., Gu, Y.: A hybrid and fast deep learning framework for Covid-19 detection via 3D chest CT images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 508–512, October 2021
26. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
27. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
28. Loh, A., et al.: Supervised transfer learning at scale for medical imaging. arXiv preprint arXiv:2101.05913 (2021)
29. Mhaske, D., Rajeswari, K., Tekade, R.: Deep learning algorithm for classification and prediction of lung cancer using CT scan images. In: 2019 5th International Conference On Computing, Communication, Control and Automation (ICCUBEA), pp. 1–5 (2019)
30. Miron, R., Moisii, C., Dinu, S., Breaban, M.E.: Evaluating volumetric and slice-based approaches for Covid-19 detection in chest CTs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 529–536, October 2021
31. Mukherjee, H., Ghosh, S., Dhar, A., Obaidullah, S.M., Santosh, K., Roy, K.: Deep neural network to detect Covid-19: one architecture for both CT scans and chest x-rays. Appl. Intell. **51**(5), 2777–2789 (2021)
32. Polsinelli, M., Cinque, L., Placidi, G.: A light CNN for detecting Covid-19 from CT scans of the chest. Pattern Recogn. Lett. **140**, 95–100 (2020)
33. Rao, P., Pereira, N.A., Srinivasan, R.: Convolutional neural networks for lung cancer screening in computed tomography (CT) scans. In: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 489–493 (2016)
34. Revel, M.P., et al.: Study of thoracic CT in Covid-19: the stoic project. Radiology (2021)
35. Roth, H.R., et al.: Rapid artificial intelligence solutions in a pandemic-the Covid-19-20 lung CT lesion segmentation challenge. Res. Square (2021)
36. Ruiz, J., Mahmud, M., Modasshir, Md., Shamim Kaiser, M., Alzheimer's Disease Neuroimaging Initiative: 3D DenseNet ensemble in 4-way classification of Alzheimer's disease. In: Mahmud, M., Vassanelli, S., Kaiser, M.S., Zhong, N. (eds.) BI 2020. LNCS (LNAI), vol. 12241, pp. 85–96. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59277-6_8
37. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
38. Seibold, C.M., Reiß, S., Kleesiek, J., Stiefelhagen, R.: Reference-guided pseudo-label generation for medical semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 2171–2179 (2022)

39. Shakeel, P.M., Burhanuddin, M., Desa, M.I.: Automatic lung cancer detection from CT image using improved deep neural network and ensemble classifier. Neural Comput. Appl. 1–14 (2020)
40. Suganyadevi, S., Seethalakshmi, V., Balasamy, K.: A review on deep learning in medical image analysis. Int. J. Multimedia Inf. Retrieval **11**(1), 19–38 (2022)
41. Tan, W., Liu, J.: A 3D CNN network with BERT for automatic Covid-19 diagnosis from CT-scan images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 439–445, October 2021
42. Tang, Y., et al.: Self-supervised pre-training of swin transformers for 3D medical image analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20730–20740 (2022)
43. Wang, J., Zhu, H., Wang, S.H., Zhang, Y.D.: A review of deep learning on medical image analysis. Mob. Netw. Appl. **26**(1), 351–380 (2021)
44. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 432–448. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_26
45. Zhang, L., Wen, Y.: A transformer-based framework for automatic Covid19 diagnosis in chest CTs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 513–518, October 2021