BetterCOVIDSD-Net: Improving COVID-19 Severity Classification through Chest X-ray

Vuong Ho
University of Rochester
vho7@u.rochester.edu

Minh Khoi Nguyen Do
University of Rochester
mnguyend@u.rochester.edu

Phuc Lam
University of Rochester
plam6@u.rochester.edu

Abstract

As the COVID-19 pandemic is affecting many parts of the world, many studies have also been rapidly conducted to combat this virus in any way possible. One way computer vision plays into this is by developing classification system to detect COVID-19 in patient through chest X-ray. Even though many models have been developed to do a binary classification of whether a patient is contracted or not, few has been created to classify his or her severity level, as thus the accuracy for this task have been low. This project aims to improve upon a previous model COVID-SDNet that achieves this COVID-19 severity level classification task. With an accuracy of 61.80% ± 5.49% *in detecting Mild COVID-19 severity level by COVID-SDNet on the COVIDGR-1.0 dataset, there is definitely room of improvement here. We replace the original base Resnet-50 with other 6 CNN base including EfficientNet B1, EfficientNet B7, GoogLeNet, VGG-16, Resnet-101 to observe the changes in per-class accuracy. We found that EfficientNet B1 works best in place of Resnet-50 in terms of runtime and accuracy. Our improved model, called BetterCOVID-SDNet, using EfficientNet B1 as the classifier, achieves an overall accuracy of 71.97%, with 30.24%, 69.18%, 89.74%, and 98.68% for Normal-PCR+, Mild, Moderate, and Severe COVID-19 severity level, compared to the baseline 28.77%, 60.37%, 85.69%, and 98.02%.*

1. Background and related work

1.1. COVID-19 detection

The COVID-19 pandemic has been a global threat since its discovery in late 2019, but thanks to strict preventative measures and rapidly increasing vaccination rate, the COVID-19 effect has somewhat been mediated. However, work must continue to combat the rate of hospitalization by COVID-19. Many computer vision-focus paper has been published that has developed models for the task of detecting COVID-19 in chest X-ray images of affected patients. Rahaman et al. (2020) compared 15 different pre-trained CNN models like VGG16, ResNet-50, Xception, and MobileNet for this COVID-19 detction task and found that VGG-19 obtains the highest accuracy of 89.3% [8]. Chowdhury et al. (2020) developed ECOVNet, a CNN based on EfficientNet, which is reported to have a classification accuracy of 97% accuracy [3]. Wang et al. (2021) developed a 4-class (COVID-19, Normal, Viral pneumonia, Bacterial pneumonia) classification model MCFF-Net based on PCAF system that achieves highest classification accuracy of 94.66%, with COVID-19 detection rate of 100% [12]. Garzón et al. (2021) made use of VGG19 and U-Net to build a binary classification model that determines positive or negative COVID-19 [11]. The model makes use of lung segmentation technique to remove unrelated information from the chest X-ray images and achieves a detection accuracy of COVID-19 around 97%.

1.2. COVID-19 severity classification

Despite the multitude of research in COVID-19 detection, there has been very few papers and datasets that tackle the problem of classifying severity level of COVID-19. Tabik et al. (2020) is the first paper to address this multi-classification problem with COVID-SDNet [11]. The model they proposed is comprised of a Resnet-50 based initialized with ImageNet weights for transfer learning. However, the classification accuracy for severity levels has proven to be underwhelming, with the accuracy of only 97.72%±0.95%, 86.90%±3.20%, 61.80±5.49% in severe, moderate and mild COVID-19 severity levels. Subsequent papers on this topic like Monaco et al. (2020), Yasin and Gouda (2020), Singh et al. (2021), Setiawati et al. (2021) examines ways to implement a severity levels scoring system for chest X-ray images, but none actually proposed an improved deep learning models to increase the previous accuracy of COVID-SDNet [6, 7, 10, 13].

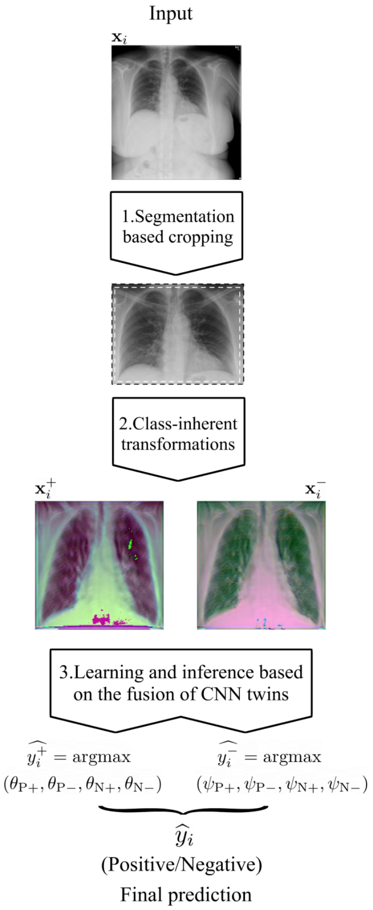

Input
$\mathbf{x}_i$
1.Segmentation based cropping
2.Class-inherent transformations
$\mathbf{x}_i^+$
$\mathbf{x}_i^-$
3.Learning and inference based on the fusion of CNN twins
$\widehat{y_i^+} = \mathrm{argmax}\ (\theta_{\mathrm{P}+}, \theta_{\mathrm{P}-}, \theta_{\mathrm{N}+}, \theta_{\mathrm{N}-})$
$\widehat{y_i^-} = \mathrm{argmax}\ (\psi_{\mathrm{P}+}, \psi_{\mathrm{P}-}, \psi_{\mathrm{N}+}, \psi_{\mathrm{N}-})$
$\widehat{y}_i$
(Positive/Negative)
Final prediction

Figure 1. Pipeline of the COVID-SDNet methodology, image provided in Tabik et al. (2020)

S (Severity level)	accuracy (S) (%)
Normal-PCR+	28.42 ± 2.58
Mild	61.80 ± 5.49
Moderate	86.90 ± 3.20
Severe	$97.720.95$

Table 1. Results of COVID-SDNet by severity level, as reported in Tabik et al. (2020)

2. Method

2.1. Previous approach

As we have mentioned above, we are looking to improve the COVID-SDNet to classify COVID severity by Tabik et al. (2020). Here we give a summary of this model composition and their performance on the COVIDGR-1.0 dataset, which are severity labelled X-ray datasets reported in the same paper. The pipeline of the COVID-SDNet is provided in Fig. 1.

Before the data is fed into the classifier, it was preprocessed to remove unnecessary information. The paper uses a UNet segmentation-based croppping approach to find the mask of the lung and crop the image down using this technique. Due to the rather small and unbalanced dataset, instead of feeding them directly to classifier networks, the authors instead make use of a GAN-based network called FuCiTNet (Rey-Area et al. (2020)) to increase the discriminability of the dataset with learned class-inherent transformations. The CNN classifier was Resnet-50 model pretrained on ImageNet weight. Again, due to the unbalanced dataset, the COVID-SDNet method binary classified first between positive and negative cases, and within the positive cases it further classified into 4 classes of severity. The final stage is swapped out for the appropriate classification type when needed. The performance of COVID-SDNet is reported in the paper in Tab. 1. As observed, the accuracy of the severity class Normal-PCR+ and Mild is quite low, and so in this final project we look to improve COVID-SDNet performance on these severity classes.

Class	Severity	#images	women	men
Negative		426	239	187
COVID-19		426	190	236
	Normal-PCR+	76		
	Mild	100		
	Moderate	171		
	Severe	79		

Table 2. A brief summary of COVIDGR-1.0 dataset, provided in Tabik et al. (2020)

2.2. Dataset

Before we explained our approach, it might be more fruitful to first introduce the dataset in which our model will train on. We will be working with COVIDGR-1.0, a lung X-ray images dataset annotated with COVID-19 severity labels. This dataset came in the same paper by the COVID-SDNet was published from. Since we are improving this model, it makes sense to use the same dataset. Tab. 2 explains the makeup of the COVIDGR-1.0 dataset. The labels was hand-made by four highly trained radiologists from Hospital Universitario Clínico San Cecilio, Granada, Spain.

From observing the summary of the dataset, we notice that the X-ray with severity label are somewhat unbalanced. Images labelled Moderate (171 images) are more than double those labelled Normal-PCR+ (76) and Severe (79). Also, it was impossible to train on all 5 classes, as we planned originally, since the Negative labels are much more than all the other labels. This was the reason we separate this severity prediction to two stages: binary classification of Negative and Positive, and 4-class classification of the severity labels.

2.3. Our approach

From observing the make-up of the pipeline of COVID-SDNet, we speculated that one way that might need improvement on in this network is in the CNN classification part. Due to the small and unbalanced dataset, we expect that there might not be much we can do to improve the class-inherent transformations nor the segmentation based cropping part, and so we focused our efforts on tweaking the CNN classification part. The Resnet-50 network's accuracy, as explored and tested by Bianco et al. (2018) [2], has its Top-1 and Top-5 accuracy (76% and 93%) and relatively low compared to other CNN, such as Resnet-101 (77% and 93.5%), Inception-v4 (80% and 94%), DenseNet-201 (77% and 93.5%) among others. In this project, we experimented with using different CNN classification model in place of the Resnet-50 and compare their accuracy to select one that achieves a higher performance for classifying severity level. Here we focused on only increasing the accuracy of the model for the dataset, so speed of the model will not be taken into account in the comparison, though there will be comments on this aspect.

2.4. Our pipeline

Our pipeline is similar to the that of COVID-SDNet, but we add another stage to it. Instead of classifying either severity or negative and positive in the previously last stage, we decide to dedicate this stage to binary classify negative and positive, and the X-ray images classified as Covid-19 positive will be further classified into 4 severity levels. Fig. 3 shows the diagram of our pipeline. The following sections detail our design choice for each stage.

2.4.1 Segmentation based cropping

We initially tried to fit the uncropped data into the a classification network, but the result was not very good. Taking a look at the dataset, we notice that the X-ray images themselves contain a lot of unnecessary information, for example, other body parts such as arms, neck, stomach (see Figs. 4a and 4b which take up the majority of the X-ray images, different X-ray equipments artifacts (see Figs. 4b and 4c, and cropping errors (see Fig. 4c) which leave a lot of black spots in the X-ray. As suggested in Tabik et al. (2020), we decide to crop these X-ray images by first segmenting the lung using a U-Net segmentation model pretrained on Tuberculosis Chest X-ray Image datasets and RSNA Pneumonia CXR challenge dataset, then finding the smallest bounding box around the masks, add 2.5% pixels to each side, then cropping it [4, 5, 11]. The green bounding box in Fig. 4 closely resembles what the results of the cropping look like. All images are resize to 512 × 512 before cropping (which is about 2-2.5 times smaller than the uncropped images); however, this should not affect the training.

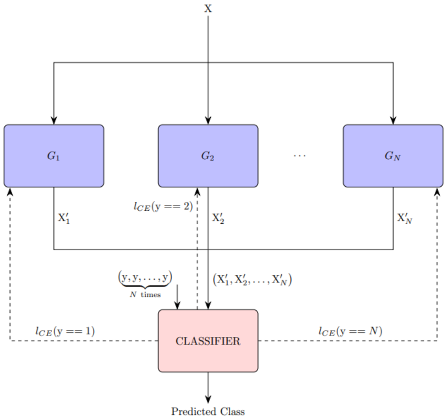

X
G_1
G_2
$\cdots$
G_N
$l_{CE}(\mathrm{y} == 2)$
X'_1
X'_2
X'_N
$\underbrace{(\mathrm{y}, \mathrm{y}, \ldots, \mathrm{y})}_{N \text{ times}}$
$(\mathrm{X}'_1, \mathrm{X}'_2, \ldots, \mathrm{X}'_N)$
$l_{CE}(\mathrm{y} == 1)$
CLASSIFIER
$l_{CE}(\mathrm{y} == N)$
Predicted Class

Figure 2. Flowchart of FuCiTNet during training. The input image X is broadcast to every generator, which it then produces a transformed image X'. The classifier computes the cross-entropy loss which is transferred back to the generator commissioned to enhance features of the class given by the input's ground truth label y. Image and caption are from Rey-Area et al. (2020).

2.4.2 Class-inherent transformations network

In order to increase the discriminability of the small dataset, especially when classifying severity labels, we feed the image through FuCiTNet, a GAN-inspired class inherent transformations network (see Fig. 2).

The number of generators are decided by the number of classes of the classification task. For the binary classification, we feed the images into the two generators and then put those transformed images to the CNN classifier. Due to the different design, FuCiTNet uses a mix of Mean Squared Error and classifier loss. According to Rey-Area (2020) [9], the loss function of each generator are defined as follows:

$$\mathcal{L}_{gen_k} = l_{MSE} + 0.006 \cdot l_{Perceptual} + \lambda l_{CE}(y == k)$$

where:

$\mathcal{L}_{gen_k}$: loss function of generator k

l_{MSE}: pixel-wise MSE loss

$l_{Perceptual}$: perception MSE loss

λ: weighted factor to change outcome of generator

l_{CE}: classifier loss

The Python code for FuCiTNet are taken from [5] and adjusted for our use.

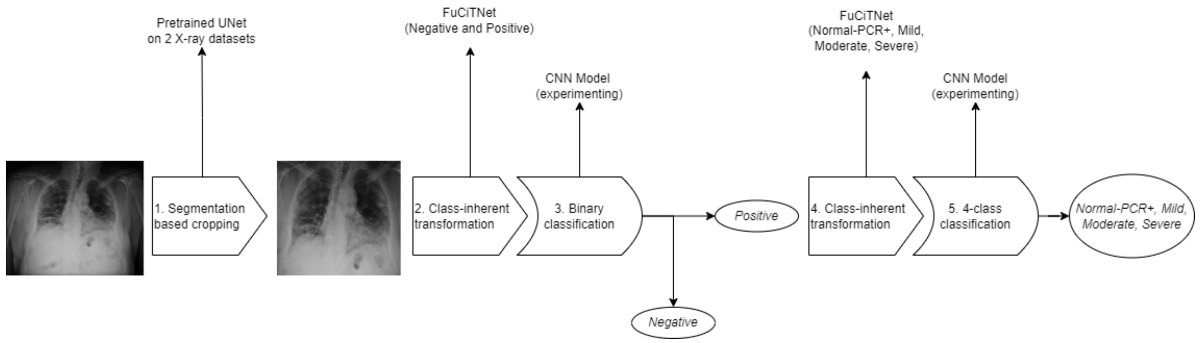
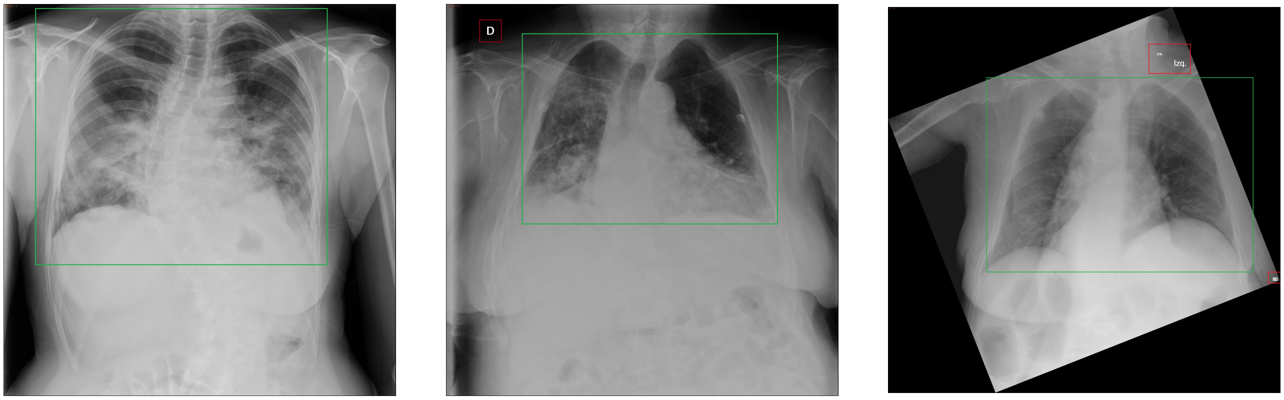


Figure 3. The diagram of the pipeline of our BetterCOVID-SDNet, an improvement over the original COVID-SDNet. We try to improve the accuracy of the classification stage (stage 3 and 5 on the diagram) by experimenting with other CNN models. Due to the small dataset and low accuracy when feeding the X-ray images directly to the classification network, we followed the original COVID-SDNet approach and used the generator-based FuCiTNeT to increase the discriminability of the data before classification.



(a) X-ray with unnecessary information

(b) Unnecessary informations and artifacts

(c) Cropping erros and artifacts

Figure 4. Uncropped X-ray images from the COVIDGR-1.0. The part in the green bounding box is the most important part of a lung X-ray image, and anything outside of that (arms, stomach, neck, etc.) is considered unnecessary information. The parts marked by the red bounding box (Figs. 4b and 4c) are artifacts by the X-ray machine. All these extra parts in the X-ray might heavily affect the performance of the classification model.

2.4.3 Image classification

Here are where most of our experiments will be conducted on. Originally, Tabik et al. (2020) uses Resnet-50, pretrained with ImageNet dataset, as the classifier [11]. As it turns out, there might be other CNN model that might yield a higher accuracy, as explained in the first section of our approach. Here are the lists of the classifiers we experimented with:

- Resnet-50 (base)
- Resnet-101
- VGG-16
- EfficientNet B1
- EfficientNet B7
- GoogLeNet

We do not specifically choose those that are shown to perform better than Resnet-50 by Bianco et al. (2018), but any models that we deem fit to compare the performance against. All of these models are pretrained on ImageNet and available by PyTorch [1]. Most of the architecture of the classifiers remain the same, except for the last fully-connected layer, where we swap it out for our own fully-connected layer with output 2 or 4 to make the models produce suitable results for the classification task.

3. Experiments and results

In this section we provide the details in which we setup our experimental environment for testing the BetterCOVID-SDNet pipeline, displays the results we have from testing with the above classifiers, and the discussions of the impacts.

3.1. Experimental setup

There are two periods to our model training. We first try to classify without using the class-inherent transformations network FuCiTNet to gauge the performance, and then we use FuCiTNet in our pipeline. Since the experimental setup for each of the period are different, we will report it separately in the following section.

3.1.1 Without FuCiTNet

For the period without the class-inherent transformations, we try both version of the dataset, that is, with and without cropping, to see how the removing the unnecessary information affects the performance. The pretrained classifiers are from PyTorch distribution, which are pretrained on ImageNet. We then swapped the final FC layer with our 3 more FC layers, with the last layer having an output of either 2 or 4, using Softmax as our activation function. We use a train-test split of 70-30 and a batch size of 64. We use SGD optimizer with a learning rate of 0.01 and cross entropy loss. In subsequent experiments, we also add an exponential learning rate with gamma 0.9 and a weight decay of 0.01 since we detected overfitting. We run for a total of 20 epochs each training, or until we observe that the accuracy on the test set has converged.

3.1.2 With FuCiTNet

For the period with the class-inherent transformations, we use the FuCiTNet code that comes with Rey-Area et al. (2020) [9]. We swap the classifier models by directly modifying the code. Same as the previous period, we use a train-test split of 70-30 and a batch size of 64. We use the default lambda class parameter, which is 1. We run for a total of 20 epochs each training, or until we observe that the accuracy on the test set has converged. The loss function is as described above.

3.2. Results and Analysis

In this section, we will present the result we receive from our experiments with our proposed pipeline. Due to the difference between the setup of the first and second period, we will be displaying the result separately. We also take a look at per-class severity classification accuracy. Similar in Tabik et al. (2020), we have also experimented with removing Normal-PCR+ and see how the accuracy among the severity labels change [11]. Observing that EfficientNet B1 and B7 yield the highest accuracy of the 5 extra models, we decide to only experiment with these two when we try removing Normal-PCR+ label.

3.2.1 Without FuCiTNet

The results of both Positive/Negative and severity classification is shown in Tab. 3a. For negative/positive classification, EfficientNet B7 seems to produce the highest accuracy among the 6 classifiers, 3.78% higher than the baseline Resnet-50 with cropped data. As expected, VGG-16 performs the weakest, with an accuracy of 61.33% on uncropped data and 63.86% on cropped data. For severity labels, we can clearly see the effect of the small and unbalanced dataset to the performance of the CNN classifiers. Accuracy among all 6 models average around 43.49% with uncropped data and 44.41% with cropped data, showing how little the cropping do to help with this type of dataset. Again, EfficientNet B1 and B7 perform the best compared to the other models, with B1 with a slightly higher accuracy than B7.

3.2.2 With FuCiTNet

The results of both Positive/Negative and severity classification is shown in Tab. 3b. As shown in the accuracy table, the effects of class-inherent transformation are obvious for this dataset. For negative/positive classification, the average accuracy increases from 72.68% to 78.72% with uncropped data (6.04% increase), and from 75.22% to 81.1% with cropped data (5.88% increase). This proves the usefulness of FuCiTNet for the COVID-GR1.0 dataset. The EfficientNet model still achieves the highest overall accuracy. The same applies to severity classification, where we can see the increase in the average accuracy when using class-inherent transformation network. Overall, we conclude that the most effective replacement for Resnet-50 is EfficientNet B1 or EfficientNet B7, as it has raised the overall accuracy for both classification task. We are also interested in seeing this affects the accuracy of the Mild and Normal-PCR+ label.

3.2.3 Per-class accuracy

The results of per-class accuracy is displayed in Tab. 4. Compared to the baseline Resnet-50, the two EfficientNet achieves similar accuracy in Moderate and Severe labels, but improves somewhat in both the Normal-PCR+ and Mild labels. As observed in Tab. 4a, regarding the accuracy of Normal-PCR+, EfficientNet B1 accuracy is 1.47% higher than the baseline and EfficientNet B7 is 6.36% higher; regarding the accuracy of Mild, EfficientNet B1 is 8.81% higher than the baseline, while EfficientNet B7 is 13.06% higher. This is a very impressive accuracy, considering that the X-ray images of Normal-PCR+ and Mild is very similar to each other. An interesting thing happens in the Severe accuracy, with accuracy of EfficientNet B7 being lower than

Classifier	Resnet-50	Resnet-101	VGG-16	EfficientNet B1	EfficientNet B7	GoogLeNet
Acc. (uncropped, NvP)	71.52	74.85	61.33	76.53	**77.92**	73.92
Acc. (cropped, NvP)	76.37	77.52	63.86	77.28	**80.15**	76.13
Acc. (uncropped, sev)	40.15	43.78	28.18	50.07	**52.27**	45.93
Acc. (cropped, sev)	41.88	43.26	28.33	**53.69**	53.29	46.03

(a) Table showing the accuracy of BetterCOVID-SDNet without using class-inherent transformation, in percent (%).

Classifier	Resnet-50	Resnet-101	VGG-16	EfficientNet B1	EfficientNet B7	GoogLeNet
Acc. (uncropped, NvP)	78.45	80.00	68.05	85.79	**86.02**	74.02
Acc. (cropped, NvP)	81.37	82.74	70.17	**87.82**	87.49	77.01
Acc. (uncropped, sev)	66.07	66.04	36.70	**70.02**	69.34	65.89
Acc. (cropped, sev)	68.25	69.93	38.97	72.08	**73.45**	68.67

(b) Table showing the accuracy of BetterCOVID-SDNet with using class-inherent transformation, in percent (%).

Table 3. Tables showing the accuracy of experiment with and without using class-inherent transformation. ”Uncropped” or ”cropped” denotes the status of the images that were used. ”NvP” means the positive/negative classification, while ”sev” means severity classification. The bold value means that it is the highest accuracy within the respective classification tasks.

Classifier	Resnet-50	EfficientNet B1	EfficientNet B7
Normal-PCR+	28.77	30.24	**35.13**
Mild	60.37	69.18	**73.43**
Moderate	85.69	89.74	**90.34**
Severe	98.02	**98.68**	97.66
Overall	68.21	71.97	**74.14**

(a) Table showing the accuracy of individual severity, including Normal-PCR+, in percent (%).

Classifier	Resnet-50	EfficientNet B1	EfficientNet B7
Mild	52.43	**54.18**	52.26
Moderate	84.63	**87.21**	85.33
Severe	97.58	96.35	**98.06**
Overall	71.54	**79.24**	78.21

(b) Table showing the accuracy of individual severity, excluding Normal-PCR+, in percent (%).

Table 4. Tables showing the individual accuracy of experiment with severity classification using class-inherent transformation, including and excluding Normal-PCR+. EfficientNet B1 and B7 are chosen to be shown here since they are two classifiers with the highest accuracy in preliminary testing. The bold value means that it is the highest accuracy within the respective severity label.

the baseline. Although this might be a one-time off, there might be some implications that can be made from this.

We are also interested in seeing the effect of removing Normal-PCR+ from the dataset on the accuracy of the other labels. The results of this experiment is shown in Tab. 4b. Similar to what Tabik et al. (2020) shows, when removing Normal-PCR+ from the images pool, the accuracy for Mild drops about 14% and the accuracy for Moderate drops about 1% [11]. One interesting observation is that EfficientNet B1 seems to outperform EfficientNet B7 when Normal-PCR+ is removed.

4. Conclusion

The results of the experiment shows that EfficientNet B1 and B7 improves quite a bit on the COVID-SDNet. Regarding the accuracy of Normal-PCR+, EfficientNet B1 accuracy is 1.47% higher than the baseline and EfficientNet B7 is 6.36% higher; regarding the accuracy of Mild, EfficientNet B1 is 8.81 higher than the baseline, while EfficientNet B7 is 13.06% higher. However, one thing to note about using EfficientNet B7 is that its parameters is much more than all the other models we looked at, so the training time might be very slow. Those who are looking to improve their image classificatoin pipeline should weigh the tradeoff between accuracy and run time to choose the best classifiers. That being said, we do not necessarily have to use EfficientNet B7. EfficientNet B1 achieves similar accuracy to B7, but its parameters are much smaller, allowing for faster training time. Overall, from the result of this experiments, to improve upon COVID-SDNet, we suggests using EfficientNet B1 to increase the COVID-19 severity classification.

References

[1] Models and pre-trained weights. 4

[2] Simone Bianco, Rémi Cadène, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *CoRR*, abs/1810.00736, 2018. 3

[3] Nihad Chowdhury, Ashad Kabir, Md. Muhtadir Rahman, and Noortaz Rezoana. Ecovnet: a highly effective ensemble based deep learning model for detecting covid-19. *PeerJ Computer Science*, 7:e551, 05 2021. 1

[4] Guo R;Passi K;Jain CK;. Tuberculosis diagnostics and localization in chest x-rays via deep learning models. 3

[5] Stefan Jaeger, Sema Candemir, Sameer Antani, Yì-Xiáng J. Wáng, Pu-Xuan Lu, and George Thoma. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery*, 4(6), 2014. 3

[6] Chris Kawatsu, Frank Koss, Andy Gillies, Aaron Zhao, Jacob Crossman, Benjamin Purman, Dave Stone, and Dawn Dahn. Gesture recognition for robotic control using deep learning. 08 2017. 1

[7] Cristian Monaco, Federico Zaottini, Simone Schiaffino, Alessandro Villa, Gianmarco Pepa, Luca Carbonaro, Laura Menicagli, Andrea Cozzi, Serena Carriero, Francesco Arpaia, Giovanni Di Leo, Davide Astengo, Ilan Rosenberg, and Francesco Sardanelli. Chest x-ray severity score in covid-19 patients on emergency department admission: a two-centre study. *European Radiology Experimental*, 4, 12 2020. 1

[8] Md Rahaman, Chen Li, Yudong Yao, Frank Kulwa, Mohammad Rahman, Qian Wang, Shouliang Qi, Fanjie Kong, Xuemin Zhu, and Xin Zhao. Identification of covid-19 samples from chest x-ray images using deep learning: A comparison of transfer learning approaches. *Journal of X-ray science and technology*, 28, 07 2020. 1

[9] Manuel Rey-Area, Emilio Guirado, Siham Tabik, and Javier Ruiz Hidalgo. Fucitnet: Improving the generalization of deep learning networks by the fusion of learned class-inherent transformations. *CoRR*, abs/2005.08235, 2020. 3, 5

[10] Aparajita Singh, Yoke Hong Lim, Rajesh Annamalaisamy, Shyam Sunder Koteyar, Suresh Chandran, Avinash Kumar Kanodia, and Navin Khanna. Chest x-ray scoring as a predictor of covid-19 disease; correlation with comorbidities and in-hospital mortality. *Scottish Medical Journal*, 66(3):101–107, 2021. PMID: 34176342. 1

[11] Siham Tabik, Anabel Gómez-Ríos, J. Martin-Rodriguez, I. Sevillano-Garcia, Manuel Rey-Area, David Charte, Emilio Guirado, J. Suarez, Julián Luengo, M. Valero-Gonzalez, P. Garcia-Villanova, E. Olmedo-Sanchez, and Francisco Herrera. Covidgr dataset and covid-sdnet methodology for predicting covid-19 based on chest x-ray images. *IEEE Journal of Biomedical and Health Informatics*, 24:3595–3605, 12 2020. 1, 3, 4, 5, 6

[12] Wei Wang, Yutao Li, Ji Li, Peng Zhang, Xin Wang, and Nian Zhang. Detecting covid-19 in chest x-ray images via mcff-net. *Intell. Neuroscience*, 2021, jan 2021. 1

[13] Rabab Yasin and Walaa Gouda. Chest x-ray findings monitoring covid-19 disease course and severity. *Egyptian Journal of Radiology and Nuclear Medicine*, 51, 10 2020. 1