Exercise Project, TKO_2027-3001

Classifying Healthy, Pneumonia and Covid-19 cases based on X-ray images with CNN

A work report

Oona Leppänen
Information and Communication Technology,
Data Analytics (Tech)
1800509
oklepp@utu.fi

Instructor: Tapio Pahikkala

Table of Contents

Description and Analysis of the Task

The chosen task to implement in this project was to create an Artificial Intelligence (AI) solution to differentiate Covid-19 patients from healthy and pneumonia patients based on human chest X-ray images. Convolutional neural networks (CNN) were chosen to be the AI model to carry out the task. The idea was to train the first version of the model with X-ray images only taken from chests of healthy and pneumonia patients [1]. The second version of the model was supposed to be trained with all patients: healthy, pneumonia and covid-19 patients [7]. Because of the small size of the dataset having the covid-19 patients (1823 images) a data augmentation method was planned to use to ensure better learning of the second model. Later the augmentation was expanded to concern also the dataset without covid-19 patients because this dataset is quite small as well (4192 images). The dataset with covid-19 patients was noticed to partially overlap with the dataset without covid-19 patients so a better dataset was created based on that dataset and another one with covid-19 patients [6].

The dataset without covid-19 patients will be referred as Pneumonia dataset [1] and the dataset with covid-19 patients will be referred as Covid-19 dataset from now on. The Pneumonia dataset (named Pneumonia X-Ray Images in Kaggle) is available in Kaggle with CC BY 4.0 license [3] by Paulo Breviglieri. This dataset consists of X-ray images of healthy chests and chests with pneumonia. The dataset has 5856 images divided into training, validation and test sets with 4192, 1040 and 624 images, respectively. The training set consists of 1082 healthy cases and 3110 pneumonia cases, the validation set consists of 267 healthy and 773 pneumonia cases and the test set consists of 234 healthy and 390 pneumonia cases. All images are jpeg formatted grayscale images with different sizes.

The Covid-19 dataset consists of mostly from Covid-19 Image Dataset [6] which is available in Kaggle with CC BY-SA 4.0 license [4] by Pranav Raikote enabled by University of Montreal. The Covid-19 dataset also consists of covid images in COVID CXR Image Dataset [7] with CC0 license [2] by Manu Siddhartha. Covid-19 Image Dataset has 317 images divided into training set and test set with 251 and 66 images, respectively. The training set consists of 70 healthy cases, 70 pneumonia cases and 111 covid cases and the test set consists of 20 healthy cases, 20 pneumonia cases and 26 covid cases. COVID CXR Image Dataset has 1823 images total but only 536 images of them are covid cases and are therefore used in the project. The healthy and pneumonia images in that dataset overlap with the Pneumonia dataset and therefore can't be used in this project. All images in both datasets are grayscale images with different sizes and have different formats for example jpeg, png and jpg.

The Pneumonia and Covid-19 datasets are used to train the model. Part of the Covid-19 dataset is not used to train the model but evaluate it and test the correctness of the prediction capacity of it. The model outputs predictions for images it gets as inputs. Therefore, outcome of this project is a model that can predict the right condition of a patient in an input image. User can use the model by giving an X-ray image of either healthy patient or a patient with pneumonia or covid-19 to the model and the model gives a prediction that can be "Healthy", "Pneumonia" or "Covid-19".

However, the user must go and change the file path in the end of the code to get their image to be predicted. They also must read the prediction from a print the code provides. In other words, the code does not have any kind of user interface (UI) to use. To extend the project work a proper UI should be done to ease the usage of the model. The model should be also tested with another set of X-ray images containing covid-19 patients to ensure that the model works well with similar data and has not learn the data used to train the model too well. If that would be the case, then the model should be trained a bit more or different to counter the problem. Also, the accuracy of the model should be able to be raised higher [5] with a bit more complex model so that should be considered to carry out in the future as well.

Solution Principles

The focus of the solution is on building and fine-tuning a CNN model to differentiate healthy, pneumonia and covid-19 X-ray images from the chests of human patients. The solution consists of several parts: creating the Covid-19 dataset based on Covid-19 Image Dataset and COVID CXR Image Dataset, loading the data, building and training a base model with image augmentation, fine-tuning the model with image augmentation and evaluating those two models. After all that the created model can make predictions.

As mentioned, a Covid-19 dataset is created based on Covid-19 Image Dataset (project files use the name *Covid19-dataset*) and COVID CXR Image Dataset (Research) (*COVID_IEEE* in project files). Because the pneumonia and healthy case images in COVID CXR Image Dataset are taken from Pneumonia dataset they can't be used in the fine-tuning phase to train, validate or test the model. That's why only the covid-19 cases are used from that dataset by combining them into the Covid-19 Image Dataset.

Validation set isn't built-in to the Covid-19 Image Dataset so the training set must be split into training set and validation set. Then the covid images of the COVID CXR Image Dataset are divided into training set, validation set and test set. Both training sets, validation sets, and test sets are combined and added into a same new folder called *Covid-19_dataset*. All this happens in a separate ipynb file called *Ex_project_forming_proper_Covid19_dataset*.

After the Covid-19 dataset has been created and the data loaded, the base CNN model is built. Rescaling of the images and the data augmentation are done inside the CNN model before the actual AI solution. The built base model takes the Pneumonia dataset and trains itself based on images and their classes ('healthy', 'pneumonia'). Those classes are the ones the base model tries to learn to differentiate. Because a CNN model can consist of many layers and the layers can vary and their parameters can vary too, different kinds of versions of the model should be tried to find out a good base model. By varying those three things a good version was found based on some metrics and evaluation of the base model. However, in the code only one version has been left – the one which is used as a base model.

Next the model is fine-tuned with Covid-19 dataset. This means that a few layers are added to the base model, those layers are trained with the Covid-19 dataset and as a result the model can also differentiate covid images. The rescaling and data augmentation are also used for the Covid-19 dataset images. Different base models were also tried for fine-tuning to get a good, fine-tuned model.

With the fine-tuned model ready, predictions can be done. First, the test set of the Covid-19 dataset is used for predicting the classes of the test set images to see how well the model can predict. At the end of the file is a place where a user can add their own X-ray images of human chests and the fine-tuned CNN mode predicts a class for the input image.

Description of the Program and Its Components

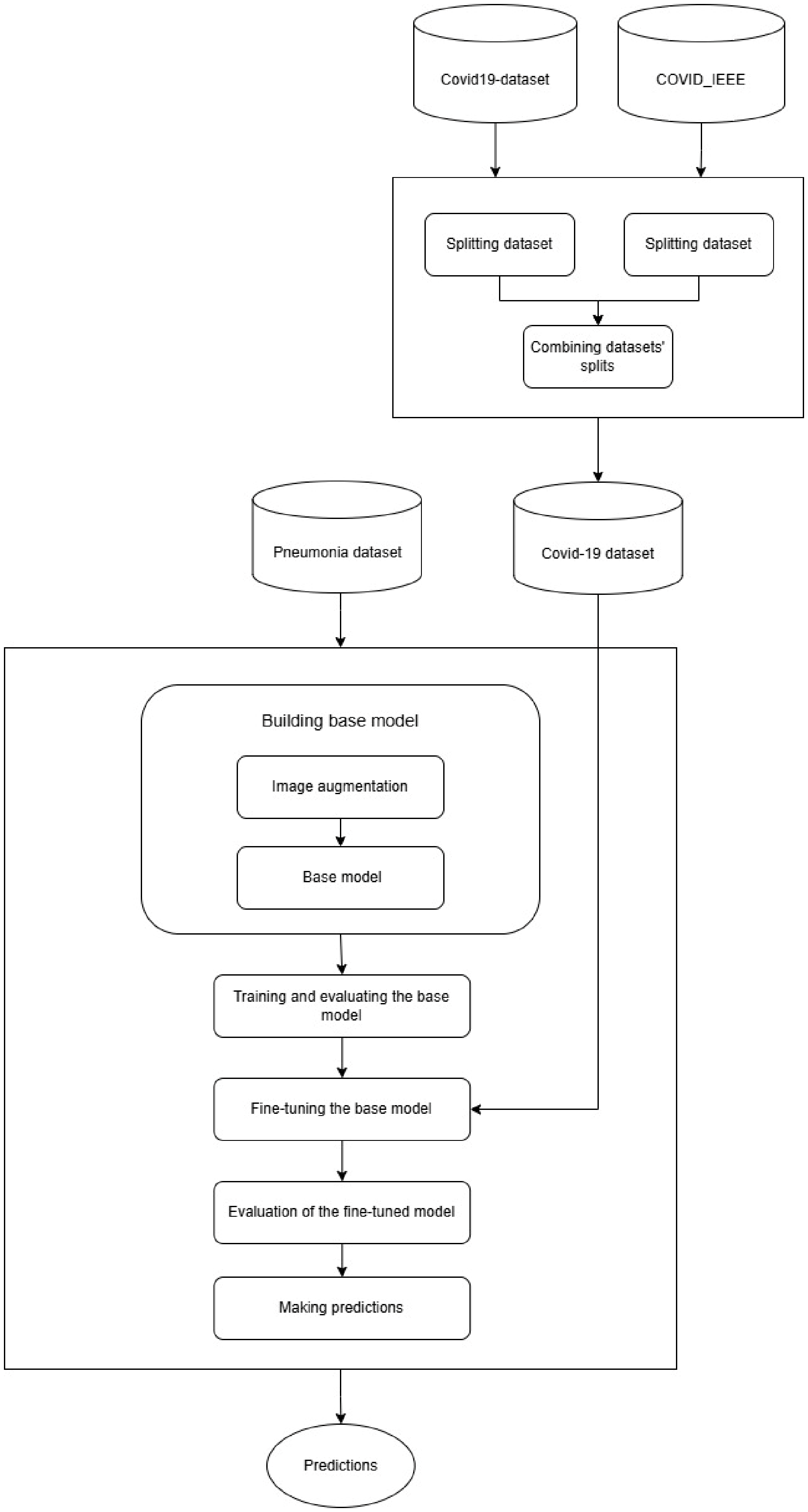


Figure 1. Structure diagram of forming the solution.

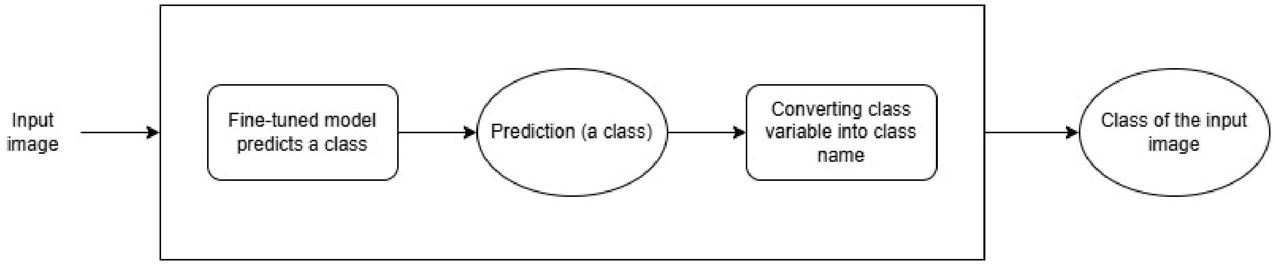

Input image
Fine-tuned model predicts a class
Prediction (a class)
Converting class variable into class name
Class of the input image

Figure 2. Structure diagram of usage of the solution.

Testing Arrangements

In this project it is very important to verify that the loading of the datasets and the models work correctly. To verify the loading of the images the number of images and some of the images with their classes are printed when they have been loaded. To verify that the models work as they should, three different testing routines have been performed during the development of the program. Both models (base model and fine-tuned model) must be evaluated in some way to ensure that the models learn well enough. Accuracy has been used as the metric to evaluate the performance of the models. The best performing version of the model in terms of accuracy is saved and evaluated using the test set of Pneumonia dataset and utilized later as the base for the fine-tuned model.

The best version of the model is tracked using validation accuracy. The models are trained with training data and then evaluated with validation data. The data is from the Pneumonia dataset. The training accuracy shows how well the model performs on known images and the validation accuracy tells how well the model should perform with unseen data. Because of this, validation accuracy is the best way to find out the best version of a model. A rough boundary for a good model that learns well is 90% accuracy or above. In this case the model does not learn if the accuracy is 73.41% or below and performs badly under 85% accuracy. At 73.41% (0.7341) accuracy the model always guesses a class pneumonia. The variation of the accuracy and loss function during training are shown in two figures to help evaluate the developments of the loss and accuracies.

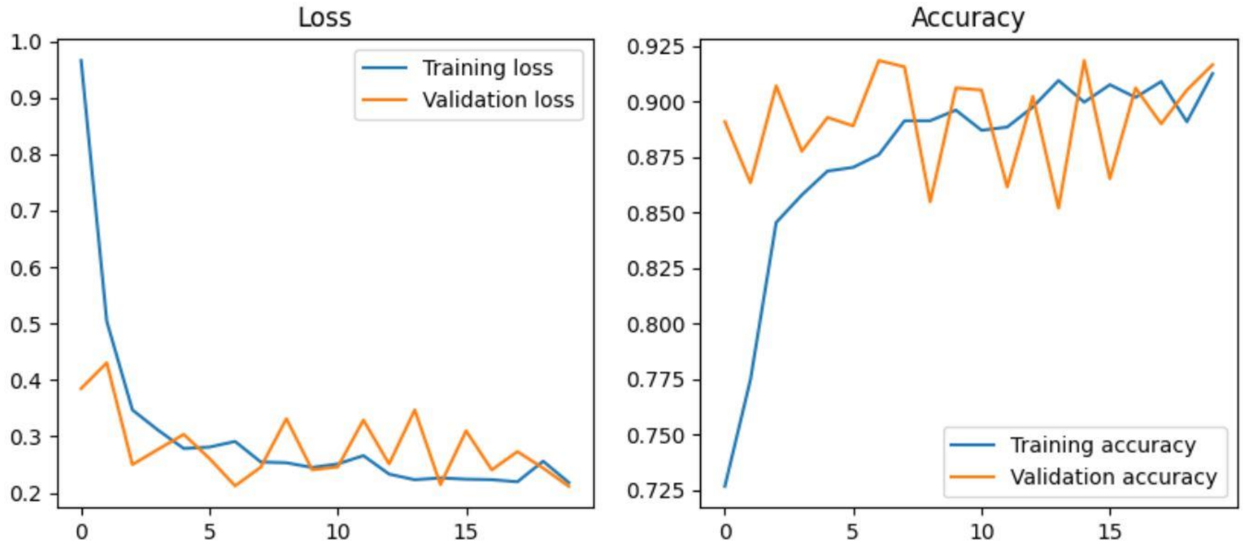


Figure 3. Losses and accuracies of the base model. On the left are the training and validation losses. On the right are the training and validation accuracies.

Because the best version of a model is chosen by accuracy, different base model versions have been tested to find the best validation accuracy among them – and therefore a high evaluation accuracy for the model - but taking into account the largest difference between the best and worst accuracy, the variation of the accuracy and behaviour of the loss function. The base model is varied by changing the number of layers, their parameters and adding different layers. The variations with results can be found in Table 1. However, the exact value of different accuracies varies a bit among different runs. The training and validation accuracies in the Table 1 are not unfortunately the best ones from the tested models but the last accuracies resulted from the training phase. These numbers don't also tell anything about how steadily the loss got down or how steadily the accuracies of each model changed during training and validating.

Filters in convolutional layer	**Number of neurons in dense layer**	**Training accuracy**	**Validation accuracy**	**Evaluation accuracy**
32, 64, 128	100	0.7419	0.7341	0.625
32, 64, 128	500	0.8922	0.9155	0.8494
32, 64, 128	500, 100	0.8712	0.8993	0.8878
32, 32, 64	100	0.8867	0.9022	0.7724
32, 32, 64	500	0.8872	0.9031	0.8622
32, 32, 64, 128	100	0.9008	0.9117	0.8622
32, 32, 64, 128	500	0.8791	0.9145	0.8590
32, 32, 64, 128	500, 100	0.9008	0.8841	0.8750
32, 64, 64	100	0.8903	0.9240	0.8462
32, 64, 64	500	0.7533	0.7341	0.6250

Table 1. Comparing different base model versions by different accuracies. Different layers, their parameters and number of layers were tried in the base model. Here is those variations and the last accuracies of the training accuracy and validation accuracy. The best validation accuracy was not compared here.

As can be seen from the Table 1 the best evaluation accuracy is 0.8878 with pretty good validation accuracy as well. That model has three convolutional layers with filters 32, 64 and 128, respectively, and two dense layers with 500 and 100 neurons showed pretty good stability in validation accuracy and its loss function led to it to be the chosen one for next studies. That model was tested with fine-tuned model. Also, it was altered a little and got a competitor from a model done by Madhav Mathur in Kaggle [5]. The results can be found from Table 2.

Filters in convolutional layer	**Batch normalization layer after every layer?**	**Dropout used?**	**Number of neurons in dense layer**	**Evaluation accuracy of the fine-tuned model**
32, 64, 128	No	No	500, 100	0.78
32, 64, 128	Yes	No	500, 100	0.71
32, 64, 64, 128, 256	Yes	Yes	500, 100	0.71
32, 64, 64, 128, 256	Yes	No	500, 100	0.79

Table 2. Comparing the best base model found out, its variation, a bit more complex model (base of it got from [5]) and a variation of a more complex model.

The top two models according to the evaluation accuracy of the fine-tuned model shown in the Table 2 were tested out with fine-tuned model and compared more comprehensively (Table 3). The test set of Covid-19 dataset was used to evaluate the fine-tuned model. As can be seen from Table 3 the less complex model gets better accuracies through the training and evaluation and was therefore chosen to be the base model to be used.

Metrics	Small model	More complex model
Model structure	CL 32, CL 64, CL 128, Dense 500, Dense 100	CL 32, BN, CL 64, BN, CL 64, BN, CL 128, BN, CL 256, BN, Dense 500, Dense 100
Base model best validation accuracy	0.9383	0.9164
Base model evaluation accuracy with best version of the base model	0.8381	0.7837
Fine-tuned model, best validation accuracy	0.8464	0.7031
Fine-tuned model, evaluation accuracy with best version of the fine-tuned model	0.8283	0.6785

Table 3. Comparing the top two best models. CL = convolutional layer, Dense = dense layer, BN = batch normalization layer.

After all this work the original covid dataset used to train the fine-tuned model (COVID CXR Image Dataset by Manu Siddhartha, [7]) was noticed to contain same pneumonia and healthy case images from the Pneumonia dataset and therefore could not be used during the training or evaluation of the fine-tuned model or testing the predicting quality of the program. However, the results give a direction to go to and so the smaller model was decided to be the base model to build upon.

The last testing routine is to ensure the fine-tuned model (finished model) produces reasonable and meaningful outputs. For that purpose, the test set of Covid-19 dataset has only been used for evaluating the fine-tuned model and making the test predictions. Predictions and correct classes are being printed for an approximate comparison. Also, 30 images with corresponding predictions and true classes of images are being printed. For non-approximate estimations two confusion matrixes are being printed. The first one shows numbers of correctly differentiated classes and numbers of incorrectly differentiated classes by classes (Figure 4). For example, it shows how many of the covid-19 cases are truly differentiated as covid-19 cases, how many of them are differentiated as pneumonia cases and how many as healthy cases. The second confusion matrix includes the same information but expressed in precents (Figure 5). The better the fine-tuned model differentiates classes correctly the better the model is.

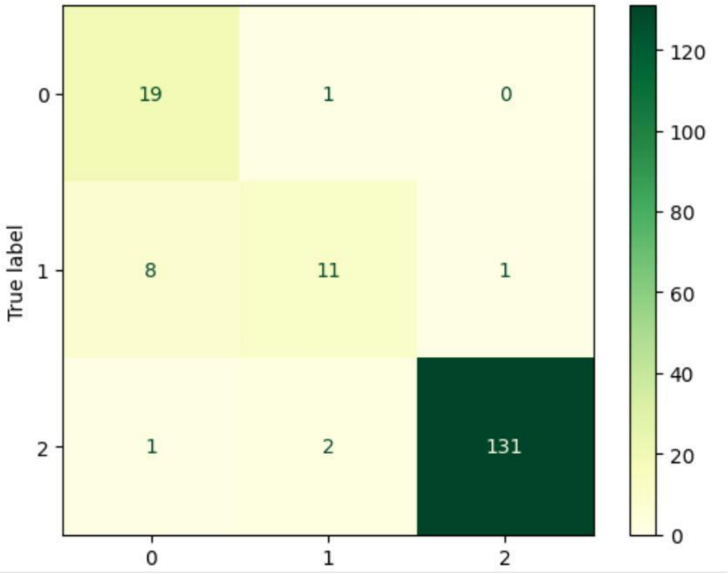
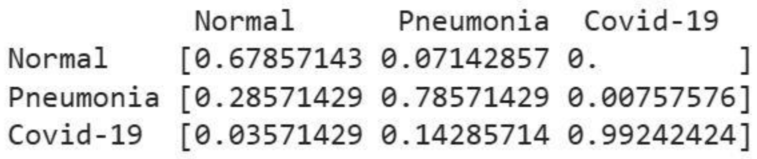


Figure 4. Confusion matrix with numbers of correctly and incorrectly differentiated classes.



```
           Normal      Pneumonia  Covid-19
Normal    [0.67857143 0.07142857 0.        ]
Pneumonia [0.28571429 0.78571429 0.00757576]
Covid-19  [0.03571429 0.14285714 0.99242424]
```

Figure 5. Same confusion matrix as above except the numbers are percentages.

Almost all covid-19 cases are predicted correctly during all runs of the program and the same goes for pneumonia cases. The healthy cases are harder to predict correctly but most of them are always predicted correctly. When the program must predict what is the class of a truly healthy case it mixes the healthy and pneumonia cases often. For example, as can be seen from the Figure 5 the program predicts the true healthy class to be healthy by about 68 percentage of time, pneumonia 29 percentage of time and covid-19 class by 3.6 percentage of time.

References

[1] Breviglieri, Paulo (updated 2020, referred 4.7.2024), Pneumonia X-Ray Images, CC BY 4.0 license (ATTRIBUTION 4.0 INTERNATIONAL), https://www.kaggle.com/datasets/pcbreviglieri/pneumonia-xray-images?resource=download&select=val, Adaptation from ‘Chest X-Ray Images (Pneumonia)’ dataset by Paul Mooney, https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[2] Creative Commons (referred 4.7.2024), CC0 1.0 (CC0 1.0 UNIVERSAL) Deed, CC0 1.0 Deed | CC0 1.0 Universal | Creative Commons

[3] Creative Commons (referred 4.7.2024), CC BY 4.0 ATTRIBUTION 4.0 INTERNATIONAL Deed, CC BY 4.0 Deed | Attribution 4.0 International | Creative Commons

[4] Creative Commos (referred 4.7.2024), CC BY-SA 4.0 (ATTRIBUTION-SHARELIKE 4.0 INTERNATIONAL) Deed, CC BY-SA 4.0 Deed | Attribution-ShareAlike 4.0 International | Creative Commons

[5] Mathur, Madhav (2024, referred 4.7.2024) Pneumonia Detection using CNN(92.6% Accuracy), Apache 2.0 open source license https://www.apache.org/licenses/LICENSE-2.0, https://www.kaggle.com/code/madz2000/pneumonia-detection-using-cnn-92-6-accuracy

[6] Raikote, Pranav (updated 2020, referred 4.7.2024), Covid-19 Image Dataset, CC BY-SA 4.0 license (ATTRIBUTION-SHARELIKE 4.0 INTERNATIONAL), dataset got from the University of Montreal, Alteration for this project: The train set of Covid-19 Image Dataset has been divided into training set and validation set in a way that the validation set has 20 precent of the images from the former training set and the new training set having the rest of them. This way born dataset has been saved into completely new folder ‘Covid-19_dataset_new_split’ having folders train, valid and test. Also, the covid-19 images from COVID CXR Image Dataset (Research) by Manu Siddhartha [] has been divided into those three folders in ‘Covid-19_dataset_new_split’. 20 precent of them going into the test set, another 20 percent from the rest of the data going into the validation set and rest going into training set., https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset/data

[7] Siddhartha, Manu (updated 2021, referred 4.7.2024), COVID CXR Image Dataset (Research), CC0 1.0 (CC0 1.0 UNIVERSAL) license, https://www.kaggle.com/datasets/sid321axn/covid-cxr-image-dataset-research, acknowledgements for

1. D. Kermany, K. Zhang, M. Goldbaum, Large dataset of labeled optical coherence tomography (oct) and chest x-ray images, Mendeley Data, v3, https://data.mendeley.com/datasets/rscbjbr9sj/3 (2018).
2. C. J. P, P. Morrison, D. L, Covid-19 image data collection, arxiv, arXiv preprint arXiv:2003.11597 (2020). URL https://github.com/ieee8023/covid-chestxray-dataset Z.-H. Chen, Mask-rcnn detection of covid-19 pneumonia symptoms by employing stacked autoencoders in deep unsupervised learning on low-dose
high resolution ct (2020). doi:10.21227/4kcm-m312.URL https://ieee-dataport.org/open-access/mask-rcnn-detection-covid-19-pneumonia-symptoms-employing-stacked-autoencoders-deep
3. D. S. et.al, Covid19action-radiology-cxr (2020). doi:10.21227/s7pw-jr18.
URL https://ieee-dataport.org/open-access/covid19action-radiology-cxr