Computer Vision and Deep Learning in medical imaging

Detecting Covid-19 on chest CT scans

Frank Rust [1]

[1] MSc Electro & Computer Engineering, University of Birmingham (UK)
fxr908@student.bham.ac.uk

Abstract: In the fight against the Covid-19 virus, enough testing capacities are crucial for quick isolation of infected patients. However, testing capacities are limited due to the finite supply of testing kits for RT-PCR and similar tests, and the time these methods can take until the result is delivered. This work evaluates the use of chest CT scans as an alternative method for Covid-19 screening and develops a computer aided diagnostic system to increase testing accuracy and speed. The best results for automated diagnostic solutions were achieved using CNNs for the binary classification task, with ResNet being the best performing architecture. To face the sparse data problem due to limited availability of Covid-19 chest CT scans, transfer learning and fine-tuning were used, which was substantial for the final network ResNet101 to achieve an accuracy of 97%, substantially exceeding average radiologist performance. The performance evaluation was done on a very diverse, publicly available Covid-19 CT dataset which was part of a machine learning challenge at grand-challenge.org, making the results rather solid and likely to generalize well. The achieved results make this work the top contender on the grand-challenge leader board.

1. Introduction

At the end of September 2020, the Covid-19 virus had spread around the world and has infected over 30 million people in almost every country worldwide. It has already caused over a million deaths and numbers are still rising sharply, according to the official information provided by the World Health Organization [1].

The main measures that are recommended to slow down the spread of the virus are social distancing and increased personal hygiene. Of similar importance however is the strict isolation of infected people, which makes an early detection of the virus crucial. The main Covid-19 testing method is the RT-PCR test. However, the testing capacities can be quite limited regionally and getting the test results can take some time. By introducing additional testing methods, these problems can be alleviated. Since Covid-19 symptoms are often visible in the patient's lung, chest CT scans are becoming a popular resource for Covid screening and monitoring. However, while it can be fast and rather cheap to take a vast amount of CT scans, evaluating the images takes a lot of time and medical expertise. This paper will analyse how this CT scan analysis bottleneck can be alleviated by introducing Computer Vision to the task to develop a Computer Aided Diagnostic system for Covid-19.

2. Covid-19 testing

Diagnostic tests are used to test for a currently active coronavirus infection. Due to its high accuracy, the RT-PCR (reverse-transcription polymerase chain reaction) test is the most popular testing option. Using a collection kit with a specialized swab on the patient's nose or throat, a sample is collected which can then be evaluated in a laboratory. Receiving the test results can take up to a week [2].

Besides the time necessary to receive the result, the much bigger issue in many regions, especially those with very high infection rates, is test availability, since the test kits and laboratory capacities are limited. This is why additional testing strategies would be very beneficial in the fight against the virus.

2.1. CT scans for Covid-19 screening

More and more research is being published describing the effects of the virus on human lungs and some papers specifically describe the sensitivity of chest CT scans for Covid detection. Studies with subjects that have positive PT-PCR tests monitored the patient's lungs and showed a very high likelihood of the development of ground-glass opacities and/or consolidation. The images below show cases with bilateral ground-glass and consolidative opacities marked by white arrows [3]:

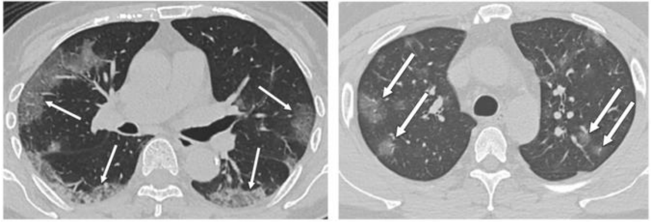


Fig. 1 Covid-19 symptoms on chest CT

According to the research in [3], developing these symptoms during the infection is very likely, with roughly 80% of the 121 symptomatic patients in the study showing either ground-glass opacities or consolidation. This portion further increases with the duration of infection, which Figure 2 shows. After just a few days, when the patient is usually also starting to develop other noticeable symptoms like a cough or a fever, which is when s/he might want to get tested for Covid, the likelihood of lung symptoms is even higher than 80% [3]:

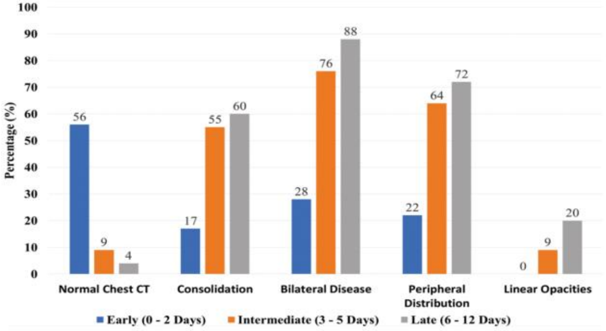


Fig. 2 Likelihood and duration of symptom development

Other studies suggest a general likelihood of ground-glass opacities caused by the virus of over 90% [4]. One study with a direct comparison of RT-PCR and CT results even determined a 97% sensitivity of chest CT image features for Covid on 601 patients with positive RT-PCR tests [5].

This makes chest CT scans a good screening method for Covid detection

3. Computer Vision for CT-scan analysis

Computer vision is becoming increasingly popular in many medical imaging tasks, where large amounts of images need to be analysed in a repetitive, structured environment. From X-Rays to MRIs, Ultrasound and CT scans, computers can make the detection of health problems much faster and easier [6]. Machines are often able to detect even slightest abnormalities that physicians might miss. Especially deep learning techniques are making great advancements in the medical sector for pattern and object recognition and localization in natural images [7] [8].

3.1. Computer Aided Covid CT Diagnostics

When it comes to detecting Covid-19 on chest CTs and distinguishing it from other lung diseases like viral pneumonia, radiologists face a difficult task. Studies like [9] show an average sensitivity of 80% and specificity of 84% for Chinese and American radiologists. Additional research shows that introducing AI to medical image analysis can not only help radiologists make diagnosis faster and easier, often AI systems simply outperform human diagnosis. One paper dealing with AI augmentation of radiologist performance on detecting Covid and distinguishing it from other lung disease [10] shows an average Covid detection accuracy by radiologist of 85% which was successfully increased to 90% with AI assistance.

This sets the goal for the following work: to develop a classification algorithm that detects Covid-19 on chest CTs with higher sensitivity and specificity than radiologist performance, to make diagnosis not only faster, but also more accurate and even exceed 90%.

3.2. Image Data Considerations

One of the most important factors for the development of a good image classification algorithm is having enough informative and trusted data. Especially in the medical domain, Computer Vision engineers often have to rely on medical experts to provide enough data that is professionally analysed and labelled. If a Computer algorithm is developed with insufficient data that is not labelled correctly, it can generally not perform well.

3.2.1 Sparse Data: Given the requirements mentioned above, collecting enough CT scan data for Covid detection can become a difficult task for multiple reasons:

- Covid-19 is a rather new condition of which not a lot of image data has been collected and published yet.
- In the medical field, publishing patient data must be in line with the patient's right to privacy, which further limits the amount of data being published.
- Covid can be hard to detect on CT scans by radiologists with insufficient experience, which can lead to wrong labelling. Ideally, all cases detected as Covid on lung CTs should go through additional confirmation via other testing methods, ideally RT-PCR.

3.2.2 Grand-challenge dataset: The main image dataset that will be used for this project was published on https://grand-challenge.org/, a platform focussing on machine learning solutions in biomedical imaging. The CT scan data was published as part of a competition "to encourage the development of effective Deep Learning techniques to diagnose COVID-19 based on CT images" [11]. The challenge also has a leader board where participants can list their results, which will be mentioned again in a later part of this report. The published dataset consists of 349 CT images with clinical findings of Covid-19 from 216 patients and a similar amount of Non-Covid CT images. The images are collected from many different resources, captured in many different health institutes, which makes the dataset very diverse and therefore very good for testing, since the results are likely to generalize well. The data is already split into image sets for training, validation, and testing, which further eliminates any bias. More information on the dataset can be found in [12].

3.2.3 Variances and disturbances on lung CTs: As to be expected when dealing with human subjects, chest CT scans can show a lot of variance even between healthy subjects since every lung is different. On top of that, additional variances and disturbances can appear depending on the health institution where the images are captured, which is shown here using some examples from the mentioned Covid-CT dataset:

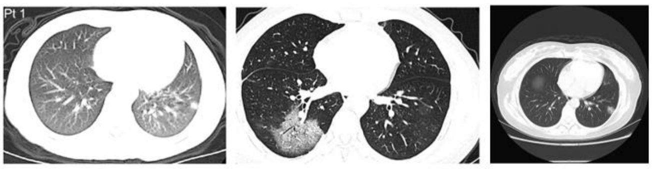


Fig. 3 Diversity in chest CT images

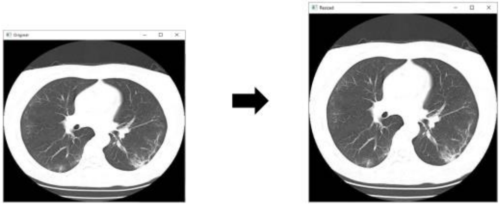
The images vary a lot in image quality, size, brightness, contrast, and scope.
Since these irregularities can be difficult for a classification algorithm to deal with, image pre-processing can be very beneficial for performance enhancement.

3.3. Image Pre-processing

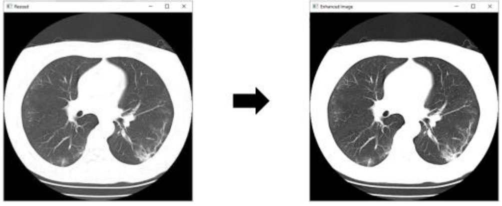
Image pre-processing can help in classification tasks by making the relevant information more dominant or by removing unnecessary variance and disturbances from the images.

To remove some of the CT scan irregularities including differences in image size, brightness scope and especially any disturbances on the outside of the lungs, an automatic pre-processing algorithm was developed. This algorithm uses convolution filters and pixel value thresholds to detect the lung boundaries. It then crops the image accordingly and applies a custom lung mask. It also slightly increases the brightness of CT scans that are underexposed. The whole process is described in these steps:

1) Resize Image:
A common size for the CT images is 512x512 pixels to which all images are resized:



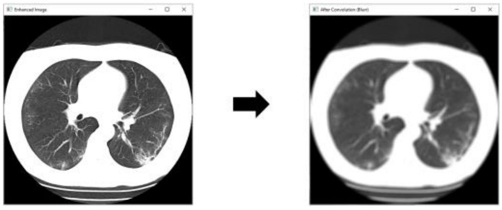
2) Enhance Contrast and Brightness:
Images with low brightness and contrast need to be enhanced to apply pixel value thresholds in the next steps:



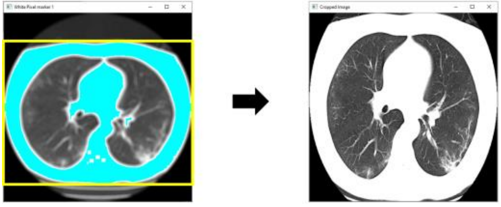
3) Convolution Blur:
To find the lung boundaries, a convolution filter is applied that blurs the images by taking the average of a 9x9 kernel:

$$\mathbf{K} = \frac{1}{81} * \begin{bmatrix} 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

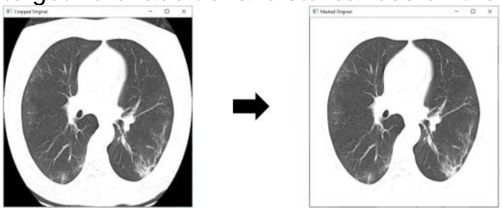
The kernel is moved across the image to take the average of the pixels underneath to replace the central element. This reduces the brightness of smaller white areas outside of the lungs (which are in this case noise) so they can be discarded in the next step since they will fall below the threshold. The size of the kernel was chosen to optimally match the task:



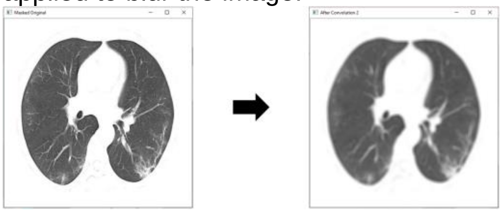
4) Crop White Boundaries:
A high pixel value threshold is now applied and outer areas that fall below the threshold get cropped out. The turquoise mask shows which pixels exceeded the threshold after convolution, getting rid of the two lines at the bottom:



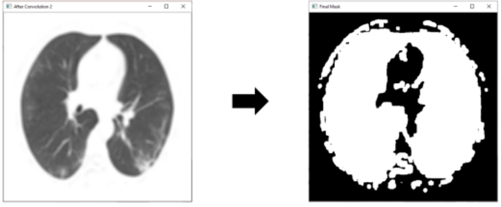
5) Apply Lung Mask to cropped Original:
The same cropping area is now applied to the original image and a custom lung mask is applied to get rid of additional disturbances on the outside:



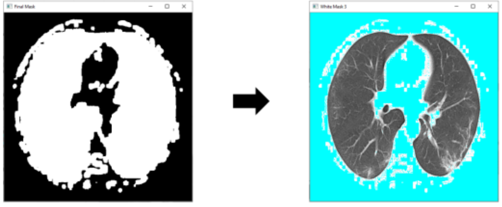
6) Second Convolution Blur:
Since the area between the two lobes of the lung can still include some minor disturbances and darker patches, another convolutional filter is applied to blur the image:



7) Extract White Threshold Mask:
After blurring again, a pixel value threshold is applied to create the final binary mask for the image:



8) Apply Mask to Original:
This mask is now applied to the cropped and resized original. The pixels covered by the binary mask become fully white. These pixels are again shown in turquoise below to show that only very few irregularities outside the lung are left:



The result after completing all the steps can be seen here:

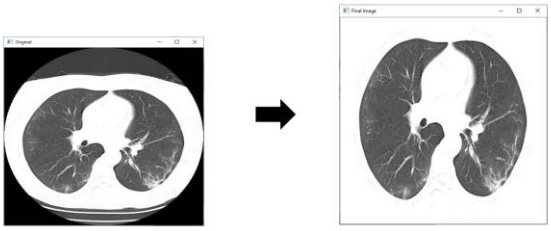


Fig. 4 Result of automated Image Pre-processing

The algorithm was successfully applied to whole image batches. While it finds the lung boundaries very reliably, in some images, smaller parts of the outer lung boundaries are left and can be a bit darker. However, getting rid of those could lead to also losing information on the inside of the lungs, which needs to be avoided at all cost, because that might lead to eliminating disease symptoms and therefore wrong classification.

This pre-processing will help classification algorithms to achieve higher accuracies by reducing the amount of irrelevant information in the images.

3.4. Initial classification algorithm design

During this work, multiple different classification algorithms have been developed and tested. Initially, the algorithms used rather simple feature vectors for classification, analysing features like grayscale distribution and Haralick texture [13]. These initial algorithms have shown that there are multiple issues when it comes to accurate Covid detection:

- Difficulties with correct distinction between lungs with Covid-19 and lungs with other abnormalities and diseases
- Irregularities on the outside of the lungs when using the automated pre-processing algorithm that is not 100% accurate on the lung boundaries which can affect classification
- Determining specific classification features and thresholds for Covid since the symptoms can be very diverse and even for a trained radiologist can be very hard to detect. There is no clear cut-off for the simple feature vectors for accurate binary classification.

Especially the last point leads to the assumption that a more complex algorithm would be appropriate for the task. Ideally an algorithm, that finds the relevant features for classification automatically, which is why the next paragraph will introduce deep convolutional neural networks to the task.

4. Deep Convolutional Neural Networks

Deep Convolutional Neural Networks (CNNs) are a special type of neural network that have shown increasingly good performance in computer vision tasks throughout the past years. The main advantage of a CNN is its “powerful learning ability due to the use of multiple feature extraction stages that can automatically learn representation from the data” [14]. The advantages of the network complexity can be exploited more and more with increasing data availability and improvement in hardware technology and thereby processing power.

CNNs are feedforward hierarchical structures combining convolutional layers, non-linear processing units and subsampling layers [15]. Convolutional layers apply convolutional kernels to perform multiple transformations that help to extract useful features from locally connected data points. Their output is then forwarded to the activation function which generates responses corresponding to the differences in the images. This output is then usually subsampled to summarize the results. This automatic feature extraction ability means additional computational effort, but it reduces the need for a separate feature extractor.

While there are many options to design different activation and loss functions for parameter optimization and regularization, the main factor for network performance is choosing a CNN architecture that fits the task.

4.1. Network architectures

The option of exploiting different network depths and widths, multi-path information processing and the use of structural block units consisting of multiple layers opens the door for the development of many different architectures. The following graph tries to introduce some of the most popular CNN architectures by classifying them as part of seven different categories based on depth, width, spatial exploration, multi-path, feature-map exploitation, channel boosting and attention [14]:

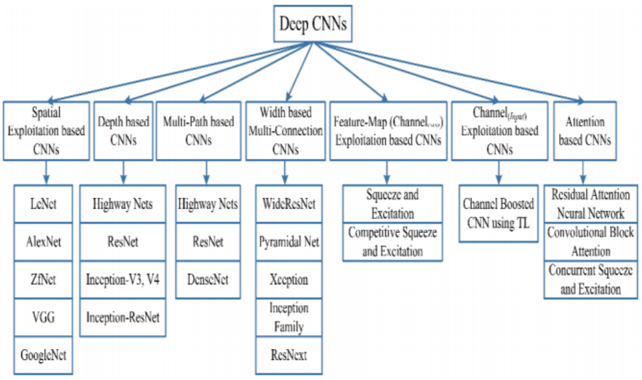

Deep CNNs
Spatial Exploitation based CNNs
Depth based CNNs
Multi-Path based CNNs
Width based Multi-Connection CNNs
Feature-Map (Channel_Input) Exploitation based CNNs
Channel(Input) Exploitation based CNNs
Attention based CNNs
LeNet
AlexNet
ZfNet
VGG
GoogleNet
Highway Nets
ResNet
Inception-V3, V4
Inception-ResNet
Highway Nets
ResNet
DenseNet
WideResNet
Pyramidal Net
Xception
Inception Family
ResNext
Squeeze and Excitation
Competitive Squeeze and Excitation
Channel Boosted CNN using TL
Residual Attention Neural Network
Convolutional Block Attention
Concurrent Squeeze and Excitation

Fig. 5 Overview Deep CNN architectures

Looking at all the options that could be applied to the task, it is important to look at their performance on benchmark datasets and other vision-related datasets possibly in related domains. A lot of useful information on network performance can be found in [14]. Additionally, looking through other papers that apply CNNs in the medical domain can help guide towards good, popular options that were chosen for related tasks like in [16].

After a few initial tests on the main dataset, the top contenders for this task that will be further analysed are ResNet, DenseNet and InceptionV3. The training setup and results are presented in the following part.

4.2. CNN Application

All the different CNN architectures were trained and tested under similar conditions. First, a pretrained model is initialized and the final layers are reshaped to match the number of outputs to the number of classes for the task, in this case two; Covid and Non-Covid. More information on using pretrained models can be found in part 5.5.

The basic concept for the CNN training process is shown in the following diagram:

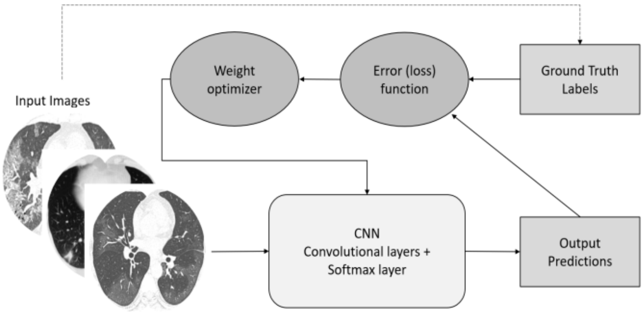


Fig. 6 Basic CNN training process

This process is usually the same independently of the task for which the CNN is trained, as shown in [17]. The images fed to the CNN for this specific task are labelled as Covid and Non-Covid and its predictions are constantly compared to the ground truth as part of the loss function to optimize the network weight parameters via backpropagation to achieve highest possible prediction accuracy. Throughout the training process, the multilayered CNN structure learns to extract low, mid and high-level features that are then used for the classification task.

The initial training process is run for 20 epochs. After each training epoch, the model performance is evaluated on the validation data and the model with the highest validation accuracy is returned. The graph below shows the model performance of the three different architectures ResNet50, InceptionV3 and DenseNet with the grand-challenge image data for training and validation:

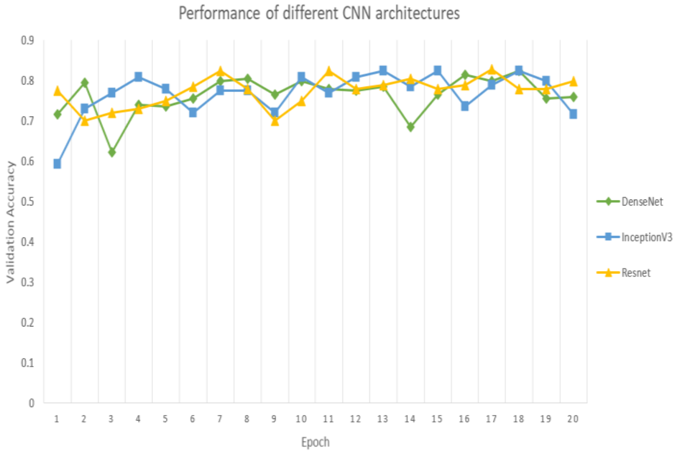


Fig. 7 Performance comparison of CNN architectures

As can be seen, all networks perform similarly well on the given task. When looking at the performance development over 20 epochs it is hard to tell which CNN performs best; all results are good for an initial test run, already exceeding the performance of the previous algorithms that were not relying on CNNs. The following table compares the average and best validation accuracies from the test:

Table 1: CNN architecture performance comparison

Architecture	Best validation accuracy	Average validation accuracy
DenseNet	0.8227	0.7623
InceptionV3	0.8227	0.7665
ResNet50	0.8276	0.7722

Looking at these numbers, the best performing network structure is the ResNet50 which slightly beats the other two networks in average and top accuracy. However, it is important to mention that the training algorithm and the optimizer always have a random element to them, so repeating the same test could lead to slightly different results. Nevertheless, the good ResNet performance has also been confirmed in other similar projects like [19] where a “Covid-ResNet” has been used for Covid detection with a total accuracy of 96%. However, their dataset only included 68 Covid-19 radiographs from 45 patients and focussed on chest X-Ray instead of CT. Nevertheless, using a ResNet structure increased performance compared to [20] where a different network structure was designed for the same task and the accuracy was only 92%.

4.3. ResNet

Having the benefit of exploiting very deep neural networks for multi-level feature recognition is crucial in challenging visual recognition tasks. However, training very deep networks can be difficult especially due to a degradation problem [21] where accuracy gets saturated with increased network depths and even starts decreasing after saturation. To deal with this degradation problem, ResNets use a special learning method called “residual learning”. The idea is to change the mapping function for the layers from the original, unreferenced mapping to residual mapping which is easier to optimize. Residual mapping is achieved by introducing shortcut connections between blocks of layers to realize additional feedforward functions that are used for identity mapping, where input and output are the same, which can be used as a precondition for the otherwise difficult to optimize output of multiple convolutional layers. “If the optimal function is closer to an identity mapping than to a zero mapping, it should be easier for the solver to find the perturbations with reference to an identity mapping, than to learn the function as a new one” [21]. The residual blocks end up looking like this [21]:

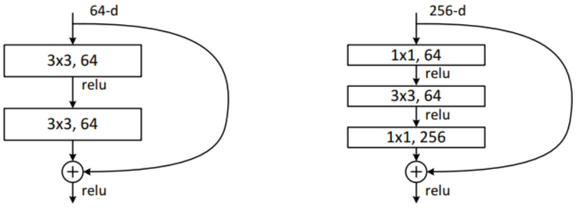
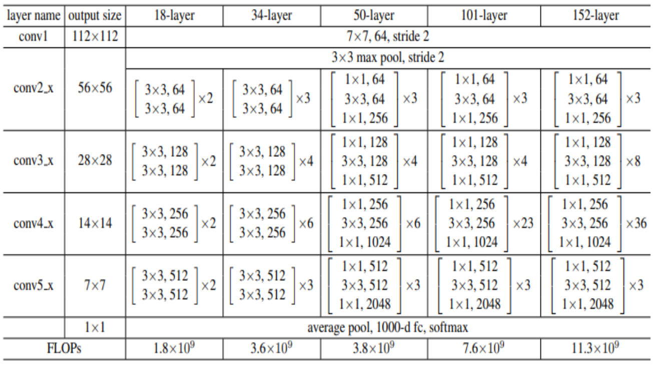


Fig. 8 Residual blocks in ResNet

While in the smaller ResNet architectures only 2 layers are stacked using this method, deeper structures stack up to three layers which can help to reduce training times. The ResNet structure itself is available in different depths from 18-layer to 152-layer ResNet [21]:



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 9 Different ResNet architectures

The stacked blocks used are shown in brackets and the total amount is noted next to it. The number of FLOPs (Floating Point Operations per second) measures the complexity of the networks, which even for the 152 layer model at 11.3 billion is still reasonably low compared to other structures like VGG-16/19 nets with 15.3/19.6 billion FLOPs [21]. This means reduced computational effort is necessary for training.

A variety of ResNet depths will be tested under the same training and evaluation conditions as the three architectures before (see 5.3). These are ResNet18, ResNet50 and ResNet101. The results will show why going even deeper (ResNet152) would not be beneficial for the given task.

The resulting accuracies are shown below. The networks were trained for 40 instead of 20 epochs for this test which further increased the ResNet performance compared to the attempt in 5.3:

Table 2 ResNet performance comparison

ResNet type	Validation accuracy	Training time
ResNet18	0.822660	115 min
ResNet50	0.837438	280 min
ResNet101	0.827586	536 min

The table shows the best validation accuracies and the computation times for training with the main training and test set. ResNet101 takes the longest time for training since it is the deepest structure with the most parameters to tune. It needs almost double the training times of ResNet50 with 536 minutes instead of 280. However, the results for the given task are best using ReNet50, exceeding the validation accuracies of other two structures by nearly 1%. This could be explained by the ResNet18 structure not being complex enough to recognize the diverse patterns optimally, while the ResNet101 might be too complex to be optimally tuned with the limited image data available.

4.4. Transfer learning and Fine-tuning

As mentioned before, the neural networks applied to the task above were pretrained. This means the weights and biases have initially been trained for other classifications tasks on huge image databases. In this case, the ImageNet database [22] allowed training on 1.2 million images of a thousand different categories [18] [23]. These classes are of many different domains and include animals, vehicles, and hundreds of other every-day objects. The categories are therefore not necessarily related to the given task in the medical domain. However, by making use of transfer learning, the network still profits severely from the initialized weights especially on the lower layers to detect basic low- and mid-level features like lines, edges and blobs.

The final output layer of the network needs to be changed from an output of a thousand classes to fit the binary classification task with only two outputs for Covid and Non-Covid. After that, the network can be fine-tuned for the specific task with further training to adapt the weights more precisely. The example below shows a direct comparison of the training process on the main dataset of a ResNet50 architecture that is pre-trained (blue) vs trained from scratch (orange). Both networks were trained for 40 epochs and the graphs show the validation accuracy for each epoch:

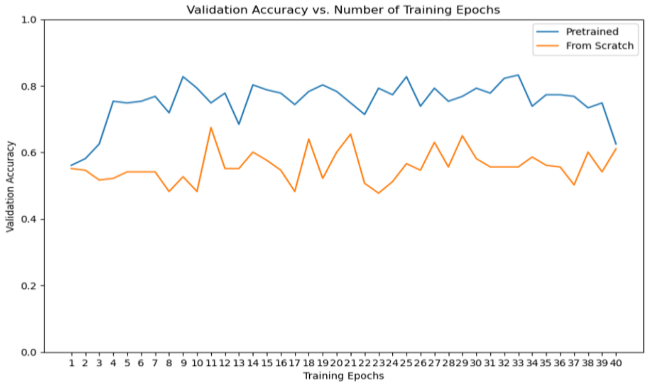


Fig. 10 Comparison Pretrained vs Non-Pretrained ResNet50

As can be seen immediately, the pre-trained network performs significantly better. The non-pre-trained ResNet50 reaches its top validation accuracy at 0.6749, while the pre-trained network reaches a validation accuracy of 0.8374. Further tests have shown that the untrained network struggles to find any reliable upwards trends and instead mainly relies on random improvements. This is not surprising since the network structure is highly complex and when being initialized with random weights, it struggles to find any reliable patterns with the limited training data available. Since the pre-trained network however already knows basic pattern and object detection, it makes transfer learning a very beneficial strategy especially on classification tasks that have to deal with sparse data problems.

4.5. Result evaluation

As shown by the results above, using deep neural networks for the given classification task can provide good results. Using pre-trained networks and fine-tuning with the limited data already achieved accuracies of up to 84%, which is significantly better than the mentioned initial algorithms. Nevertheless, to be applicable in the medical context the results still need improvement.

4.6. Data expansion

Due to the complexity and depths of the neural networks, it is common for the network training to be done with many thousands or even millions of images, rather than just a few hundred, like in this case. More training data can greatly improve the accuracy and is also very important to avoid overfitting on specific training sets that include many similar images. Overfitting would in the given case mean that the network parameters are tuned specifically for best performance on the given dataset while performing significantly worse on other random datasets. However, since the main dataset provided by grand-challenge.org combines CT scans from many different sources, it is very diverse, making the results generalise well. Nevertheless, it will now be expanded with additional images collected from other publicly accessible online resources, to provide more training data and make the results more solid.

As mentioned before, finding publicly available CT images for Covid classification can be difficult for multiple reasons. However, after searching through many different online resources, a few hundred additional images were found and added to the dataset from:

- Zenodo.org [24]
- Sirm.org [25]
- Mosmed.ai [26]

Which lead to a total of over 1,100 Covid and 1,200 Non-Covid images from over 1000 different patients. They were mixed into the training and validation sets in different ways, which will be explained and evaluated in the following sections. While there are more public resources available, manually selecting and combining trusted sources with images that are labelled correctly can be challenging.

Some of the images in the medical sector, especially CT scans, are often provided in the DICOM format, which stores three-dimensional views through the lungs and requires special software to open. However, using a python script allowed automated image conversion to a two dimensional .png format, which can also expand the number of images significantly since every single layer of the 3D images can be used as a separate slice for training. However, using many similar images, especially multiple from a single patient's lung, can again lead to overfitting which should be avoided by carefully balancing the data.

4.7. Improved results

There are four main training and testing strategies to be analysed:

1. *Training with all combined data evaluated on random test set:*

After combining images from grand-challenge.org, zenodo.org, sirm.org and mosmed.ai, the total image count was roughly 1100 Covid and 1200 Non-Covid images. This was after already significantly reducing the images per patient form mosmed.ai, since after the conversion from the 3D format into 2D slices, some patients were represented with up to 100 images, which is not beneficial as will be confirmed later. The classification accuracy was evaluated on 100 Covid and 100 Non-Covid images which were randomly selected from the combined data.

The highest validation accuracy that was achieved during training was 97%. But since the data was not balanced very well and most images were from mosmed.ai and thereby very similar, the good performance might be achieved due to overfitting, which will be confirmed in the next attempt.

2. *Training with all combined data evaluated on grand-challenge test set:*

When using the same combined images for training but evaluating the prediction accuracy only on the very diverse grand-challenge test set, the best achievable performance dropped significantly to 83%, which confirms the overfitting issue in attempt 1. Therefore, the data needs better balancing.

3. *Training with more balanced data evaluated on grand-challenge test set:*

To make the data more balanced, the number of mosmed.ai images had to be further reduced to avoid overfitting. The images per patient and the total amount of mosmed.ai images were reduced significantly. This resulted in a new combined total of roughly 700 Covid and 700 Non-Covid images for training. When evaluating the performance on the grand-challenge data, an accuracy of 87% was achieved. However, during training the training accuracies were consistently higher at around 95%, which shows the slight disbalance between the training data from all sources and the test data only from grand-challenge.org.

4. *Training with more balanced data evaluated on random test set:*

Instead of using the grand-challenge data for validation, a random batch of 100 Covid and 100 Non-Covid images was used for validation this time and a top accuracy of 95% was achieved. Since these were good results on balanced data, the trained network was saved and then applied for classification on the original grand-challenge task. The resulting predictions for this test are shown in the following confusion matrix:

Table 3 Confusion Matrix ResNet50 on challenge test set

Medical annotation Detected as	Covid	Non-Covid
Covid	95	3
Non-Covid	3	102

And the following performance measurements:

Table 4 Performance measurement ResNet50

Total accuracy	0.9704
Sensitivity	0.9694
Specificity	0.9714
F1	0.9694
AUC	0.9704

As these results are very satisfying, the same training and testing principles were then also applied to other network structures for comparison, since the initial testing in [5.3] showed similar performance. InceptionV3 did perform almost equally well on the random validation set with 94.5% accuracy, however it did not translate well after being saved and applied to the grand-challenge test set with a prediction accuracy of only 88%. DenseNet achieved only 93.5% validation accuracy on the randomly mixed data and a prediction accuracy of 86% on the grand-challenge test set.

The results of the different network architectures are summarized in the following table:

Table 5 Final Comparison of top CNN architectures

Architecture	Validation acc.	Test acc.
ResNet50	0.95	0.97
InceptionV3	0.945	0.88
DenseNet	0.935	0.86

Again, validation was done on a random subset of all combined images, while testing was done on the grand-challenge test set to see if results translate and generalize well.

Since ResNet performed best and the data has now been expanded, another attempt on a more complex ResNet structure; ResNet101 was done where a validation accuracy of 95% was achieved again and the applied model predictions on the grand-challenge test set were:

Table 6 Confusion Matrix ResNet101 on challenge test set

Medical annotation Detected as	Covid	Non-Covid
Covid	96	4
Non-Covid	2	101

This is similar to the performance of ResNet50 in terms of total accuracy, but due to slightly reduced false negatives, the performance is even better in the given context. The improved performance measurements with increased sensitivity are:

Table 7 Performance measurement ResNet101

Total accuracy	0.9704
Sensitivity	0.9796
Specificity	0.9619
F1	0.9697
AUC	0.9708

4.8. Result analysis

Since the networks can be saved and loaded to do predictions on single images or batches of images, the exact decisions can now be analysed. Especially the wrong classifications are of interest. While it is hard to fully understand how the network makes its decisions, it can at least be checked if the wrong classifications might be especially difficult cases or if the network seems to make severe errors.

Looking at the output of the best performing network, Resnet101, below are the two Covid images that got falsely classified as Non-Covid:

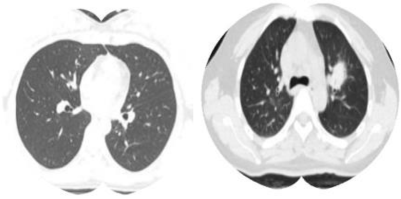


Fig. 11 False negative predictions by ResNet101

The first image hardly shows any characteristic symptoms and seems very similar to a healthy lung. The second image shows one strong white patch in the right lung; however, the tone and pattern of the patch seems much more solid than those usually occurring as Covid symptoms. These two cases can therefore be checked off as rather difficult cases to classify. Here a comparison with the predictions of a professional radiologist would be very interesting. The lungs might have been labelled falsely, or the patient might have had Covid but the lungs might just not have been affected (yet).

The four Non-Covid images classified falsely as Covid are also presented here:

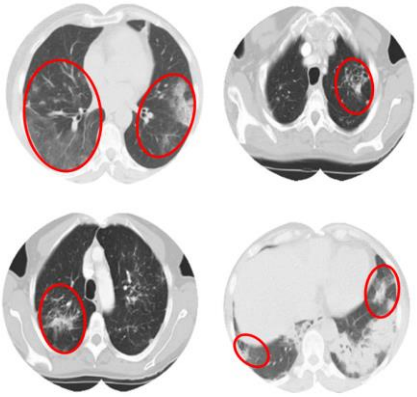


Fig. 12 False positive predictions by ResNet101

The red circles were manually added to mark areas with patches that look similar to the Covid symptoms described in the beginning. Again, the wrong classifications do not seem like severe errors and instead would be worth a second look by a radiologist.

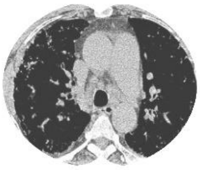
Analysing all wrong predictions and seeing that they are all to some extend justifiable, makes the network performance even more satisfactory.

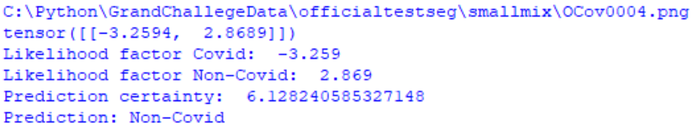
Since the data used for testing from grand-challenge.org was provided as part of a public challenge with a leader board, the results with the performance measurements listed in 5.8 were actually submitted. While the submission is not yet listed at the current time, the achieved accuracy would currently place at first place on the leader board at https://covid-ct.grand-challenge.org/Leaderboard/.

5. Additional feature: Decision Certainty

To make the performance analysis even more advanced, instead of just evaluating the binary decision accuracy, one can measure the network's decision certainty. Ideally this certainty would be below average for the falsely classified images, which would make the model performance even more solid.

An output for decision certainty can be generated by accessing the two output values in the final CNN layer, which stand for the likelihood of each class: Covid and Non-Covid. The network makes its decisions by comparing these two values and the final prediction depends on which of the two class values is higher. Ideally, if the value for one of the classes is high, the other one should be low, which provides a high certainty factor. The following example shows the tensor output for a Non-Covid image:

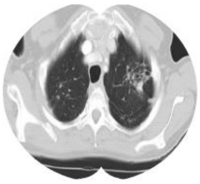


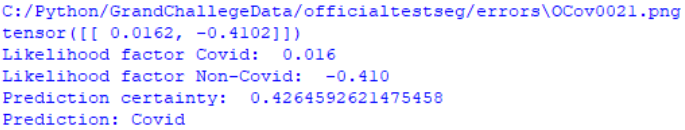


```
C:\Python\GrandChallegeData\officialtestseg\smallmix\OCov0004.png
tensor([[-3.2594,  2.8689]])
Likelihood factor Covid:  -3.259
Likelihood factor Non-Covid:  2.869
Prediction certainty:  6.128240585327148
Prediction: Non-Covid
```

For this example, the likelihood factor for the analysed lung to have Covid symptoms is roughly -3.26 while the likelihood factor for the image to be a Non-Covid lung is roughly 2.87. This means the image gets classified as Non-Covid. Taking the difference between these two values provides the prediction certainty. If the likelihood factors of the two classes are very close together, the model is not very certain of its decision, which is an important output to have with the final prediction.

The following example shows the outputs for one of the falsely classified Non-Covid images:





```
C:/Python/GrandChallegeData/officialtestseg/errors\OCov0021.png
tensor([[ 0.0162, -0.4102]])
Likelihood factor Covid:  0.016
Likelihood factor Non-Covid:  -0.410
Prediction certainty:  0.4264592621475458
Prediction: Covid
```

As can be seen, the final prediction is false, but the certainty factor is quite low, which is similar for most of the other false classifications shown in the previous paragraph. This is a valuable information to have as a user. Predictions with low certainty should be carefully analysed.

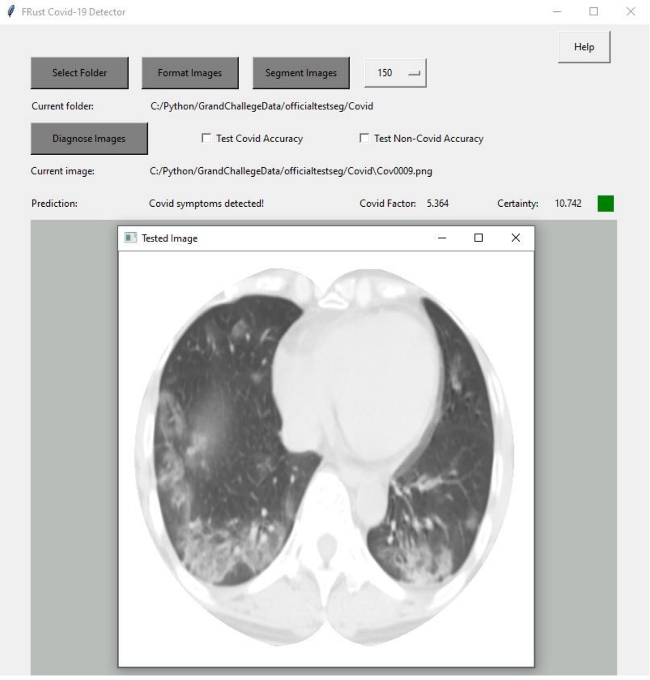
6. User Interface & Accessibility

Since the results are promising and it would be desirable to make the model accessible for testing and implementation in a real world environment, a user interface needs to be designed to make it easy for radiologist and other users without a programming background to apply the model to do predictions on their images.

The focus for the GUI (graphical user interface) development is functionality, so it will be kept rather simple. The main features are:

- Selecting the file path for the stored images
- Formatting the images and applying the pre-processing algorithm for segmentation with adjustable segmentation brightness
- Doing automatic classification of all images with user feedback showing the image, the prediction, the Covid severity, and the decision certainty

Tkinter was used for the GUI design with the following outcome:



Current folder: C:/Python/GrandChallegeData/officialtestseg/Covid

☐ Test Covid Accuracy ☐ Test Non-Covid Accuracy

Current image: C:/Python/GrandChallegeData/officialtestseg/Covid\Cov0009.png

Prediction: Covid symptoms detected! Covid Factor: 5.364 Certainty: 10.742 🟩

The system also includes functionality for accuracy testing on labelled Covid or Non-Covid images. The Help button provides a user manual in more detail.

The model will be uploaded to https://marketplace.arterys.com/model/frustcovidCT soon, where it will be fully available for research use.

7. Critical review and outlook

Throughout this work, different classification algorithms have been developed and tested. The simpler algorithms classifying grayscale distribution and Haralick texture did not provide good enough results and were very susceptible to any disturbances or irregularities in the images. Making use of deep learning and different CNN structures however provided improved results that were optimized to reach accuracies of up to 97%. While many different network architectures were compared, the clear favourite for the task was ResNet.

Since the image data for this task was limited, these results should always be analysed cautiously. But due to the diversity in the grand-challenge data mixed with additional resources, severe overfitting should not be an issue and the results should generalize well. However, it is always recommended to further expand the data for training and testing as it becomes available, to make the network performance more solid and maybe even further increase its accuracy.

When using an algorithm like this in the actual medical context, it is important to use it as supplementation for otherwise limited testing capacities, not as a stand-alone solution. When using this method for Covid testing, positive CT results, which can be obtained comparably fast, can be used for the patient to immediately go into strict self-isolation and later confirm the results with another testing method as testing capacities become available. Patients with severe symptoms with negative CT results should also try an alternative testing method to confirm the results if their symptoms are not relieved after a few days.

If applied correctly, this method can alleviate the Covid-19 testing bottleneck and thereby help to reduce the spread of the virus.

The diagnostic system will be published so it can be tested and used by researchers and health institutes in a real-world environment while collecting feedback and possibly further improving the system.

8. Acknowledgments

Firstly, I would like to thank Dr Neil Cooke, senior lecturer at the University of Birmingham, for supervising this project and providing guidance and advice throughout this work. He strongly increased my interest in the field of computer vision.

Another person to be mentioned is Ghaith Sankari, a deep learning engineer in the healthcare sector, working on the grand-challenge competition as well. He showed great interest in my work and we shared some resources with additional image data and research.

Additionally, I would like to thank the people behind grand-challege.org for providing exciting and relevant competitions like the Covid-19 CT diagnosis challenge and the included image data which built the basis for this project.

Furthermore, all institutions that publish their Covid-19 related data to promote fast solution development in the fight against the virus deserve to be thanked as well.

9. References

[1] World Health Organization, “WHO Coronavirus Disease (COVID-19) Dashboard,” [Online]. Available: https://covid19.who.int/.

[2] FDA U.S Food & Drug Administration, “Coronavirus Testing Basics,” [Online]. Available: https://www.fda.gov/consumers/consumer-updates/coronavirus-testing-basics.

[3] A. Bernheim, “Chest CT Findings in Coronavirus Disease 2019 (COVID-19): Relationship to Duration of Infection,” Radiological Society of North America, 2020.

[4] H. X. Bai, “Performance of radiologists in differentiating COVID-19 from viral pneumonia on chest CT,” Radiological Society of North America, 2020.

[5] T. Ai, “Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases,” Radiological Society of North America, 2020.

[6] HealthEuropa, “Computer vision can improve accuracy of medical diagnostics,” 2019. [Online]. Available: https://www.healtheuropa.eu/computer-vision-accuracy-of-diagnostics/93650/.

[7] F. Altaf, “Going Deep in Medical Image Analysis: Concepts, Methods, Challenges, and Future Directions,” IEEE, 2019.

[8] H. Greenspan, “Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique,” IEEE, 2016.

[9] H. X. Bai, “Performance of Radiologists in Differentiating COVID-19 from Non-COVID-19 Viral Pneumonia at Chest CT,” Radiological Society of North America, 2020.

[10] H. X. Bai, “Artificial Intelligence Augmentation of Radiologist Performance in Distinguishing COVID-19 from Pneumonia of Other Origin at Chest CT,” Radiological Society of North America, 2020.

[11] Grand Challenge, “CT diagnosis of COVID-19 Challenge,” [Online]. Available: https://covid-ct.grand-challenge.org/.

[12] X. Yang, “COVID-CT-Dataset: A CT Image Dataset about COVID-19,” 2020.

[13] R. M. Haralick, “Textural Features for Image Classification,” IEE, 1973.

[14] A. Khan, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks,” SpringerLink, 2020.

[15] K. Jarrett, “What is the best multi-stage architecture for object recognition?,” IEEE, 2010.

[16] H. S. Vardhan, “Deep Convolutional Neural Networks for Classification of Interstitial Lung Disease,” SSRN, 2020.

[17] Z. Akku, “Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions,” SpringerLink, 2017.

[18] N. Inkawhich, “Finetuning torchvision models,” 2017. [Online]. Available: https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html.

[19] M. Farooq, “COVID-ResNet: A Deep Learning Framework for,” arXiv.org, 2020.

[20] L. Wang, “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images,” arXiv.org, 2020.

[21] K. He, “Deep Residual Learning for Image Recognition,” Microsoft Research, 2015.

[22] Stanford & Princeton Univerity, “ImageNet,” [Online]. Available: http://www.image-net.org/.

[23] CS321n, “Transfer Learning,” [Online]. Available: https://cs231n.github.io/transfer-learning/.

[24] M. Jun, “COVID-19 CT Lung and Infection Segmentation Dataset,” 20 April 2020. [Online]. Available: https://zenodo.org/record/3757476#.X06bNMgzaUl.

[25] Societa Italiana di Radiologia Medica e Interventistica, “COVID-19 DATABASE,” 2020. [Online]. Available: https://www.sirm.org/category/senza-categoria/covid-19/.

[26] MosMed.ai, “MosMedData: Chest CT Scans with COVID-19 Related Findings,” 2020.