Covid – 19 Detection with CNNs

FINAL EXAM REPORT
2023-2024

A.A 2023/2024
Prof: SEBASTIANO BATTIATO
Team members name:
Maria Teresa Nicotra 1000049782
Salvatore Damiano Calabrese 1000055827
Mahtab Alghasi 1000039465
Fabrizio L'Episcopo 1000055645
Alessandro Vicari 1000051804

Index

Introduction

Over the past few years, the COVID-19 pandemic has profoundly impacted our lives, reshaping our daily routines. Significant advancements in medical research and scientific endeavors have been pivotal in swiftly and accurately detecting the presence of the virus during its early stages. Neural networks, heralded as the cornerstone of computer vision, offer a unique advantage in discerning intricate patterns within images, a feat often challenging for human observation or traditional machine learning algorithms. Visual interpretation of CT scans by physicians can become complicated and time-consuming and requires expert knowledge and experience. The challenges in visually interpreting CT scans have led to the development of artificial intelligence-based algorithms to support physicians and radiologists in the diagnosis of COVID-19 with a fully automated solution by CNN.

This report endeavors to develop a non-invasive method for detecting COVID-19 through the utilization of a Deep Learning algorithm applied to chest computed tomography images (CT scans). The initial phase of the study began by exploring the dataset in depth. As part of this exploration, the dataset was segmented into a training set and a set for validation purposes. This step is crucial in machine learning tasks as it allows for the evaluation of model performance on unseen data to assess generalization capabilities and prevent overfitting. We've explored a variety of CNN architectures to enhance our accuracy in image processing. Our efforts transcend mere implementation; we're focused on optimizing and refining these structures to maximize their performance. By doing so, we aim to elevate their efficacy across diverse contexts in image processing applications.

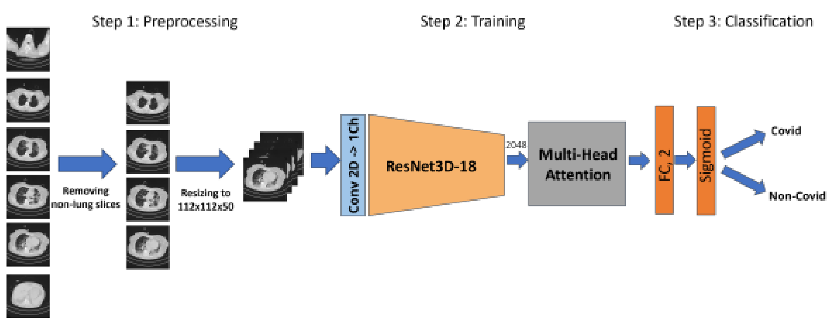
1. Dataset

The COV19-CT-DB dataset was used to evaluate the performance of the proposed approach for automated COVID-19 detection from chest CT scans of patients. This dataset consists of 3D CT scans of the chest labelled by experts, corresponding to a large number of patients. In total, the dataset contains 3746 CT scans, with 1147 scans labelled as Covid and 2599 scans labelled as non-Covid. Each scan in this dataset contains a different number of axial slices, and these slices were provided as JPEG images of size 512 × 512, that we have accurately resized according to the dimensions required by the CNN we tested. The following table, shows the initial dataset composition into training and validation sets.

Annotation	Train Data	Validation Data	Total
Covid	922	225	1147
Non-Covid	2110	489	2599
Total	3032	714	3746

2. **Data Pre-processing**

Before training the model, the data underwent a cleaning, extraction, and preparation process. This involved removing any inconsistencies, errors, or irrelevant information from the dataset to ensure that the model's training process is not adversely affected by noise or biases in the data. In addition, in order to apply each CNN architecture, different sizes of images were needed. All these processes were performed using Python with specific code, a popular programming language in data science and machine learning. Moreover, specific code was used in order to replicate the entire analysis using MATLAB, another widely used tool in scientific computing and machine learning. The Deep Network Designer Toolbox leveraged within MATLAB for this purpose, indicates the use of deep learning techniques in this analysis.



CNN Training

The report explores the use of various pre-trained neural networks, dealing with the problem of binary classification: Covid/NonCovid. In particular, we tried some experiments with fine-tuning parameters on the following networks, with the main goal of finding out some good performances in terms of validation accuracy:

1) AlexNet
2) GoogLeNet
3) ResNet-18
4) ResNet-50
5) VGG-19

We tried to well integrate all these networks with attention mechanisms to enhance the detection of Covid-19. In this perspective, the convolutional neural networks adopted in the analysis work as features extractor for classification tasks, while attention mechanisms are employed to improve the models' capacity to prioritize crucial regions within images. Now, we can go deeper by showing the CNNs involved in, and the main results that we achieved.

1. AlexNet

In the landscape of deep learning, AlexNet stands as a milestone in the history of convolutional neural networks (CNNs). Its pioneering architecture and remarkable performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 sparked a resurgence of interest in deep learning and laid the foundation for subsequent advancements in the field. In this report, we explore the application of AlexNet for the binary classification task of detecting COVID-19 infection from chest computed tomography images (CT scans).

AlexNet Overview: AlexNet, proposed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012, is a deep CNN architecture notable for its depth, use of convolutional layers, and utilization of techniques such as data augmentation and dropout regularization. It comprises eight layers: five convolutional layers followed by three fully connected layers. The convolutional layers are responsible for feature extraction, while the fully connected layers perform classification.

Task Suitability: AlexNet is well-suited for image classification tasks, including medical image analysis. Its deep architecture allows it to learn complex hierarchical features from input images, making it effective for tasks requiring high-level feature representation, such as identifying patterns indicative of COVID-19 infection in chest computed tomography images (CT scans).

Description and Theory: The architecture of AlexNet consists of alternating convolutional and max-pooling layers, followed by three fully connected layers. The convolutional layers employ small receptive fields and stride values to capture local features at different spatial scales. Max-pooling layers are interspersed between convolutional layers to downsample feature maps, reducing computational complexity and providing translation invariance.

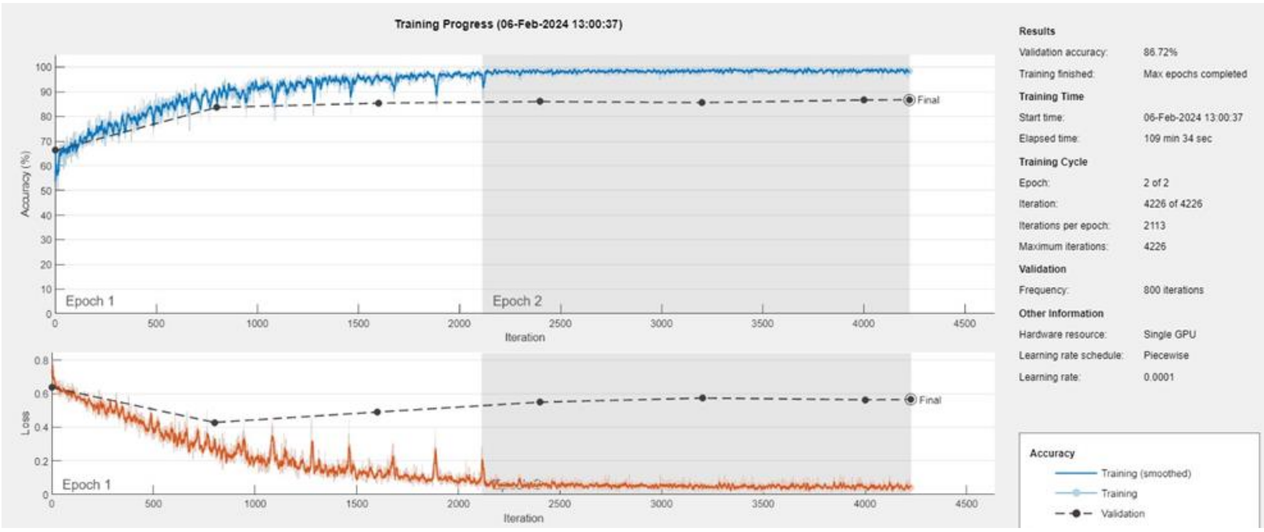
One of the key contributions of AlexNet is the introduction of Rectified Linear Units (ReLU) as the activation function, which accelerates convergence during training by mitigating the vanishing gradient problem. Additionally, AlexNet incorporates techniques such as local response normalization (LRN) and dropout regularization to prevent overfitting and improve generalization performance.

In what follows, the parameter settings and the results are shown:

Hyperparameters Table

	Initial Learning Rate	MiniBatch Size	Validation Frequency	Momentum	L2Regularization	Validation Accuracy

Try 1	**0.001**	**256**	**800**	**0.87**	**0.001**	**86.72%**



The relative simplicity of the AlexNet model compared to others has proven to be advantageous, delivering excellent results while also ensuring a rapid training phase. After several attempts, the achieved results were highly satisfactory, with the model already converging to a very high accuracy after just one epoch. However, to validate the results thoroughly, we deemed it appropriate to complete the second epoch. This decision ensures a comprehensive assessment of the model's performance and stability.

2. GoogLeNet

GoogLeNet Overview: GoogLeNet is a 22-layer deep convolutional neural network (CNN) aimed at solving computer vision tasks. Properly, it’s a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google. It was introduced in 2014, in the paper titled “Going deeper with convolutions”, by providing a significant decrease in error rate with respect to previous models, such as AlexNet (proposed in 2012) and ZF-Net (proposed in 2013).

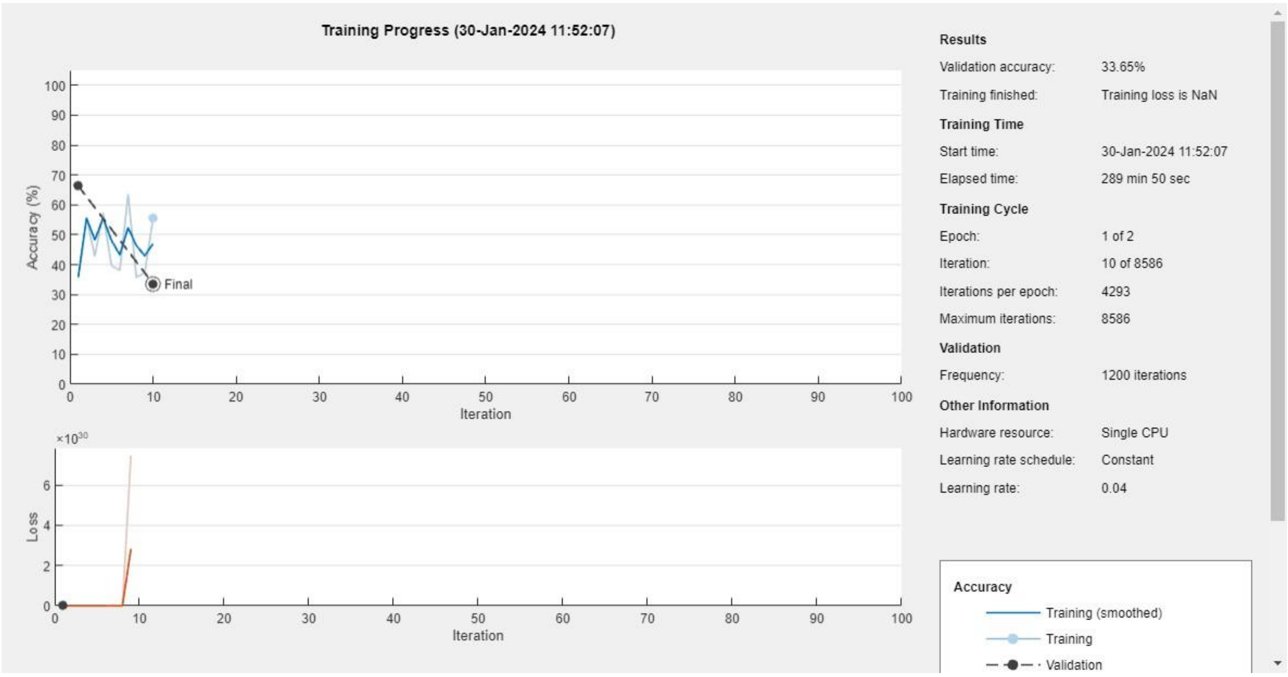
Task Suitability: GoogLeNet is particularly indicated for **image recognition** and **classification tasks,** where the main goal is to assign a label to an input image from a predefined set of classes. So, it is very suitable for the aim of this analysis, where the goal is to distinguish 2 different classes: covid and non-covid. Moreover, the GoogLeNet can be also used for **object detection** and for **recognizing scenes and patterns into images**. Finally, it’s important to underline the fact that GoogLeNet is often used for **transfer learning**, where a pre-trained model on a large dataset is fine-tuned on a smaller dataset for a specific task.

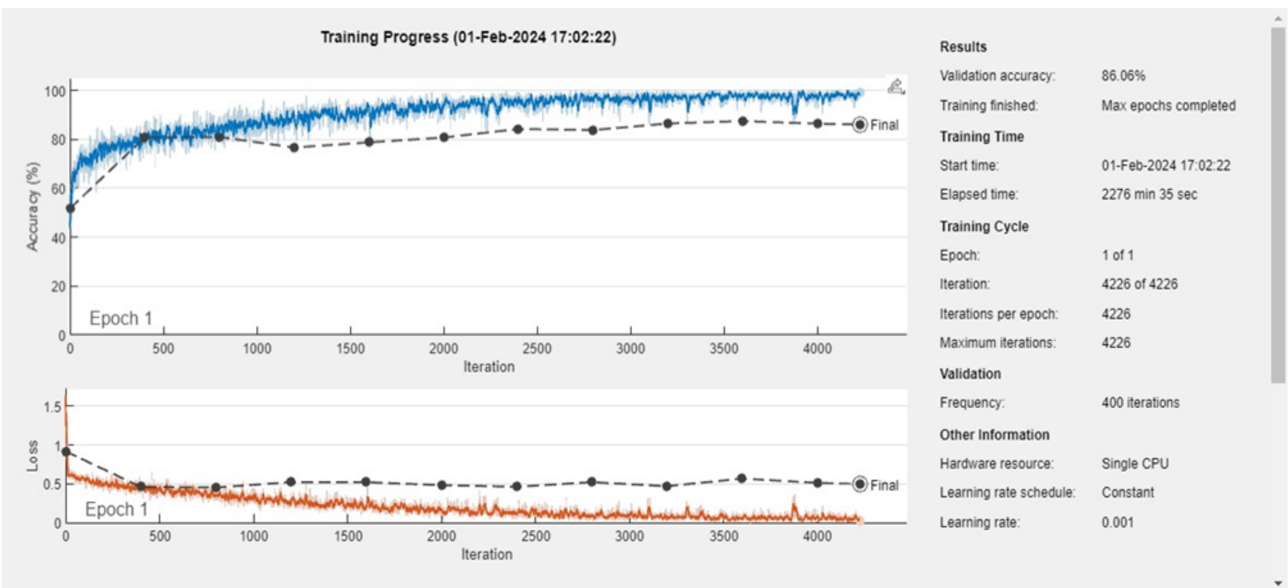
Description and Theory: The GoogLeNet architecture, is very different from previous state-of-the-art architectures such as AlexNet and ZF-Net. It takes in input images of size 224x224, and all the convolutions inside this architecture use Rectified Linear Units (ReLU) as their activation functions. The entire architecture of GoogLeNet consists of multiple stacked Inception modules, followed by global average pooling and a softmax layer for classification. So, by focusing on its architecture, it's possible to distinguish:

1. **Inception Modules**: These modules employ a combination of 1x1, 3x3 and 5x5 convolutions, as well as max-pooling operations, in parallel, and concatenate the outputs. Because of this, the network is able to capture features at multiple scales and improves the efficiency of computation.
2. **1x1 Convolution**: These small convolutions are used for dimensionality reduction, that is to say, to decrease the number of parameters (weights and biases) of the architecture. By contrast, by reducing the parameters, the depth of the architecture is increased, preserving important features and reducing the computational cost.
3. **Global Average Pooling**: What is really interesting about GoogLeNet is the fact that rather than using fully connected layers at the end of the network structure, it employs global average pooling. This involves taking the average of each feature map in the final convolutional layer, resulting in a single vector for each feature map and this aspect is fundamental, because it reduces the number of parameters involved in and helps to prevent overfitting.
4. **Auxiliary Classifiers for Training**: At intermediate layers, GoogLeNet is characterized by the so-called auxiliary classifiers, helpful to address the vanishing gradient problem during the training phase. In particular, they are used during training only, contribute to the overall loss function and help with gradient flow during backpropagation.
5. **Multiple Scales**: The use of multiple scales in the Inception modules allows the network to capture fine details as well as more global features. This helps the model in guaranteeing good performances on a variety of object recognition tasks.

Hyperparameters table

	Initial Learning Rate	**MiniBatch Size**	**Validation Frequency**	**Momentum**	**Learning Rate Drop Factor**	**L2Regularization**	**Validation Accuracy**
Try 1	**0.04**	**128**	**1200**	**0.9**	**0.1**	**0.0001**	**33.65%**
Try 2	**0.001**	**126**	**300**	**0.9**	**0.1**	**0.0001**	**84.34% ***
Try 3	**0.001**	**128**	**400**	**0.9**	**0.1**	**0.0001**	**86.06%**





In the first trial the network stopped after just 10 iterations in the first epoch: clearly, it's not a suitable result, neither a reliable one, with a validation accuracy of 33.65%, because it can be supposed that the model has been skipping over important features during the training. Because of this, the Initial Learning Rate was decreased in the next experiments, by setting it to 0.001. The Validation Frequency was also changed accordingly. In the second try the network seemed to perform quite good, but it occurred in a graphical bug. Therefore, it was stopped manually after the first epoch, by achieving a validation accuracy of 84.34%. Lastly, the training process was launched again with the same network with almost the same parameters, reaching the best result achieved with the GoogLeNet for the analysis, with a validation accuracy of 86.06%.

3. ResNet – 18

ResNet - 18 Overview: ResNet - 18 is a convolutional neural network (CNN) architecture introduced by Microsoft Research as part of the ResNet family. It is specifically designed to address the challenge of vanishing gradients in very deep neural networks. This architecture incorporates residual connections, which allow for the direct flow of information across layers, facilitating the training of extremely deep networks.

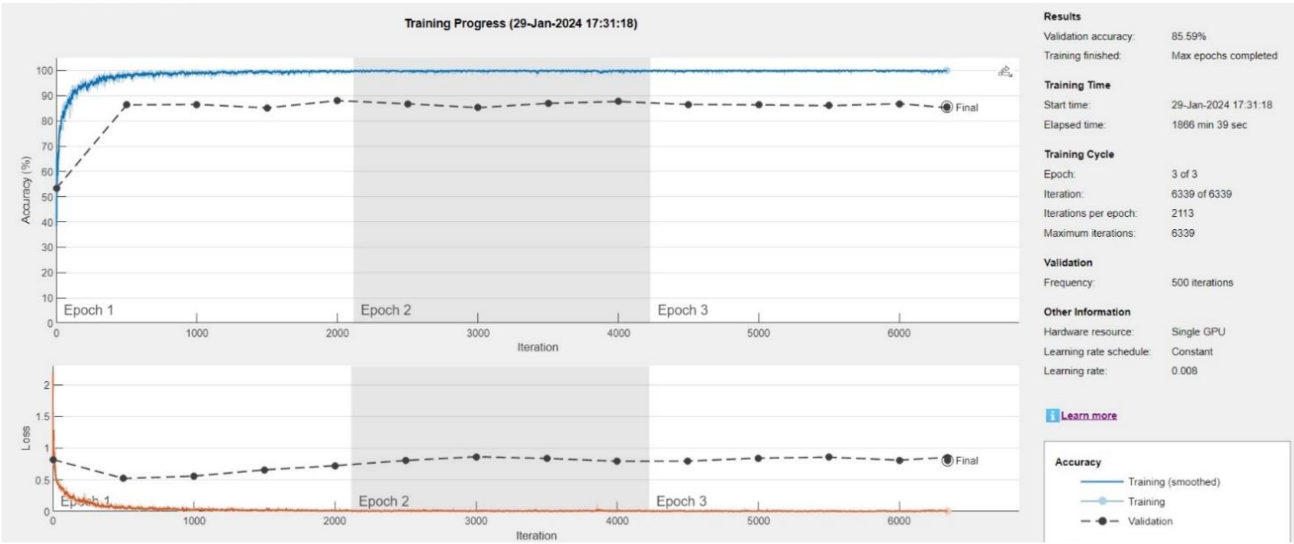
Task Suitability: ResNet - 18 is particularly well-suited for tasks involving image classification, including medical imaging tasks such as the detection of abnormalities in X CT-scans. Its moderate depth and computational efficiency make it an attractive choice for various classification tasks, especially when computational resources are limited.

Description and Theory: ResNet - 18 consists of 18 layers, including convolutional layers, batch normalization layers, activation functions (typically ReLU), and pooling layers. The key innovation of ResNet architectures, including ResNet - 18, lies in the introduction of residual blocks. These blocks contain shortcut connections (or skip connections) that skip one or more layers, allowing the network to learn residual functions rather than attempting to directly approximate the desired underlying mapping. Mathematically, the output of a residual block can be expressed as the sum of the input to the block and the output of the block's internal layers. This formulation enables the network to effectively propagate gradients during training, mitigating the vanishing gradient problem and facilitating the training of very deep networks. In ResNet - 18, the architecture consists of several residual blocks, each containing multiple convolutional layers with batch normalization and ReLU activation functions. Additionally, max-pooling layers are interspersed between blocks to downsample feature maps and increase the receptive field of the network.

Overall, ResNet - 18 offers a compelling combination of depth, accuracy, and computational efficiency, making it an excellent choice for various image classification tasks, including the detection of COVID-19. In the subsequent sections of this report, we delve into the specifics of training ResNet - 18 for the binary classification of COVID-19 infection and discuss the experimental results and implications.

Hyperparameters table

	Initial Learning Rate	**MiniBatch Size**	**Validation Frequency**	**Momentum**	**Learning Rate Drop Factor**	**L2Regularization**	**Validation Accuracy**
Try 1	**0.1**	**128**	**60**	**0.9**	**0.2**	**0.0006**	**62.34%**
Try 2	**0.1**	**512**	**1000**	**0.9**	**0.3**	**0.01**	**67.89%**
Best Results	**0.008**	**256**	**500**	**0.87**	**0.3**	**0.001**	**85.59%**



The training of ResNet - 18 demonstrated remarkably fast convergence, which, while initially promising, raised concerns regarding the stability of the trained model. With the chosen hyperparameters and dataset characteristics, the model converged rapidly, requiring only a single epoch for training to reach satisfactory performance levels. However, it's important to note that such rapid convergence can sometimes indicate instability in the training process. Due to time constraints, we accepted the attained results, which were nonetheless satisfactory, and proceeded with the training of subsequent networks. The swift convergence of ResNet18, while advantageous in terms of computational efficiency, necessitates careful consideration of potential overfitting and the robustness of the trained model. Despite these concerns, the obtained performance serves as a promising foundation for further exploration and refinement in subsequent stages of model development.

4. ResNet – 50

ResNet - 50 Overview: ResNet - 50 is a deep convolutional neural network (CNN) architecture that was developed by Microsoft Research in 2015. It is a variant of the popular ResNet architecture, which stands for “Residual Network.” The “50” in the name refers to the number of layers in the network, which is 50 layers deep.

Task Suitability: ResNet - 50 is a popular choice for medical image analysis, such as breast cancer detection, due to its ability to extract deep features from images and achieve high accuracy . It is a convolutional neural network (CNN) architecture that has been pre-trained on a large dataset, allowing it to learn complex patterns and features. ResNet - 50 has been shown to effectively detect suspicious lesions in breast MRI images, with high sensitivity and specificity. It has also been used for feature extraction in histopathology images, achieving state-of-the-art performance in classifying malignant and benign tumors.

Description and Theory: ResNet - 50 is a powerful image classification model that can be trained on large datasets and achieve state-of-the-art results. One of its key innovations is the use of residual connections, which allow the network to learn a set of residual functions that map the input to the desired output. These residual connections enable the network to learn much deeper architectures than was previously possible, without suffering from the problem of vanishing gradients.

The architecture of ResNet - 50 is divided into four main parts: the convolutional layers, the identity block, the convolutional block, and the fully connected layers. The convolutional layers are responsible for extracting features from the input image, while the identity block and convolutional block are responsible for processing and transforming these features. Finally, the fully connected layers are used to make the final classification.

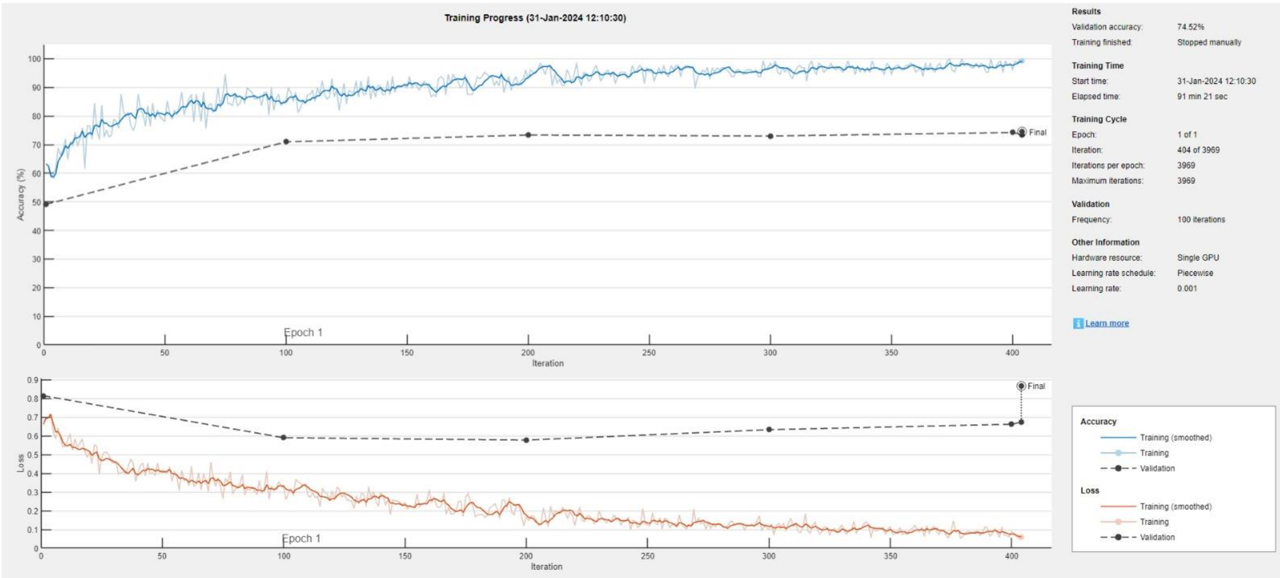
The convolutional layers in ResNet - 50 consist of several convolutional layers followed by batch normalization and ReLU activation. These layers are responsible for extracting features from the input image, such as edges, textures, and shapes. The convolutional layers are followed by max pooling layers, which reduce the spatial dimensions of the feature maps while preserving the most important features.

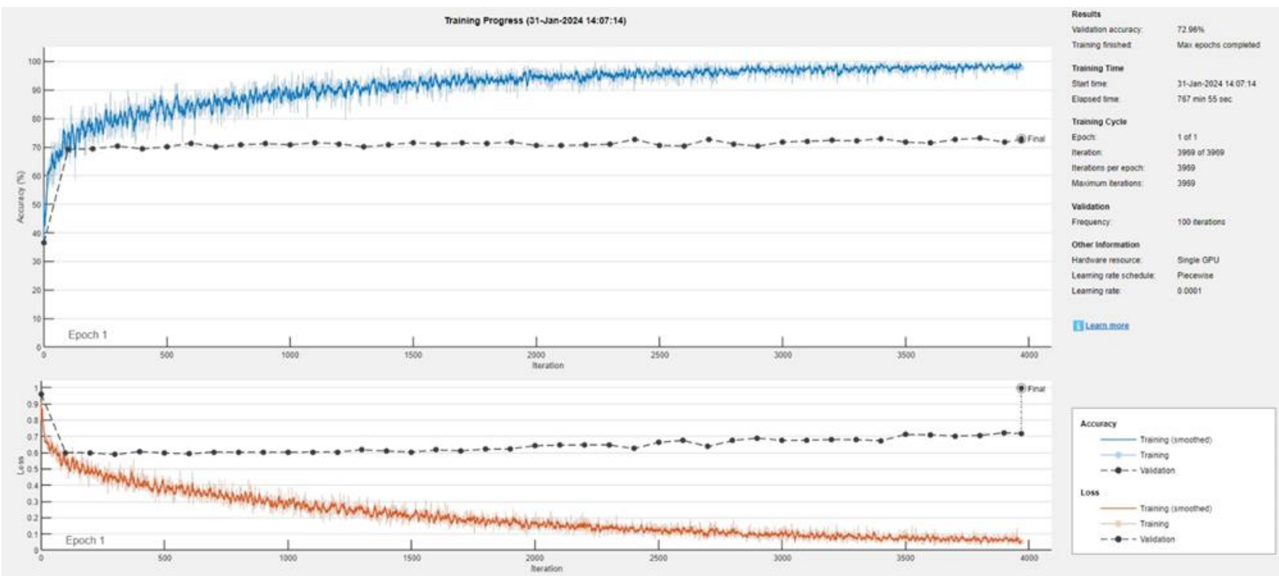
The identity block and convolutional block are the key building blocks of ResNet - 50. The identity block is a simple block that passes the input through a series of convolutional layers and adds the input back to the output. This allows the network to learn residual functions that map the input to the desired output. The convolutional block is similar to the identity block, but with the addition of a 1x1 convolutional layer that is used to reduce the number of filters before the 3x3 convolutional layer.

The final part of ResNet - 50 is the fully connected layers. These layers are responsible for making the final classification. The output of the final fully connected layer is fed into a softmax activation function to produce the final class probabilities.

Hyperparameters table

	Initial Learning Rate	MiniBatch Size	Validation Frequency	Momentum	Learning Rate Drop Factor	L2Regularization	Validation Accuracy
Try 1	**0.001**	**128**	**100**	**0.8**	**0.3**	**0.01**	**74.52%**
Try 2	**0.0001**	**128**	**100**	**0.8**	**0.3**	**0.01**	**72.96%**





The first try was stopped manually as it seemed to have plateaued at around 74% validation accuracy while the training loss kept increasing. It was decided to start a new training cycle after reducing the learning rate from 0.001 to 0.0001 to try and reduce the overfitting, but it seems that the network starts overfitting almost immediately after only 100 iterations. It's likely that, being the task binary, the architecture overfits as it's more suitable for more complex tasks.

5. VGG – 19

VGG - 19 Overview: VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The “deep” refers to the number of layers with VGG -

16 or VGG - 19 consisting of 16 and 19 convolutional layers respectively. The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGG - Net also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures.

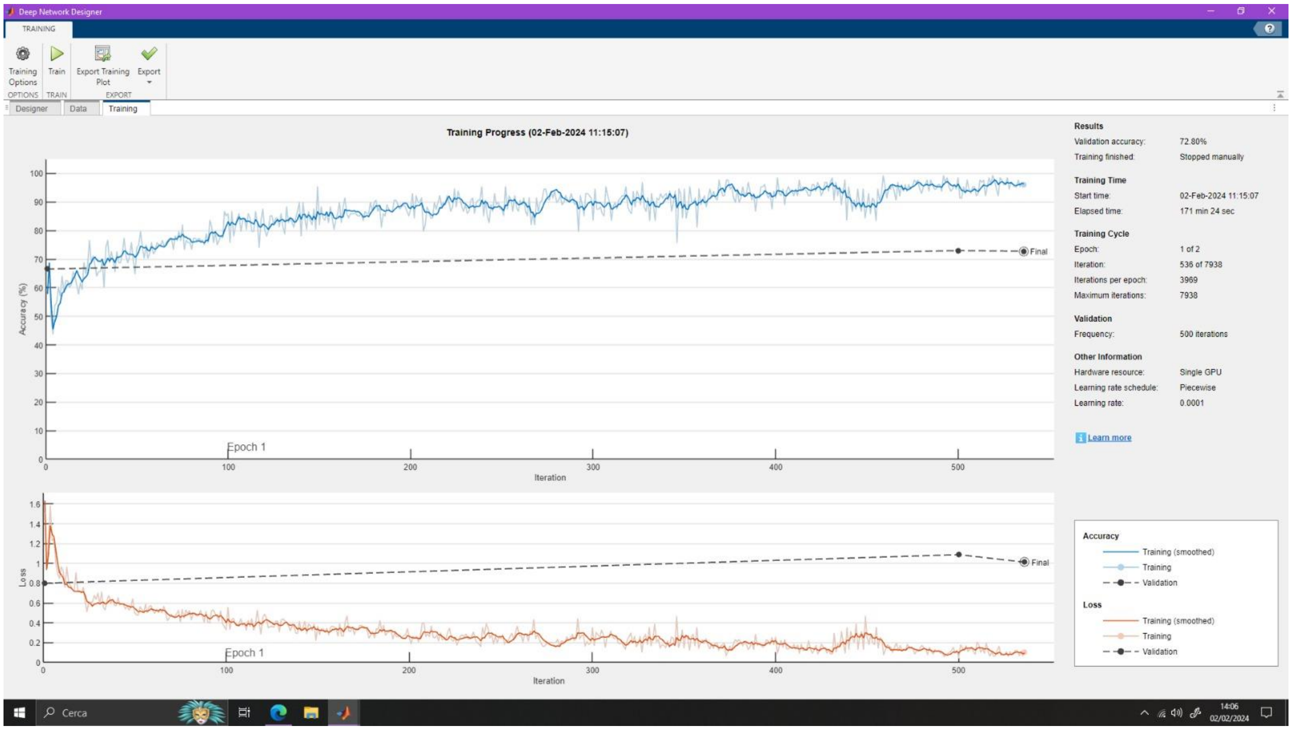
Task Suitability: VGG - 19 excels in Image Classification, Object Recognition, and Computer Vision Tasks.

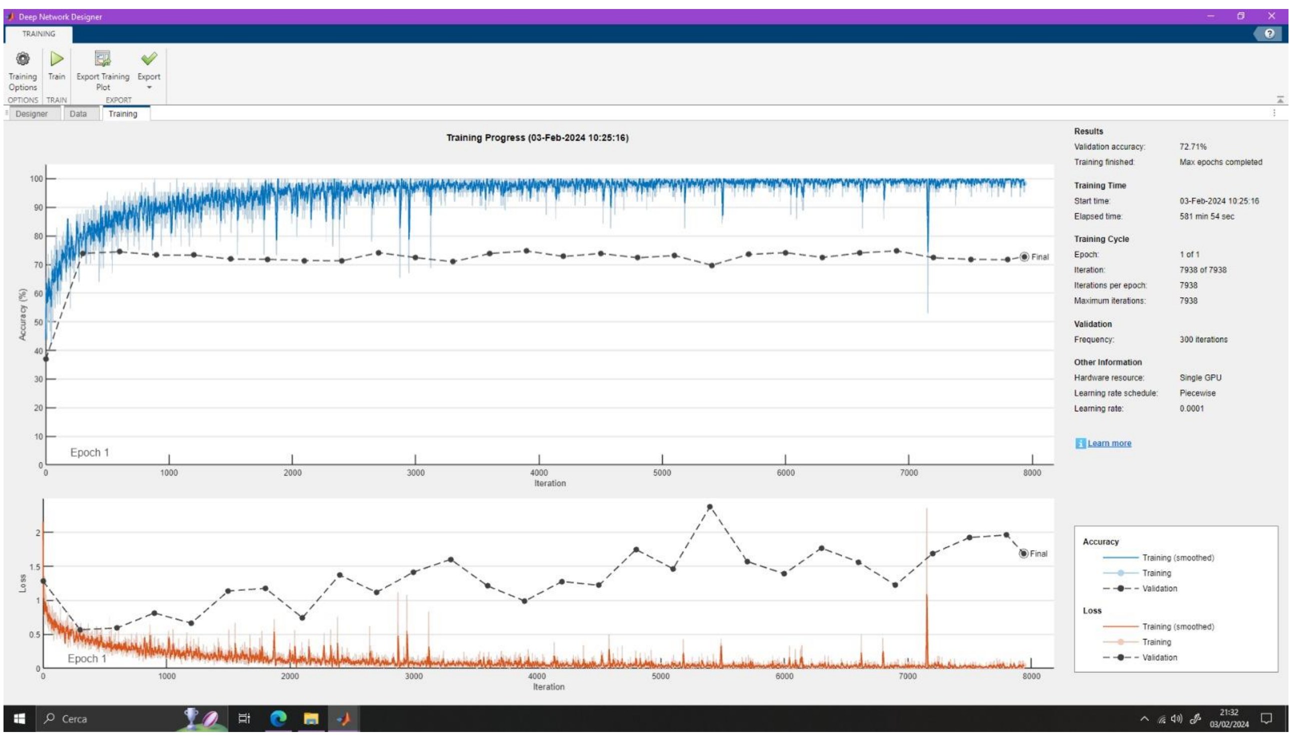
Description and Theory: The VGG network is constructed with very small convolutional filters. The architecture is composed as follows:

Input: The VGG - Net takes in input an image sized 224×224. For the ImageNet competition, the creators of the model cropped out the center 224×224 patch in each image to keep the input size of the image consistent. Convolutional Layers: VGG's convolutional layers leverage a minimal receptive field, i.e., 3×3, the smallest possible size that still captures up/down and left/right. Moreover, there are also 1×1 convolution filters acting as a linear transformation of the input. This is followed by a ReLU unit, which is a huge innovation from AlexNet that reduces training time. ReLU stands for rectified linear unit activation function; it is a piecewise linear function that will output the input if positive; otherwise, the output is zero. The convolution stride is fixed at 1 pixel to keep the spatial resolution preserved after convolution (stride is the number of pixel shifts over the input matrix). Hidden Layers: All the hidden layers in the VGG network use ReLU. VGG does not usually leverage Local Response Normalization (LRN) as it increases memory consumption and training time. Moreover, it makes no improvements to overall accuracy. Fully-Connected Layers: The VGG - Net has three fully connected layers.

Hyperparameters table

	Initial Learning Rate	**MiniBatch Size**	**Validation Frequency**	**Momentum**	**Learning Rate Drop Factor**	**L2Regularization**	**Validation Accuracy**
Try 1	**0.008**	**256**	**500**	**0.87**	**0.3**	**0.001**	**32.14%**
Try 2	**0.0001**	**512**	**500**	**0.9**	**0.3**	**0.01**	**72.80%**
Try 3	**5e-05**	**64**	**300**	**0.89**	**0.3**	**0.01**	**70.66%**
Try 4	**0.0001**	**64**	**300**	**0.5**	**0.3**	**0.1**	**72.71%**





Although the second try registers a slightly higher validation accuracy, the training process was stopped early after only 600 iterations, as it seemed it was plateauing at around 70% and the validation loss had already gone higher than 1. In the last attempt the validation loss keeps increasing throughout the epoch, and it's clear that there is overfitting going on. In the various attempts, learning rate was decreased, regularization increased and momentum decreased, but the network kept overfitting the data. It could be that the task is too simple for the architecture.

Conclusions

This paper presents a fully automated method for detecting Covid-19 from chest CT scans, leveraging deep learning techniques. The study compared the effectiveness of different deep neural networks architectures in the classification of pulmonary medical images.

The applied models demonstrated distinct abilities in recognizing relevant features in CT scans, enabling accurate classification of cases despite some differences in accuracy:

- AlexNet: demonstrated good competence in recognizing features of pulmonary tomographies with a validation accuracy of 86.72%.
- GoogLeNet: shows high computational efficiency. The best result obtained has a validation accuracy of 86.06%.
- ResNet18: exhibited excellent performance with a validation accuracy of 85.59%.
- ResNet50: showed a decent performance with a validation accuracy of 74.52%, although it tends to overfit the training data.
- VGG-19: showed good discrimination ability with a validation accuracy of 72.80%, although it may be sensitive to overfitting effects.

Therefore, we can confidently state that AlexNet emerges as the best model for discriminating Covid-19 and Non Covid -19 classes, but ResNet - 18 and GoogLeNet could well compete with AlexNet.

These results are of great importance as they could benefit clinical practice, with neural networks being usable for COVID-19 diagnosis through the analysis of pulmonary tomographies. However, further research is needed to ensure the robustness and reliability of such systems in the medical context.