Universidad de Concepción
Dirección de Postgrado
Facultad de Ingeniería - Programa de Magíster en Ciencias de la Ingeniería con Mención en Ingeniería Eléctrica

DOMAIN ADAPTATION APPLIED TO THE CLASSIFICATION OF COVID-19 CHEST CT IMAGES.

Tesis para optar al grado de
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA CON MENCIÓN EN INGENIERÍA ELÉCTRICA

POR
VALENTINA PASKAL FUENTEALBA FERNÁNDEZ
CONCEPCIÓN, CHILE
Julio, 2023

Profesor guía: GUILLERMO FELIPE CABRERA VIVES
Departamento de Ingeniería Informática y Ciencias de la Computación
Facultad de Ingeniería
Universidad de Concepción

Contents

List of Figures **iv**

Chapter 1 Introduction **5**

Chapter 2 State of the Art **7**

2.1 Convolutional Neural Networks 7

2.2 Transfer learning and Domain Adaptation 8

2.3 Covid-19 Classification and Transfer learning 11

Chapter 3 Dataset Description **14**

Chapter 4 Hypothesis, Goals and Methodology **15**

4.1 Hypothesis 15

4.2 Goals 15

4.2.1 General Goal 15

4.2.2 Specific Goals 15

4.3 Methodology 16

4.3.1 Preprocessing 16

4.3.2 Models architectures 18

4.3.3 Training process 21

4.3.4 Model evaluation 24

Chapter 5 Results **27**

5.1 Data selection for classification 27

5.2 Performance of non segmented framework 30

5.2.1 ResNet-50 based pipelines. 30

5.2.2 DenseNet-201 based pipelines. 34

5.3 Performance of segmented framework 38

5.3.1 ResNet-50 based pipeline. 39

5.3.2 DenseNet-201 42

Chapter 6 Conclusions and discussion **47**

References **49**

List of Figures

2.1 Example of CNN used for image classification 8

4.1 (a) Original CT Slice (b) Original Mask (c) Masked lung 17
4.2 Training process of nn-Unet. 18
4.3 ResNet building block. 19
4.4 Dense Block with 5 layers and a growth rate of $k = 4$ 20
4.5 Grad-CAM overview. 25

5.1 Example of image used in both frameworks. (a) Original CT image. (b) Segmented CT image where the lung parenchyma is preserved. 28
5.2 Example of image used only on Non-segmented classification framework. (a) Original CT image. (b) Segmented image where the lung parenchyma isn't preserved. 28
5.3 Example of CT slice used in both frameworks. (a) Non segmented image. (b) Segmented image using the U-Net. 29
5.4 Example of CT Slice used only on Non-segmented classification framework. (a) Non segmented image. (b) Segmented image where lung parenchyma isn't preserved. 30
5.5 Visualization of the framework pipeline. 30
5.6 Scheme of the classification pipeline. 31
5.7 Training process on CT Covid curated dataset. 31
5.8 Training process of fine-tuning in HCUCH dataset. 31
5.9 Covid-19 testing image. 32
5.10 Non Covid testing image. 32
5.11 Scheme of the classification pipeline. 33
5.12 Training process on CT Covid curated dataset, fine-tuned on Imagenet dataset. 33
5.13 Training process of fine-tuning in HCUCH dataset. 33

5.14 Covid-19 testing image. 34
5.15 Non Covid testing image. 34
5.16 Scheme of the classification pipeline. 34
5.17 Training process on CT Covid curated dataset. 35
5.18 Training process of fine-tuning in HCUCH dataset. 35
5.19 Covid-19 testing image. 36
5.20 Non Covid testing image. 36
5.21 Scheme of the classification pipeline. 36
5.22 Training process on CT Covid curated dataset. 37
5.23 Training process of fine-tuning in HCUCH dataset. 37
5.24 Covid-19 testing image. 38
5.25 Non Covid testing image. 38
5.26 Visualization of the framework pipeline. 38
5.27 Scheme of the classification pipeline. 39
5.28 Training process on CT Covid curated dataset. 40
5.29 Training process of fine-tuning in HCUCH dataset. 40
5.30 Covid-19 testing image. 40
5.31 Non Covid testing image. 40
5.32 Scheme of the classification pipeline. 41
5.33 Training process on CT Covid curated dataset. 42
5.34 Training process of fine-tuning in HCUCH dataset. 42
5.35 Covid-19 testing image. 42
5.36 Non Covid testing image. 42
5.37 Scheme of the classification pipeline. 43
5.38 Training process on the CT Covid curated dataset. 43
5.39 Training process of fine-tuning in HCUCH dataset. 43
5.40 Covid-19 testing image. 44
5.41 Non Covid testing image. 44
5.42 Scheme of the classification pipeline. 44
5.43 Training process on the CT Covid curated dataset. 45

5.44 Training process of fine-tuning in HCUCH dataset. 45
5.45 Covid-19 testing image. 46
5.46 Non Covid testing image. 46

"I took a deep breath and listened to the old brag of my heart. I am, I am, I am." — Sylvia Plath, The Bell Jar

Acknowledgments

I thankfully acknowledge the support from Agencia Nacional de Investigación through the grant COVID0733.

Abstract

One of the most common uses of convolutional neural networks is image classification. This can be performed in a supervised, semi-supervised and unsupervised manner depending on the available data. This task is specially important in the field of Computer Aided Diagnosis, where many biomedical applications of deep learning are developed and used to aid medical professionals.

As the Covid-19 progressed across the globe, multiple image classification models were developed, most of them using well known networks such as ResNet and DenseNet as a backbone, and in order to save time, were pre-trained on the ImageNet dataset and then fine-tuned to the Covid-19 disease images.

We used a public Covid-19 dataset composed of 7305 subjects in a pre-training and fine-tuning based training framework to classify the disease and GradCAM to visualize saliency maps and visually evaluate if the pipelines in the framework use medically relevant information in the CT slices to make the classification choice.

While the models perform along the line of the state of the art in terms of accuracy, with over 0.9 accuracy, the visualization of saliency maps show that they do not necessarily find relevant imaging findings in the images. This shows the need for more transparency when showing classification results in tasks related to the medical imagining domain, in order to build trust with medical professionals and advance in the computer aided diagnosis field. Other classification methods are proposed, such as adversarial learning and attention based learning, in order to overcome the shortcut learning problem to improve the saliency maps, such as adversarial and self supervised learning.

Chapter 1

Introduction

COVID-19 is a respiratory infectious disease, caused by the novel coronavirus SARS-CoV-2. The common symptoms include fever, sore throat, fatigue and dry cough, although they may vary patient to patient, and their severity depends on multiple factors, such as pre-existing conditions [1].The gold standard to diagnose the disease is the Reverse Transcription Polymerase Chain Reaction (RT-PCR) test, but different imaging techniques, such as computed tomography (CT) may be useful in early assessment and progression of it. The most common findings are expression of acute interstitial lung damage, such as Ground Glass Opacity (GGO), focal consolidation and crazy paving, frequently located on the posterior region of the lungs [2].

The use of Artificial Intelligence (AI) has become a tool in the management of the pandemic, with uses that vary from contact tracing, monitoring and early diagnosis with the ultimate goal to reduce the burden placed on healthcare professionals [3]. One of the most common applications is image classification using Convolutional Neural Networks [4]. Classification models find a mapping function from the training dataset and predicts a label for a new input based on said mapping function. In this context, a feature is a parameter in the problem set that can help build a predictive model [5].
Image classification can be performed in three manners: fully supervised, where the labels are all available during training, semi-supervised, where there is a mixture of labelled and unlabelled data and unsupervised, where there is no labelled dataset to train the model [6].
Classification models can be improved with different methods, one of them is transfer learning, a method inspired in the way that humans learn: people can solve brand new problems using knowledge acquired before [7]. One of the most commonly used datasets

for transfer learning is ImageNet [8], but it can not be blindly used for all tasks, because if the target domain is too different from it, using it to pre-train a network can decrease its performance [9].
Another important aspect of classification is explainability, particularly in the field of computer aided diagnosis (CAD). Medical professionals need to trust the algorithms in order to make decisions with them [10]. Multiple methods have been developed based on class activation mapping: Class activation mapping (CAM), Gradient-weight class activation mapping (Grad-CAM), and other approached based on these ones [4].

This thesis evaluates the results of performing classification with models fine-tuned with public data, on a locally obtained dataset CT dataset with labels Covid-19 or Non Covid in terms of classification performance (accuracy, precision, recall and f1 score) and visual results (saliency maps over the input images).

Chapter 2

State of the Art

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) (Figure 2.1) are one of the most commonly employed deep learning algorithms. They are inspired by neurons in human and animal brains: simulating a complex sequence of cells, called neurons, that form the visual cortex [11]. The CNN architecture consists of multiple layers, also called multi-building blocks [11]. Each layer is composed of:

- **Convolutional layer**: A collection of filters, also named kernels. The input image, expressed as N-dimensional metrics, is convolved with these filters to generate the output feature map. They work on a hierarchical manner: Low level features are extracted in the initial layers and high level features are extracted in the deeper ones.
- **Pooling layer**: This layer reduces the dimension of the feature maps, maintaining the majority of the dominant information while reducing computational complexity.
- **Activation function**: Defines the output of a node given an input or set of inputs. The most commonly used are: Sigmoid, ReLU [12], Tanh, leaky ReLU [13]. These functions, decide whether to fire or not a neuron in reference to a particular input by creating the corresponding output.

Another type of layer is the Fully Connected layer, located at the end of the architecture and acts as a classifier. All neurons are connected to the previous layer.

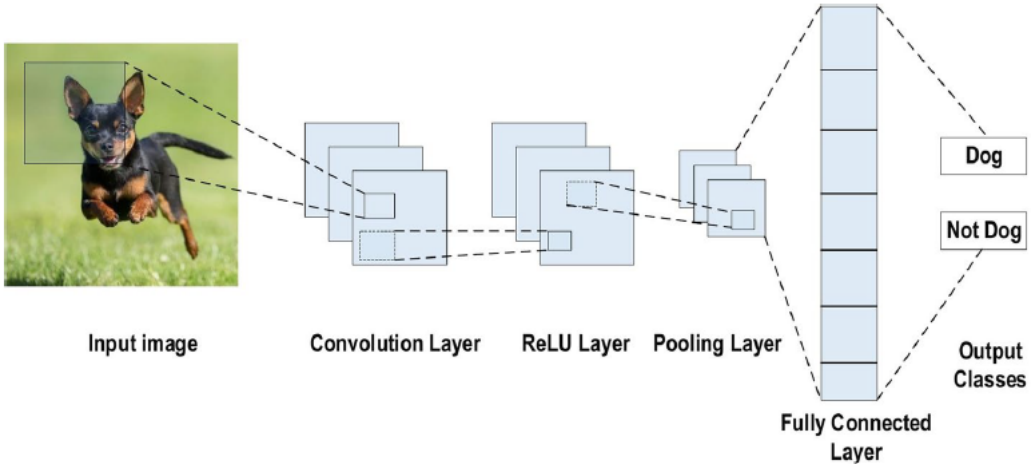


Figure 2.1: Example of CNN used for image classification
Extracted from [11].

2.2 Transfer learning and Domain Adaptation

Traditional machine and deep learning models make predictions on new data using trained models on previously collected labeled or unlabeled data. Meanwhile Transfer learning, allows domains, tasks and distributions used in training and testing to be different [7]. Domain is defined as the feature space of a specific dataset and the marginal probability distribution of features, and a task relates to the label space of a dataset and an objective predictive function [14].
There are multiple categories of transfer learning, categorized depending on the kind of source and target domains and tasks [7].

- **Inductive transfer learning**: In this setting, the target task is different from the source task, but the domains are the same.
- **Transductive transfer learning**: In this setting, the source and target tasks are the same, while the source and target domains are different but can be related.
- **Unsupervised transfer learning**: In this setting, the target task is different but related to the source task, but the focus is solving unsupervised learning tasks in the target domain.

There are multiple areas related to the different settings of transfer learning, as shown

in Table 2.1.

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels
Inductive Transfer Learning	Multi-task Learning	Available	Available
	Self-taught Learning	Unavailable	Available
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Avaible	Unavailable
Unsupervised Transfer Learning		Unavailable	Unavailable

Table 2.1: Different Settings of Transfer Learning

As seen above, Domain adaptation (DA) is a special case of transfer learning, in which it is assumed that the domain feature spaces and tasks are the same in both source and target domains [14].

Just as Transfer learning, DA can be categorized in different groups according to different problems and constrains [14].

- **Shallow and Deep DA**: Depends on the learning model, Shallow DA methods rely on traditional machine learning methods. Meanwhile, Deep methods are applied on Deep learning networks, specially those with convolutional architectures.

- **Supervised, Semi-Supervised and Unsupervised DA**: Depends on label availability on the target domain, in supervised DA there is labelled data available during training, in semi-supervised DA there is both labelled and unlabelled data and in unsupervised DA there is only unlabelled data available during training.

- **Single-Modality DA and Cross-Modality DA**: In single-modality DA, the source and target domain share the same modality, for example both domains are images. Meanwhile, cross-modality involves training using multiple source domains.

- **One-Step DA and Multi-Step DA**: In one-step DA, adaptation between source and target domains is achieved in one step due to the close relationship between them. In multi-step DA there are intermediate domains that bridge the distribution gap between domains.

In this work, we'll focus on deep domain adaptation methods on medical image analysis.
In supervised methods, a widely used strategy is to transfer models learned on the source domain onto the target domain with fine tuning. Ghafoorian *et al.* [15]. evaluates the

impact of fine tuning on brain lesion segmentation applied on CNN models pre-trained on brain MRI images, they found that using a small number of target training examples for fine tuning improves transferability of models. They also evaluate how the size of the target training set and and different network architectures influence the models performance. Most of the supervised domain adaptation with fine tuning is done using the ImageNet dataset. Samala *et al.* [16] pre-trains an AlexNet [17] based network and then fine tunes it with breast cancer lesions to classify them. Khan *et al.*[18] pre-trains a VGG Network and then uses labeled MRI data to fine-tune it for Alzheimer's Disease, they also propose to use image entropy to select the most informative training samples in the target domain. Swati *et al.* [19] also pre-trains a VGG Network, but re-trains the higher layers with MRI images for brain tumor classification. Abbas *et al.*[20] trains CNNs for x-ray classification, in order to deal with irregularities on the datasets, they propose a class decomposition method in the learning process, by partitioning each class within the image dataset into k subsets and then assign new labels to the new set.

In the semi-supervised setting, Madani *et al.*[21] proposes a generative adversarial network (GAN) based framework for chest X-ray image classification. The model takes labelled source data, unlabeled target data and generated images as input and the discriminator differentiates between those three image categories and during training, only the ones classified as generated contribute to loss computation.

In the unsupervised setting, one of the strategies is to learn domain-invariant features across domains through specific CNN architectures. Most models adopt architectures similar to the Domain Adversarial Neural Network (DANN) [22]. Yang *et al.*[23] directly uses the DANN for lung texture classification. Zhang *et al.*[24] proposes an adversarial learning based DA for Alzheimer's disease and Mild cognitive impairment classification, co-training the classifier and a domain discriminator to improve the model's transferability across domains. Shen *et al.*[25] proposes another adversarial learning based method for mammogram detection, co-training a segmentator based on a fully convolutional network and a domain discriminator. Wang *et al.*[26] proposes a method for lesion detection within

the Faster RCNN [17] framework, performing global feature alignment and extracting Region of Interest proposal to facilitate local feature alignment. Anh *et al.* [27] proposes a hierarchical auto-encoder placed atop of a pre-trained AlexNet, constraining the features from the pre-trained domain to a reliable representation of the unlabelled medical images.

2.3 Covid-19 Classification and Transfer learning

Transfer learning is widely used in the classification of COVID-19 using Convolutional Neural networks, especially in the shape of fine-tuning [4].
On CT images, Wang *et al.*[28] preprocessed the images in order to obtain ROIs based on the features of COVID-19, and trained a model based on a pre-trained GoogleNet Inception v3 CNN[29] to classify the images as COVID-19 or not. Bai *et al.*[30] conducted a multicenter study to classify patients as COVID-19 versus other pneumonia using the EfficentNet B4 [31] architecture on manually segmented lung images, the model was pre-trained using the ImageNet dataset. A heat map for important images regions that led the model to classify the case as COVID-19 or non COVID-19 was generated using Grad-CAM. Both of these works found that the use of IA frameworks improves radiologists' performances when diagnosing COVID-19 on CT images.

Li *et al.*[32] used a transfer learning-based DenseNet-121 [33] approach for the identification of COVID-19, using the CheXNet [34] as the pre-trained network on chest X-Rays and fine-tuning its weights to the small-sized dataset in the target task, the CT domain. Lahsaini *et al.*[35] collected a dataset of 4986 COVID and non-COVID images from the Tlemcen hospital and fine tuned different models pre-trained on the ImageNet dataset. They propose an explainable model based on the DenseNet-201[33] architecture and the Grad-CAM explanation algorithm to detect COVID-19. Ko *et al.* [36] developed FCONet, a 2D deep learning framework based on a single chest CT image with no segmentation applied. The model uses a ResNet-50 pre-trained on the ImageNet dataset as backbone to classify CT slices as Covid-19, Other-Pneumonia or Non-Pneumonia. They also used Grad-CAM to improve interpretability of the results.

Pathak*et al.*[37] used a transfer learning based ResNet-50 [38] architecture, using the ImageNet dataset and then fine-tuning the model to the Covid-19 and control patients CT dataset. They use the MCX-ENT [39] as loss function in order to improve the classifier's performance. This function has cost sensitive attributes, which allows it to handle noise and imbalanced datasets.

Turkoglu M.[40] developed a Multiple Kernels-ELM-based model for the binary classification of COVID-19 or not on CT images. The model uses a DenseNet-201 architecture pre-trained on the ImageNet dataset to extract the CT images' features and an Extreme Learning Machine (ELM) classifier based on different activation methods is used to calculate the architecture's performance. The activation methods used are ReLU, PReLU, and TanhReLU.

Results of the state of the art works are shown in Table 2.2.

Pre-training using the ImageNet dataset and then fine-tuning to the Covid-19 dataset is a widely used technique, but there is not much domain adaptation focused on single modality images like CT images, which are becoming more available in public datasets.

Work	Architecture	Model details	Results
Wang *et al.*	GoogleNet Inception	Preprocesses the images to obtain Covid-19 Regions of interest.	0.895 accuracy and 0.88 sensitivity.
Bai *et al.*	EfficentNet B4	Manually segmented lung images.	0.96 accuracy, 0.95 sensitivity and 0.96 precision.
Li *et al.*	DenseNet-121	Fine-tuned the weights of the ChexNet to the target task.	0.87 accuracy
Lahsaini *et al.*	DenseNet-201	Fine-tuned multiple models.	0.9818 accuracy, 0.982 precision and 0.982 sensitivity.
Ko *et al.*	ResNet-50	Proposed the FCONet framework, based on a single chest CT with no segmentation applied.	0.9987 accuracy and 0.9958 precision.
Pathak *et al.*	ResNet-50	Uses the MCX-ENT as a loss function.	0.9705
DenseNet-201	CT Covid Curated	Segmented	0.9649 accuracy, 0.929 sensitivity and 0.9345 precision.
Turkoglu *et al.*	DenseNet-201	Extracts features using the DenseNet-201 and uses an Extreme Machine Learning classifier based on different activation methods.	0.9836 accuracy.

Table 2.2: State of the art models performance.

Chapter 3

Dataset Description

Three different medical image datasets will be used in this project:

- **NSCLC-radiomics dataset** [**41**]: This dataset contains images of 422 non-small cell lung cancer (NSCLC) patients, it's made of CT scans in DICOM format with a resolution of 512x512 a 3.0 mm thickness taken in a Siemens Biograph 40 / 64 True Point PET CT Scanner. This dataset also includes lung and nodule annotations for all subjects.

- **Curated Covid CT** [**42**]: This dataset contains CT images with varying sizes in png format of 466 Covid-19 positive patients and 6839 normal subjects, making in total 14486 slices obtained from 7 public datasets widely used in literature. The images were selected focusing on the presence of lung parenchyma.

- **Locally extracted dataset (HCUCH dataset)**: 44 non-contrast and 21 contrast-enhanced CTs of Covid-19 patients were extracted, acquired between March 2020 and June 2021. There were 24 CT and 14 contrast enhanced CTs of patients with non Covid-19 diagnosis acquired before December 2019. Acquisition was accomplished through a SOMATON Definition Edge CT scanner from Siemens with a resolution of 512x512 with 1.5 mm thickness at the Hospital Clínico de la Universidad de Chile (HCUCH). The dataset collection was authorized by the ethics comitee at the Hospital Clínico de la Universidad de Chile, associated to the project COVID0733 funded by the Asociación Nacional de Investigación.

Chapter 4

Hypothesis, Goals and Methodology

4.1 Hypothesis

Using a convolutional neural network based classifier it is possible to identify 2D chest CT images as Covid-19 or Non Covid for a real case scenario over the HCUCH dataset, with the use of domain adaptation from publicly available datasets to improve classification results. The results can be shown in a explainable manner in order to provide an explainable classification model.

4.2 Goals

4.2.1 General Goal

Adapt the domain of a deep learning classification model using publicity available Covid-19 and Non Covid CT images in order to improve the model's performance on the HCUCH dataset.

4.2.2 Specific Goals

SG1 Preprocess and select CT images from public and HCUCH datasets.

SG2 Fine-tune classification models on the HCUCH dataset, using architectures pre-trained on the ImageNet dataset as backbone.

SG3 Fine-tune models on the HCUCH dataset, using architectures pre-trained on public dataset.

SG4 Fine-tune models pre-trained on the ImageNet dataset to the public dataset and

then fine-tune over the HCUCH dataset.

SG5 Compare model performance and apply Grad-CAM to visualize salient maps.

4.3 Methodology

To complete this work the following steps were performed:

4.3.1 Preprocessing

Data processing

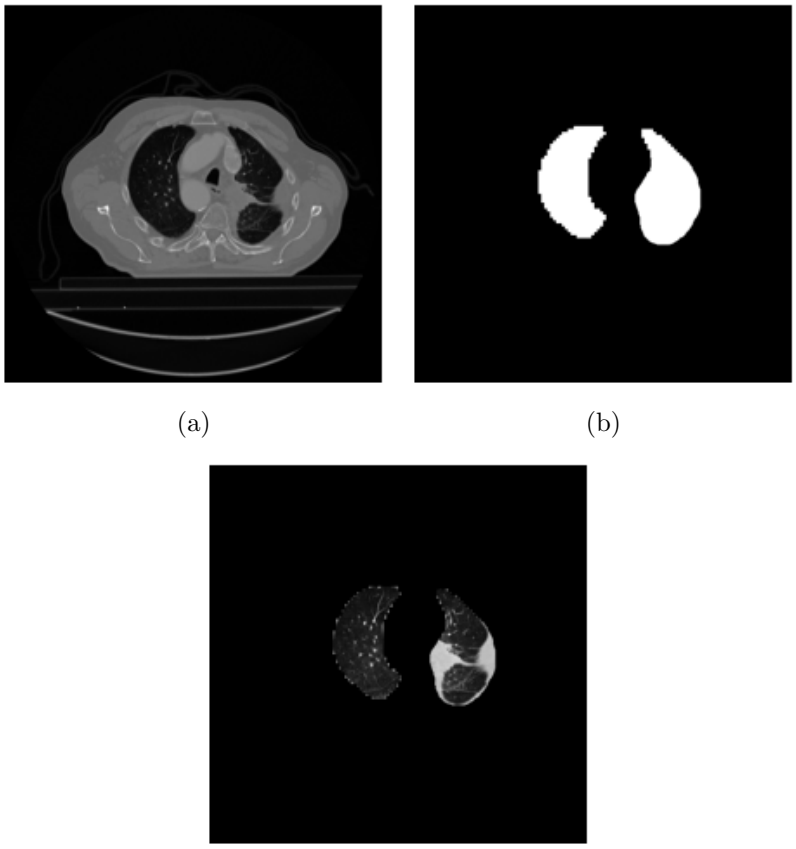
The CT scans from the NSCLC and HCUCH datasets were adjusted to a -1250 to 250 Hounsfield Unit (HU) lung window to have better view of the lung parenchyma and then turned into individual 2D slices normalized between 0 and 1 in nifti format to perform segmentation, and in png format to perform classification.
The Curated Covid CT images were already adjusted to the hounsfield unit interval and in png format, with this in mind, they were normalized between 0 and 1 and transformed to nifti format to perform segmentation, and the images that were not 512x512 were resized.

Image Segmentation

To perform image segmentation, the nnUnet [43] framework was selected due to its nature: a semantic segmentation method that automatically adapts to a dataset. The method, in a step called experiment planning, takes intrinsic information from the data, such as median shape of the images, distribution of spacing, intensity distribution and image modality to select optimal parameters. Some parameters such as Learning rate, loss function, optimizer and data augmentation are set by default, but others like network topology, patch size and batch size depend on median shape, target spacing and GPU limit: The patch size is initialized to the median image shape and iteratively reduced while adapting the network topology until the network can be trained with a batch size according to GPU memory [44] (see Table 4.1).

The model was trained on the NSCLC dataset, selecting 8866 CT slices for training,

886 for validation and 866 for testing. The NSCLC images were originally labelled for lung parenchyma and cancer nodule, but with the goal of lung segmentation, these labels were fused together (see Figure 4.1). The model achieved a 0.97 dice coefficient in training (see Figure 4.2) and 0.95 in validation and 0.93 in the test dataset.



(a) (b)

(c)

Figure 4.1: (a) Original CT Slice (b) Original Mask (c) Masked lung

Parameter	**Value**
Batch Size	12
Pooling kernel size	2x2
Convolutional kernel size	3x3
learning rate	0.001
Loss function	Dice and cross-entropy
Optimizer	Adam 0.003
Number of epochs	100

Table 4.1: U-Net training parameters parameters

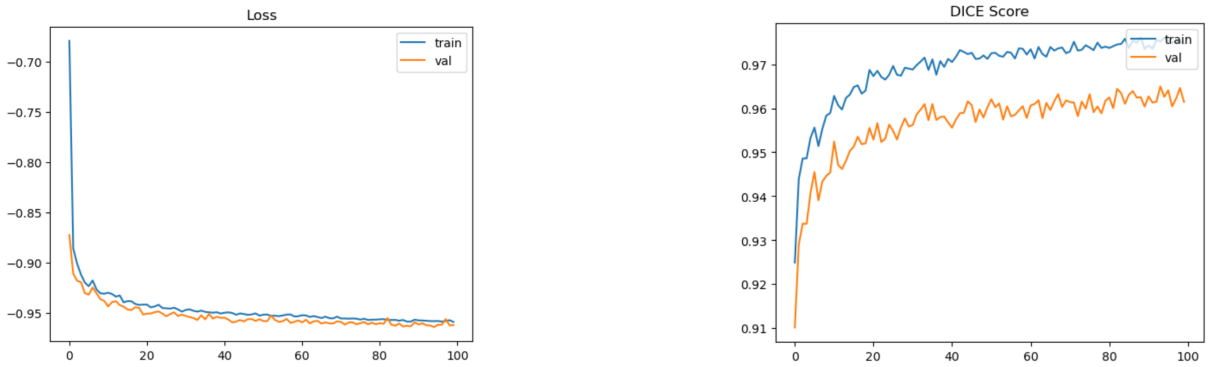


Figure 4.2: Training process of nn-Unet.

A blind segmentation is performed on the CT Covid curated and HCUCH datasets due to the lack of lung annotations for these datasets, and to select the images used during the segmented classification, a visual selection is performed, this is described in dept in section 5.1.

4.3.2 Models architectures

Based on the state of the art review (see Chapter 2), two of the most used model architectures were used on this project. Both of these networks are well suited for image classification, and they solve the vanishing gradient problem in an explicit manner.

Residual Neural Network

He *et al.* [38] designed the Residual Neural Network (ResNet) in order to create a very deep model without the vanishing gradient issue. This is a conventional feed-forward network with a residual connection (see Figure 4.3).

A building block is defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{4.1}$$

In Equation 4.1, x and y are the input and output vectors of the layers considered and the function $\mathcal{F}(x, \{W_i\})$ represents the residual mapping to be learned. The operation

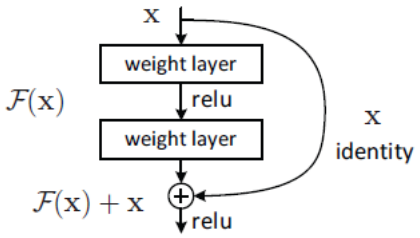


Figure 4.3: ResNet building block.
Extracted from [38].

$\mathcal{F}+x$ is performed by a shortcut connection and element-wise addition. This doesn't add extra parameters nor computation complexity.
For the example of Figure 4.3 that has two layers, $\mathcal{F} = W_2\sigma(W_1x)$ in which σ denotes ReLU and the biases are omitted to simplify notation. It adopts the second nonlinearity after the addition ($\sigma(y)$).
The dimensions of $\mathcal{F}$ and x must be equal, if they are not, a linear projection can be performed W_s by the shortcut connection to match the dimensions:

$$y = \mathcal{F}(x, \{W_i\}) + W_sx \tag{4.2}$$

The function $\mathcal{F}(x, \{W_i\})$ can represent multiple convolutional layers. The element wise addition is performed on two feature maps, channel by channel.

Multiple types of ResNet were developed based on the number of layers, starting from 34 and up to 1202, but the most common type is ResNet-50, made of 49 convolutional layers and a single fully connected layer.

Densely Connected Convolutional Network

Huang *et al.* [33] developed a Densely Connected Convolutional Network (DenseNet) as a way to solve the problem of the vanishing gradient. To ensure maximum information flow between the layers in the network, they connect all layers (with matching feature-map sizes) directly with each other (see Figure 4.4).

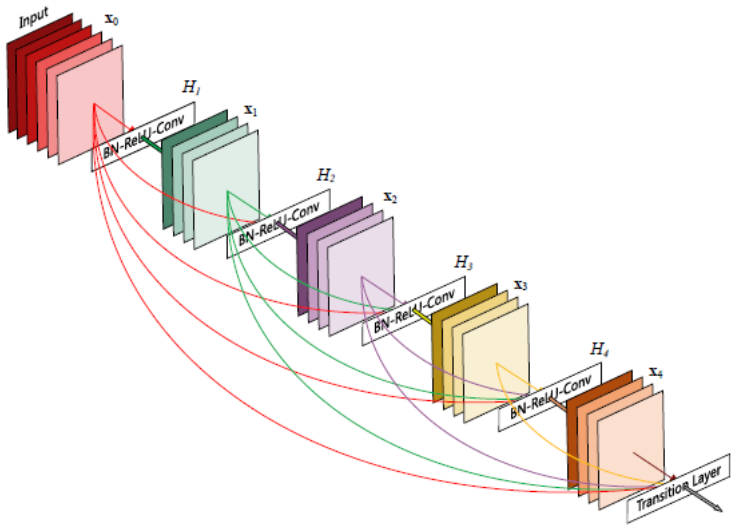


Figure 4.4: Dense Block with 5 layers and a growth rate of $k = 4$
Extracted from [33].

Considering a single image x_0 that is passed through a convolutional network of L layers, each of which implements a non-linear transformation $H_l(\cdot)$. This can be a composite function of operations such as ReLu, Convolution (conv), Batch Normalization (BN) [45] or Pooling [46]. The output of l^{th} layer is denoted as x_l. Figure 4.4 illustrates the direct connection between layers. The l^{th} layer receives the feature-maps of all preceding layers as inputs,

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]) \tag{4.3}$$

where $[x_0, x_1, ..., x_{l-1}]$ is the concatenation of the feature-maps produced in layers 0,1,...,l-1. $H_l(\cdot)$ is defined as a composite function of three consecutive operations: Batch Normalization, ReLU and a 3x3 convolution. The transition layers ensure the downsampling of the feature-maps, doing convolution and pooling. The growth rate (k) regulates how much new information each layer contributes to the global state. In the original work it was shown how a small growth rate allows state-of-art results on the tested datasets. There are multiple developed and widely used DenseNet based on the number of layers going from 121 to 264.

4.3.3 Training process

The classification models based on the ResNet-50 and DenseNet-201 were trained using the Google Colaboratory Pro platform (Google, California), using GPU NVIDIA T4 with 32 GB RAM.

Model adjustments

The architectures are already implemented on the Keras library and pre-trained on the ImageNet dataset. Considering this, the models were stripped of their in built fully connected layer and new layers were added to fit the problem (see Table 4.2).

Layer name	Description
Avgpool	2D Global average pooling
Dense1	Fully connected layer of 1024 neurons with a sigmoid function activation
Dense	Fully connected layer of 1 neuron with a sigmoid activation function, outputs "Covid-19" or "Non Covid"

Table 4.2: Layers added to the baseline architectures.

Optimizer

Adam is an optimizer algorithm that's used to update network weights based in training data. The name is derived from Adaptive moment estimation [47]. The method computes individual adaptive learning rates for different parameters from estimates of the first and second moments of the gradients, the mean and the uncentered variance [47]. It requires the parameters:

- α: Step size, larger values result in faster initial learning before the rate is update, and smaller values result in slow learning during training.
- β_1: The exponential decay for the first moment estimates. Must be between 0 and 1.
- β_2:The exponential decay rate for the second-moment estimates. Must be between 0 and 1.
- ϵ: Small number to prevent division by 0.

The algorithm is explained in pseudo code in the table 4.3.

Proposed algorithm for stochastic optimization. g_t^2 indicates element wise $g_t \odot g_t$. All operations are element wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power of t.
Require: $f(\theta)$: Stochastic function with parameters θ.
Requiere: θ_0 Initial parameter vector.
$m_0 \leftarrow 0$ (initialize first moment vector)
$v_0 \leftarrow 0$ (initialize second moment vector)
$t \leftarrow 0$ (initialize time step)
while θ_0 not converged **do**:
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Table 4.3: Adam algorithm.
Extracted from [47].

Loss Function

The training stage is governed by a loss function, Binary Cross-Entropy loss with the aim to minimize the probability of a negative class by maximizing an expected value of some function on the training data [6].

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{4.4}$$

Where y is the label and $p(y)$ is the predicted probability of the label being positive for all N points.

Data Augmentation

Since there is a limited amount of images, data augmentation was performed using the in built Keras package functions. This function allows to perform horizontal and vertical flips, rotate the images in 10° angle and randomly change brightness on a scale from 0.2 to 1.0 on the original image. This function uses nearest neighbour interpolation to fill the

images.

4.3.4 Model evaluation

Accuracy

For tasks like classification, Accuracy is one of the most used measuring methods, the proportion of examples for which the model produces the correct output [6].
For binary classification is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

Where True Positive (TP) is the data that was labelled as positive and classified as such, True Negative (TN) is the data is labelled as negative and was classified as such, False Positive (FP) is the negative data misclassified as positive and False Negative (FN) is data labelled as positive was wrongly classified as Negative.

Precision and Recall

The precision expresses how much a model can be trusted when it predicts a positive outcome [48].

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

Meanwhile, Recall measures the model's predictive accuracy for the positive class [48]. In the medicine related research field, this measure is also called *Sensitivity*, it measures the probability to correctly label a sick patient (data sample) as such [49].

$$Recall = \frac{TP}{TP + FN} \tag{4.7}$$

F1-Score

The F1-score can be interpreted as the harmonic mean of the precision and recall [48].

$$F1 - score = 2 \cdot (\frac{Precision \cdot Recall}{Precision + Recall}) \tag{4.8}$$

Gradient-weighted Class Activation Mapping

Selvaraju *et al.* [50] proposes Gradient-weighted Class Activation Mapping (Grad-CAM), a class-discriminative localization technique that can generate visual explanations from any CNN network. The goal is to make explainable models that users can trust and rely on, this is specially important in the computer aided diagnosis field, and in the case of this project: A visual explanation of the classification can cue physicians where the disease COVID-19 is in the lung parenchyma without delving into semantic segmentation models, which are more expensive to train due to the time it takes to obtain manually segmented images.

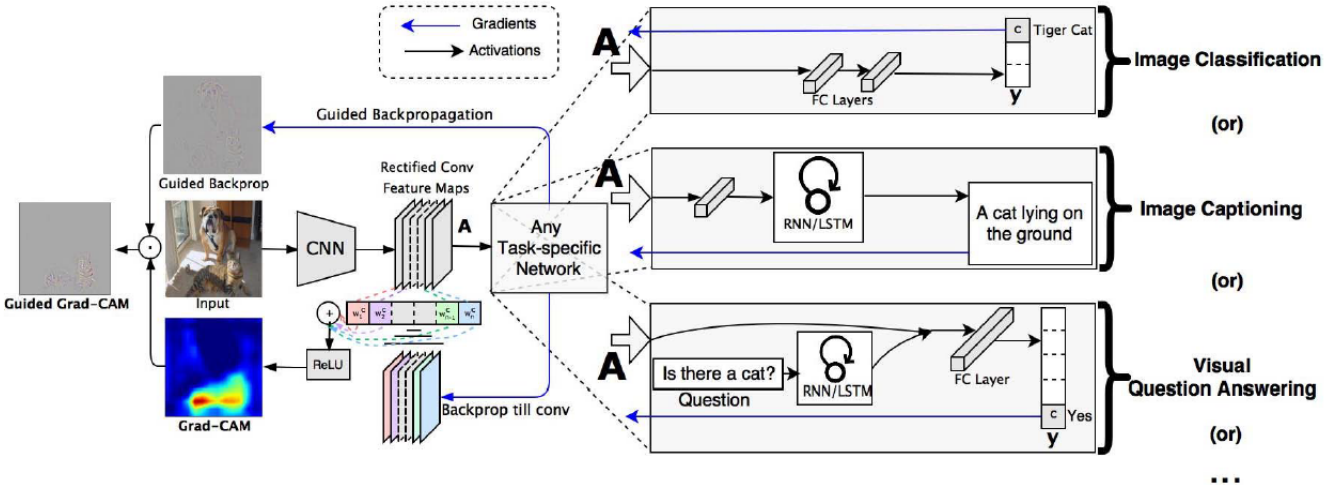


Figure 4.5: Grad-CAM overview.
Extracted from [50].

As shown in Figure 4.5, in order to obain the class discriminative localization map Grad-CAM of width u and height v for any class c, the gradient of the score for class c, y^c is computed (before the softmax), with respect to the feature maps A^k of a convolutional layer. These gradients flowing back are average-pooled to obtain the neuron importance weights α_k^c:

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{i\,j}^k}}_{\text{gradients via backpropagation}} \tag{4.9}$$

The weight α_k^c represents a partial linearization of the deep network downstream from A,

and captures the importance of feature map k for a target class.

With a weighted combination of forward activation maps, followed by a ReLU a coarse heat-map of the same size as the convolutional feature maps is obtained. This function is applied to the linear combination because the interest is only in the features that have a positive influence on the class of interest. Negative pixels are likely to belong to other categories in the image.

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k) \tag{4.10}$$

Chapter 5

Results

5.1 Data selection for classification

CT Covid curated dataset

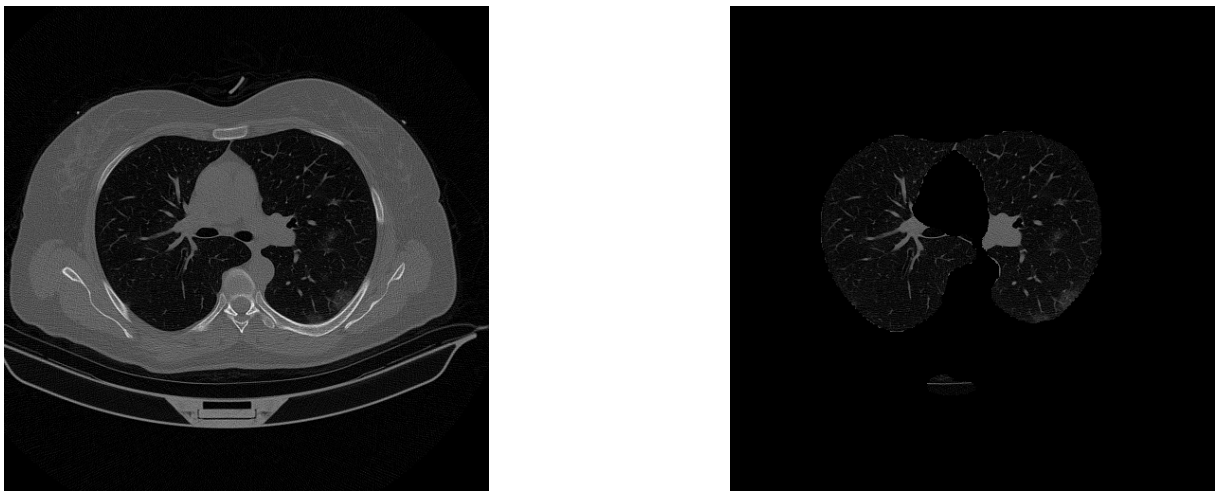
Since all the images contain lung parenchyma, and in the case of the covid subset, relevant patterns to the disease, there was no need to visually select images in the non-segmented classification framework (see Table 5.1). For the segmented image classification framework, a few images were excluded due to segmentation errors detected visually, such as no preservation of the lung outline or other types of artifacts (see Figures 5.1 and 5.2). In total, 7590 Covid-19 and 6893 Non Covid images were used in the Non-segmented image classification framework. Meanwhile, 7430 Covid-19 and 6856 Non Covid images were used in the Segmented image classification framework. All the images were resized to 256 x 256 pixels in order to reduce execution time.

Table 5.1: Non-segmented framework

Split	**Number of images**
Training	11587
Validation	1448
Testing	1448

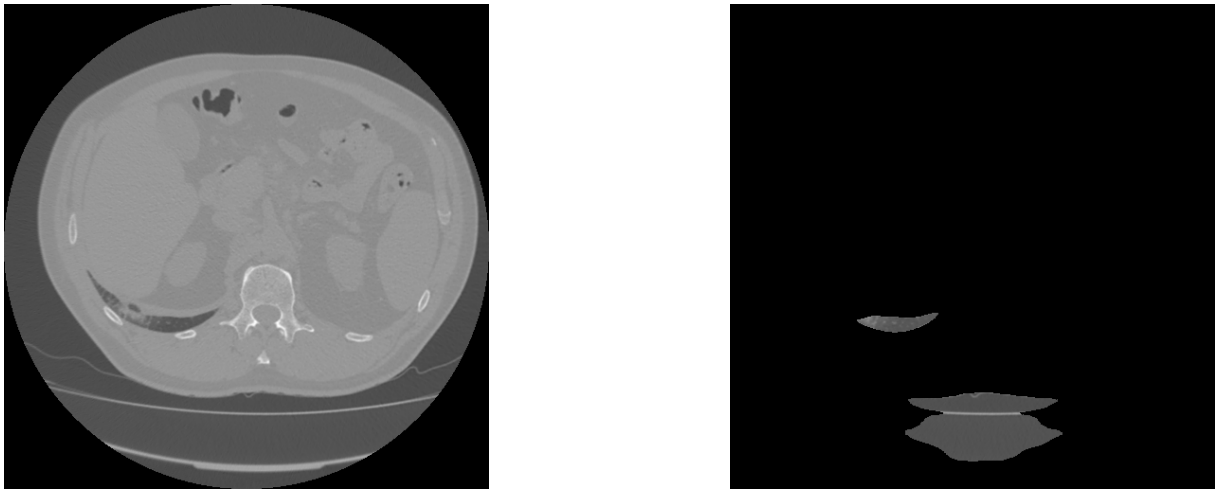
Table 5.2: Segmented framework

Split	**Number of images**
Training	11390
Validation	1448
Testing	1448



(a) (b)

Figure 5.1: Example of image used in both frameworks. (a) Original CT image. (b) Segmented CT image where the lung parenchyma is preserved.



(a) (b)

Figure 5.2: Example of image used only on Non-segmented classification framework. (a) Original CT image. (b) Segmented image where the lung parenchyma isn't preserved.

HCUCH dataset

For the Non-Segmented classification framework, CT slices were selected if there was any lunch parenchyma present (see Figure 5.3). This was necessary because at the beginning and the end of the 3D CT scans were structures of no interest for this project, such as slices of the neck of the patient. Meanwhile, for the Segmented classification framework,

the images previously selected were segmented with the U-Net, and the ones that didn't pass visual supervision, where the shape of lung is lost or elements outside of it remain after segmentation, were deleted (see Figure 5.4). All the images were resized to 256x256 pixels to reduce training time.

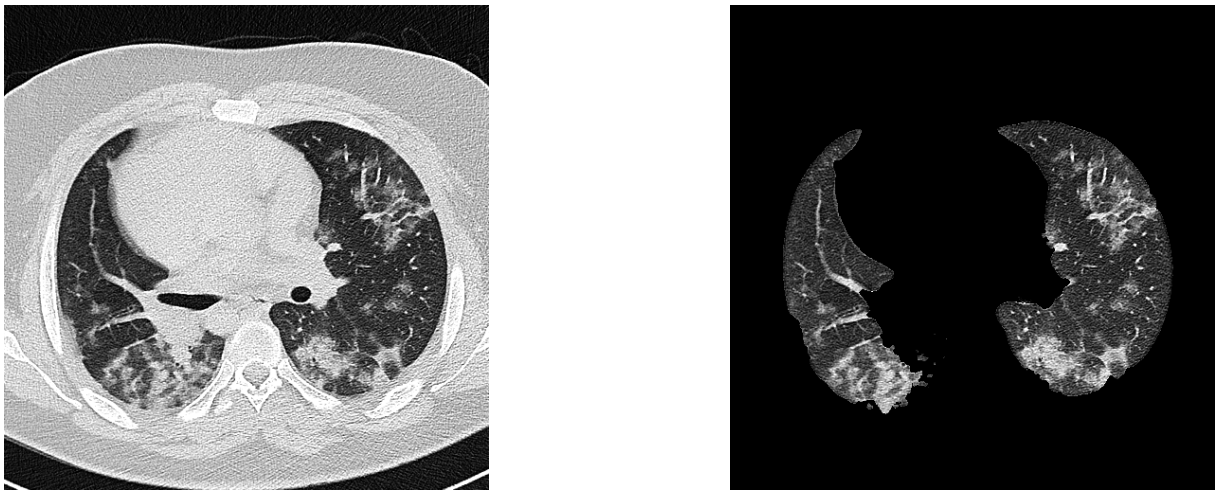
In total 4631 Covid-19 and 4515 Non Covid images were used in the Non-segmented classification framework (see Table 5.3). Meanwhile, 4540 Covid-19 and 4482 Non Covid images were used in the Segmented classification framework (see Table 5.4).

Table 5.3: Non-segmented framework

Split	**Number of images**
Training	7317
Validation	915
Testing	914

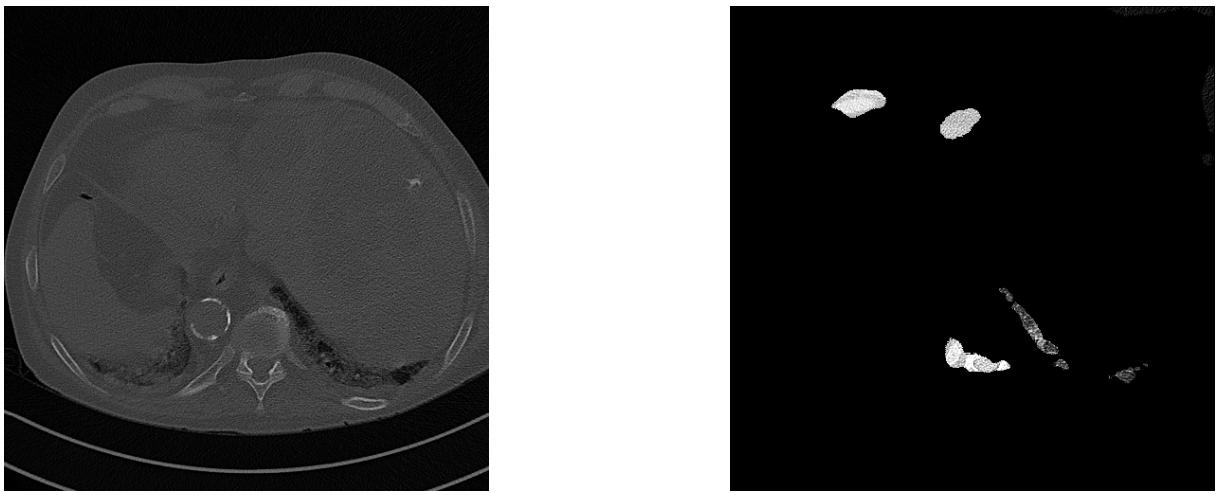
Table 5.4: Segmented framework

Split	**Number of images**
Training	7220
Validation	909
Testing	909



(a) (b)

Figure 5.3: Example of CT slice used in both frameworks. (a) Non segmented image. (b) Segmented image using the U-Net.



(a) (b)

Figure 5.4: Example of CT Slice used only on Non-segmented classification framework. (a) Non segmented image. (b) Segmented image where lung parenchyma isn't preserved.

5.2 Performance of non segmented framework

This framework has two pipelines: One uses ResNet-50 based architectures and the other uses DenseNet-201 based architectures (see Figure 5.5), both consider Non segmented images as input, and evaluate on the previously described datasets with metrics and GradCam visualization. All the models used a starting learning rate of 0,000001.

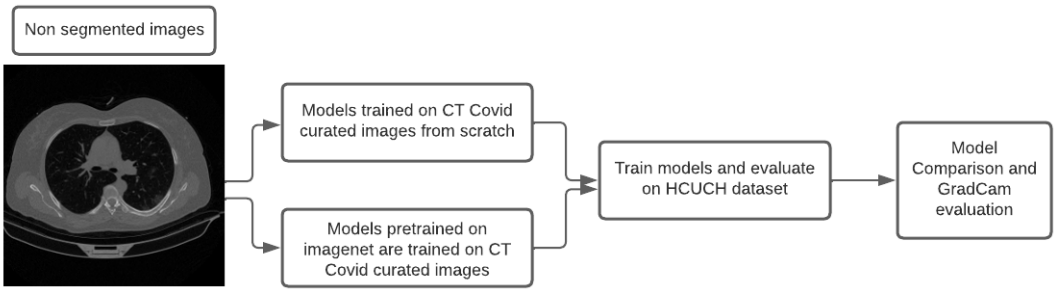


Figure 5.5: Visualization of the framework pipeline.

5.2.1 ResNet-50 based pipelines.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

This pipeline (see Figure 5.6) consists of two steps:

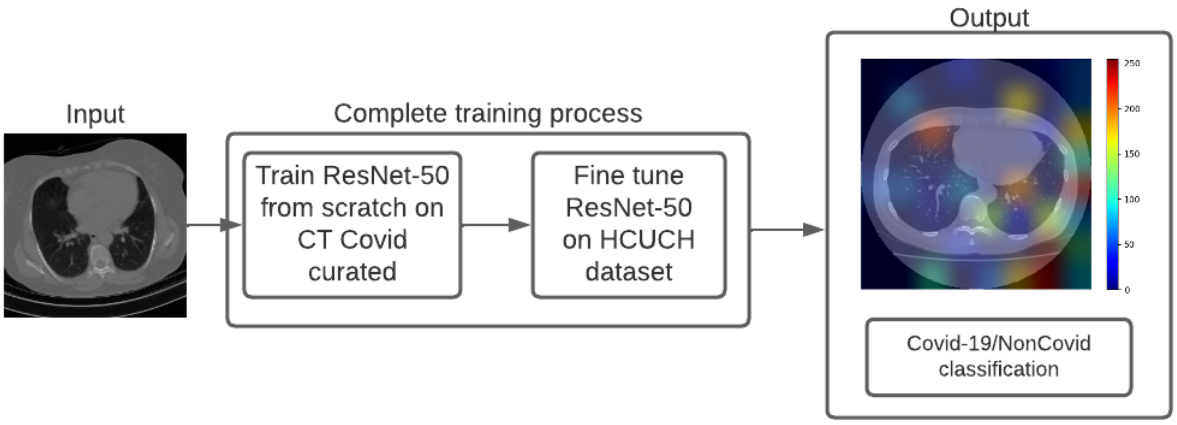


Figure 5.6: Scheme of the classification pipeline.

- Step 1: Train ResNet-50 based architecture from scratch on the CT Covid curated dataset (see Figure 5.7 and Table 5.5).
- Step 2: Fine tune ResNet-50 based architecture on the HCUCH dataset, on the weights obtained in the previous step (see Figure 5.8 and Table 5.6).

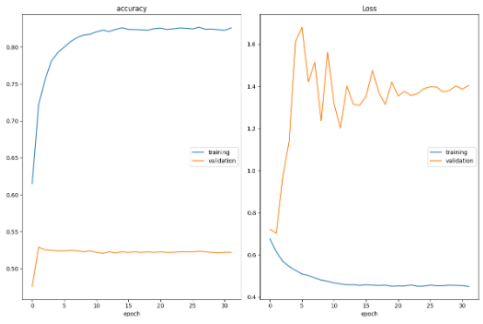


Figure 5.7: Training process on CT Covid curated dataset.

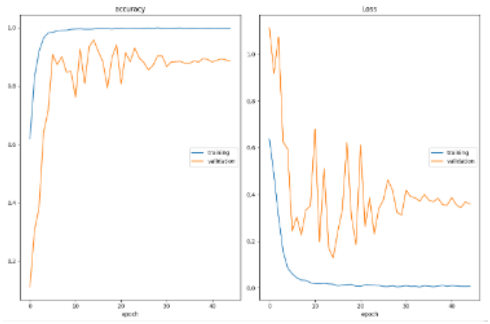


Figure 5.8: Training process of fine-tuning in HCUCH dataset.

As for performance on the testing sets, step 1 has an accuracy of 0.73, with a 0.9262 precision, 0.6790 recall and 0.7836 F1-Score. This means that the model can be trusted to correctly classify Covid-19 images, but has issues with the Non Covid ones. Following this, the final step has an overall accuracy of 0.9639, with a 0.9300 recall, 1.0 precision and 0.9293 F1 score.

GradCam was applied on the testing images, on Figure 5.9 the method was applied on a Covid-19 image and on Figure 5.10 was applied on a Non Covid image. The saliency maps do not give any input on medically relevant pixels in the image, and indicates pixels outside of the lunch parenchyma to make classification decisions.

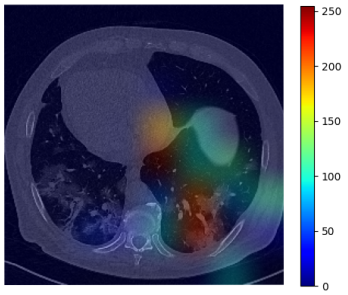


Figure 5.9: Covid-19 testing image.

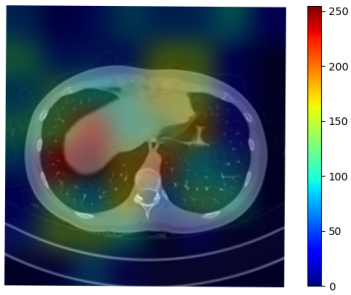


Figure 5.10: Non Covid testing image.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9262	0.0738
	Non Covid	0.4978	0.5022

Table 5.5: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1.0	0
	Non Covid	0.0732	0.9268

Table 5.6: Confusion matrix of the first step of this pipeline.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

This pipeline (see Figure 5.11) consists of two steps:

- Step 1: Train ResNet-50 based architecture the CT Covid curated dataset, fine-tuning from the Imagenet dataset. (see Figure 5.12 and Table 5.7).
- Step 2: Fine tune ResNet-50 based architecture on the HCUCH dataset, on the weights obtained in the previous step (see Figure 5.13 and Table 5.8).

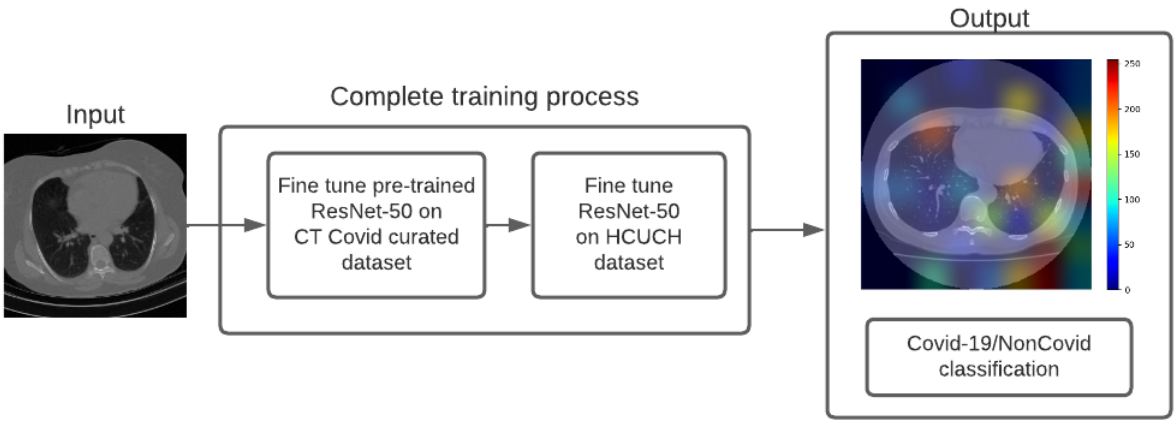


Figure 5.11: Scheme of the classification pipeline.

The first step of this pipeline has a 0.9540 overall accuracy on the testing dataset, with a 0.9149 Recall, 1.0 Precision and 0.9556 F1 score. After this, the performance on the testing data of the HCUCH dataset in step two is of 0.9758 accuracy, 0.9809 recall, 0.9710 precision and 0.9710 F1 score.

After GradCam was applied, an example of Covid-19 and Non Covid is seen in Figures 5.14 and 5.15. The saliency maps tend to focus more in the lung parenchyma, but still highlight pixels outside of it. In the Covid-19 image, the saliency maps highlights non-lung organs to make the classification decision and ignores disease markers present in the image, at the back of the lung, near the vertebrae.

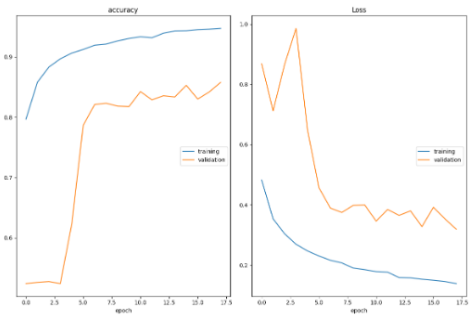


Figure 5.12: Training process on CT Covid curated dataset, fine-tuned on Imagenet dataset.

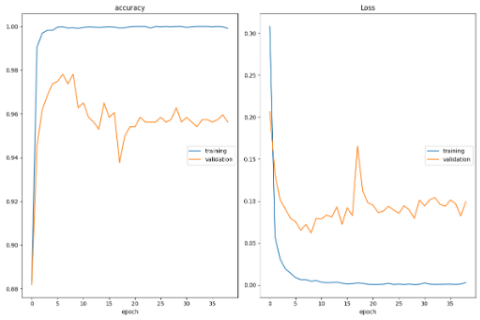


Figure 5.13: Training process of fine-tuning in HCUCH dataset.

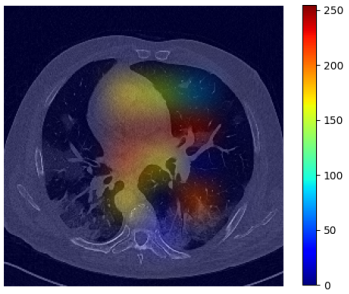


Figure 5.14: Covid-19 testing image.

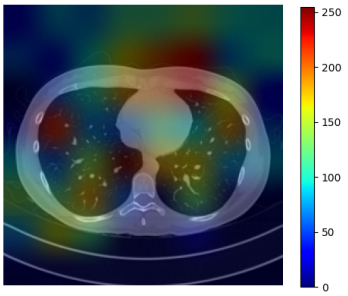


Figure 5.15: Non Covid testing image.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9710	0.0290
	Non Covid	0.0189	0.92811

Table 5.7: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0.0937	0.9069

Table 5.8: Confusion matrix of the second step of this pipeline.

5.2.2 DenseNet-201 based pipelines.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

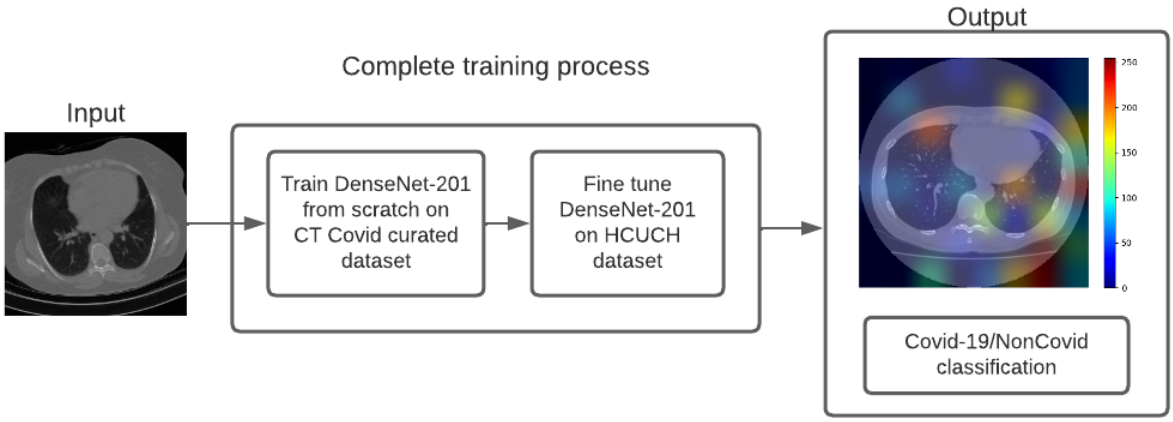


Figure 5.16: Scheme of the classification pipeline.

This pipeline (see Figure 5.16) consists of two steps:

- Step 1: Train DenseNet-201 based architecture from scratch on the CT Covid curated dataset (see Figure 5.17 and Table 5.9).
- Step 2: Fine tune DenseNet-201 based architecture on the HCUCH dataset, on the weights obtained in the previous step (see Figure 5.18 and Table 5.10).

The performance on the first step on the testing dataset is 0.9454 overall accuracy, with a 0.8971 recall, 1.0 precision and 0.9458 f1 score. After the second step, the framework performs with a 0.9519 accuracy, 0.9111 recall, 1.0 precision and 0.9535 on the testing data of the HCUCH dataset.
An example of GradCam results are shown in Figures 5.19 and 5.20. The saliency maps mostly focus on pixels outside of the lung parenchyma, and it's relevant to note that the red pixels are all outside of it; this means that the most relevant pixels to classification in this model are not biologically relevant.

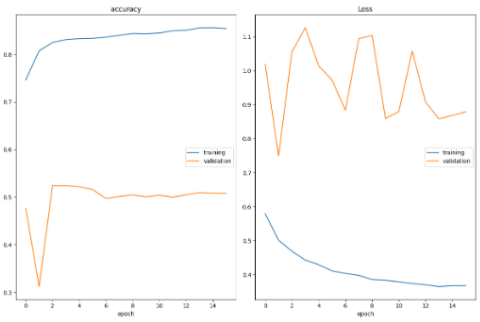


Figure 5.17: Training process on CT Covid curated dataset.

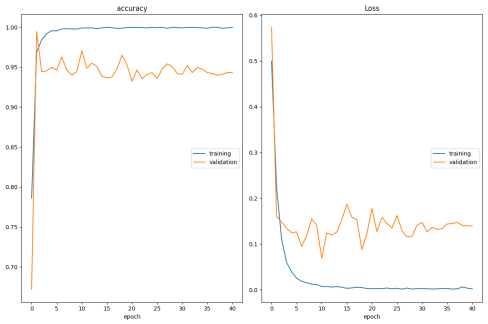


Figure 5.18: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0.01147	0.8853

Table 5.9: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0.0976	0.9024

Table 5.10: Confusion matrix of the first step of this pipeline.

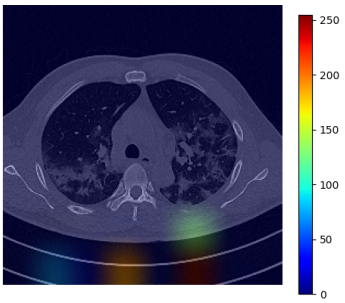


Figure 5.19: Covid-19 testing image.

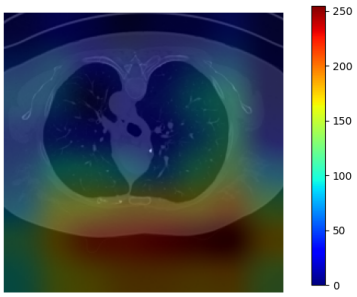


Figure 5.20: Non Covid testing image.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

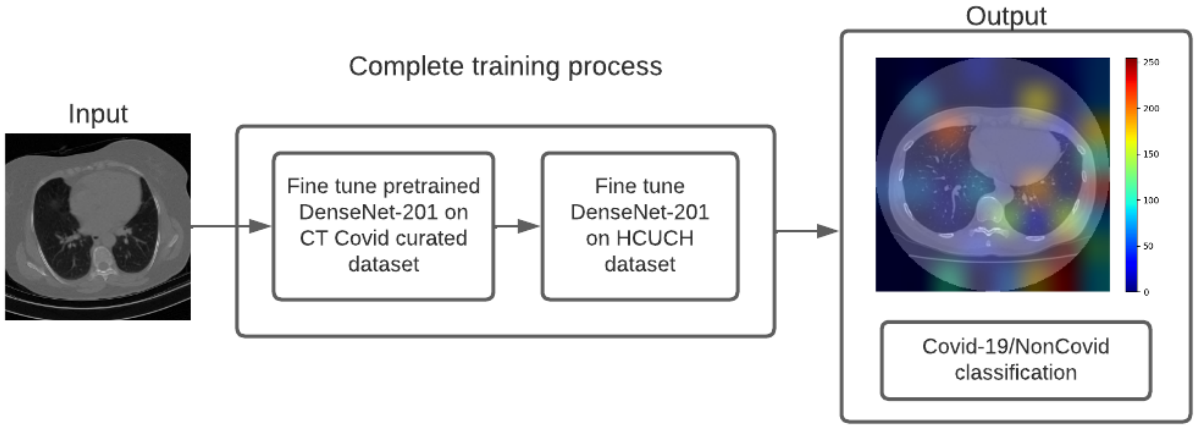


Figure 5.21: Scheme of the classification pipeline.

This pipeline (see Figure 5.21) consists of two steps:

- Step 1: Train DenseNet-201 based architecture from scratch on the CT Covid curated dataset (see Figure 5.22 and Table 5.11).
- Step 2: Fine tune DenseNet-201 based architecture on the HCUCH dataset, on the

weights obtained in the previous step (see Figure 5.23 and Table **??**).

The first step has a 0.9903 accuracy, 0.9829 recall, 0.9974 precision and 0.9901 F1 score. After the second step, the framework performs with a 0.9705 accuracy, 0.9435 recall, 1.0 precision and 0.9709 on the testing data of the HCUCH dataset.

An example of GradCam results are shown in Figures 5.24 and 5.25. The saliency maps show that the architecture in the second step do not use anatomically relevant pixels to make the classification decision: in the Covid-19 image the map is completely outside of the lung, and in the non Covid image while the map is over the lung parenchyma, it also highlights areas outside of it as high importance (colored red).

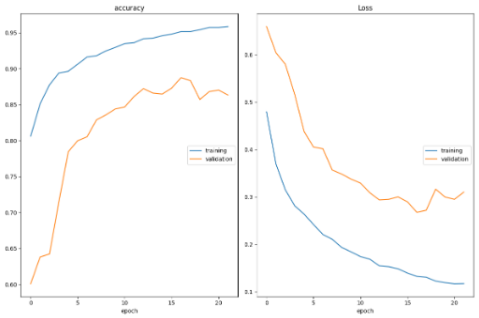


Figure 5.22: Training process on CT Covid curated dataset.

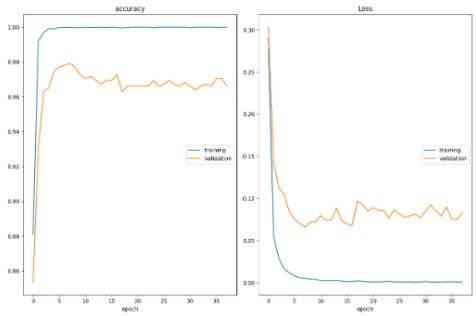


Figure 5.23: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9974	0.0026
	Non Covid	0.0174	0.9826

Table 5.11: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0.0599	0.9401

Table 5.12: Confusion matrix of the first step of this pipeline.

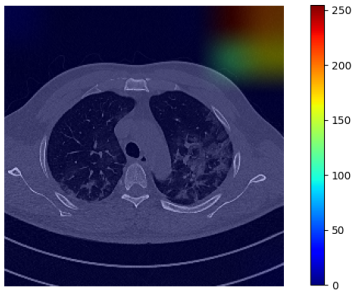


Figure 5.24: Covid-19 testing image.

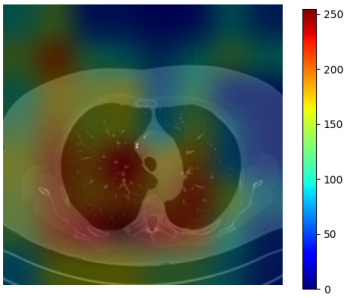


Figure 5.25: Non Covid testing image.

5.3 Performance of segmented framework

This framework (see Figure 5.26) considers the images segmented with the U-Net as input, and evaluates the ResNet-50 and DenseNet-201 based models on the previously described datasets. All the models used a starting learning rate of 0,000001.

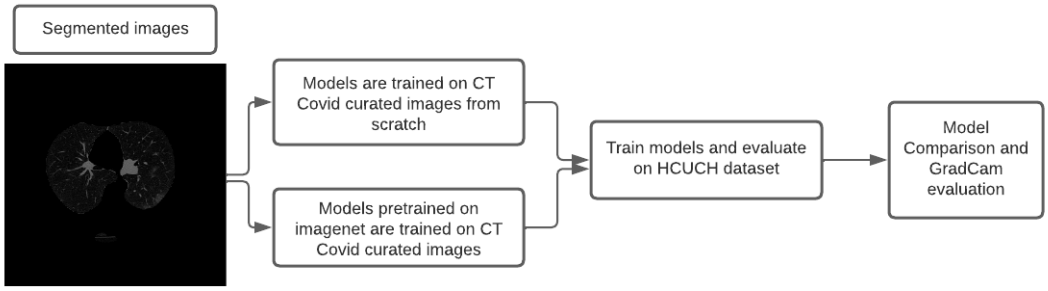


Figure 5.26: Visualization of the framework pipeline.

5.3.1 ResNet-50 based pipeline.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

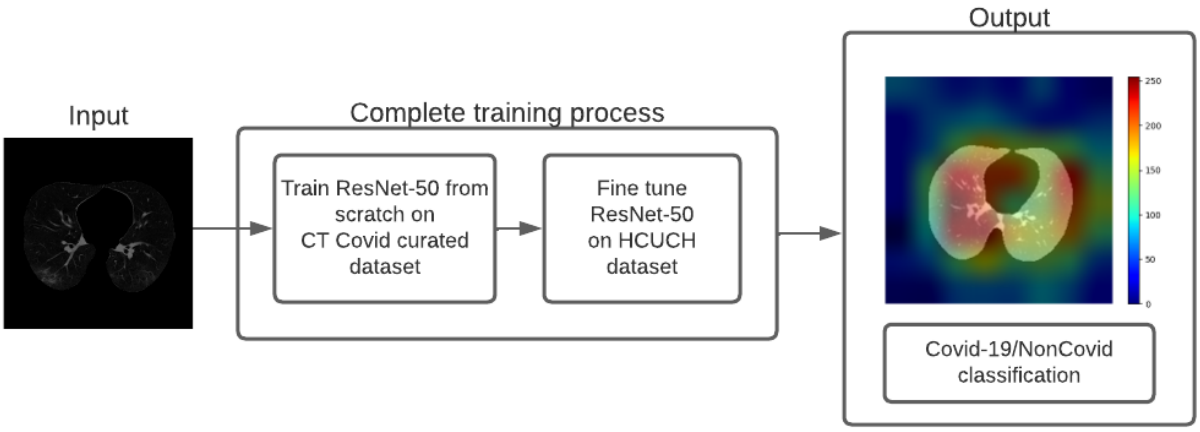


Figure 5.27: Scheme of the classification pipeline.

Same as in the previous section, this pipeline is made of two steps:

- Step 1: The ResNet-50 based architecture is trained from scratch on the CT Covid curated dataset (see Figure 5.28 and Table 5.13).
- Step 2: The weights of the previous step are used to fine tune a ResNet-50 based architecture trained on the HCUCH dataset (see Figure 5.29 and Table 5.14)

The first architecture performs with 0.7583 accuracy, 0.6635 recall, 0.9987 precision and 0.7973 F1 score on the testing set of the CT Covid curated data. After this, the step two of the pipeline is implemented, and performs with a 0.9120 accuracy, 0.9050 recall, 0.9207 precision and 0.9128 F1 score on the testing set of the HCUCH dataset.
The Figures 5.30 and 5.31 show examples of the GradCam results on testing data. The saliency maps still hightlight pixels outside of the lunch parenchyma and in both images they focus on different areas outside of it, but in the Covid-19 also highlight medically relevant tissues in green and blue.

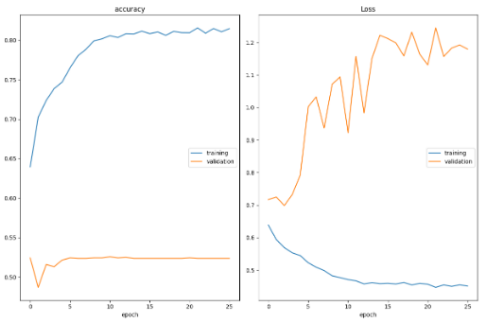


Figure 5.28: Training process on CT Covid curated dataset.

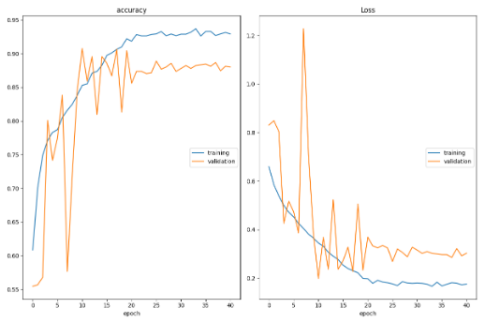


Figure 5.29: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9987	0.0013
	Non Covid	0.5065	0.4935

Table 5.13: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9207	0.0793
	Non Covid	0.0967	0.9033

Table 5.14: Confusion matrix of the first step of this pipeline.

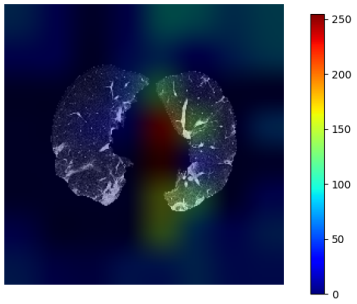


Figure 5.30: Covid-19 testing image.

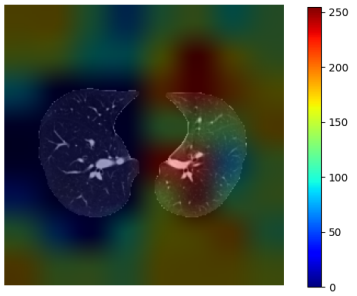


Figure 5.31: Non Covid testing image.

Pipeline based on fine tuning of models with use of the Imagenet dataset.

This pipeline is performed in two steps (see 5.32):

- Step 1: The ResNet-50 based architecture is fine tuned on the CT Covid curated

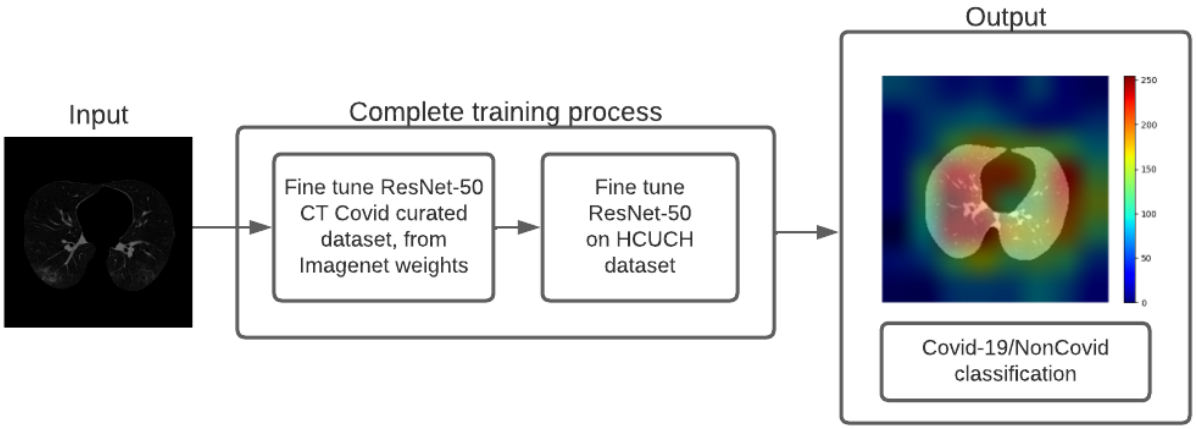


Figure 5.32: Scheme of the classification pipeline.

data with the Imagenet weights and evaluated on the testing data (see Figure 5.33 and Table 5.15).

- Step 2: The HCUCH data is used to fine tune a ResNet-50 based architecture with the weights of the previous step and evaluated on testing data (see Figure 5.34 and Table 5.16).

The performance of the step 1 in the CT Covid curated testing set is 0.9586 accuracy, 0.9489 recall, 0.09967 precision and 0.9722 f1 score. Following the second step, the pipeline performs with a 1.0 accuracy, recall precision and f1 score on the testing data of the HCUCH dataset.
GradCam is performed on the test HCUCH data, as some examples are shown in Figures 5.36 and 5.40. The saliency maps don't focus inside of the lung parenchyma, but outside of it with little attention to medically relevant pixels, specially on the Covid-19 image.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9684	0.0316
	Non Covid	0.0522	0.9478

Table 5.15: Confusion matrix of the first step of this pipeline.

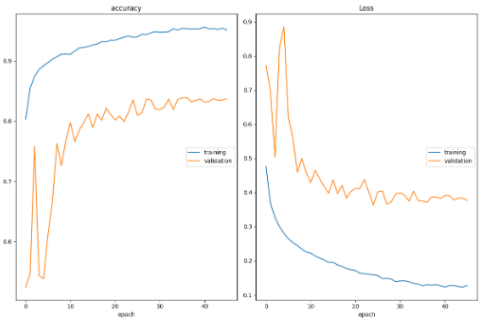


Figure 5.33: Training process on CT Covid curated dataset.

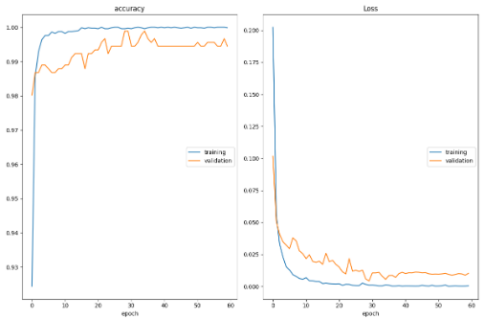


Figure 5.34: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	1	0

Table 5.16: Confusion matrix of the first step of this pipeline.

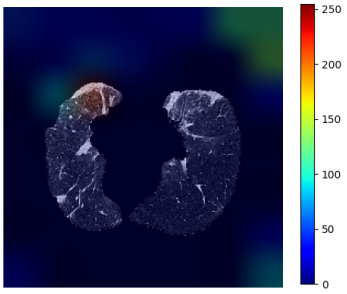


Figure 5.35: Covid-19 testing image.

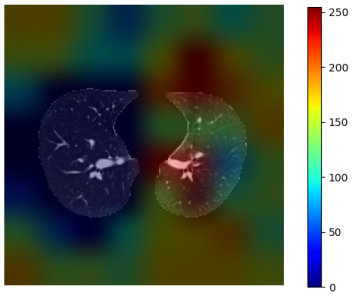


Figure 5.36: Non Covid testing image.

5.3.2 DenseNet-201

Pipeline based on fine tuning of models without use of the Imagenet dataset.

This pipeline is composed of two steps (see 5.37):

- Step 1: The DenseNet-201 based architecture is trained on the CT Covid curated data from scratch and evaluated on the testing data (see Figure 5.38 and Table 5.17).

- Step 2: The HCUCH data is used to fine tune the DenseNet-201 based architecture

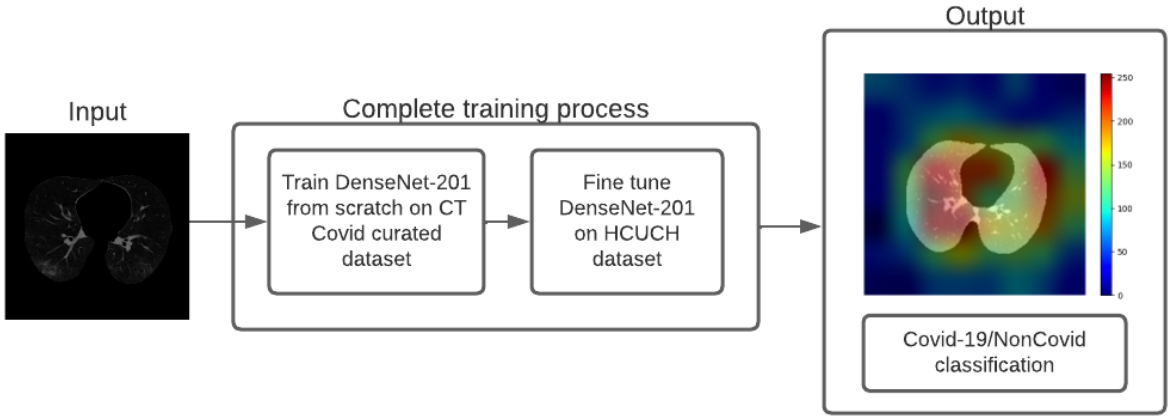


Figure 5.37: Scheme of the classification pipeline.

with the weights of the previous step and evaluated on testing data (see Figure 5.39 and Table 5.18).

Step 1 of the process performs with a 0.4758 accuracy, indetermined recall, 0 precision and 0 f1 score on the CT Covid curated test set. After this, the second and final step performs with a 0.9956 accuracy, 0.9913 recall, 1.0 precision and 0.9956 f1 score on the testing data subset of the HCUCH dataset.

GradCam is applied on the test HCUCH images and Figures 5.40 and 5.41 are examples of the results. The saliency maps highlight the most important pixels to perform classification inside the lung parenchyma, but there is a lot of attention outside of it, particularly in the non Covid-19 image. In the Covid-19 image, anatomically relevant pixels are highlighted in red, showing lung tissue affected by the disease.

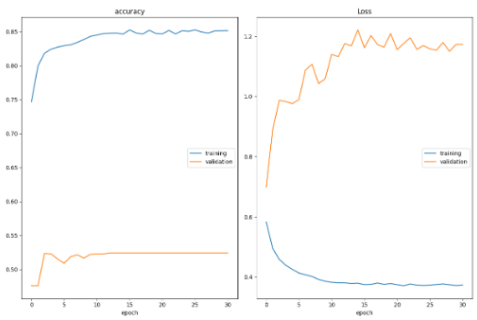


Figure 5.38: Training process on the CT Covid curated dataset.

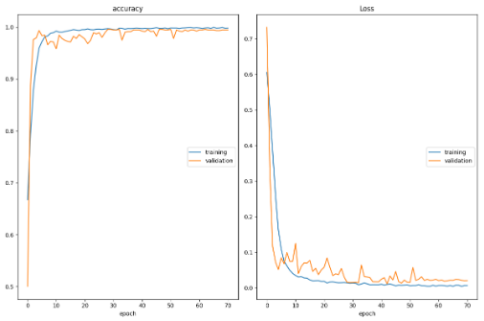


Figure 5.39: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0	1
	Non Covid	0	1

Table 5.17: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0.0088	0.9912

Table 5.18: Confusion matrix of the first step of this pipeline.

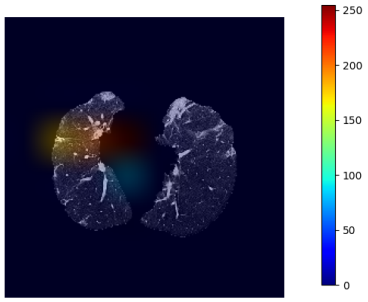


Figure 5.40: Covid-19 testing image.

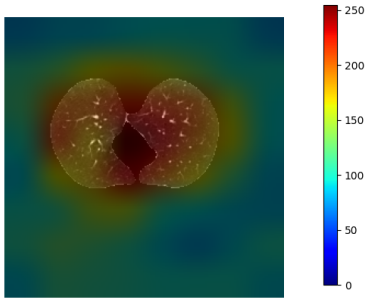


Figure 5.41: Non Covid testing image.

Pipeline based on fine tuning of models without use of the Imagenet dataset.

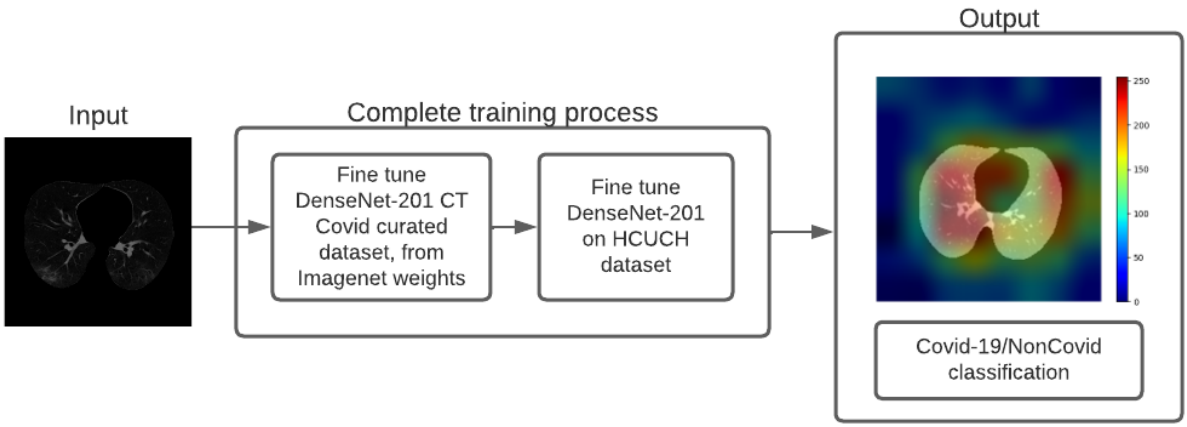


Figure 5.42: Scheme of the classification pipeline.

The last pipeline implemented is shown in the Figure 5.42. The steps are:

- Step 1: Use a pretrained on the Imagenet Dataset DenseNet-201 based architecture and fine-tune it to the CT Covid curated dataset (see Figure 5.43 and Table 5.19).
- Step 2: The same DenseNet-201 based architecture is fine-tuned to the HCUCH dataset, using the weights from the previous step (see Figure 5.44 and Table 5.20).

The first step of the pipeline performs with a 0.9454 accuracy, 0.9079 recall, 0.9868 precision and 0.9457 f1 score on the test set of the CT Covid curated dataset. After this, the final architecture performs with a 1 accuracy, recall, precision and f1 score on the HCUCH dataset.

Two examples of GradCam are shown in Figures 5.45 and 5.46. The saliency maps show divided results: in the Covid-19 image there are no highlighted pixels over the lung parenchyma and in the non Covid image the lung is highlighted, but the outside of it has high relevance in the classification decision.

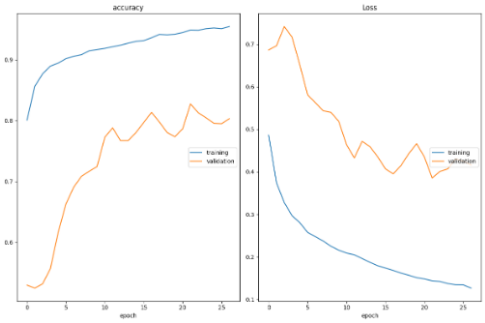


Figure 5.43: Training process on the CT Covid curated dataset.

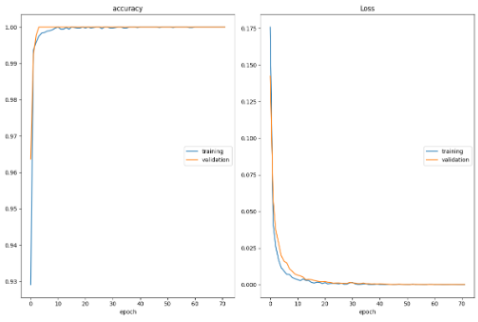


Figure 5.44: Training process of fine-tuning in HCUCH dataset.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	0.9868	0.0132
	Non Covid	0.1001	0.8999

Table 5.19: Confusion matrix of the first step of this pipeline.

		Predicted Label	
		Covid-19	**Non Covid**
True label	**Covid-19**	1	0
	Non Covid	0	1

Table 5.20: Confusion matrix of the first step of this pipeline.

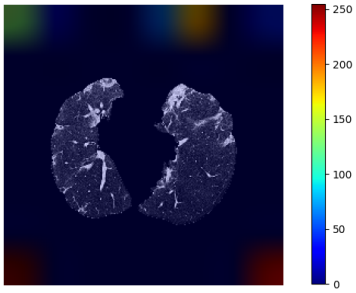


Figure 5.45: Covid-19 testing image.

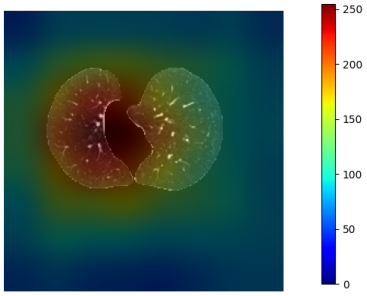


Figure 5.46: Non Covid testing image.

Chapter 6

Conclusions and discussion

Overall, both frameworks end up performing on par with the state of the art models for Covid-19 classification. All of the models fine-tuned on the HCUCH dataset have strong precision, which mean they can confidently classify 2D CT scans of sick patients.
It's relevant to note that in all frameworks, when step 1 of the pipeline was performed from scratch, meaning no fine-tuning to the Imagenet dataset, all architectures overfit during training, with their validation accuracy becoming static and the validation loss increasing while training loss decreases. This is a practical example of why fine-tuning is a very powerful tool to get reliable models with little data, considering that the Imagenet is composed of more than 14 million annotated images, versus the 7590 Covid-19 and 6893 Non Covid images in the CT Covid curated dataset. Another practical example in this work is that fine-tuning (no matter if the Imagenet dataset weights were used in the step one of the pipeline) improves performance in the architectures that were trained on the HCUCH dataset.

While it's important to have good metrics, in the Computer aided diagnosis field the priority is to show comprehensible results to medical doctors. In this work GradCam was selected because it's one of the most widely used tools in Covid-19 classification, but in the state of the art it's never thoroughly analyzed: little to no explanation is given to the results. As we can see on the different experiments, a good accuracy, recall, precision and f1 score in the testing dataset doesn't mean that the model is finding medically relevant patterns in the images. All saliency maps shown in this work fail to highlight medically relevant areas, at best they roughly highlight the lung parenchyma, and at worst they highlight the corners of the image. There is no major difference in the non segmented and segmented frameworks.

With the tools at our disposal, we can correctly classify Covid-19 no matter if the lung parenchyma was segmented or not, but GradCam is a powerful truth that show us what is behind the scenes of the CNN, giving us a glimpse inside of the black box. GradCam shows that while the metrics are good, these classification frameworks aren't enough to deliver medically relevant explanations to the results of the task.

The frameworks are likely to be falling to the shortcut learning issue [51], where deep learning architectures fail to generalize in more complex domains, such as medical images. These images have complex structures and most importantly, their quality depends on the machine used for sampling. One way to solve this is to have a more varied and robust pre-training dataset, but those are few and sparse in the medical image domain due to patient privacy concerns. An alternative to this is the use of generative neural networks to synthesize images and perform classification [52].
A supervised way to solve this problem could be attention-based neural networks [53], architectures that use channel and spacial attention information to improve generalization. Another way to approach this is to step away from the supervised domain adaptation focus of this task, and jump to more complex methods such as self supervised learning [54] and adversarial learning [55].

References

[1] Vikram rao Bollineni, Koenraad Hans Nieboer, Seema Döring, Nico Buls, and Johan de Mey. The role of CT imaging for management of COVID-19 in epidemic area: early experience from a university hospital. *Insights into Imaging*, 12(1), jan 2021.

[2] Gianluca Pontone, Stefano Scafuri, Maria Elisabetta Mancini, Cecilia Agalbato, Marco Guglielmo, Andrea Baggiano, Giuseppe Muscogiuri, Laura Fusini, Daniele Andreini, Saima Mushtaq, Edoardo Conte, Andrea Annoni, Alberto Formenti, Antonio Giulio Gennari, Andrea I. Guaricci, Mark R. Rabbat, Giulio Pompilio, Mauro Pepi, and Alexia Rossi. Role of computed tomography in COVID-19. *Journal of Cardiovascular Computed Tomography*, 15(1):27–36, jan 2021.

[3] Neelima Arora, Amit K Banerjee, and Mangamoori L Narasu. The role of artificial intelligence in tackling COVID-19. *Future Virology*, 15(11):717–724, nov 2020.

[4] Priya Aggarwal, Narendra Kumar Mishra, Binish Fatimah, Pushpendra Singh, Anubha Gupta, and Shiv Dutt Joshi. Covid-19 image classification using deep learning: Advances, challenges and opportunities. *Computers in Biology and Medicine*, 144:105350, may 2022.

[5] Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. Supervised classification algorithms in machine learning: A survey and review. In Jyotsna Kumar Mandal and Debika Bhattacharya, editors, *Emerging Technology in Modelling and Graphics*, pages 99–111, Singapore, 2020. Springer Singapore.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, oct 2010.

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[9] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2018.

[10] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. Explainable deep learning models in medical image analysis. *Journal of Imaging*, 6(6):52, jun 2020.

[11] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), mar 2021.

[12] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML 2010*, pages 807–814, 2010.

[13] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015.

[14] Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: A survey. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185, mar 2022.

[15] Mohsen Ghafoorian, Alireza Mehrtash, Tina Kapur, Nico Karssemeijer, Elena Marchiori, Mehran Pesteie, Charles R. G. Guttmann, Frank-Erik de Leeuw, Clare M. Tempany, Bram van Ginneken, Andriy Fedorov, Purang Abolmaesumi, Bram Platel, and William M. Wells. Transfer learning for domain adaptation in MRI: Application in brain lesion segmentation. In *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, pages 516–524. Springer International Publishing, 2017.

[16] Ravi K Samala, Heang-Ping Chan, Lubomir M Hadjiiski, Mark A Helvie, Kenny H Cha, and Caleb D Richter. Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms. *Physics in Medicine &amp Biology*, 62(23):8894–8908, nov 2017.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[18] Naimul Mefraz Khan, Nabila Abraham, and Marcia Hon. Transfer learning with intelligent training data selection for prediction of alzheimer's disease. *IEEE Access*, 7:72726–72735, 2019.

[19] Zar Nawab Khan Swati, Qinghua Zhao, Muhammad Kabir, Farman Ali, Zakir Ali, Saeed Ahmed, and Jianfeng Lu. Brain tumor classification for MR images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*, 75:34–46, jul 2019.

[20] Asmaa Abbas, Mohammed M. Abdelsamea, and Mohamed Medhat Gaber. Detrac: Transfer learning of class decomposed medical images in convolutional neural networks. *IEEE Access*, 8:74901–74913, 2020.

[21] Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1038–1042, 2018.

[22] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *Domain Adaptation in Computer Vision Applications*, pages 189–209. Springer International Publishing, 2017.

[23] Jie Yang, Thomas Vetterli, Pallavi P. Balte, R. Graham Barr, Andrew F.Laine, and Elsa D. Angelini. Unsupervised domain adaption with adversarial learning (udaa) for emphysema subtyping on cardiac ct scans: The mesa study. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 289–293, 2019.

[24] Jing Zhang, Mingxia Liu, Yongsheng Pan, and Dinggang Shen. Unsupervised conditional consensus adversarial network for brain disease identification with structural MRI. In *Machine Learning in Medical Imaging*, pages 391–399. Springer International Publishing, 2019.

[25] Rongbo Shen, Jianhua Yao, Kezhou Yan, Kuan Tian, Cheng Jiang, and Ke Zhou. Unsupervised domain adaptation with adversarial learning for mass detection in mammogram. *Neurocomputing*, 393:27–37, 2020.

[26] Jing Wang, Yi He, Wangyi Fang, Yiwei Chen, Wanyue Li, and Guohua Shi. Unsupervised domain adaptation model for lesion detection in retinal OCT images. *Physics in Medicine &amp Biology*, 66(21):215006, oct 2021.

[27] Euijoon Ahn, Ashnil Kumar, David Dagan Feng, Michael J. Fulham, and Jinman Kim. Unsupervised deep transfer feature learning for medical image classification. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1915–1918, 2019.

[28] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, and Bo Xu. A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *European Radiology*, feb 2021.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[30] Harrison X. Bai, Robin Wang, Zeng Xiong, Ben Hsieh, Ken Chang, Kasey Halsey, Thi My Linh Tran, Ji Whae Choi, Dong-Cui Wang, Lin-Bo Shi, Ji Mei, Xiao-Long Jiang, Ian Pan, Qiu-Hua Zeng, Ping-Feng Hu, Yi-Hui Li, Fei-Xian Fu, Raymond Y. Huang, Ronnie Sebro, Qi-Zhi Yu, Michael K. Atalay, and Wei-Hua Liao. Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT. *Radiology*, 296(3):E156–E165, sep 2020.

[31] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019.

[32] Chun Li, Yunyun Yang, Hui Liang, and Boying Wu. Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. *Knowledge-Based Systems*, 218:106849, apr 2021.

[33] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[34] Pranav Rajpurkar, Jeremy A. Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, C. Langlotz, Katie S. Shpanskaya, Matthew P. Lungren, and A. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *ArXiv*, abs/1711.05225, 2017.

[35] Ilyas LAHSAINI, Mostafa ELHABIBDAHO, and Mohamed Amine CHIKH. Deep transfer learning based classification model for covid-19 using chest ct-scans. *Pattern Recognition Letters*, 152:122–128, 2021.

[36] Hoon Ko, Heewon Chung, Wu Seong Kang, Kyung Won Kim, Youngbin Shin, Seung Ji Kang, Jae Hoon Lee, Young Jun Kim, Nan Yeol Kim, Hyunseok Jung, and Jinseok Lee. COVID-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest CT image: Model development and validation. *Journal of Medical Internet Research*, 22(6):e19569, jun 2020.

[37] Y. Pathak, P.K. Shukla, A. Tiwari, S. Stalin, S. Singh, and P.K. Shukla. Deep transfer learning based classification model for COVID-19 disease. *IRBM*, 43(2):87–92, apr 2022.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[39] Subhankar Ghosh, Palaiahnakote Shivakumara, Prasun Roy, Umapada Pal, and Tong Lu. Graphology based handwritten character analysis for human behaviour identification. *CAAI Transactions on Intelligence Technology*, 5(1):55–65, feb 2020.

[40] M. Turkoglu. COVID-19 detection system using chest CT images and multiple kernels-extreme learning machine based on deep neural network. *IRBM*, 42(4):207–214, aug 2021.

[41] Hugo J. W. L. Aerts, Leonard Wee, Emmanuel Rios Velazquez, Ralph T. H. Leijenaar, Chintan Parmar, Patrick Grossmann, Sara Carvalho, Johan Bussink, Ren Monshouwer, Benjamin Haibe-Kains, Derek Rietveld, Frank Hoebers, Michelle M. Rietbergen, C. Ren Leemans, Andre Dekker, John Quackenbush, Robert J. Gillies, and Philippe Lambin. Data from nsclc-radiomics, 2019.

[42] Maede Maftouni, Andrew Chung Chee Law, Bo Shen, Yangze Zhou, Niloofar Ayoobi Yazdi, and Zhenyu Kong. A robust ensemble-deep learning model for covid-19 diagnosis based on an integrated ct scan images database. 06 2021.

[43] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, dec 2020.

[44] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, dec 2020.

[45] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[46] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[47] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[48] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *ArXiv*, abs/2008.05756, 2020.

[49] Margaret Sampson, Jessie McGowan, Elise Cogo, Jeremy Grimshaw, David Moher, and Carol Lefebvre. An evidence-based practice guideline for the peer review of electronic search strategies. *Journal of Clinical Epidemiology*, 62(9):944–952, sep 2009.

[50] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017.

[51] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, nov 2020.

[52] Kazuhiro Koshino, Rudolf A. Werner, Martin G. Pomper, Ralph A. Bundschuh, Fujio Toriumi, Takahiro Higuchi, and Steven P. Rowe. Narrative review of generative adversarial networks in medical and molecular imaging. *Annals of Translational Medicine*, 9(9):821–821, may 2021.

[53] Menghua Zheng, Jiayu Xu, Yinjie Shen, Chunwei Tian, Jian Li, Lunke Fei, Ming Zong, and Xiaoyang Liu. Attention-based cnns for image classification: A survey. *Journal of Physics: Conference Series*, 2171(1):012068, jan 2022.

[54] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P. Lungren, Serena Yeung, and Akshay S. Chaudhari. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *npj Digital Medicine*, 6(1), apr 2023.

[55] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, 2021.