ISTANBUL BILGI UNIVERSITY
INSTITUTE OF GRADUATE PROGRAMS
ELECTRICAL AND ELECTRONICS ENGINEERING MASTER'S DEGREE PROGRAM

DIAGNOSIS AND PROGNOSIS OF COVID-19 FROM MEDICAL IMAGES USING DEEP LEARNING

NOURAN FADLALLAH
119815015

PROF. DR. AHMET DENKER

ISTANBUL
2022

ACKNOWLEDGEMENTS

This endeavor would not have been possible without my professor Dr. Ahmet Denker with his continuous support, patience and feedback. I'm also extremely grateful to my defense committee Dr. Elena Battini Sonmez, and Dr. Sertan Serte, who generously provided their time and knowledge.

I would like to extend my sincere thanks to my classmates and friends for their recommendations, feedback, and motivation.

ABSTRACT

Covid-19, with its high death rate, was discovered nearly three years ago. New variants resistant to vaccines still emerge while travel and export restrictions can not be held for longer. An accurate and fast diagnosis of such a disease is crucial to reducing its global spread. Computed Tomography CT scans have shown to be the most precise method for covid-19 diagnosis. However, it is a slow process to read and diagnose a disease from a CT scan due to the scarcity of skilled radiologists and the limited information and data available about covid-19. Computer vision has been successfully used in assisting professionals in diagnosis tasks both in terms of speed and accuracy when trained on large datasets. This work is an effort to develop a fast and accurate AI model for covid-19 diagnosis trained on a small dataset. We developed an ensemble model consisting of a 3D CNN LeNet-based model and a 2D Convolutional-Like Vision transformer to diagnose CT scans as covid-19 and healthy. A total of 508 CT scans were used to train the model as a subset of the publicly available MosMed dataset. This results in an accuracy of 90%, specificity of 92%, and a sensitivity of 88%.

ÖZET

Yüksek ölüm oranına sahip olan Covid-19, yaklaşık üç yıl önce tespit edildi. Seyahat ve ihracat kısıtlamalarının daha uzun süre tutulmasının mümkün olmayacağı durumdayken, aşılara dirençli yeni varyantlar hala ortaya çıkmaya devam etmekte. Böyle bir hastalığın doğru ve hızlı teşhisi, küresel yayılımını azaltmak için çok önemlidir. Bilgisayarlı Tomografi (BT) taramalarının Covid-19 tanısı için en hassas yöntem olduğu gösterilmiştir. Ancak, kalifiye radyologların azlığı ve covid-19 hakkında sınırlı bilgi ve veri olması nedeniyle BT taramasını okuyarak hastalığı teşhis etmek uzun süren bir işlemdir. Büyük veri kümeleri üzerinde eğitilen bilgisayar görüntüsü, tanılama süreçlerinde uzmanlara destek amaçlı olarak hem hız hem de doğruluk açısından başarılı bir şekilde kullanılmıştır. Bu çalışma, küçük bir veri kümesi üzerinde eğitilmiş covid-19 tanısı için hızlı ve doğru bir AI modeli geliştirmesi üzerinedir. BT taramalarından covid-19 veya sağlıklı teşhisi yapan, 3D CNN LeNet tabanlı bir model ve 2D Evrişimli-benzeri görüntü dönüştürücüden oluşan bir kolektif model geliştirdik. Modeli eğitmek için, açık kaynak olan MosMed veri kümesinden bir alt küme olarak toplam 508 BT tarama kullanılmıştır. Bu, % 90 doğruluk, % 92 özgüllük ve % 88 hassasiyet ile sonuçlanmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Deep Learning for Medical Imaging

Medical imaging is a term used to refer to images or visual representations of the interior of the body taken for medical analysis or intervention. Accurate analysis of these images is necessary as they are widely used for many diseases' diagnosis and treatment planning. There are many types of medical images; the most used ones include radiography (X-Ray), ultrasound, magnetic resonance imaging (MRI), and tomography (CT, PET). The large number of medical images taken each year, the longer amount of time needed to accurately analyze them due to their complexity, and the limited amount of skilled radiologists all derive the need for a technological solution. The use of computers in medical image diagnosis automation has been studied since the 1960s [1,2]. However, systematic development of a computer-aided diagnosis system using machine learning image processing techniques started as late as the 1980s [3,4]. The main goal of a CAD system, unlike an automated diagnosis system, is to assist radiologists in diagnosis as a second opinion. This approach is more suitable for a medical setting, as well as more achievable with the hardware and technological limitations. The development of CAD systems continued using traditional image processing techniques such as difference-image and edge enhancement. But since the results of using these methods were not accurate enough, they had a very limited presence in clinics. In later years, deep learning proved its superior performance to traditional methods in many different tasks and presented a new opportunity for developing highly performing CAD systems. Deep learning systems can analyze a huge amount of data in a very short amount of time while keeping high accuracy and precision. In a study performed by Stanford Academic Medical Center, while radiologists labeled 420 images in 240 minutes on average, the AI model used in their study labeled the same data in 1.5 minutes [5]. Its high accuracy and precision are also reliably consistent since it doesn't get tired or distracted. It also learns to diagnose new diseases

relatively fast. That can be observed by the number of published scientific papers and AI models created for the novel coronavirus (covid-19) diagnosis within a month after it was declared a pandemic. That fast-forwarded delivers an accurate diagnosing tool even to remote areas.

1.2. Convolutional Neural Networks

1.2.1. Architecture

CNNs are created to mimic how the human brain recognizes objects. The building blocks of CNNs are called artificial neural networks. Similar to the brain, each neuron holds specific information which in total can understand the characteristic features of an image. For example, consider a CNN model that recognizes pictures of cats and dogs. The model consists of different layers. The first layer is the input layer, taking in cats' and dogs' photos. In the training process, these photos are labeled as 'cat' or 'dog'. In the next layer, the model would be looking for certain features in the image. The first neuron of that layer might be looking for triangular ears; the second could be the shape of the tail, or whiskers. The next layer of neurons could be looking for even finer details. These are called hidden layers since developers cannot exactly know what each neuron is measuring. In each layer, the mathematical operation called “convolution” is applied. In mathematics, convolution between two functions produces a third function that shows how the shape of one function is changed by the other. In CNNs, the first function is the input of the layer, the second function is the weights in the neurons of the layer, and the resulted function is the output of the layer where the input has been affected by the weights. Therefore, each time a new image passes through these layers it gets a score of how identical each part of it is to the feature of the neuron. Typically, each CNN layer has a nonlinear activation function followed by a pooling layer. The max-pooling layer modifies the output to be more invariant to small changes in the input layer by taking the maximum output within a rectangle. At the end of a CNN model and after the CNN layers are usually “Dense” or fully

connected layers where a total score is computed for which category the image is closest to.

CNN model architectures vary from one model to the other, depending on the number of layers, neurons, the connections between the layers, and the types of layers used. Different architectures yield different results in different applications. A commonly used classification model is AlexNet. AlexNet architecture consists of 5 convolutional layers and 3 fully connected layers. In between the convolutional layers, there are max-pooling layers for generalization and each layer uses ReLU as an activation function for non-linearity. The depth of the convolutional layers increases and the height and width decrease as we go deeper in the network. The size of the used filters for convolution also decreases. Therefore, the network converts the 2D weights, step by step, into a vector without losing the spatial information of the images. Finally, we reach the fully connected layers ending with a softmax activation function to find the correct class.

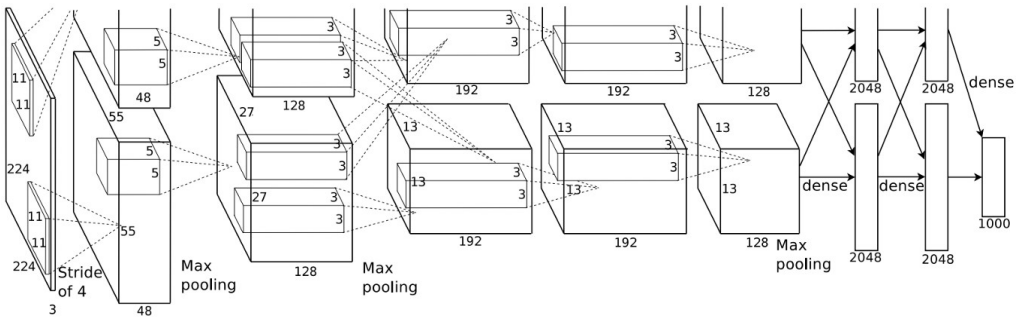

Figure 1.1. Architecture of AlexNet, after [11]

1.2.2. Training

Training a CNN network uses a method called back-propagation. As previously explained, each layer of the model contains weights that express the importance of a feature towards the overall score of the input. To find the best values of these weights, we train the model by feeding in the labeled input data. A score is

calculated and then compared against the given label through a loss function. Then back-propagation is used to modify the weights according to the loss function results. These steps are repeated till the end of training. There are some commonly used loss functions for CNNs including Mean Squared Error Loss (MSE), Mean Absolute Error Loss (MAE), Binary and categorical cross-entropy, etc. In addition to model architecture and loss function, several parameters should be considered when training to achieve the best performance of the model. These are usually called hyperparameters and they include learning rate, number of epochs, and batch size. Unfortunately, no formula can compute the best parameter values for a specific application. The Hyperparameter tuning process was handled using a set of trials. Lately, several frameworks were developed both commercially and open source that can help speed up the tuning process by training the model with different parameter values simultaneously and comparing the results.

1.2.3. History

The beginning of the Convolutional Neural Networks dates back to the early 1980s. The first trained CNN model was created by LeCun et al. [6] to classify handwritten digits. CNNs were quickly adopted by the industry, the AT&T research group developed a CNN model to read checks [7] then used by the NEC, and Microsoft used some CNN-based OCR and handwriting recognition systems [8]. To this day, Facebook, Google, and IBM among a list of companies remain some of the biggest contributors to CNN research. In medical applications, CNN was first introduced in 1993 for lung nodule detection in chest radiographs [9]. In the same year, CNNs were used for the recognition of mammographic microcalcifications [10]. These models would easily replace traditional image processing techniques as CNNs are insensitive to noise, blur, contrast, etc. CNN architectures since then became increasingly complicated, deeper models were created, pooling layers were added and rectified linear unit was used as activation function rather than the typical sigmoid function. To decrease the chance of over-fitting, dropout layers and batch normalization were introduced. Also transferred learning was introduced, increasing

the accuracy of applications with limited training data available. The first most commonly used CNN architecture for transfer learning was introduced in 2012 as AlexNet [11]. Different CNN architectures and their uses will be discussed next.

1.3. Transfer learning

Training CNN architecture from scratch is not always the best choice. When dealing with medical imaging in most cases the amount of labeled data is limited. In that case, transfer learning with an off-the-shelf model is usually used.

Transfer learning is one of the commonly used techniques to ensure better accuracy when using a small amount of data for training. The idea of transfer learning is to use a pre-trained CNN model, freeze the first number of layers and then train the model on new data for the specific application. This makes better use of the limited available data as it is used only to train the model on the specific features that the dataset contains rather than waste most of it on training the model to extract generic features. Sometimes referred to as transfer of knowledge, the method uses the pre-trained model's knowledge of extracting general features and then uses the small dataset to train the model specifically for the new application. This is based on Yosinski et al. [12], who show that the first layers of CNNs contain generic features and then the features become more specific as we go to deeper CNN layers.

1.4. Classification

Classification models are CNN models trained to classify images to a set of labels. Different classification models differ in depth, layer sizes, layer types, and layouts. There are many famous classification architectures and pre-trained models than proved high accuracy in different applications. These models include VGG, ResNet, Inception net, etc. For example, VGG-16 [13] architecture consists of 11 convolutional layers split into 5 stages, the width and length dimensions of convolution layers are decreased as we go into the next stage while the depth

increases. Activation functions for all convolutional layers are ReLU and every stage is followed by a max-pooling layer. The stages are then followed by 3 fully connected layers and a softmax activation function at the output layer.

1.5. Detection

Object detection models are different from classification models since they find the place of the detected object rather than classifying the whole image with a label. This also allows the model to find and classify more than one object in an image if present. Object detectors use bounding boxes to mark the object's place in an image, therefore, the output of the model is the prediction and the size and coordinates of the bounding box for each found object.

Detection methods can be divided into two main categories based on their way of finding objects in an image, region proposal methods and sliding window methods. The first group of object detectors tries to find regions of interest before applying the classification problem. These methods consist of several stages, at least two models. The first stage has two networks, a backbone network is classification architecture such as ResNet or VGG, and a region proposal network that proposes a huge number of regions, in this stage the model finds the region of interest. The second model classifies the objects and finds their bounding boxes. Whether it has two or more models, each model has to be trained separately, making it slower and harder to train. Moreover, since these methods work in different consecutive stages their speed in real-time applications is relatively slow compared to direct classification methods. In this family of object detectors come R-CNN, SPP-net, Fast R-CNN, and Faster R-CNN.

The second group of object detectors aims to reduce time expense by applying global regression/ classification directly to the image without searching for regions of interest. This object recognition method is relatively faster than the prior since they do not have many stages and do not try to find the exact boundaries of an object.

Therefore, they are mainly used for applications where speed is more important than accuracy. These methods include YOLO, SSD, DSSD, and DSOD. Most of these models depend on single-shot learning, where the data go through a single forward propagation of the network.

1.6. Segmentation

1.6.1. Semantic Segmentation

Semantic segmentation classifies objects in an image based on pixels rather than a bounding box, which results in a precise location of the predicted object. The most commonly used model in that category is U-net. U-net is an encoder-decoder CNN model, where the encoder part classifies an object and the decoder finds its exact location. The encoder part can be a classification model such as VGG, or ResNet. V-net is a 3D implementation of the U-net architecture.

1.6.2. Instance Segmentation

Instance segmentation is a recent approach that integrates the goals of object detection with semantic segmentation. It detects object labels with bounding boxes while segmenting a mask for each object instance. Thus, instance segmentation models can detect multiple instances of the same label objects and give their pixel-precise location in the image. This is possible by using ROI-based detection model architecture, ex: faster R-CNN, then adding a branch for predicting segmentation masks on each ROI. Famous models of this type are Mask R-CNN and Deeplab.

1.7. Data

Since in supervised learning the model learns from the given annotated data, it is especially important to create a suitable dataset for the application. The first step of creating a dataset is data collection. There are multiple available datasets for

medical applications such as INbreast [14], The Lung Image Database Consortium (LIDC), and the Image Database Resource Initiative (IDRI) lung nodules dataset [15], etc. Some of these datasets can be used directly, others need some pre-processing. However, not every application has an online public dataset available. In that case, data need to be collected from hospitals or medical institutions. A dataset should also be split, typically into train, test, and validation. The train and test sets are used during training for updating weights through back-propagation and testing the accuracy of the model while training. The validation set is used to evaluate the performance of the model after training is finished.

1.7.1. Augmentation and Synthetic Data

In many cases, data augmentation is used to increase the amount of data in our dataset. Data augmentation creates data by altering the images in the original dataset by shifting, rotating, etc. These transformations can be slightly done to every image numerous times, producing a large amount of data. Augmented data help increase the accuracy of training but also can contribute to over-fitting since the variation between the data is slight. Another approach to increase the amount of data in a dataset is to use synthetic data. Unlike augmentation, synthetic data are not just slight alterations to existing data, but it is the creation of new data after understanding how the existing data is structured. There are two main approaches two creating synthetic data, one uses CNNs and the other uses GANs.

1.7.2. Preprocessing

After all the previous steps are taken and the dataset is completed, there is one last step to do before feeding the data directly to the model for training, which is data pre-processing. Pre-processing is the process of editing the images to fit into the model architecture and simplify the numeric representation of the image. The first and most common step of preprocessing is resizing and/or cropping the images

to drop the resolution for faster training, focus on the region of interest, or most importantly match the size of the input layer of the model. Standardization is also important to make all images' pixel values lie in the same range, ex: [0, 1]. Other common edits are histogram equalization or normalization for eliminating the effects of different lighting, and noise filtering for removing blur or extra sharpness. The pre-processing function is then applied to the dataset before training as well as any new input for prediction when using the model after training.

2. PURPOSE

Since the discovery of the novel Coronavirus Disease (SARS-Cov-2), to date, there have been over 520 million confirmed cases worldwide and over 6 million deaths as reported by the World Health Organization [16]. The disease was declared a pandemic in February 2020, and health ministries all over the world started dealing with the exponentially rising number of cases. Computed Tomography (CT) scans proved to be the most reliable imaging to screen the disease and its development in patients [17, 18]. However, recognizing the effects of the new disease correctly from scans as well as its severity is a difficult task even for trained experts. The amount of time it takes radiologists and other health care providers to learn about the new disease and its manifestations on different types of medical imaging is dangerously long for such a critical situation. The use of deep learning can help quickly transfer the information needed to diagnose and give a prognosis to the new disease internationally. Moreover, deep learning-based diagnostic systems proved to be faster than radiologists without sacrificing accuracy [19]. The performance of radiologists using a deep learning diagnostic system is much superior in terms of speed, accuracy, and recall. All of these facts make AI-based systems extremely desirable in the current medical situation.

The main purpose of this dissertation is to create a deep learning model for covid-19 diagnosis. The CNN-based model is to perform the classification process on CT scans with accuracy on par with radiologists and faster performance. The model should be beneficial to be used by radiologists to increase diagnosis accuracy and save experts' and patients' time. It is considered a second opinion system or a computer-aided diagnostic system as it does not aim to replace the role of a radiologist but rather to support the expert's decision-making process by providing fast reliable predictions. There are five main components of such a system; data collection, image processing, model architecture and training, optimization methods, and finally validation of results.

The first component, data, needs to be discussed in terms of acquisition and augmentation. Training a high accuracy model requires relatively high amounts of medical imaging data. However, especially for a new disease, the available data is scarce. To guarantee the true performance of the model, the used data needs to be verified or acquired from an approved source. The requirement of huge amounts of data can easily result in false high accuracy if the training data contains duplicates or incorrectly manipulated or labeled images. For example, in the Digital Mammography Dream challenge in 2016, a huge dataset was presented, and the winning teams created highly accurate deep learning but with a false positive rate much higher than a radiologist. This resulted from the low-quality labeling of the data [20]. Thus, one good source of public data would be one approved by a known medical or research institution. Acquiring data directly from hospitals is another reliable method; however, hospitals need to comply with the laws and regulations of patient privacy, which differ from one country or region to another. This includes redacting some patient information from the data, and/or getting written consent.

After acquiring the base data for training, to reduce the overfitting of the model and increase accuracy, data augmentation needs to be applied. The traditional way of augmenting data is adding copies of manipulated base images by changing their rotation, scale, shear, etc. Many libraries achieve this, including Opencv and Keras. This increases the amount of training data thus increasing accuracy by allowing longer training epochs and decreasing over-fitting since it enables the model to be trained on different modifications of the image. Traditional data augmentation is generally useful in many applications however in medical applications its effect is different. Medical images and especially CT scans taken by the same machine are usually taken in specific orientation and scale, the difference between the machines regarding scale, rotation, and shear is minimal. Therefore, traditional augmentation should only be applied to medical images with limitations to account for different machines however its effects on increasing accuracy for detecting infected lesions on CTs taken by the same machine is negligible.

A more suitable data augmentation technique for medical images is using CNNs, such as UNet or GAN. GANs are used to create synthetic data by understanding the shapes and occurrences of infected lesions in the original data and then imitating them into its generated data. Unlike traditional data augmentation methods, GANs create new images according to the ROI of the images rather than just the rotation, scale, and so on. This produces better results on medical images both in terms of over-fitting and accuracy [21, 22].

The second component of the system is data pre-processing or image processing. Medical images, in general, are produced by different brands of machines with different calibrations and technologies, producing some apparent differences in the produced image, especially in terms of the number of pixels, pixel value ranges, and overall brightness and contrast of the produced image. This requires a data preprocessing step to be done for both the training data and any data used for testing or later for prediction. Histogram equalization is one of the most used methods to deal with brightness and contrast. Sometimes a filter is used to reduce noise. Resizing is used to ensure the same pixel size for all images. Finally, normalization and standardization are used to bring the pixel value range into a specific interval that is most suitable for training. In CT scans Hounsfield units (HU) are used to eliminate unwanted parts of each slice since they express the material of different parts of the scan with specific values. For example, the densest material in the human body is represented by 2000HU, so any values greater than 2000HU are unwanted as they represent objects outside the patient's body seen by the machine, and therefore should be eliminated [23]. Since 1981, transfer learning has been proved by Stevo Bozinovski to be an effective way to build accurate models with relatively little data [24]. Using a pre-trained model on a large generic dataset and then fine-tuning it using application-related data has many benefits. First, accuracy improves as the used application-related data all go towards training the model on the specifics of the ROIs to be detected rather than wasting a portion of it on training the model from scratch. Furthermore, it offers a solution for applications with limited available data, which is usually the case for many medical applications. Another benefit is

reducing the training time of the model drastically, which not only helps with the developing process of the model to reach better performance faster, it also makes it cheaper and more environmentally friendly. Many pre-trained models can be used for medical applications; most of them are trained on colored generic object images such as ResNetx150 which is based on the ResNet model architecture [25]. Other pre-trained models are designed specifically for medical images such as MedicalNet [26] which is a 3D black and white ResNet-based model.

Thus far, all the discussed components are the proven factors for making a CNN-based model perform a highly accurate classification. However, the process of developing such a model also depends highly on trial-and-error experiments. This is where the optimization component takes place. Although we understand generally how CNNs work, and how the mathematics of convolutional filters and gradient descent work at formulating the classification problem, there are many factors that we cannot determine how exactly they affect the training process or the final performance of the model. This includes all of the model's hyper-parameters such as batch size, size of used filters, number of layers, etc. Fortunately, in recent years many tools have been developed to decrease the amount of time spent on experimenting with hyper-parameters before reaching the desired performance. Such tools make use of search algorithms such as grid search and Bayesian optimization to help find the best parameters for the application. These frameworks include Optuna [27] and Ray tune [28]. Moreover, some dashboards help visualize the training process and its results using graphs and statistical analysis and make use of parallel training such as TensorBoard.

The final component of the system is validation. Although in recent years many deep learning-based models are created for medical applications, only a few are used in the health care system. This has many factors, but one of the most important factors that help a model become actively used in hospitals is the validation process. The first aim of the validation process is to determine the performance of a diagnostic model after training in terms of accuracy, sensitivity, specificity, and

some predictive values. Accuracy is the most commonly used evaluation metric although it is not enough to determine the performance of a model. A model that always gives a negative output can have high accuracy when tested with data of mostly negative cases. Sensitivity and specificity fix this problem; they show the rate of true positives and true negatives respectively. But to calculate these results the model needs to be tested on new data then the results should be validated according to ground truth. In a medical setting, ground truth can be reached either by consensus voting of a group of experts or by performing additional medical testing such as PCR tests for covid-19 or a biopsy for tumor-related diseases. Secondly, the validation process extends after the production of the model to its use in clinics. Acceptance testing must be done before using the model in a clinic to make sure its performance is not changing according to the local patient population [20]. Moreover, the quality of the model's performance must be tested for some time to detect any malfunctions, such as biases or wrong predictions when presented with new data features.

Finally, building up the model in the light of the previously discussed components is a process of mixing and matching to find the best combination for the application at hand. Next, the chosen parts are discussed in detail in terms of data collection and augmentation, data pre-processing, choosing model architecture, and using transfer learning for better results, optimization tools, and validation of results.

3. LITERATURE REVIEW

3.1. Covid-19 Datasets

Since the outbreak of covid-19, several CT-scan datasets have been collected and/or created to be used in Artificial intelligence research. Due to the different privacy laws in different countries and the lengthy official procedures to get permission for data collection from hospitals, many of the datasets used in developing deep learning models are not available for public use. However, there are some publicly available datasets with different amounts of patient data, augmentation techniques, and labels. Arranged by the number of citations are the following publicly available datasets.

Song et al. [29] collected a small dataset from 275 patients in China with covid-19, bacterial pneumonia, and healthy cases. Zhang et al. [30] constructed a large dataset from the China Consortium of Chest CT Image Investigation (CC-CCII) of a total of 617,775 CT images of healthy, pneumonia, and novel covid pneumonia complete scans collected from 4,154 patients. Wu et al. [31] collected a classification and segmentation dataset in China consisting of a total of 144,167 CT scan images of covid positive and negative from 750 patients. Yang et al. [32] collected from China a total of 812 CT images for covid-19 and non-covid-19 patients. Rahmizadeh et al. [33] collected a total of 63,846 CT images from Iran of covid-19 positive and negative from 377 patients. Morozov et al. [34] collected 1110 full CT scans from Russia with 5 labels; non-covid-19, mild, moderate, critical, and severe covid-19 pneumonia. Vaya et al. [35] collected 163 annotated CT studies from Valencia. Yan et al. [36] collected a total of 165,667 annotated CT images from 861 patients in China. Wang et al. [37] collected from five hospitals in China a total of 1418 CT scans of covid-19 positive and negative from 1391 patients. Afshar et al. [38] collected a total of 305 scans of covid-19, healthy and community-acquired pneumonia from Iran. Ning et al. [39] collected 19,685 CT slices from

1521 patients in China. Yan et al. [40] collected from two hospitals a total of 828 CT scans of 618 patients with covid-19 and non-covid-19 pneumonia from China and Canada. Tsai et al. [41] collected two datasets, one with 31,856 annotated CT images of 110 patients, and the second with 21,220 CT images of 117 patients. Gunraj et al. [42] constructed a large dataset of different publicly available datasets to form 201,103 CT images of normal, common pneumonia and covid-19 of 4,501 patients from at least 15 countries.

3.2. Covid-19 Classification Models

Deep learning has been used to facilitate many tasks in the fight against the pandemic. Nguyen et al. [43] show different AI models used for diagnosis from medical images, data analysis for covid-19 modeling, computational biology for vaccine and treatment development as well as Internet of Things IOT solutions that help screen and trace patients, and Natural Language Processing models to analyze sentiment and awareness of disease prevention policies. In the medical imaging field, many models have been developed to perform or support diagnosis and prognosis tasks in hospitals. These models vary both in terms of architecture and goal. Here we list the relevant work of classification models trained on CT imaging.

Classification models can perform both diagnosis and prognosis tasks. The diagnosis task determines whether a scan is of a covid-19 positive or negative patient. While the prognosis task determines the severity of the disease to help professionals plan for treatment accordingly. Diagnosis tasks are carried out with binary classification models as follows. Yang et al. [44] developed a self-supervised model using pre-trained DenseNet-169 [45] and ResNet-50 [25] models on the ImageNet dataset [46]. Jaiswal et al. [47] use a pre-trained Dense-Net-201 [45] on the ImageNet dataset. Yang et al. [48] trained a DenseNet-based architecture on high-resolution ct scans. Wang et al. [49] use a pre-trained GoogleNet Inception v3 [50] on the ImageNet dataset. Bai et al. [51] used a pre-trained EfficientNet B4 [52] on the ImageNet dataset. Pathak et al. [53] use a pre-trained ResNet-50 on the

ImageNet dataset. Serte and Demirel [54] created an ensemble model of several ResNet-50 models combining results using majority voting. Mishra et al. [55] developed an ensemble model combining results from 5 modes: VGG16 [56], InceptionV3, ResNet50, DenseNet121, and DenseNet201 using majority voting. Rahimzadeh et al. [57] created a model using ResNet50V2 as the backbone with a Feature Pyramid Network (FPN) and classification layers. Goel et al. [58] used the ResNet50 model after generating data using a generative adversarial network (GAN) optimized by the whale optimization algorithm (WOA). Wu et al. [59] developed a multi-view fusion deep learning model that uses the axial, coronal, and sagittal views of a scan. He et al. [60] proposed a Self-Trans model, a self-supervised model with transfer learning comparing different large datasets that are commonly used for transfer learning. Chen et al. [61] developed a contrastive learning model with a pre-trained encoder.

Other classification models are used to differentiate between covid and other pneumonia-related diseases. Others are used for prognosis as they could classify the severity of covid in the lungs. These models use categorical classifications as follows. Ning et al. [62] use CT scans alongside other clinical findings such as blood and urine tests to train a VGG-16-based model integrated with an ANN model to classify the severity of the disease. Singh et al. [63] proposed an ensemble model of DenseNet201, ResNet152V2, and VGG16 to classify scans as covid-19, tuberculosis, pneumonia, or healthy. Xu et al. [64] developed a ResNet-18-based architecture combined with a location-attention mechanism to classify scans as covid-19, influenza-A viral pneumonia (IAVP), and irrelevant to infection (ITI). Wang et al. [65] designed a novel prior-attention residual learning block by coupling two 3D ResNet models and integrating prior-attention mechanisms to classify scans as covid-19, interstitial lung disease (ILD), and non-pneumonia. Polsinelli et al. [66] developed a SqueezeNet-based [67] model to classify scans as covid-19, community-acquired pneumonia, and healthy. Ouyang et al. [68] created an online attention module with a 3D CNN to classify scans as covid-19, CAP, and healthy.

Yan et al. [69] use a multi-scale convolutional neural network model (MSCNN) to classify scans as covid-19 or CAP.

Haseeb et al. [70], show their extensive survey results in an informative table, similarly we represent the literature review results in Tables 3.1 and 3.2.

Table 3.1. List of top publicly available covid-19 CT scan datasets

Author's Name	Location	Data Structure
Song et al. [29]	China	275 patients with covid-19, bacterial pneumonia, and healthy cases
Zhang et al. [30]	China Consortium of Chest CT Image Investigation (CC-CCII)	617,775 CT images of healthy, pneumonia, and covid-19 full scans collected from 4,154 patients
Wu et al. [31]	China	144,167 CT scan images of covid positive and negative from 750 patients
Yang et al. [32]	China	of 812 CT images for covid-19 and non-covid-19 patients
Rahmizadeh et al. [33]	Iran	63,846 CT images from of covid-19 positive and negative from 377 patients.
Morozov et al. [34]	Russia	1110 full CT scans from with 5 labels; non-covid-19, mild, moderate, critical, and severe covid-19
Vaya et al. [35]	Valencia	163 annotated CT studies of covid-19
Yan et al. [36]	China	165,667 annotated CT images from 861 patients
Wang et al. [37]	China	1418 CT scans of covid-19 positive and negative from 1391 patients

Afshar et al. [38]	Iran	305 scans of covid-19, healthy and community-acquired pneumonia
Ning et al. [39]	China	19,685 CT slices from 1521 patients
Yan et al. [40]	China and Canada	828 CT scans of 618 patients with covid-19 and non-covid-19
Tsai et al. [41]	USA	Two datasets, one with 31,856 annotated CT images of 110 patients, and the second with 21,220 CT images of 117 patients
Gunraj et al. [42]	At least 15 countries	201,103 CT images of normal, common pneumonia and covid-19 of 4,501 patients

Table 3.2. List of top covid-19 binary classification models using CT scans

Source/Author	Dataset Information	Framework/Approach	Performance
Yang et al. [44]	49 covid-19 CT images from 216 patients, and 463 non-COVID-19 CTs	self-supervised model using pre-trained DenseNet-169 and ResNet-50 on the ImageNet dataset	Accuracy: 0.89 F1-score: 0.90
Jaiswal et al. [47]	COVID-CT-Dataset	a pre-trained Dense-Net-201 on the ImageNet dataset	Accuracy: 0.85 F1-score: 0.86
Yang et al. [48]	146 covid-19 patients, and 149 normal patients, High Resolution CT scans	a DenseNet-based architecture on high-resolution CT scans	Accuracy: 0.92 Sensitivity: 0.97 Specificity: 0.87 F1-score: 0.93
Wang et al. [49]	453 COVID CT images	a pre-trained GoogleNet Inception v3 on the ImageNet dataset	Accuracy: 0.829 Sensitivity: 0.81 Specificity: 0.84 F1-score: 0.77

Bai et al. [51]	521 covid-19, and 665 non-covid-19 pneumonia	a pre-trained EfficientNet B4 on the ImageNet dataset	Accuracy: 0.96 Sensitivity: 0.95 Specificity: 0.96
Pathak et al. [53]	413 covid-19 images, and 439 images of normal or non-covid-19 CT scans	a pre-trained ResNet-50 on the ImageNet dataset	Accuracy: 0.93 Specificity: 0.95 Sensitivity: 0.91
Serte and Demirel [54]	214 covid-19 CT scans, and 105 normal CT scans	an ensemble model of several ResNet-50 models combining results using majority voting	Accuracy: 0.84 Sensitivity: 0.1 Specificity: 0.8
Mishra et al. [55]	360 covid-19 CT scans, and 397 normal CT scans	an ensemble model combining results from 5 modes: VGG16, InceptionV3, ResNet50, DenseNet121, and DenseNet201 using majority voting	Accuracy: 0.883 F1-score: 0.867
Rahimzadeh et al. [57]	95 covid-19 CT scans, and 282 normal CT scans	a model using ResNet50V2 as the backbone with a Feature Pyramid Network (FPN) and classification layers	Accuracy: 0.985 Sensitivity: 0.95
Goel et al. [58]	1252 covid-19 CT images, and 1230 non-covid-19 CT images	ResNet50 model after generating data using GANs optimized by WOA	Accuracy: 99.22 Sensitivity: 99.78 Specificity: 97.78 F1-score: 98.79
Wu et al. [59]	368 covid-19 CT scans, and 127 non covid-19 pneumonia	a multi-view fusion deep learning model	Accuracy: 0.7 Sensitivity: 0.73 Specificity: 0.615
He et al. [60]	216 covid-19 CT scans, and 133 normal CT scans	a self-supervised model with transfer learning	Accuracy: 0.86 F1-score: 0.85

Chen et al. [61]	216 covid-19 CT scans, and 171 normal CT scans	a contrastive learning model with a pre-trained encoder	Accuracy: 0.868 Sensitivity: 0.872
Ouyang et al. [68]	3389 covid-19 CT images, and 1593 CAP CT images	an online attention module with a 3D CNN	Accuracy: 0.875 Sensitivity: 0.869 Specificity: 0.9 F1-score: 0.82
Yan et al. [69]	416 covid-19 CT scans, and 412 non-covid-19 pneumonia CT scans	a multi-scale convolutional neural network model	Accuracy: 0.875 Sensitivity: 0.89 Specificity: 0.857

Table 3.3. List of top non-binary covid-19 classification models using CT scans

Source/Author	Dataset Information	Framework/Approach	Performance
Ning et al. [62]	1,521 patients with negative, mild and severe covid-19 CT scans with 130 clinical features	a VGG-16-based model integrated with an ANN model to classify the severity	Accuracy: 0.95, 0.83, 0.88 Sensitivity: 0.85, 0.88, 0.71 Specificity: 0.998, 0.79, 0.93
Singh et al. [63]	3038 healthy, 2890 non-covid-19 pneumonia, 3193 tuberculosis, and 2373 covid-19 CT images	an ensemble model of DenseNet201, ResNet152V2, and VGG16	Accuracy (overall): 0.988 Sensitivity (overall): 0.988 Specificity (overall): 0.988 F1-score (overall): 0.98
Xu et al. [64]	175 healthy, 224 influenza-A, 219 covid-19 CT scans	a ResNet-18-based architecture combined with a location-attention mechanism	Accuracy (overall): 0.87

Wang et al. [65]	936 healthy, 2406 ILD, 1315 covid-19 CT scans	a novel prior-attention residual learning block by coupling two 3D ResNet models and integrating prior-attention mechanisms	Accuracy: 0.915, 0.89, 0.93 Sensitivity: 0.82, 0.885, 0.876 Specificity: 0.935, 0.9, 0.955
Polsinelli et al. [66]	A total of 397 CT scans of healthy and non-covid-19 pneumonia, and 360 covid-19 CT scans	SqueezeNet-based model	Accuracy (overall): 0.85

4. MEDICAL INFORMATION

4.1. Diagnosis

The main goal of the AI system is to correctly diagnose covid-19 patients from their CT scans with accuracy on par with radiologists. Thus, it is needed to analyze the problem from both medical and technological sides. First, we discuss CT scans, why they are a good medium to study covid-19 effects, the manifestations of covid-19 on CT scans, and how radiologists diagnose it.

Computed Tomography (CT) scans are a type of medical imaging where a computer is used to process a combination of X-ray measurements taken from different angles. The result of a lung CT scan is a large number of cross-sectional X-ray images of the lung, arranged in order from the top to the bottom of the patient's chest. This results in a highly detailed volume of images of organs, tissues, bones, and other elements that cannot be otherwise seen without invasive procedures. CT scan tests include an amount of radiation, however, some developed software can help get a highly detailed scan with reduced radiation dosages.

In the case of covid-19, the current standard test to definitively diagnose the disease is the transcription-polymerase chain reaction assay (rt-PCR). However, to understand the extent of the damage produced by the disease, track its progress, and create treatment plans, analyzing radiological images is necessary. Unlike CT scans, Chest X-ray (CXR) has less radiation but is not sensitive enough for pulmonary abnormalities detection, especially at the early stage of the disease. CT scan is proven to be effective for distinguishing covid-19 abnormalities as well as estimating the evolution of the disease [71]

To date, many studies have been published about specific covid-19 findings in CT scans [72]. However, most of these studies rely on experience as there is a deficiency in radiologic-pathologic correlations studies [73]. This makes the collective knowledge of diagnosing covid-19 from CT scan based mainly on experience or in technological terms; data. Similar to a deep learning model, radiologists learn some features of covid-19 ct scans from confirmed cases and diagnose new scans by searching for similar findings.

The CT scans of covid-19 patients present the effects or damage the disease causes in the patient's lungs which have some unique characteristics that we can differentiate from the effects of pneumonia or other lung-damaging diseases. Radiologists screen for the following when assessing a potential covid-19 ct scan: ground-glass opacities (hazy or grey areas caused by air displacement by fluid), consolidation (a region of lung tissue that became of airless solid consistency), reticular pattern (a collection of small linear opacities that can appear like a net without significant ground-glass opacity), mixed pattern (combination of all previously stated findings) and honeycomb pattern [74]. The most common findings of covid-19 are ground-glass opacities, bilateral abnormalities (on both lungs), lower lobe involvement, and posterior predilection [73].

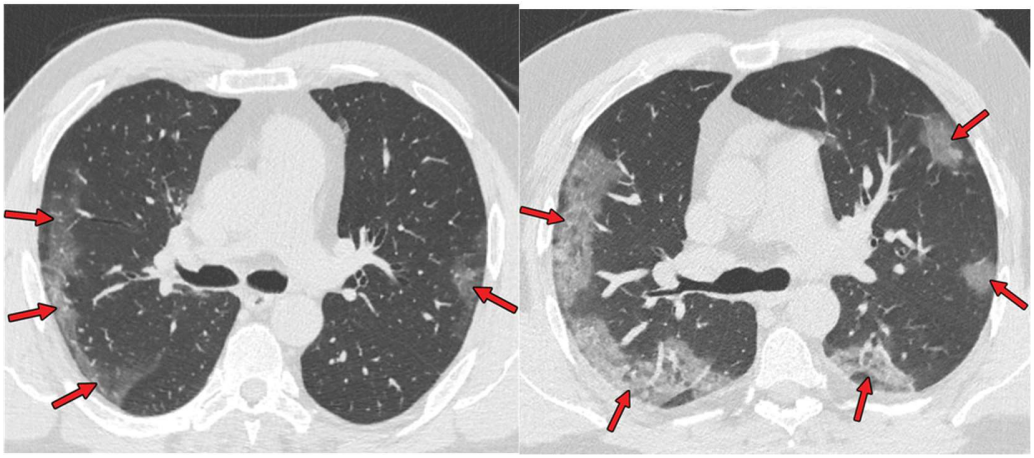

Figure 4.1. Covid-19 pneumonia in two patients showing bilateral areas of ground-glass opacities (arrows) in a peripheral distribution. Adapted from [73].

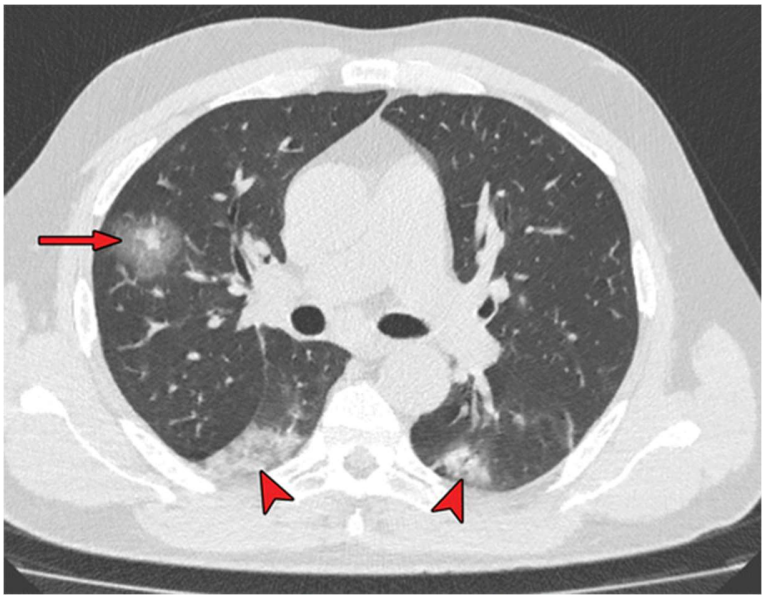

Figure 4.2. RT-PCR-test–proven covid-19 patient shows consolidation surrounded by ground-glass opacity in the right lobe (arrow) and consolidation in both lower lobes (arrowheads). After [73]

4.2. Scoring

After recognizing the discussed patterns, a scoring system is needed to quantify the severity of the disease. Similar to diagnosis there are many publications with different methods for scoring covid severity from medical imaging [71]. As Chest X-ray (CXR) is typically used for monitoring the disease progression since it exposes the patient to fewer radiations, there are several scoring systems based on X-ray images. For CT images, one method is called Chest Computed Tomography Severity Score (CT-SS) which uses lung opacification as an equivalent for extension of the disease in the lungs. The lung is divided into 20 regions, each region is evaluated and given a score of 0,1 or 2 depending on the parenchymal opacification involved: 0%, 1-50%, or 51-100%, respectively. The scores are then added up to get a total score of range 0-40 points [75]. Another method is Total Severity Score (TSS)

where both lungs are divided into 5 lobes, each assessed for inflammatory abnormalities including ground-glass opacities and consolidation, then gives a score for each lobe in the range of 0-4 points depending on the percentage of the involved lobe: 0 (0%), 1 (1-25%), 2 (26-50%), 3 (51-75%), or 4 (76-100%). The TSS is then calculated by summing the lobes' points [76].

4.3. Temporal Development

Disease progression starting from first symptoms prevalence can be roughly divided into 4 stages; early-stage days 0-5 where ground-glass opacity is prominent, progressive stage days 6-8 where ground glass opacities increase and crazy-paving appears, peak stage days 9-13 where consolidation increases and late-stage 14 days and more where consolidation and ground-glass opacities gradually decrease [73]. In a more detailed study, it is shown that before symptoms onset only 4 out of 10 patients present abnormalities in CT scans, 2 present pure ground-glass opacities, and 2 present consolidations [74]. Illness days 0-5 after symptoms onset, ground-glass opacity is the most prominent with a percentage of 62%, crazy-paving pattern comes in second place with 24%, and consolidation with 23%. Illness days 6-11 present decreasing ground-glass opacities, decreasing crazy-paving pattern, and consistent consolidation with 24%. Illness days 12-17 ground-glass opacity drops to 45% and a large increase in mixed patterns occurs from 1% to 38%. The reported results are consistent with other temporal studies [77,78].

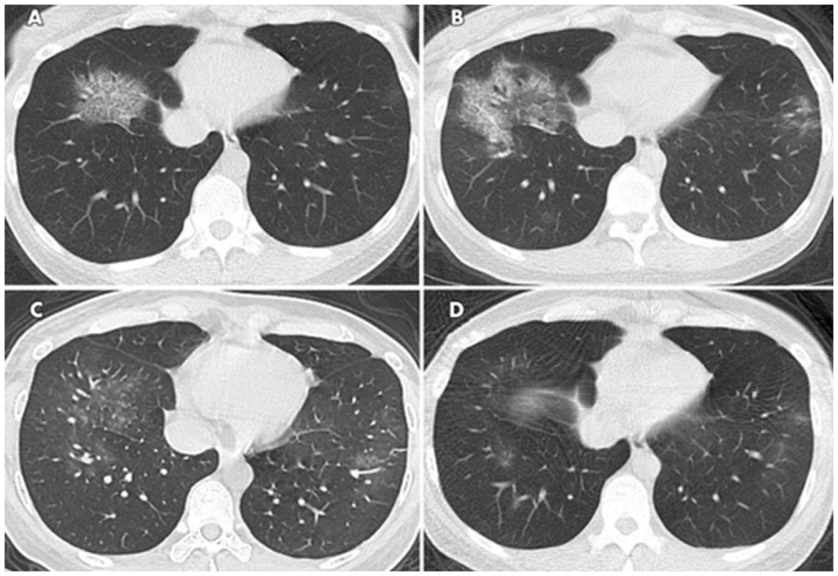

Figure 4.3. CT scans of coronavirus 2019 pneumonia progression. ***A*** Scan obtained on illness day 3 shows ground-glass opacity with a crazy-paving pattern in the right lower lobe. ***B***, Scan obtained on illness day 7 shows crazy-paving pattern superimposed on ground-glass opacity with increased extent. Note that patchy ground-glass opacity is newly developed in the left lower lobe. ***C***, Scan obtained on illness day 12 shows the absorption of abnormalities, with pure ground-glass opacity left in both lower lobes. ***D***, Scan obtained on illness day 17 shows obvious absorption of abnormalities. Only small pure ground-glass opacity is observed in both lower lobes. The patient was discharged on illness day 20. The day of initial symptom onset was defined as illness day 0. After [74].

5. METHODS

5.1. Data

The model uses chest CT scans as input. 3D chest CT scans are acquired using computed tomography scanners that take X-ray images of the chest from different angles, then process them to produce a number of highly detailed 2D slices that constitute a volume or a 3D image of the patient's chest. The 2D slices are arranged to show the lungs and chest area from top to bottom. The sequence of the slices is important when using the data in a 3D format to keep the volumetric information. The thickness of the tissue represented in each slice varies depending on the used machine; therefore, the total number of 2D slices in a CT scan varies accordingly.

5.1.1. Dataset

The used dataset for this project is the publicly available MosMed dataset published by the Center of Diagnostics and Telemedicine in Russia [34]. It consists of 1110 chest CT scans of 1110 anonymized patients, obtained between March and May of 2020, and provided by municipal hospitals in Moscow, Russia. The scans are split into 4 folders based on the severity of the disease: CT0 folder contains 254 scans of healthy lung images (non-consistent with pneumonia including covid-19), CT1 contains 684 scans of mild covid-19 infection or less than 25% ground-glass opacities involvement in lungs, CT2 contains 125 scans of moderate covid-19 infection with ground-glass opacities involvement between 25% and 50%, CT3 contains 45 scans of severe covid-19 infection with between 50% and 75% ground-glass opacities and consolidation involvement, lastly CT4 contains 2 scans of critical covid-19 infection with more than 75% ground-glass opacities, consolidation, and reticular changes.

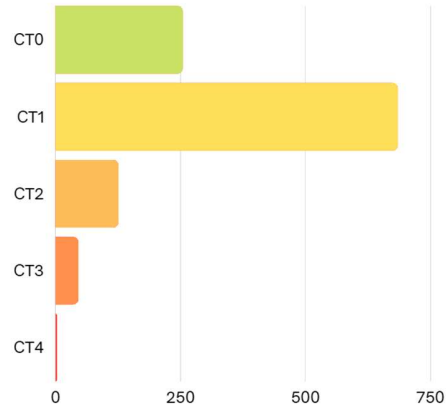

Figure 5.1. MOSMED dataset number of scans according to each class

The data is uploaded in NIfTI format which was converted directly from the original DICOM format of the scans. DICOM is the standard format for medical imaging while NIfTI is a file format usually used in neuroimaging but sometimes used for other types of medical imaging. The main difference between DICOM and NIfTI formats is that DICOM stores a scan as 2D slices, whereas NIfTI stores a scan as a 3D volume. Currently, the most efficient Python library to process NIfTI data is Nibabel [79].

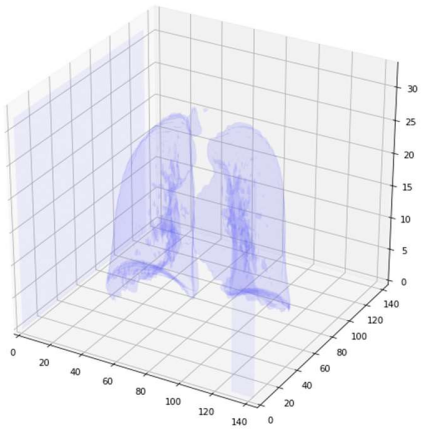

Figure 5.2. 3D visualization of a covid-19 patient chest ct scan acquired from [34] after segmentation.

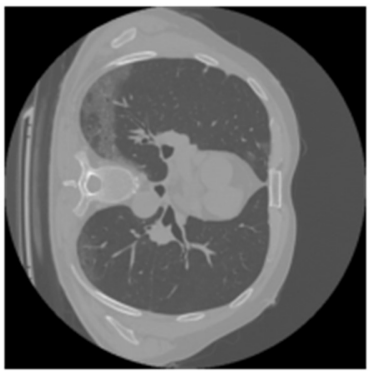

Figure 5.3. Original 2D slice number 27 of the same covid-19 patient acquired from [34].

5.1.2. Pre-Processing

The data pre-processing phase of this project consists of multiple functions, both on the 2D and 3D levels. On the 2D level, the following methods are applied to each slice individually. The first is used to omit unwanted parts of the slice and create a better system for the pixel values. CT scans use Hounsfield units (HU) to express the material of different parts of the scan. Different tissues in the human body absorb different amounts of the scanner's emitted X-ray. Hounsfield units are used to represent the amount of X-ray absorption, and therefore the material of an element in the scan. For example, air is expressed as -1000 HU, water at 0 HU, and very dense bones at 2000 HU [79]. Notice in figure 2, the heart and other soft tissues have a similar gray shade while bones have lighter almost white color, and areas inside the lungs are darker as it is mostly air. From this information, it is safe to set pixels of values greater than 2000 to zero, since there couldn't be a material in the patient's body higher than 2000 HU which means these pixels are out of the scan.

The second function is further segmentation to get the region of interest (ROI) or saliency information of each scan. Skimage python library is used to find

different regions in each slice using the region props function. The result is a segmented slice with only the ROI for covid cases, which are the lungs only.

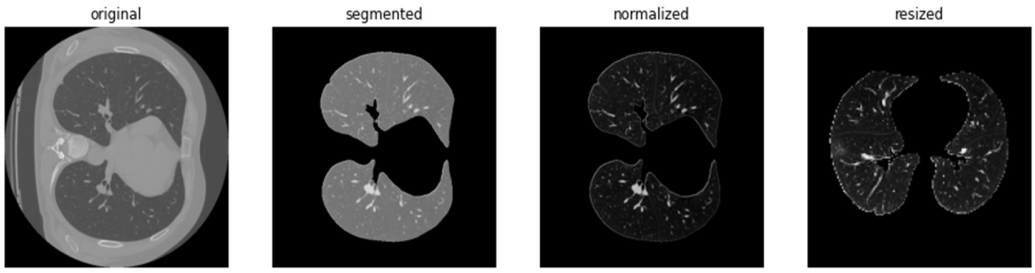

Figure 5.4. Segmented 2D slice number 27 of the same covid-19 patient.

The third function is Normalization. After getting rid of unwanted pixel data, and keeping only the region of interest pixels, it is important to normalize data images to ensure that each pixel has a similar data distribution. This eliminates data inconsistency and makes the convergence of the model faster when training [80]. The normalization function calculates the mean and standard deviation of the image and subtracts pixels by the mean and divides by the standard deviation. Thus, returning the image with pixels ranging from 0 to 1.

Next, on the 3D level, it is important to prepare the scans to fit into the input of the 3D CNN model. First, since each patient's scan can have a different number of slices, we can omit some of the slices at the beginning and end of a scan as the furthest top and bottom parts of the lung do not usually contain any ROI resulting in a dark image after segmentation. Therefore, we set a specific size for the depth of a scan (number of slices). Secondly, we divide each scan's slices (depth) into batches for easier learning of the model. So, we set a specific depth size for each batch, in this case, 40, to have all 3D inputs of the same size. Then, we rescale the other two dimensions height and width into 144 and 144. At this point, batches of 3D data are ready to be input into the model.

The last step before training is splitting the processed dataset into train, test, and validation sets. The first two sets are used in training and producing the model's

initial evaluation metrics. Then the validation set is used to evaluate the model on new data after training. The dataset is split into 60% training (304 cases), 20% test (104 cases), and 20% validation (100 cases). However, for K-fold cross-validation, we combine training and validation sets and use sklearn library's Kfold algorithm to split data randomly into train and validation for each fold.

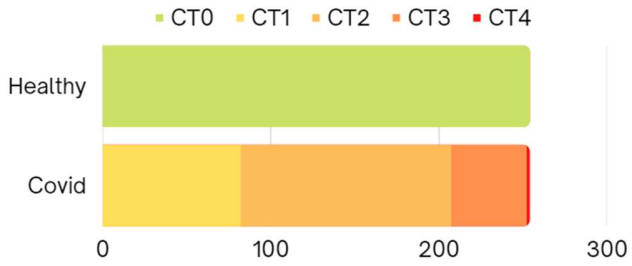

Figure 5.5. Our used balanced subset of MOSMED dataset for classification tasks

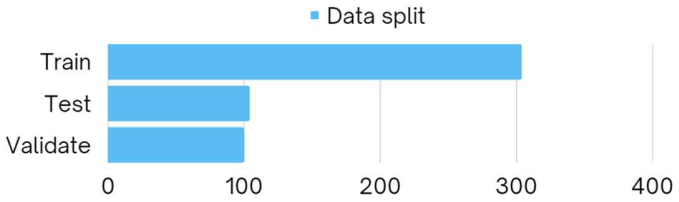

Figure 5.6. Data split of the balanced subset of MOSMED dataset used for classification

5.2. Models

As mentioned before, many popular models created for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) perform well on many applications by using the trained model directly or after using transfer learning. Convolutional models and Attention models have been performing increasingly better, proving their capabilities in image classification. In order to choose an appropriate model for

diagnosing covid-19 in CT scans, two aspects of covid-19 diagnosis need to be taken into consideration. The first is the features of covid-19 patterns such as ground-glass opacities and consolidations. The second is the placement of the found features in the lungs. The proposed model is an ensemble model that covers the most important points for covid-19 diagnosis by combining a 3D CNN model for considering volumetric information, and a 2D attention-based model Convit for recognizing covid-19 patterns in detail in slices of lung scan. Such a model consists of three important components, a 3D model, a 2D model, and the combination of the results of both models. First, we train the 3D CNN and ConViT models separately using the same pre-processed and sampled dataset as shown in Figure 5.7. Then we train a linear regression model using the classification output of 3D CNN on a full CT scan as covid-19 or healthy, and the classification output of ConViT on 11 middle slices of the same scan. This builds up the prediction flow as shown in Figure 5.8.

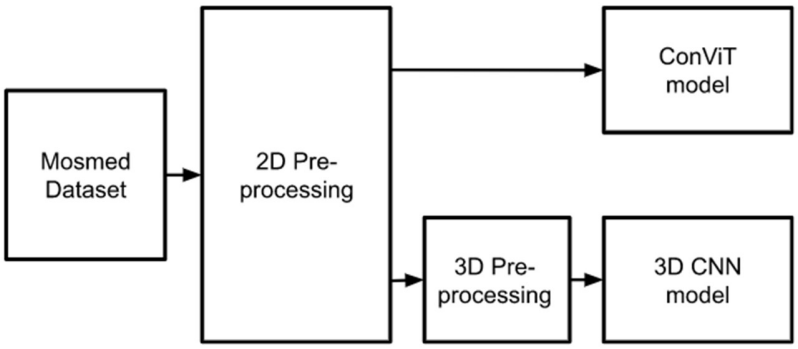

Figure 5.7. Training flow of the proposed model

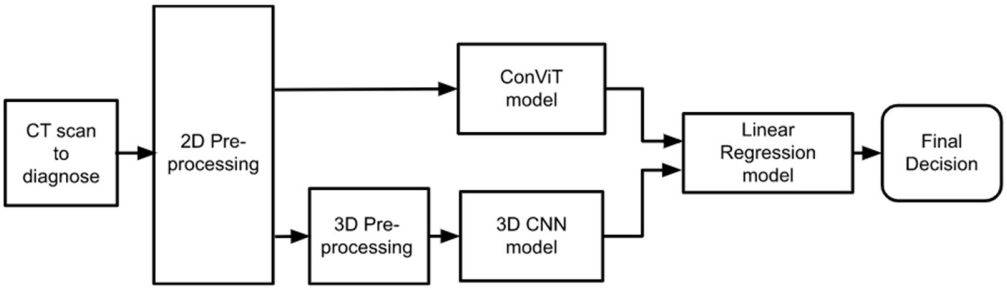

Figure 5.8. Prediction flow of the proposed model

5.2.1. Volumetric Convolutional Neural Networks

To build our diagnosis 3D CNN model architecture, a LeNet-based architecture was chosen for 3D CNN classification. A combination of different numbers of CNN layers and filters were tested, as well as the use of different optimizing layers. The most efficient architecture consists of 3 convolutional layers each followed by max-pooling and batch normalization. After that, a global average pooling layer is used before a fully connected dense layer with dropout, followed by a fully connected output layer. All layers use the ReLu activation function except for the output layer which uses sigmoid. The exact sizes of each layer are shown in Figure 5.9.

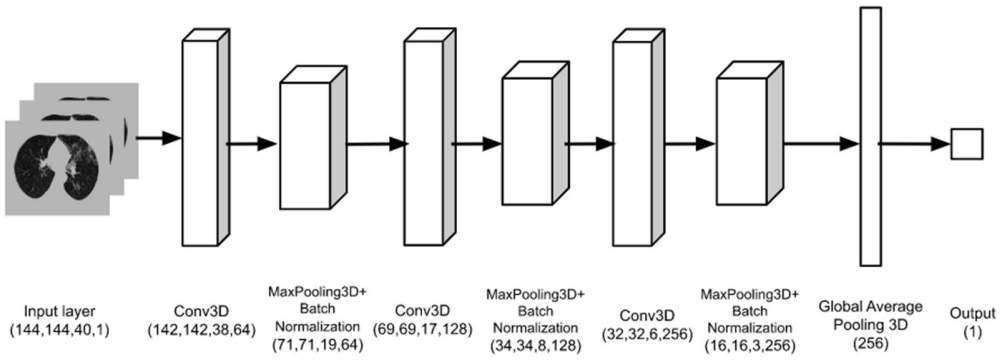

Figure 5.9. Architecture of the proposed 3D CNN model

For this model to have a balanced dataset for training we use all 254 normal cases, all CT2, CT3, and CT4 cases which add up to 172 covid-19 cases, we then add 83 cases of mild covid-19 folder CT1 to have a total of 254 covid-19 cases. The balanced dataset then undergoes the mentioned pre-processing algorithm to produce an input to the model of size (144, 144, 40).

The 3D CNN model is trained with an Adam optimizer of an initial learning rate of 0.0001, using Keras exponential decay function with a decay rate of 0.96. The loss function is binary cross-entropy, trained on 100 epochs, batch size of 2, and saved best weights. Hardware used is Google colab pro virtual machines, providing 25 GB RAM, and Tesla-P100 GPU.

The model was trained with 5-fold cross-validation giving average accuracy of **95%** then evaluated on a test set resulting in accuracy, sensitivity, and specificity of **92%**.

Table 5.1. Confusion matrix for 3D CNN model on test set

	Predicted Negative	Predicted Positive
Actual Negative	44	8
Actual Positive	8	44

Many experiments were conducted to reach this performance, by modifying data pre-processing, model architecture, and hyperparameters. The most relevant experiments are stated in Table 5.2.

Table 5.2. Performance results of proposed 3D CNN model

Exp. no	Data	Model	Hyperparameters	Results
1	Mosmed-508 254 healthy and 254 from ct1+ct2+ct3 304 train - 104 test - 100 val + New Segmentation	LeNet based with 3 3D convolutional layers **(32, 64, 128)** + Dense 256 + Dropout layer 0.7	Optimizer: Adam Learning rate: 0.0001 with decay rate 0.96 Batch size: 2 Epochs: 250	On test set: Accuracy: 0.86 Sensitivity: 0.88 Specificity: 0.84
2	Mosmed-508 254 healthy and 254 from ct1+ct2+ct3 304 train - 104 test - 100 val + New Segmentation	LeNet based with 3 3D convolutional layers (64, 128, 256) + Dense 256 + **Dense 128** + Dropout layer 0.7	Optimizer: Adam Learning rate: 0.0001 with decay rate 0.96 Batch size: 2 Epochs: 250	On test set: Accuracy: 0.875 Sensitivity: 0.9 Specificity: 0.85
3	Mosmed-508 254 healthy and 254 from ct1+ct2+ct3 304 train - 104 test - 100 val + New Segmentation	LeNet based with 3 3D convolutional layers (64, 128, 256) + Dense 256 + **Dropout layer 0.5**	Optimizer: Adam Learning rate: 0.0001 with decay rate 0.96 Batch size: 2 Epochs: 250	On test set: Accuracy: 0.86 Sensitivity: 0.75 Specificity: 0.98
4	Mosmed-508 254 healthy and 254 from ct1+ct2+ct3 304 train - 104 test - 100 val + New Segmentation	LeNet based with 3 3D convolutional layers (**64, 128, 256)** + Dense 256 + Dropout layer 0.7	Optimizer: Adam Learning rate: 0.0001 with decay rate 0.96 Batch size: 2 Epochs: 250	On test set: Accuracy: 0.92 Sensitivity: 0.92 Specificity: 0.92

A similar prognosis model was also trained by changing the output layer to 4 and loss function to categorical cross-entropy. Even after data sampling to balance the amount of data in each class, a prognosis model could not be trained on such a small dataset.

5.2.2. Convolutional-Like Vision Transformers

Since Dosovitskiy et al.[81] introduced Vision transformers, transformers have been increasingly used in image recognition and classification tasks. One approach that makes use of both convolution and vision transformers is ConViT [82]. ConViT proposes Gated Positional Self-Attention (GPSA) layers, which initially act as convolutional layers in terms of locality but can be adjusted by a gating parameter that controls the attention paid to position versus content. This creates a self-attention model with soft convolutional inductive bias, combining the CNN's ability to train on relatively small data and the great performance of flexible self-attention.

As shown in Figure 5.7, the GPSA layer is based on the combination of two ideas, multi-head self-attention, and self-attention as a generalized convolution. Multi-head self-attention uses queries W_{qry} and keys W_{key} as well as the linear projections of embed patches X_i and X_j to produce an attention filter. While convolution property of a convolutional layer with filter size $\sqrt{N_h} \times \sqrt{N_h}$ is generated using multi-head positional self-attention by applying the following conditions, as shown by Cordinnier et al. [83]:

$$\begin{aligned} v_{pos}^{h} &:= -\alpha^{h}(1, -2\Delta_1^h - 2\Delta_2^h, 0, \ldots 0) \\ r_{\delta} &:= (||\delta||^2, \delta_1, \delta_2, 0, \ldots 0) \\ W_{qry} &= W_{key} := 0,\ W_{val} := I \end{aligned} \tag{5.1}$$

$N_h$is the number of heads and learnable relative position encodings of the positional self-attention. Δ^h is the center of attention, which is the position where the head h pays most attention to. α^h is the locality strength, which controls how focused the attention is around its center. Therefore, GPSA layers initially act purely convolutional by setting the mentioned conditions, then the addition controlled by the learned gated parameter λ gives the layer freedom to escape convolutional locality.

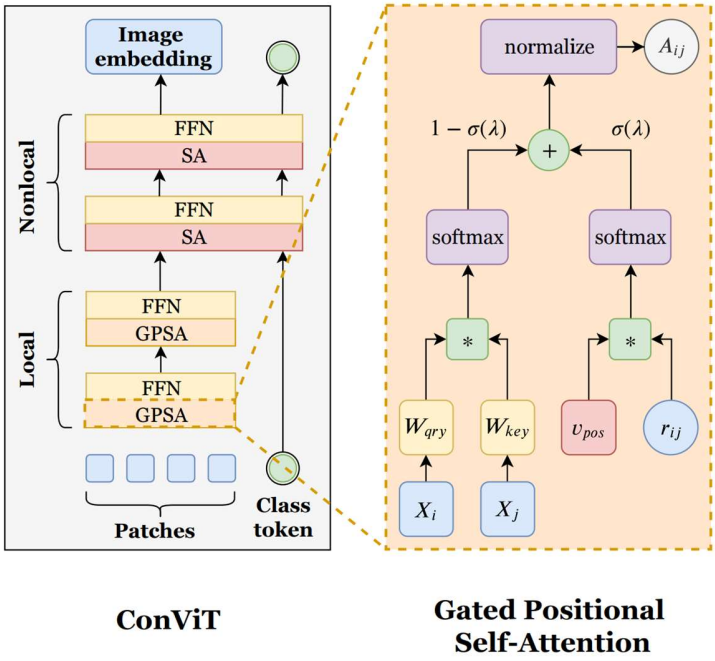

Figure 5.10. Architecture of ConViT (left) and architecture of GPSA layers (right), after [82]

Using the publicly released code and models on Github stated in the paper [82] with some modifications, multiple experiments have been conducted for Covid-19 diagnosis on Mosmed dataset as shown in Table 5.3. The training was conducted using Google colab pro virtual machines, providing 25 GB RAM, and Tesla-P100 GPU. The best performing model was trained with a batch size of 16 on 50 epochs with locality up to layer 2, locality strength of 0.5, and embed dimension of 64. Since the public code does not include a custom dataset option, we modified the ImageNet dataset function to load and normalize the CT data correctly by calculating the mean and standard deviation of the used dataset instead of using the pre-defined ImageNet values. Accuracy calculation was modified as well to obtain accuracy for binary classification rather than the top k categories method. After using the predefined ConViT model architectures such as convit-tiny, we created a smaller version with 3 self-attention heads instead of the smallest model with 4 attention heads. Figure 5.11 shows the used architecture.

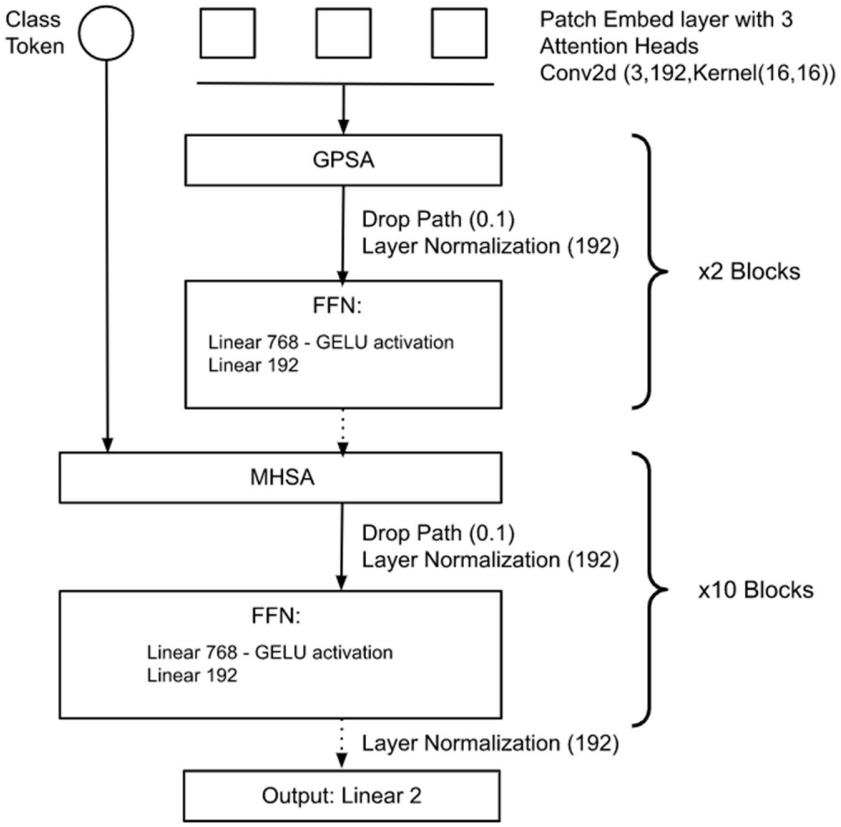

Figure 5.11. Architecture of the proposed ConViT model

Table 5.3. Performance results of proposed ConViT model

Exp. no	Data	Model	Hyperparameters	Results
1	Mosmed-508 254*5 healthy and 254*5 covid ct1+ct2+ct3 (including only 5 middle slices) + New Segmentation	ConViT tiny	Mixup: 0 Cutmix: 0 Colorjitter: 0 Batch size: 16 Epochs: 50	Val accuracy: 0.84
2	Mosmed-508 254*5 healthy and 254*5 covid	ConViT mini	Mixup: 0 Cutmix: 0	Val accuracy:

	ct1+ct2+ct3 (including only 5 middle slices) + New Segmentation		Colorjitter: 0 Batch size: 16 Epochs: 50	0.86
3	Mosmed-508 254*5 healthy and 254*5 covid ct1+ct2+ct3 (including only 5 middle slices) + New Segmentation	ConViT mini	Mixup: 0 Cutmix: 0 Colorjitter: 0 Batch size: 16 Epochs: 50 Locality strength: 0.5 Local up to layer: 2 Embed dim: 64	Val accuracy: 0.88

For testing, we load each original scan and then choose the number of slices on which we classify the scan as covid-19 or healthy. Although most publications choose middle slices directly, we found it better to choose slices a bit lower than the middle since covid-19 patterns tend to affect the lower lobes of the lungs. Since each scan contains around 40 slices, we choose slice 26 as the middle slice and we add 5 slices before and 5 slices after it to classify 11 slices. Then using the majority voting technique, we add up classification results to find the final decision. We then calculate the metrics for model performance, as shown in Table 5.4 and 5.5.

Table 5.4. Performance results on a test set of proposed ConViT model

Model	No. of slices	Accuracy	Sensitivity	Specificity
ConViT	13	0.79	0.6	0.98
ConViT	**11**	**0.84**	**0.7**	**0.98**

ConViT	7	0.83	0.74	0.92
ConViT	5	0.79	0.64	0.94

Table 5.5. Confusion matrix for ConViT model on test set

	Predicted Negative	Predicted Positive
Actual Negative	49	1
Actual Positive	15	35

A similar prognosis model was also trained by changing the number of classes to 4. Trained with and without data sampling to balance the amount of data in each class, a prognosis model could not be trained on such a small dataset.

5.2.3. Residual Networks

The most used models for covid-19 diagnosis are ResNet50 and ResNet101. For comparison, we trained both ResNet50 and ResNet101 models using the same pre-processed 2D data used for ConViT training. We used the ImageNet weights but trained all layers without freezing so it could help the model converge faster while learning the specifics of medical imaging rather than generic colored images of ImageNet. We then added a fully connected layer and an output layer with a sigmoid activation function as shown in Figure 5.12. The loss function used is binary cross-entropy with an Adam optimizer and a learning rate of 0.0001. This trains a ResNet50 model with a validation accuracy of **0.867** and a ResNet101 model with a validation accuracy of **0.872**.

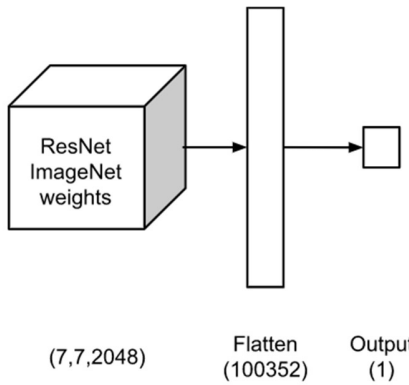

Figure 5.12. Architecture of the used ResNet model

For testing, we use the same method used for testing ConViT performance. Compared to the same number of slices, the results are shown in Tables 5.6, 5.7, and 5.8.

Table 5.6. Performance results on a test set of ResNet50 model

Model	No. of slices	Accuracy	Sensitivity	Specificity
ResNet50	13	0.83	0.80	0.86
ResNet50	**11**	**0.83**	**0.82**	**0.84**
ResNet50	7	0.78	0.80	0.76
ResNet50	5	0.80	0.80	0.8
ResNet101	13	0.87	0.88	0.86
ResNet101	**11**	**0.87**	**0.88**	**0.86**
ResNet101	7	0.85	0.90	0.80
ResNet101	5	0.83	0.90	0.78

Table 5.7. Confusion matrix for ResNet50 model on test set for 11 slices

	Predicted Negative	Predicted Positive
Actual Negative	42	8
Actual Positive	9	41

Table 5.8. Confusion matrix for ResNet101 model on test set for 11 slices

	Predicted Negative	Predicted Positive
Actual Negative	43	7
Actual Positive	6	44

5.2.4. Linear Regression

After training 2D and 3D models separately, we use a linear regression model to combine the results of the two. We use the decision of each classified slice of the 2D model along with the decision of the 3D model as input. The goal of a regression model is to find a classification decision based on the results of both models combined by best fitting weights and biases. The model is trained with a loss function of binary cross-entropy and an Adam optimizer with a learning rate of 0.01. The results of training on 50 epochs are a training and validation accuracy of 0.98.

On the test set, the model produces a specificity of **0.92**, a sensitivity of **0.88**, and an accuracy of **0.9**.

Table 5.9. Confusion matrix for linear regression model on test set

	Predicted Negative	Predicted Positive
Actual Negative	44	6
Actual Positive	5	45

6. RESULTS

The proposed ensemble model trained on very small balanced data of 508 CT scans, performs better than top published classification models trained on datasets of similar size. The model is also superior for its light weight, small total number of parameters and short time of training. The model was trained on virtual machines provided by Google's Colab Pro, providing 25 GB RAM, and Tesla-P100 GPU with total training time less than two hours.

The full proposed model on test set results an accuracy of **0.9**, sensitivity or recall of **0.88**, specificity of **0.92**, precision of **0.88** and F1-score of **0.88**.

Table 6.1. Comparing proposed model to related work

Author/model	Data	Approach	Performance
Proposed model	254 healthy CT scans, and 254 covid-19 CT scans	Ensemble model of 3D CNN and ConViT with Linear Regression	Accuracy: **0.9** Sensitivity: **0.88** Specificity: **0.92** F1-score: **0.88**
Mishra et al. [55]	360 covid-19 CT scans, and 397 normal CT scans	an ensemble model combining results from 5 modes: VGG16, InceptionV3, ResNet50, DenseNet121, and DenseNet201 using majority voting	Accuracy: 0.883 F1-score: 0.867
He et al. [60]	216 covid-19 CT scans, and 133 normal CT scans	a self-supervised model with transfer learning	Accuracy: 0.86 F1-score: 0.85

Chen et al. [61]	216 covid-19 CT scans, and 171 normal CT scans	a contrastive learning model with a pre-trained encoder	Accuracy: 0.868 Sensitivity: 0.872

7. CONCLUSION

The use of computer vision in the medical field, although increasingly beneficial in terms of speed and confidence in diagnosis and prognosis tasks, will always have the challenge of limited data. Medical images for new or rare diseases are scarce, and even for common diseases are limited due to restrictions and regulations. This makes the development of computer vision models on small datasets a critical task.

The steps taken to achieve this goal starts with preprocessing of data. Although many data augmentation techniques are available for computer vision tasks, medical images need to keep many of their features unchanged to avoid information loss. Thus, in this study, we only apply segmentation and normalization to CT scans before training. In the future, augmentation techniques with minimal alterations of medical images, such as Mixup and CutMix, can be used to increase the amount of data. To choose a model for a medical diagnosis task it is important to take into consideration the specification of the disease and its manifestation in medical imaging. Covid-19 causes ground-glass opacities, consolidation, and reticular patterns especially at the back and lower lobes of the lungs. This gives priority to the 3D information of a CT scan as well as the patterns in 2D slices. This study explored different architectures and principles in computer vision to achieve this task and proposed combining models in order to achieve good performance using a very small dataset.

Convolutional Neural Networks have been developed for decades for computer vision tasks and are the most expected to perform well. The 3D CNN model makes use of the volumetric information of CT scans which is essential as position in the third dimension plays a critical role in diagnosis. We used a 3D LeNet-based CNN model to achieve this goal. Lately, Attention-based models have been increasingly used in computer vision tasks. Vision transformers perform well on images but are not used widely in medical imaging due to their small datasets.

We used the Convolutional-Like Vision Transformer to make use of the convolution's ability to learn from small datasets and attention's superior performance. The performance of the ConViT model in this study proves the efficiency of attention in medical imaging. With its small architecture and the much smaller number of parameters, it performs as well as the complicated architecture of the ResNet50 on a small dataset.

This study proposed an ensemble of a 2D attention based ConViT model and a 3D CNN model with linear regression, although trained on only 508 cases split into train, validation, and test sets produces an accuracy of 90% with a specificity of 92% and sensitivity of 88%. Such a model proves that computer-aided diagnosis systems can be trained and used for diseases with limited data.

REFERENCES

1. Lodwick, G.S., Haun, C.S., Smith, W.E., Keller, R.F., Robertson, E.D., "Computer Diagnosis of Primary Bone Tumors" *Radiology,* Vol. 80:2, pp. 273-275, 1963.

2. Winsberg, F., Elkin, M., Macy, J., Bordaz, V., Weymouth, W., "Detection of radiographic abnormalities in mammograms by means of optical scanning and computer analysis." *Radiology,* Vol. 89, pp. 211–215, 1967.

3. Ishida, M., Kato, H., Doi, K., Frank, P.H., "Development of a new digital radiographic image processing system.", *Proc SPIE*, Vol. 347, pp. 42–48, 1982.

4. Giger, M.L., Doi, K., "Investigation of basic imaging properties in digital radiography. I. Modulation transfer function." *Med Phys*, Vol. 11(3), pp. 287-95, 1984.

5. Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., … Lungren, M. P., "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists." *PLoS Medicine*, Vol. 15(11), 2018.

6. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D., "Backpropagation applied to handwritten zip code recognition." *Neural computation,* Vol 1(4), pp. 541-551, 1989.

7. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86(11), pp. 2278-2324, 1998.

8. Simard, D., Steinkraus, P. Y., and Platt, J. C., “Best practices for convolutional neural networks.” ICDAR, 6 August 2003, Vol 3.

9. Lo, S.C.B., Lin, J.S., Freedman, M.T., Mun, S.K., “Computer-assisted diagnosis of lung nodule detection using artificial convolution neural network.” *Proc SPIE* Vol. 1898, pp. 859–869, 1993.

10. Chan, H.P., Lo, S.C.B., Helvie, M.A., Goodsitt, M.M., Cheng, S.N.C., Adler, D.D., “Recognition of mammographic microcalcifications with artificial neural network.”, *Radiology,* Vol. 189, pp. 318, 1993.

11. Krizhevsky, A., Sutskever, I., Hinton, G., “ImageNet Classification with Deep Convolutional Neural Networks.” *Advances in neural information processing systems,* 25, 2012.

12. Yosinski, J., Clune, J., Bengio, Y., Lipson, H., “How transferable are features in deep neural networks?” *Proceedings of the Advances in neural information processing systems* (NIPS’14), pp 3320–3328, 2014.

13. Simonyan, K., Zisserman, A., “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *arXiv preprint,* arXiv: 1409.1556, 2014.

14. Moreira, I., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M., Cardoso, J. “INbreast: toward a fullfield digital mammographic database.” *Acad Radiol,* 19(2), pp. 236–248, 2012.

15. Armato, S. G., 3rd, McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D. R., Henschke, C. I., Hoffman, E. A., Kazerooni, E. A., MacMahon, H., Van Beeke, E. J., Yankelevitz, D., Biancardi, A. M., Bland, P. H., Brown, M. S., Engelmann, R. M., Laderach, G. E., Max, D., … Croft, B. Y., “The Lung Image Database

Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans." *Medical physics*, Vol. 38(2), pp. 915–931, 2011.

16. WHO Coronavirus Disease (COVID-19) Dashboard [online], (2022, June 16), World Health Organization: https://covid19.who.int/

17. Borakati, A., Perera, A., Johnson, J., Sood, T., "Diagnostic accuracy of X-ray versus CT in COVID-19: a propensity-matched database study", *BMJ Open*, Vol. 10(11), 2020.

18. Hossein, H., Ali, K. M., Hosseini, M., Sarveazad, A., Safari, S., & Yousefifard, M., "Value of chest computed tomography scan in diagnosis of COVID-19; a systematic review and meta-analysis", *Clinical and translational imaging*, Vol. 8(6), pp. 469-481, 2020.

19. Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., … Lungren, M. P., "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists." *PLoS Medicine*, Vol. 15(11), 2018.

20. Heung-Il Suk, "An Introduction to Neural Networks and Deep Learning", in S. K. Zhou, H. Greenspan, D. Shen (eds.), *Deep Learning for Medical Image Analysis*, Pages 3-24, Academic Press, 2017.

21. Sandfort, V., Yan, K., Pickhardt, P.J., et al., "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks", *Scientific reports*, Vol. *9*(1), pp. 1-9, 2019.

22. Han, C., Murao, K., Noguchi, T., Kawata, Y., Uchiyama, F., Rundo, L., ... & Satoh, S. I., "Learning more with less: conditional PGGAN-based data augmentation for brain metastases detection using highly-rough annotation

on MR images", *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management,* pp. 119-127, 3 November 2019.

23. Bolliger, S. A., Oesterhelweg, L., Spendlove, D., Ross, S., Thali, M. J., "Is differentiation of frequently encountered foreign bodies in corpses possible by Hounsfield density measurement?", *Journal of forensic sciences*, Vol. 54(5), pp. 1119–1122, 2009.

24. Bozinovski, S., "Reminder of the First Paper on Transfer Learning in Neural Networks", 1976. *Informatica*, Vol. 44, 2020.

25. He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

26. Chen, S., Ma, K., & Zheng, Y., "Med3D: Transfer Learning for 3D Medical Image Analysis", *arXiv preprint*, arXiv:1904.00625, 2019.

27. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., "Optuna: A Next-generation Hyperparameter Optimization Framework", *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

28. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I., "Tune: A Research Platform for Distributed Model Selection and Training", *arXiv preprint*, ArXiv:1807.05118, 2018.

29. Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., ... Yang, Y., "Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images", *IEEE/ACM transactions on computational biology and bioinformatics*, Vol. 18(6), pp. 2775-2780, 2021.

30. Zhang, K., Liu, X., Shen, J., Li, Z., Sang, Y., Wu, X., ... Wang, G., "Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of COVID-19 pneumonia using computed tomography", *Cell*, Vol. 181(6), pp. 1423-1433, 2020.

31. Wu, Y. H., Gao, S. H., Mei, J., Xu, J., Fan, D. P., Zhang, R. G., Cheng, M. M., "Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation", *IEEE Transactions on Image Processing*, Vol. 30, pp. 3113-3126, 2021.

32. Yang, X., He, X., Zhao, J., Zhang, Y., Zhang, S., Xie, P., "COVID-CT-dataset: a CT scan dataset about COVID-19", *arXiv preprint,* arXiv:2003.13865, 2020.

33. Rahimzadeh, M., Attar, A., & Sakhaei, S. M., "A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset", *Biomedical Signal Processing and Control*, Vol. 68, pp. 10258, 2021.

34. Morozov, S. P., Andreychenko, A. E., Pavlov, N. A., Vladzymyrskyy, A. V., Ledikhova, N. V., Gombolevskiy, V. A., ... & Chernina, V. Y., "Mosmeddata: Chest ct scans with covid-19 related findings dataset", *arXiv preprint,* arXiv:2005.06465, 2020.

35. Vayá, M. D. L. I., Saborit, J. M., Montell, J. A., Pertusa, A., Bustos, A., Cazorla, M., ... & Salinas, J. M., "Bimcv covid-19+: a large annotated dataset of rx and ct images from covid-19 patients", *arXiv preprint,* arXiv:2006.01174, 2020.

36. Yan, Q., Wang, B., Gong, D., Luo, C., Zhao, W., Shen, J., ... & You, Z., "COVID-19 chest CT image segmentation--A deep convolutional neural network solution", *arXiv preprint*, arXiv:2004.10987, 2020.

37. Wang, B., Jin, S., Yan, Q., Xu, H., Luo, C., Wei, L., ... & Dong, J., "AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system", *Applied Soft Computing*, Vol. 98, 106897, 2021.

38. Afshar, P., Heidarian, S., Enshaei, N., Naderkhani, F., Rafiee, M. J., Oikonomou, A., ... & Mohammadi, A., "COVID-CT-MD, COVID-19 computed tomography scan dataset applicable in machine learning and deep learning", *Scientific Data*, Vol. 8(1), pp. 1-8, 2021.

39. Ning, W., Lei, S., Yang, J., Cao, Y., Jiang, P., Yang, Q., ... & Wang, Z., "Open resource of clinical data from patients with pneumonia for the prediction of COVID-19 outcomes via deep learning", *Nature biomedical engineering,* Vol. 4(12), pp. 1197-1207, 2020.

40. Yan, T., Wong, P. K., Ren, H., Wang, H., Wang, J., & Li, Y., "Automatic distinction between COVID-19 and common pneumonia using multi-scale convolutional neural network on chest CT scans", *Chaos, Solitons & Fractals*, Vol. 140, 110153, 2020.

41. Tsai, E. B., Simpson, S., Lungren, M. P., Hershman, M., Roshkovan, L., Colak, E., ... & Wu, C. C., "The RSNA international COVID-19 open radiology database (RICORD)" *Radiology*, Vol. 299(1), E204-E213, 2021.

42. Gunraj, H., Sabri, A., Koff, D., Wong, A., "COVID-Net CT-2: Enhanced deep neural networks for detection of COVID-19 from Chest CT images through bigger, more diverse learning", *arXiv preprint,* arXiv:2101.07433, 2021.

43. Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Hsu, E. B., Yang, S., Eklund, P., "Artificial intelligence in the battle against coronavirus (COVID-19): a survey and future research directions", *arXiv preprint,* arXiv:2008.07343, 2020.

44. Yang, X., He, X., Zhao, J., Zhang, Y., Zhang, S., Xie, P., "COVID-CT-dataset: a CT scan dataset about COVID-19", *arXiv preprint,* arXiv:2003.13865, 2020.

45. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q., "Densely connected convolutional networks", *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 4700-4708, 2017.

46. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L., "Imagenet: A large-scale hierarchical image database", *In 2009 IEEE conference on computer vision and pattern recognition,* pp. 248-255, June 2009, IEEE, 2009.

47. Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., Kaur, M., "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning", *Journal of Biomolecular Structure and Dynamics*, Vol. 39(15), pp. 5682-5689, 2021.

48. Yang, S., Jiang, L., Cao, Z., Wang, L., Cao, J., Feng, R., ... Shan, F., "Deep learning for detecting corona virus disease 2019 (COVID-19) on high-resolution computed tomography: a pilot study", *Annals of translational medicine*, Vol. 8(7), 2020.

49. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., ... Xu, B., "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)", *European radiology*, Vol. 31(8), pp. 6096-6104, 2021.

50. Szegedy C, Vanhoucke V, Ioffe S, Shelns J, Wojna Z., “Rethinking the inception architecture for computer vision”, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), pp 2818–2826, 2016.

51. Bai, H. X., Wang, R., Xiong, Z., Hsieh, B., Chang, K., Halsey, K., ... Liao, W. H., “Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT”, *Radiology*, Vol. 296(3), E156-E165, 2020.

52. Tan, M., & Le, Q., “Efficientnet: Rethinking model scaling for convolutional neural networks”, *In International conference on machine learning,* pp. 6105-6114, May 2019, PMLR, 2019.

53. Pathak, Y., Shukla, P. K., Tiwari, A., Stalin, S., & Singh, S., “Deep transfer learning based classification model for COVID-19 disease”, *Irbm*, 2020.

54. Serte, S., & Demirel, H., “Deep learning for diagnosis of COVID-19 using 3D CT scans”, *Computers in biology and medicine*, Vol. 132, 104306, 2021.

55. Mishra, A. K., Das, S. K., Roy, P., & Bandyopadhyay, S., “Identifying COVID19 from chest CT images: a deep convolutional neural networks based approach”, *Journal of Healthcare Engineering*, 2020.

56. Simonyan, K., & Zisserman, A., “Very deep convolutional networks for large-scale image recognition”. *arXiv preprint*, arXiv:1409.1556, 2014.

57. Rahimzadeh, M., Attar, A., & Sakhaei, S. M., “A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct

scan dataset", *Biomedical Signal Processing and Control*, Vol. 68, 102588, 2021.

58. Goel, T., Murugan, R., Mirjalili, S., & Chakrabartty, D. K., "Automatic screening of covid-19 using an optimized generative adversarial network", *Cognitive computation,* pp. 1-16, 2021.

59. Wu, X., Hui, H., Niu, M., Li, L., Wang, L., He, B., ... & Zha, Y., "Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: a multicentre study", *European Journal of Radiology,* Vol. 128, 109041, 2020.

60. He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., & Xie, P., "Sample-efficient deep learning for COVID-19 diagnosis based on CT scans", *Medrxiv*, 2020.

61. Chen, X., Yao, L., Zhou, T., Dong, J., & Zhang, Y., "Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images", *Pattern recognition*, Vol. 113, 107826, 2021.

62. Ning, W., Lei, S., Yang, J., Cao, Y., Jiang, P., Yang, Q., ... & Wang, Z., "Open resource of clinical data from patients with pneumonia for the prediction of COVID-19 outcomes via deep learning", *Nature biomedical engineering*, Vol. 4(12), pp. 1197-1207, 2020.

63. Singh, D., Kumar, V., & Kaur, M., "Densely connected convolutional networks-based COVID-19 screening model", *Applied Intelligence*, Vol. 51(5), pp. 3044-3051, 2021.

64. Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., ... & Li, L., "A deep learning system to screen novel coronavirus disease 2019 pneumonia", *Engineering*, Vol. 6(10), pp. 1122-1129, 2020.

65. Wang, J., Bao, Y., Wen, Y., Lu, H., Luo, H., Xiang, Y., ... & Qian, D., "Prior-attention residual learning for more discriminative COVID-19 screening in CT images", *IEEE Transactions on Medical Imaging*, Vol. 39(8), pp. 2572-2583, 2020.

66. Polsinelli, M., Cinque, L., & Placidi, G., "A light CNN for detecting COVID-19 from CT scans of the chest", *Pattern recognition letters*, Vol. 140, pp. 95-100, 2020.

67. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size", *arXiv preprint*, arXiv:1602.07360, 2016.

68. Ouyang, X., Huo, J., Xia, L., Shan, F., Liu, J., Mo, Z., ... & Shen, D., "Dual-sampling attention network for diagnosis of COVID-19 from community acquired pneumonia", *IEEE Transactions on Medical Imaging*, Vol. 39(8), pp. 2595-2605, 2020.

69. Yan, T., Wong, P. K., Ren, H., Wang, H., Wang, J., & Li, Y., "Automatic distinction between COVID-19 and common pneumonia using multi-scale convolutional neural network on chest CT scans", *Chaos, Solitons & Fractals*, Vol. 140, 110153, 2020.

70. Hassan, H., Ren, Z., Zhao, H., Huang, S., Li, D., Xiang, S., ... & Huang, B., "Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks", *Computers in biology and medicine*, 2021.

71. Wasilewski, P. G., Mruk, B., Mazur, S., Półtorak-Szymczak, G., Sklinda, K., & Walecki, J., "COVID-19 severity scoring systems in radiological

imaging - a review:, *Polish journal of radiology,* Vol. 85, e361–e368, 2020.

72. Adams, H. J., Kwee, T. C., Yakar, D., Hope, M. D., & Kwee, R. M., "Chest CT imaging signature of coronavirus disease 2019 infection: in pursuit of the scientific evidence", *Chest*, Vol. *158*(5), pp. 1885-1895, 2020.

73. Kwee, T. & Kwee, R., "Chest CT in COVID-19: What the Radiologist Needs to Know", *RadioGraphics*, Vol. 40, pp. 1848-1865, 2020.

74. Wang, Y., Dong, C., Hu, Y., Li, C., Ren, Q., Zhang, X., Shi, H., & Zhou, M., "Temporal Changes of CT Findings in 90 Patients with COVID-19 Pneumonia: A Longitudinal Study", *Radiology,* Vol. 296:2, E55-E64, 2020.

75. Yang R, Li X, Liu H, et al., "Chest CT severity score: an imaging tool for assessing severe COVID-19". *Radiology*, 2020.

76. Kunwei Li, Yijie Fang, et al., "CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19)", *Eur Radiol*, 2020.

77. Yongxing Yun, Ying Wang, Yuantao Hao, Lin Xu, Qingxian Cai, "The time course of chest CT lung changes in COVID-19 patients from onset to discharge". *European Journal of Radiology Open*, Vol. 8, 100305, 2021.

78. Guan, C. S., Wei, L. G., Xie, R. M., Lv, Z. B., Yan, S., Zhang, Z. X., & Chen, B. D. , "CT findings of COVID-19 in follow-up: comparison between progression and recovery", *Diagnostic and interventional radiology*, Vol. 26(4), pp. 301–307, 2020.

79. Dance, D. R., “Diagnostic radiology physics: a handbook for teachers and students”, *International Atomic Energy Agency,* 2014.

80. Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L., “Normalization Techniques in Training DNNs: Methodology, Analysis, and Application”, *arXiv preprint* arXiv:2009.12836, 2020.

81. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N., “An image is worth 16x16 words: Transformers for image recognition at scale”, *arXiv preprint*, arXiv:2010.11929, 2020.

82. D’Ascoli, S., Touvron, H., Leavitt, M. L., Morcos, A. S., Biroli, G., & Sagun, L., “Convit: Improving vision transformers with soft convolutional inductive biases”, *In International Conference on Machine Learning,* July 2021, pp. 2286-2296, PMLR, 2021.

83. Cordonnier, J. B., Loukas, A., & Jaggi, M., “On the relationship between self-attention and convolutional layers”, *arXiv preprint*, arXiv:1911.03584, 2019.