REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

DETECTION OF LUNG DISEASES FROM RADIOGRAPHY IMAGES

17011038 — Onur PORSUK
17011080 — Engin Deniz ÇAĞLAR

SENIOR PROJECT

Advisor
Dr. Ahmet ELBİR

January, 2022

ACKNOWLEDGEMENTS

It is amazing to have done this research and development work in senior project in Yıldız Technical University and Computer Engineering department. This was an opportunity to practice what we have learned in Image Processing and experience further. We would like to sincerely appreciate consideration and guidance of our advisor lecturer Dr. Ahmet Elbir.

This study allowed us to have more experience in traditional and modern ways of image processing. Classification in medical field is quite important for human health and by this project, we achieved the goal to classify the images in the dataset used. During the process, we carried out research and implementation on the system that we created to be able to detect lung diseases. We would like to express our gratitude to our advisor lecturer for always being with us in all the problems we encountered during the project, informing us about the possible reasons of the problems and suggesting solutions.

We would also like to thank all other professors who provided this project course and evaluated its output.

Onur PORSUK
Engin Deniz ÇAĞLAR

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ResNet	Residual Neural Network
VGG	Visual Geometry Group
MLP	Multi Layer Perceptron
DNN	Deep Neural Network
PNG	Portable Network Graphics
ReLU	Rectified Linear Unit
RGB	Red Green Blue
SGD	Stochastic Gradient Descent
GLCM	Gray Level Co-Occurrence Matrix
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
XGBoost	Extreme Gradient Boosting
GPU	Graphics Processing Unit
CPU	Central Procesing Unit

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

DETECTION OF LUNG DISEASES FROM RADIOGRAPHY IMAGES

Onur PORSUK
Engin Deniz ÇAĞLAR

Department of Computer Engineering
Senior Project

Advisor: Dr. Ahmet ELBİR

In this time, users create and share huge amount of image data on the internet either for entertainment or serious purposes, for instance for medicine. Technology and its advantages are used in medicine field for many years and by these advances, detection and treatment of illnesses are significantly developed. In this study, by using classical image processing and deep learning methods to extract features from medical images, it is aimed to classify them with different algorithms including machine learning models.

Before running all the processes, it is vital to preprocess the dataset. By applying some steps such as changing type, normalizing numerical values, reshaping for the proper data structure and encoding of label information, the system is ready to operate on the dataset.

To obtain better success of the system, machine and deep learning are used as hybrid models. After doing the relevant operations on classical processing and deep learning methods, the features that are extracted by both classical ways and CNN models are used to be classified. In this part, CNN's softmax, machine learning algorithms SVM and XGBoost are implemented to perform the task. Analysis of classical ways, modern ways and hybrid models of deep and machine learning are done to indicate the results.

Keywords: Image processing, preprocessing, convolutional neural network, machine learning algorithms, svm, xgboost, data classification

ÖZET

RADYOGRAFİ GÖRÜNTÜLERİNDEN AKCİĞER HASTALIKLARI TESPİTİ

Onur PORSUK
Engin Deniz ÇAĞLAR

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Dr. Ahmet ELBİR

Günümüzde kullanıcılar ister eğlence, isterse tıp ve mühendislik alanlarında kullanılmak üzere internette çok büyük miktarda görüntü verisi oluşturur ve paylaşır. Bu verilerin kullanımı ise belirli alanlarda oldukça büyük öneme sahiptir. Teknoloji tıp alanında uzun yıllardır kullanılmaktadır ve bu gelişmeler sayesinde hastalıkların tespiti ve tedavisi önemli ölçüde gelişmiştir. Bu çalışmada, tıbbi görüntülerden özellik çıkarmak için klasik görüntü işleme ve derin öğrenme yöntemleri kullanılarak, makine öğrenmesi modelleri de dahil olmak üzere farklı algoritmalarla görüntülerin sınıflandırılması amaçlanmıştır.

Sistem ana adımlarından önce veri setini ön işlemek büyük öneme sahiptir. Veri tipi değiştirme, sayısal değerlere normalizasyon uygulama, uygun veri yapısı için yeniden şekillendirme ve sınıf (etiket) bilgilerini sayısal hale getirme gibi adımlar uygulanarak veri seti kullanılmaya hazır hale getirilir.

Klasik görüntü işleme ve derin öğrenme yöntemleri ile, ilgili işlemler yapıldıktan sonra çıkarılan özellikler sınıflandırılmaktadır. Bu kısımda, CNN'nin softmax algoritması, makine öğrenmesi algoritmalarından SVM ve XGBoost uygulanmaktadır. Tüm yöntemlerin başarısı analiz edilerek en yüksek başarıyı gösteren model çalışma sonunda belirtilmiştir.

Anahtar Kelimeler: Görüntü işleme, ön işleme, evrişimsel sinir ağları, makine öğrenmesi algoritmaları, svm, xgboost, veri sınıflandırma

1 Introduction

In this chapter there will be an overview of classification in computer vision and some important concepts.

1.1 What is Image Classification?

Classifying is among the most popular studies in artificial intelligence world. Image data is one of most interesting and challenging types to work on. The aim is making machine be able to distinguish some objects than others. The objects could literally be anything that is interesting to human to work on. By having or extracting some features of objects or other things (in general term images), classifier models ought to separate different data with different class labels. Each data sample has got its own features and those describe an object to be in a class. This means, experts need to be careful while providing inputs to models and they must be looking for right parameters or fine-tuning settings to allow the model to perform successfully.

1.2 What is Deep Learning?

Day by day, new methods and developments happen to enhance machines to work better. Deep Learning concept has arrived years ago and since that time, there have been dramatic change in Machine Learning field. Deep Learning is a subfield of machine learning. It is created with complex algorithms inspired by the structure and function of human brain. That inspiration is neural network, in this case artificial neural networks. Today, we use this technology in lots of fields, such as self driving cars, visual recognition, credit card fraud detection, natural language processing...

1.3 Image Classification Using Deep Learning - CNN

Users who use Google may have wondered ever how Google Photos works that successfully. Certainly, there are many efficient algorithms in the background. All these complex methods created the state-of-the-art image classification model.

Among the popular methods, Convolutional Neural Networks (ConvNets or CNNs) have been the master algorithm in computer vision in recent years. Those created models achieved incredible performance on complex visual tasks with the increase in computational power and efficiency. That is why today some models perform better than some experts. A CNN model takes an input image or videos, process and classify it by trying to determine the input's class. Before providing input, the data must be converted to the proper format. For instance, if the input is a picture data, it is needed to export it in matrix format, the digital values of pixels in image. Technical details of CNNs will be covered in the next chapters.

2
Preview

In this part, the preview of this study will take place.

2.1 Necessity and Requirements of the Project

This project aims to create an image classifier for a specific data which is about lung diseases. By using advantages of traditional image processing and deep learning methods to process medical images, the system is created to perform well to detect some lung diseases including the worldwide problem Covid-19. With experience of image classification on this dataset, also some comparisons between methods that are going to be done will add a leading result to be understood by people who work on this field. This project requires a single computer, relevant development environments, programming tools, additional software, and libraries to be created.

2.2 Implementation Time and Budget

Many factors play important role in determination of the implementation time of the system. For this project, approximately 4-5 weeks is sufficient. In terms of budget, there is no need for a budget, the system will be available to use by installing the necessary software with a computer.

2.3 Benefits and Challenges

In this study, there are both uncountable number of benefits and challenges since this application area of Computer Vision and Deep Learning is dramatically large and crucial for today's world. From industry to e-commerce field, classification models with deep learning have performed superhuman tasks in order to make human life easier. Of course, with advantages of this field, there are so many difficulties that experts try to eliminate or solve completely. The technical part of these problems regarding to CNNs will be discussed later.

2.4 Project Implementation Decision

Considering all the requirements, advantages, and disadvantages, it was decided to do this project and present it to the Computer Vision - Image Classification field to contribute people with the experience of Deep Learning - CNN on the dataset of lung diseases.

3
Project Feasibility

In this chapter the project's studies of feasibility are covered.

3.1 Technical Feasibility

Technical feasibility is handled in two different groups as software and hardware feasibility.

3.1.1 Software Feasibility

In project's feasibility in terms of technical side, the requirement of the system in software feasibility is a strong and high-level programming language such as Python, development environment where it can be coded (for example PyCharm, Anaconda or others), libraries for the algorithms to be used, some additional software. For this project there are plenty reasons to pick Python as programming language:

- Easy to use
- Community support
- Advanced libraries for image processing, and neural networks
- Abundant online resources

3.1.2 Hardware Feasibility

There is no need for an extra hardware resource while developing the project. Personal computers can be used. The system must run on a computer with updated drivers. For hardware feasibility, there is no need for a specific hardware component with a specific feature. These are the specifications of our computers:

	Computer 1	Computer 2
RAM	16 GB	8 GB
Storage	256 GB SSD	256 GB SSD
CPU	i7-7700HQ	i5-7300HQ
GPU	GeForce GTX 1050 Ti	GeForce GTX 1050 Ti
Operating System	Windows 10 Pro 64 Bit	Windows 10 Home 64 Bit

Table 3.1 Hardware Specifications

3.2 Workforce and Time Feasibility

The diagram of the project, which is planned to be completed in 13 weeks, is shown in Figure 3.1

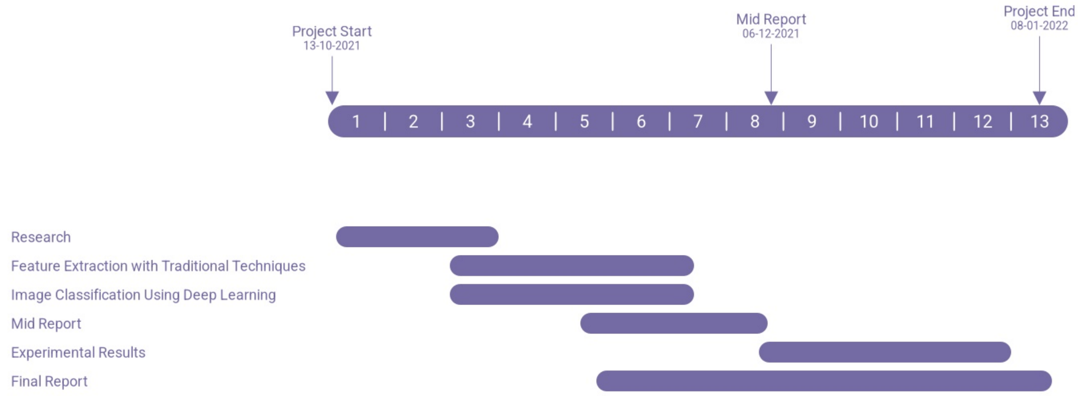

Figure 3.1 Gantt Diagram

3.3 Legal and Economic Feasibility

Since only open-source libraries and datasets were used, no extra licence is needed. Some libraries and their licences were listed as:

- numpy library BSD 3-Clause License.
- pandas library BSD 3-Clause License.
- scikit-learn BSD 3-Clause License.
- tensorflow Apache Licence Version 2.0

4
System Analysis

In this part, system analysis of the study will take place.

4.1 Requirements and Work Analysis

The problem identified in the project is the detection of lung diseases by processing medical images with both traditional and deep learning methods and with these methods it is aimed to create a program which aims to achieve this. The basic functions are extracting features and classifying samples after preprocessing. The features which are the best descriptors to distinguish between images play important role of success of system.

Different parameters and methods are used to analyze the system and some articles are searched via Google search engine to understand different models with their configuration on relevant datasets. The expected result in the output of this program is being able to classify images with a success rate which depends on various conditions, in computer term parameters. In addition to that, for both two methods of this implementation, the program can be developed easily with better configurations. Future aims of this program include implementation of new and more sophisticated algorithms or transfer learning-based models for specifically medical image data. b

5
System Design

In this chapter, the design of the program, the methods that are used and some technical details will be covered.

5.1 Software Design

On this project, it is aimed to implement a complete program to apply traditional and modern ways of image processing. In traditional way, data preprocessing, conversion to grayscale, edge detection, corner detection and other algorithms are used to extract features to be able to classify input images with some machine learning algorithms. While traditional way keeps its success at a rate, image classification with deep learning has been created the state-of-the-art methods and algorithms (AlexNet, ResNet, VGG16, Inception). Especially CNNs are the most powerful artificial neural networks that can handle with huge image dataset and can categorize that data according to features that CNN extracted. DNNs, especially CNNs, are widely applied in changing image classification problems and they have achieved significant performance since 2012. As it is said here [1], some research on medical image classification by CNN has achieved performances rivaling human experts.

In Python environment, there are uncountable number of useful packets, libraries to work on image processing. In implementation side, for traditional image processing, OpenCV; for deep learning TensorFlow and Keras are used to achieve the desired goal. All these powerful framework and libraries make the computer vision life easier so that developers can perform great tasks.

Once the development environment is set, after reading the dataset, first the image dataset should be preprocessed, which is super important to have better result in both machine and deep learning concepts. Experiments show that preprocessing plays crucial role in data science applications.

To provide some fundamentals, the important part that human concentrates on an image is known region of interest (ROI), and models are built to work on this field of images. Image means sequence of numbers in a range of 0-255 values in computer side, each value is called pixels and those pixels may be in different color spaces, one channel (dimension) for grayscale and 3 channels for RGB space colour. This study uses medical image dataset which is in grayscale. After reading the dataset and creating relevant data structures of images, next step can be processed. To prepare the data for main operations, preprocessing steps such as class label encoding, splitting dataset into training, validation and test data, data visualizing, number unit type conversion, normalization, reshaping and one-hot encoding are performed. Each step has got its own purpose to make the data more suitable for classification.

With the help of traditional image processing techniques, various types of features like color, shape and texture are extracted.

In digital image processing, one of the features which is used is color information that caused due to reflection of light. Several characteristics of color is used in this project. Color moments are used to obtain the color distribution of an image by using mean, standard deviation, and kurtosis values. Color histograms show how colors are distributed in a given range. Regardless of the locations of colors, color histograms point out how frequent a color in that image. By observing spatial distribution of histogram values, brightness and contrast of the image can be obtained. If the histogram values are concentrated on the right side specifically, the image may be called bright. With the help of histograms, threshold values are obtained. Threshold values can be used for various purposes such as edge detection and converting image to binary format.

Texture analysis is used in many different applications as remote sensing, automated inspection, and medical image processing. Texture is a repeating pattern of local variations in image intensity, and it can not be defined for a point. It is defined for a group of pixels. In texture analysis different techniques are used. GLCM characterizes the texture of an image by calculating how often pairs of pixels with specific values and in a specified spatial relationship occur in an image. It is created by gray scale image. After creating the GLCM, several statistics can be derived from matrix.

Shape is a very powerful feature, and objects may be recognised from their outline. Extracted features must not be affected from location, rotation, and scaling changes. Shape features depend on contour-based methods and region-based methods. Mostly shape boundary points are used. Boundary representation describes the closed curve surrounding the shape. Mostly, sharp color transitions are used to determine the object boundaries on image. The steps are showed in Figure 5.1 [2].

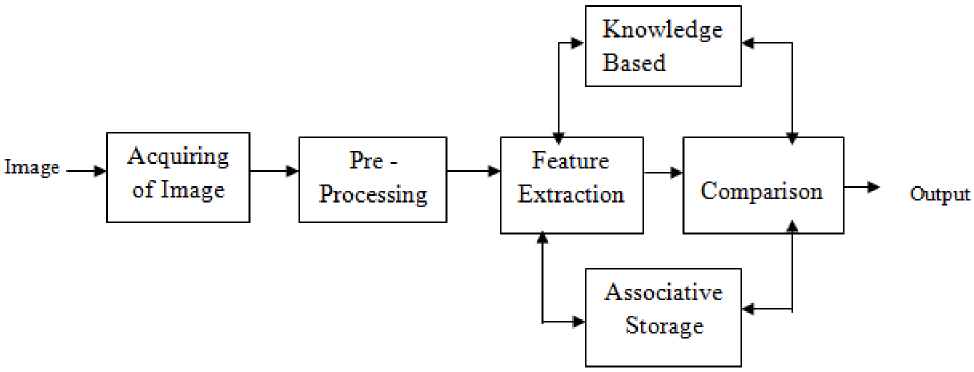

Figure 5.1 Traditional Digital Image Processing

For many years, classical way of digital image processing has been performed well. But with deep learning, computer vision studies' success reached an amazing level. On this project's second main part, CNN is built and used to perform better. A CNN consists of architecture where it has layers such as convolution, subsampling and fully connected and it mainly processes the image to extract features before the fully connected layer. As seen in Figure 5.2 [3], CNN performs some mathematical operations by using masks or kernels which are matrices with some integer values in convolution layers and then it may use subsampling methods such as max or average pooling. Each result of these layers is feature maps. After these operations, a fully connected layer takes part for classification and produces output, it is also called the last dense layer [4].

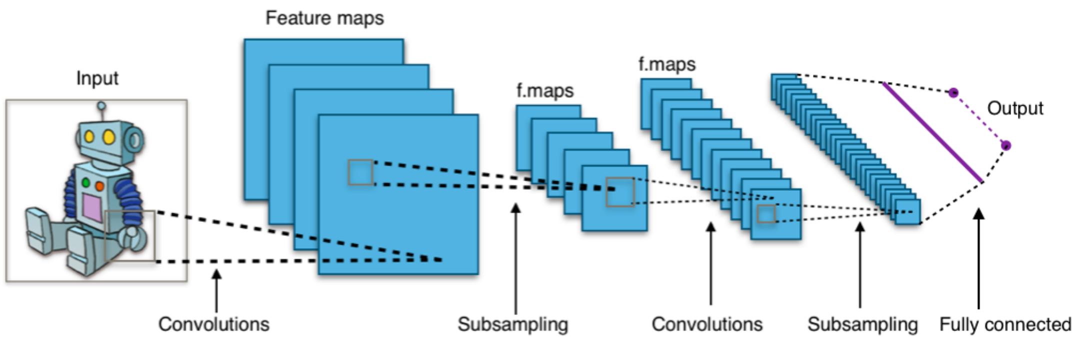

Figure 5.2 Convolutional Neural Network

In CNN, a non-trained is called an artificial neural network and a trained is called model, these are sometimes misused. CNN is driven by convolution operations with back propagation algorithm (MLP is feed-forward). This method is quite efficient and based on minimization of loss, which also means error. To be successful in relevant problem, dataset, and CNN model, it is significant how many layers are used, which techniques are preferred, how is it tried to reduce common problems such as overfitting (over-learning) or underfitting (incomplete learning) and memorization. It should be carefully thought and analysed to create the model according to the dataset and determine the parameters correctly. A parameter has the same meaning of weight which is between neurons in network, with back propagation algorithm, in each step, CNN tries to find the right parameters (weight values).

5.2 Dataset

For image data for classification of this study, a dataset that is called COVID-19 Radiography image dataset was used. Originally and as updated, the distribution of the dataset was shown in Table 5.1 with size of 299 of width and height. 4 classes of lung diseases including the Covid-19 virus that caused a global pandemic take place. Details of dataset and usage will be discussed in the following chapters.

	COVID	Lung Opacity	Normal	Viral Pneumonia	Total
Sample Number	3616	6012	10192	1345	21165

Table 5.1 Dataset Distribution

5.3 Input and Output Design

Input of the program is medical images in PNG format and the output is the log information and values of training process, prediction values on test data set for output classes, success metrics and relevant plots of both traditional and deep learning methods that are discussed. In the next chapter, implementation details will become involved.

6
Application

In this chapter, implementation of the program, methods and development will be included in detail.

6.1 Implementation

The aimed program to achieve the goal consists of four main parts. The first one is using classical digital image processing techniques and classifying the images with machine learning algorithms by using the features that traditional methods extracted. In the second step, CNN models will be built and used to extract features but instead of fully connected layer, the features will be used in classification with machine learning. Then this time CNN models' features will be used to categorize images with fully connected layer and in the last step, both traditional ways' and CNNs' features will be used by appending relevant feature data structure and they will be given to machine learning algorithms as inputs to be classified. Step by step, each process's requirements and implementation details are analyzed and brought into code.

6.1.1 Traditional Digital Image Processing

In the traditional digital image processing part, preprocessing and feature extraction techniques are used.

There were a lot of pixels which brings no information in images. After blurring and detecting the significant parts with thresholding, images are cropped. Some of the images were tagged with numbers, letters, or region names. To get rid of these tags, images are divided into regions according to the contours. Largest contour is the region of interest and rest of the contours are masked. After preprocessing operations, images are resized.

Then color characteristics are analyzed. Color moments are extracted from all pixels in images using the mean, kurtosis and standard deviation values. After applying Otsu thresholding to images, color histograms are generated from the region of interests. Since the images are gray-level, 0-255 range is used in histograms. To find the edges in the images, Canny Edge Detection method is used. Canny Edge Detection is the most used edge detector in computer vision tasks. To detect the edges it uses a multi step approach including smoothing, gradient magnitude and non-maxima suppression. Minimum and maximum threshold values are set to 5 and 25 respectively. The SIFT is used to detect and compute local features in image. If scale changes, a corner may become an edge in Harris Corner Detection. Against this, SIFT is scale invariant. SIFT returns a descriptor for each key point. Values obtained using statistical methods on the descriptors are used as features. Gray Level Co-Occurrence Matrix are used for extracting the texture features from specific parts of images. Lastly, histogram of oriented gradients method is used. To find the gradient and orientations, images are resized to 2:1 aspect. High number of cell size is chosen to prevent overfilling the feature vector with HOG features.

After extracting features from various algorithms, a feature vector is created for each image. These features are given to 7 different machine learning algorithms. Parameter optimization is used to achieve the maximum accuracy score in each algorithm. As a result, accuracy, precision, recall and Cohen's Kappa scores of algorithms are compared.

6.1.2 Convolutional Neural Networks

After the implementation of traditional ways, ConvNets are the next part to build. As it is discussed before, a CNN has got an architecture (layers), activation functions, optimizer, and some parameters in relevant functions. To show the effect of different architecture and parameters of CNNs to success, three different CNN models are created. First, with relatively small architecture and without Dropout and Batch Normalization. Dropout layer is used to prevent or at least reduce overfitting level of a model, and when the model fits too well to the training set, overfitting happens. In a general way of explanation, dropout takes part by disabling some neurons to prevent memorizing. Also, Batch normalization is included to overcome overfitting problem.

The second is relatively larger model which consists of more layers. In opposition to small model, this one has been built with dropout and batch normalization methods. Also, Leaky Relu activation function which is more balanced than Relu is added and it is among the most popular methods for deep neural networks and has proven to be more effective than the frequently used Sigmoid function [5].

The last CNN model is created with transfer learning which is a magnificent concept. It basically means a model developed for a task-dataset that was reused as the starting point for a model on a second task-dataset. This enables developers to have chance to give their data input to a pre-trained model which probably is trained with huge dataset. For instance, the state-of-the-arts methods such as AlexNet, VGG16, ResNet are built with large amount of data, and one of the most popular datasets is ImageNet which consists of about 1.2 million images for training, 50,000 for validation and 100,000 for testing, belonging to 1000 categories [6]. Transfer learning is told to be kind of brain surgery to the previous model for new problem. In this last CNN model, VGG16 is implemented.

In CNN's layers, an activation function is used which may be linear or non-linear and it aims to introduce non-linearity into the output of a neuron in CNNs, therefore it's crucial to use non-linear activation functions such as sigmoid, tanh, relu. In compiling step of models, an optimizer is passed as an argument to relevant function and its operation is trying to change the attributes of neural network such as weights and learning rate in order to reduce the losses, in other words errors. Optimizers change the learning rate adaptively and optimally to reach the min error, some of the popular ones are AdaGrad, AdaDelta, RMSprop, SGD and Adam which is also known replacement optimization algorithm for SGD [7] and in these created CNN models, Adam takes part. Figure 6.1 [8] shows the comparison of these optimizers on a dataset with MLP.

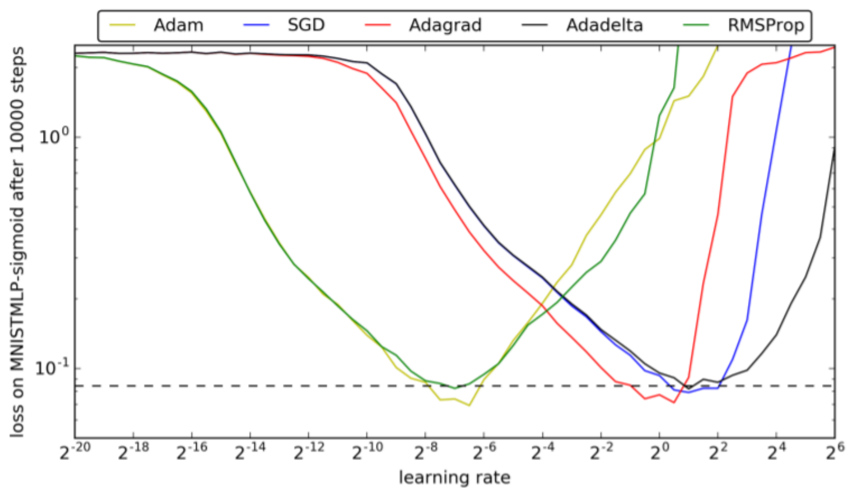

Figure 6.1 Optimizers' Learning Rate Comparison

7 Experimental Results

In this chapter, the evaluation of the outputs and the experimental results of the system will be included.

7.1 System Success

7.1.1 Traditional Methods' Results

As it was mentioned before, this project consists of 4 main parts. In the first part, some classical image processing methods were applied to extract features from images and the relevant features are passed through machine learning algorithms. In Table 7.1, the most commonly used success metrics of some powerful machine learning algorithms were included. According to the metric values, XGBoost method received the highest values of accuracy, F1-score and Cohen's Kappa.

M. Learning Models	Accuracy	Recall	Precision	F-Score	Cohen's Kappa
SVC Linear	0.799	0.768	0.825	0.793	0.685
SVC RBF	0.831	0.808	0.86	0.831	0.736
Decision Tree	0.719	0.694	0.698	0.696	0.659
XGB	0.878	0.87	0.899	0.884	0.812
Random Forest	0.845	0.823	0.875	0.846	0.759
Logistic Regression	0.803	0.785	0.819	0.8	0.694
Linear Discriminant Analysis	0.798	0.773	0.817	0.792	0.684
K-Nearest Neighbors	0.822	0.787	0.861	0.818	0.718

Table 7.1 Traditional Image Processing's Success

At first, the accuracy score of algorithms was around 0.62 with a few features from images. With the help of different methods and image processing techniques, more features were extracted and selected using forward selection. After these steps, using the selected features accuracy score is reached to 0.87. Cohen's Kappa score was analyzed to examine if the success of algorithms were random due to unbalanced dataset. Confusion matrices and Cohen's Kappa scores showed that the results were not random. Worst success rate was obtained with Decision Tree algorithm. It may have been caused by the values in the feature vector being unrelated to each other. Even though XGBoost is decision tree based, it achieved the best score as 0.87. The regularization and penalizing steps of XGBoost may have caused the difference of success with Decision Tree algorithm.

7.1.2 CNN - Fully Connected Layer Results

In the second part, 3 different CNN models which have small, large and transfer learning-based architectures were built to extract features. Then the features were given to CNN's classification method "softmax", which is the activation function of the last dense layer. Table 7.2 shows the success of CNN models with some parameters. As it can be seen, Large Model performed better than others due to its Dropout and Batch Normalization layers. Also, its execution time was much shorter and hardware resource consumption was much less than VGG16 model.

7.1.3 CNN - Machine Learning Results

Using machine learning algorithms instead of CNN's softmax took place in the third part. Since the fully connected layer classifier in the last step of CNN model does not perform that well when comparing with machine learning algorithms. Powerful algorithms SVM and XGBoost were implemented to classify images with CNN's extracted features. With this approach, CNN-SVM and CNN-XGBoost hybrid models were created to perform the task better. F1 scores of these hybrid models were showed in Table 7.3. SVM and XGBoost had quite similar success for this task but execution time of SVM was much shorter than XGBoost. Therefore, SVM is the winner of this comparison.

Instead of using the default parameters of SVM and XGBoost algorithms, some different values of parameters put to the test to have better result. To achieve this, a technique which is called Grid Search were used. For instance, C and gamma are learning factors in SVM method. These parameters are highly based on dataset that were used. It is not easy to find the best pair of c and gamma parameters at once. Therefore, Grid search helps to find the best pair for a particular dataset. Grid search takes all the given values of parameters, it runs the operations and indicates the best ones among the given values. This is quite useful for fine-tuning of classification method [9].

CNN Models	Image	Epochs	Batch Size	Val. and Test Size	Val. and Test Acc.	Duration
Small Model 2 Conv2D / MaxPooling 10 Layers	50*50	8	64	0.1	0.85 / 0.83	00:53 s
		12	128	0.2	0.87 / 0.87	00:59 s
	100*100	10	64	0.1	0.87 / 0.88	04:36 s
		15	128	0.2	0.87 / 0.87	05:17 s
Large Model 3 Conv2D / MaxPooling BatchNorm. / Dropout 21 Layers	50*50	12	32	0.1	0.88 / 0.87	02:15 s
		20	64	0.2	0.91 / 0.88	02:45 s
	100*100	12	32	0.1	0.92 / 0.91	05:33 s
		20	64	0.2	0.89 / 0.86	07:51 s
VGG16 Model 13 Conv2D / Dropout / 3 FCL 41 Layers	50*50	15	64	0.1	0.86 / 0.84	03:40 s
		25	128	0.2	0.84 / 0.83	04:45 s
	100*100	18	64	0.1	0.88 / 0.88	08:34 s
		18	128	0.2	0.90 / 0.89	08:22 s

Table 7.2 CNN Models' Success

	ML Method	Accuracy	Precision	Recall	F1-Score
Large CNN Model (0.91 F1-Score)	SVM C=0.1, kernel='linear', gamma=1	0.921	0.921	0.921	0.921
	XGBoost max_depth=9, min_child_weight=5, gamma=0.0	0.915	0.915	0.915	0.915

Table 7.3 CNN - Machine Learning Models' Success

A machine learning model is defined as a mathematical model with several parameters that need to be learned from the data. Apart from this, some parameters cannot be directly learned by the model, these are hyper parameters, and they are chosen by developers to achieve more successful prediction results. These parameters exhibit their importance by improving the performance of the model such as its complexity or its learning rate [10]. Finding the best combination of model's parameters can be seen as a search problem, where Grid search comes in. In this project's implementation, it took approximately 50 minutes to perform Grid search of SVM and 1 hour 52 minutes for XGBoost with the computer that was used. The reason of choosing SVM and XGBoost methods in machine learning is due to their pros. SVM works well in higher dimension and outlier values less affect in SVM. XGBoost does not need feature engineering that much and less prone to overfitting [11].

7.1.4 Traditional Methods - CNN - Machine Learning Results

In the last part of the study, the features that were extracted by classical methods and deep learning model CNN were combined (in the relevant data structure, they were appended column-wise) and were pass to machine learning algorithms. The aim of this approach was to experience the different methods' work and achieve to perform the task better. Success metrics of this part took place in Table 7.4. In this part, XGBoost algorithm worked slightly better than SVM.

	Method	Accuracy	Precision	Recall	F1-Score
Large CNN Model (0.89 F1-Score)	SVM	0.919	0.919	0.919	0.919
	XGBoost	0.920	0.920	0.920	0.920

Table 7.4 Combined Features Classification Success

All main steps of this study were completed. According to the experimental results of each step, combining features of CNN and classical digital image processing did not appear that successful, yet it was worth to experience and see how the machine learning models perform with different data. Also, adding some widely used methods to indicate if the success of models were random or not was a good effort. All efforts were done for the efficient and proper implementation of this study. Relevant functions, methods and libraries were. Depending purpose, each configuration had impact on the system's success. Therefore, analyzing and fine-tuning of algorithms took a dramatic part in this study. Some details will be covered in the next section.

7.2 Implementation Details

7.2.1 Parameter Optimization

There are lots of parameters in CNN models and each may have different effect on the success of models. By trying some different values of these parameters, it was aimed to have better result. First, image size of images in dataset is an important one. The best image size for CNN model highly depends on the dataset. In layers of CNN, there are two ways to choose the number of filters which extract features from inputs: down scaling way means bigger images will be down scaled. Number of pixels where the important feature is present is significantly decreased. Therefore, it becomes more difficult for model to learn these features. On the other hand, up scaling means small images are up scaled and padded with zero. Images with higher size are also slower to train and may require more memory [12]. The most proper way in choosing the right image size of dataset is picking a small number and progressively increasing it with analysis of the model's success and effort of computation by the machine.

Batch input size defines the number of samples that are used through the neural network each time. All the batch size number of inputs are given to the neural network at the same time. High batch size requires more memory. Epoch number which means how many times the model will work through the entire training dataset, and it directly affects the training time.

In CNN compile options, there is a loss function must be specified. In this study, as the aim is classifying images into four different classes, loss function was chosen as categorical cross entropy. For instance, if the output class number were two, then binary cross entropy and "sigmoid" activation function would be used instead of softmax. Categorical cross entropy must be used with one hot encoding technique. It basically converts each categorical value into a binary sequence where each integer value is represented as a binary vector. Only the relevant index is assigned with 1 and the rest are 0s. On the contrary, there is label encoding technique which represents categorical label names as integers. Label encoding uses less memory, but it may lose information. Hence one hot encoding is recommended when the close-enough predictions are important.

7.2.2 Overfitting

Overfitting is one of the most common problems of machine and deep learning algorithms. There may be other factors as well but basically overfitting happens when a model learns too much the detail and noise in the training data. Unlike overfitting, underfitting shows if the model is not complex enough to obtain relationships between features and label information. Figure 7.1 [13] shows the difference of underfitting, overfitting and robust algorithm.

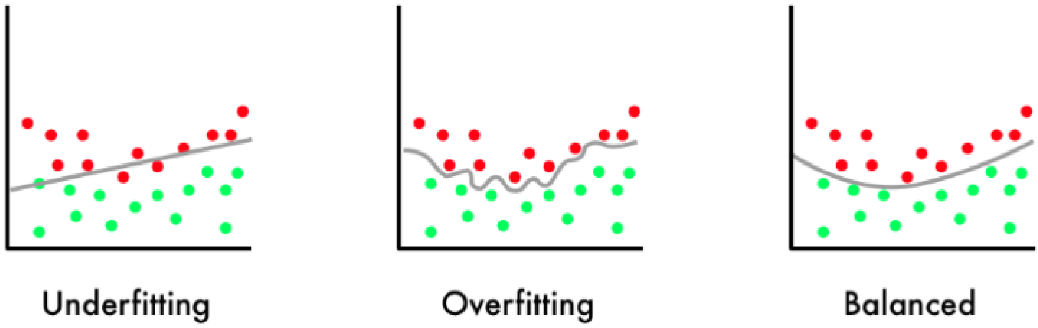

Figure 7.1 Underfitting, Robust and Overfitting Comparison

Many ways can be used to prevent this, such as regularization, simplifying model, feature selection, data augmentation and early stopping. Data augmentation is technique to artificially create new training data from the existing data and it performs this by doing some operations such as changing color, changing channel, flipping, and rotating the image, zooming, cropping horizontally or vertically etc. Early stopping is stopping the training of model when there is no more improvement, and the improvement criterion can be specified by the developer depending on the development environment. This feature is quite useful when dealing with huge training with high number of epochs. It would also reduce resource usage and it highly depends on dataset, model, and problem. There was imbalance of class samples in this study's dataset. Data augmentation was not used to create new artificial data and accuracy success metric with Cohen's Kappa coefficient were included.

7.2.3 VGG16

For the third CNN model which is transfer learning-based VGG16 model were used. It could not perform well as it was expected due to some differences of usage than normal. VGG16 is built with huge dataset ImageNet which is for visual object recognition research. VGG16 were trained with RGB images in size of 224 of width and height. But the images in this study, they have different size which would not be suitable for the model. Thus, an input size change was applied to the first layer of this transfer learning model. Also, the images are in gray scale and VGG16 requires RGB images. Changing the architecture of the model would cause performance lack since the weights of the model have been trained for a specific input configuration and changing all configuration may make the rest of the weights useless. To overcome this obstacle, the images in dataset are converted to RGB images from gray scale. This approach was more proper than changing the transfer learning model's architecture.

7.2.4 K-Fold Cross Validation

K-Fold cross validation is dataset splitting technique to train a classification model. It basically uses all train dataset for both training and validation. K stands for the number of groups which will be created by splitting train data. In each iteration, it separates one group while it trains on the remaining groups. In training part, it fits a classification model and evaluate it on the separated group. In next iterations, it runs the same operations by changing the groups sequentially and after the operations, it summarizes the performance scores of models. Figure 7.2 [14] describes the method.

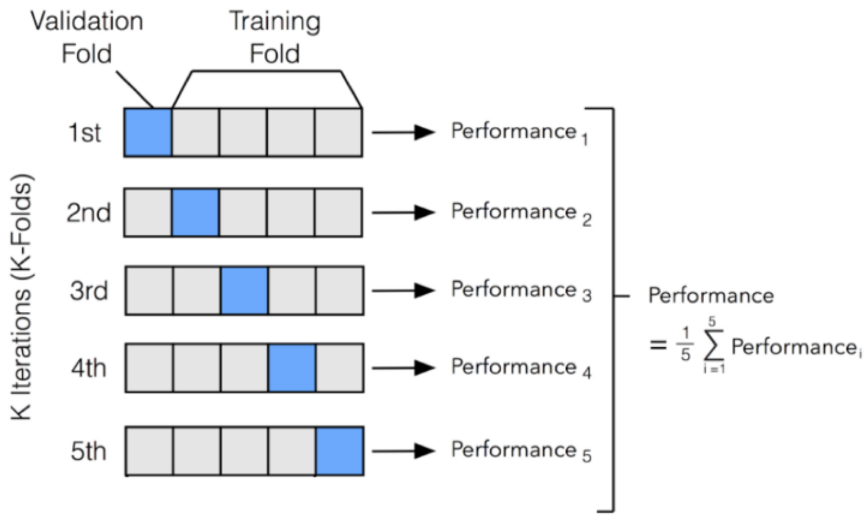

Figure 7.2 K-Fold Cross Validation

After the implementation of K-Fold cross validation technique, accuracy results of CNN Large Model for each fold were shown in Table 7.5. All iterations were run with 12 epochs on CNN Large model, also, validation split were included on CNN as 0.1 portion.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std. D.	Duration
Accuracy % (Batch size: 4)	84.72	86.35	87.72	89.75	83.42	86.38	2.22	75 mins
Accuracy % (Batch size: 16)	83.51	89.30	94.35	95.72	96.08	91.79	4.80	28 mins
Accuracy % (Batch size: 32)	83.51	90.60	94.28	93.74	96.76	91.78	4.58	23 mins

Table 7.5 K-Fold Accuracy Values

Due to the imbalance of the dataset, instead of "KFold", "Stratified KFold" were used which ensures that each fold has the same number of samples of each class and its implementation is exactly the same as KFold [15].

8
Performance Analysis

This chapter covers the performance analysis of the system that was created.

8.1 Proposed Model

During the study, different algorithms and techniques were used to have better success rate. Some of the methods required more computation power and after finding the solutions, the system worked significantly faster and more accurately by combining the appropriate parameters. The system works in an expected amount of time with satisfying results of medical image classification. Different digital image processing steps and techniques were successful to extract features and pass them to machine learning algorithms for achievement of well image classification. Comparing with other neural networks, CNNs play an amazing role in computer vision-based tasks. With its efficient mathematical approach, CNN model performed well in this study.

Working on different architectures and parameters of CNN models, these operations require powerful hardware to perform in less amount of time. To experience more and make the models train faster, GPU activation was done successfully. By installing and preparing some requirements, CNN models' training processes finished quite earlier than CPU. In Figure 8.1 [16], an example study of the training time comparison took place according to batch size.

Running the system on GPU is not easy since the machine that was used during this study provides relatively low power of GPU computation when it is compared with new GPU models. Therefore, sometimes GPU stopped working due to some memory leak and solutions for this problem were searched on the internet. One of the recommended ways is using a function which would make GPU releases the global state. This would help avoid clutter from previous models and layers [17] when more than one models were running. Also, this function frees memory and prevents slowdown situations.

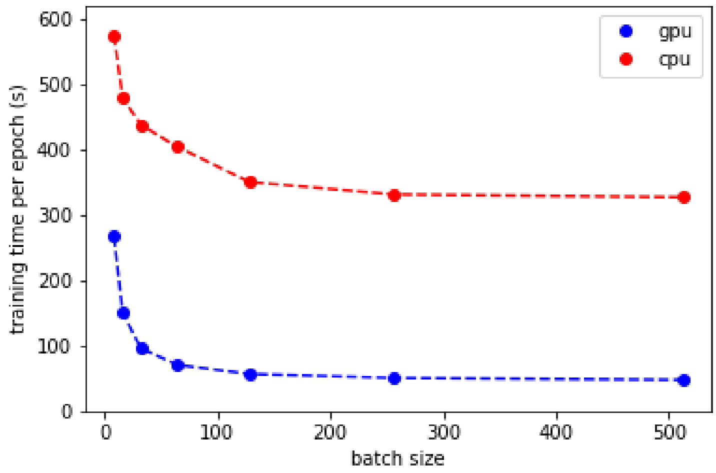

Figure 8.1 CNN Model Training Time Comparison

8.2 Comparison with Other Studies

There have been many studies that people worked on and each one has their own methods. These works include Ensemble Deep Learning approach with different pre-trained models. As a result, they have got huge number of parameters which require time and memory to perform. All details were shown in Table 8.1. This study's proposed model differs in method and dataset.

In this study's proposed method, after experiment of traditional image processing techniques, more advanced and successful method CNN was built. By implementation of different architectures and using the best one, machine learning classification was applied to CNN's extracted features. With achievement of this, there was approximately 0.01 model improvement. In addition, all the processes were run in a time and memory efficiency.

	Dataset	Method	Classification Accuracy (Multiple: M / Binary: B)
Afifi et al.(2021) [18]	11.197 Images (7.217 Normal and Other Lung pulmonary diseases, 5.451 Pneumonia, 1.056 COVID-19)	Ensemble Deep Learning (DenseNet161 with Different Weights)	M: 91.2%
Tang et al. (2021) [19]	15.477 Images (6.053 Pneumonia, 8.851 Normal, 573 COVID-19)	Ensemble Deep Learning (Different Snapshots of COVID-NET)	B: 96 %
Ghenea et al. (2021) [20]	8.088 Images (4.044 COVID-19, 4.044 NON-COVID-19)	Ensemble Deep Learning (VGG-NET based CNN1, CNN2)	B: 95.02%
Bhardwaj & Kaur (2021) [21]	10.000 Images (2.022 Other Pneumonia, 2.161 COVID-19, 5.563 Normal)	Ensemble Deep Learning (InceptionV3, DenseNet121, Xception, InceptionResNetV2)	M: 92.36%
Karacan & Eryilmaz(2021) [22]	21.165 Images (3.616 COVID-19, 6.012 Lung Opacity, 10.192 Normal, 1.345 Viral Pneumonia)	Ensemble Deep Learning (DenseNet121, MobileNetV2, Xception, InceptionResNetV2)	M: 93.98%
Proposed Model	21.165 Images (3.616 COVID-19, 6.012 Lung Opacity, 10.192 Normal, 1.345 Viral Pneumonia)	CNN-Machine Learning Hybrid Model	M:

Table 8.1 Studies’ Comparison

9
Conclusion

In the last chapter of this study, there will be a summary and finalization of the system.

In this research and implementation project, it was aimed to work on a medical image dataset with traditional image processing techniques and deep learning method. After reading all the images, a crucial step preprocessing was done to have a proper shape of the data to process. Then, by extracting features of the images with different methods, it was desired to have a great representation of the images in machine side. In order to achieve the goal, the study was splitted into four main parts. In each one, the relevant operations are completed with research of efficient ways of implementation. Also, additional studies such as machine learning performance enhancement with grid search, fine-tuning, visualization and analysis of dataset were carried on to be able to make the system work better. Comparison of methods has got significance in computer science projects and in this study, all the methods were compared with each other, and they were being improved with different approaches.

With creation of hybrid models with traditional-modern ways and machine-deep learning algorithms, development of the system and higher success rate were accomplished. Moreover, instead of CPU hardware component, running the CNN models on GPU was successfully done and by this, the computation time dramatically decreased. The goal of completing this system with high success rate of prediction and making it efficient in resource usage was succeeded.

References

[1] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, pp. 1–18, 2019.

[2] A. Africa and J. Velasco, "Development of a urine strip analyzer using artificial neural network using an android phone," *ARPN Journal of Engineering and Applied Sciences*, vol. 12, no. 6, pp. 1706–1712, 2017.

[3] C. Kone, "Introducing convolutional neural networks in deep learning," *Data Science*, 2019.

[4] A. Sharma, "Convolutional neural networks in python with keras," *DataCamp https://www. google. hu/search*, 2017.

[5] B. J. Erickson, P. Korfiatis, T. L. Kline, Z. Akkus, K. Philbrick, and A. D. Weston, "Deep learning in radiology: Does one size fit all?" *Journal of the American College of Radiology*, vol. 15, no. 3, pp. 521–526, 2018.

[6] V. Roman, *Cnn transfer learning & fine tuning*, 2020.

[7] S. Lau, "Learning rate schedules and adaptive learning rate methods for deep learning," *Towards Data Science*, 2017.

[8] K. Lv, S. Jiang, and J. Li, "Learning gradient descent: Better generalization and longer horizons," in *International Conference on Machine Learning*, PMLR, 2017, pp. 2247–2255.

[9] I. Syarif, A. Prugel-Bennett, and G. Wills, "Svm parameter optimization using grid search and genetic algorithm to improve classification performance," *Telkomnika*, vol. 14, no. 4, p. 1502, 2016.

[10] *Svm hyperparameter tuning using gridsearchcv: Ml*, Oct. 2021. [Online]. Available: `https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/`.

[11] S. Gupta, *Pros and cons of various classification ml algorithms*, Jun. 2020. [Online]. Available: `https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6`.

[12] A. Ramalingam, *How to pick the optimal image size for training convolution neural network?* Jun. 2021. [Online]. Available: `https://medium.com/analytics-vidhya/how-to-pick-the-optimal-image-size-for-training-convolution-neural-network-65702b880f05`.

[13] D. C.-E. Lin, *8 simple techniques to prevent overfitting*, Jun. 2020. [Online]. Available: `https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d`.

[14] *K fold cross validation technique.* [Online]. Available: https://inblog.in/K-Fold-Cross-Validation-Technique-NCaSQ8Kmfh.

[15] P. Kalkman, *Increase the accuracy of your cnn by following these 5 tips i learned from the kaggle community*, Feb. 2021. [Online]. Available: https://towardsdatascience.com/increase-the-accuracy-of-your-cnn-by-following-these-5-tips-i-learned-from-the-kaggle-community-27227ad39554.

[16] Moritzhambach, *Moritzhambach/cpu-vs-gpu-benchmark-on-mnist: Compare training duration of cnn with cpu (i7 8550u) vs gpu (mx150) with cuda depending on batch size*. [Online]. Available: https://github.com/moritzhambach/CPU-vs-GPU-benchmark-on-MNIST.

[17] *Tf.keras.backend.clear_session: Tensorflow core v2.7.0*, 2021. [Online]. Available: https://www.tensorflow.org/api%5C_docs/python/tf/keras/backend/clear%5C_session.

[18] A. Afifi, N. E. Hafsa, M. A. Ali, A. Alhumam, and S. Alsalman, "An ensemble of global and local-attention based convolutional neural networks for covid-19 diagnosis on chest x-ray images," *Symmetry*, vol. 13, no. 1, p. 113, 2021.

[19] S. Tang *et al.*, "Edl-covid: Ensemble deep learning for covid-19 cases detection from chest x-ray images," *IEEE Transactions on Industrial Informatics*, 2021.

[20] G.-L. Ghenea and V.-E. Neagoe, "Concurrent convolutional neural networks with decision fusion to diagnose covid-19 using chest x-ray imagery," in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2021, pp. 1–4.

[21] P. Bhardwaj and A. Kaur, "A novel and efficient deep learning approach for covid-19 detection using x-ray imaging modality," *International Journal of Imaging Systems and Technology*, vol. 31, no. 4, pp. 1775–1791, 2021.

[22] H. Karacan and F. ERYILMAZ, "Covid-19 detection from chest x-ray images and hybrid model recommendation with convolutional neural networks," *Journal of Advanced Research in Natural and Applied Sciences*, vol. 7, no. 4, pp. 486–503, 2021.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Onur PORSUK
Birthdate and Place of Birth: 27.08.1998, Ağrı
E-mail: l1117038@std.yildiz.edu.tr
Phone: 0545 441 71 56
Practical Training: Alimila Bilgi Teknolojileri ve Sağlık Hizmetleri Ltd. Yazılım Departmanı / Web Önyüz Geliştirme

SECOND MEMBER

Name-Surname: Engin Deniz ÇAĞLAR
Birthdate and Place of Birth: 09.07.1999, Ordu
E-mail: l1117080@std.yildiz.edu.tr
Phone: 0507 658 64 14
Practical Training: INFODIF YAZILIM VE BİLİŞİM TEKNOLOJİLERİ SAN. VE TİC. A.Ş. Yazılım Departmanı / Yapay Zeka

Project System Informations

System and Software: Windows / Linux İşletim Sistemi, Python
Required RAM: 4GB
Required Disk: 10GB