



Detection of COVID-19 Lung Image Infection using Convolution Neural Network

Lenin Marksia.U, Selvaganesh Srikanth.E, Esakkibabu.D, Dharshan K

Department of Computer Science and Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India.

To Cite this Article

Lenin Marksia.U, Selvaganesh Srikanth.E, Esakkibabu.D, Dharshan K, Detection of COVID-19 Lung Image Infection using Convolution Neural Network, International Journal for Modern Trends in Science and Technology, 2024, 10(02), pages. 447-457.<https://doi.org/10.46501/IJMTST1002060>

Article Info

Received: 28 January 2024; Accepted: 19 February 2024; Published: 25 February 2024.

Copyright © Lenin Marksia.U at al;. This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

*The early detection of SARS-CoV-2, the causative agent of (COVID-19) is now a critical task for the clinical practitioners. The COVID-19 spread is announced as pandemic outbreak between people worldwide by WHO since 11/ March/ 2020. In this consequence, it is top critical priority to become aware of the infected people so that prevention procedures can be processed to minimize the COVID-19 spread and to begin early medical health care of those infected persons. In this paper, the deep studying based totally methodology is usually recommended for the detection of COVID-19 infected patients using CT images. The help vector gadget classifies the corona affected CT images from others through usage of the deep features. The technique is useful for the clinical practitioners for early detection of COVID-19 infected patients. The suggested system of multi-level thresholding plus SVM presented high accuracy in classification of the infected lung with Covid-19. All images were of the same size and stored in JPEG format with 512 * 512 pixels. The average sensitivity, specificity, and accuracy of the lung classification using the proposed model results were 95.76%, 99.7%, and 97.48%, respectively.*

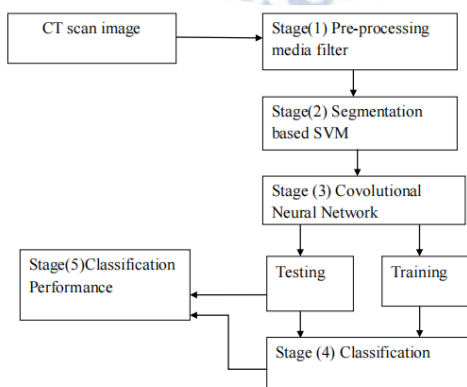
Keywords: Neural Network, Convolutional Neural Network, Deep Recurrent Neural Networks

1. INTRODUCTION

The COVID-19 pandemic situation as of March 30, 2020, comparing it with the SARS outbreak of 2003. It highlights the scale of COVID-19 in terms of confirmed cases, critical cases, and deaths worldwide. It emphasizes the novelty and diversity of coronaviruses, including their ability to cause diseases ranging from the common cold to more severe diseases like SARS and MERS. The symptoms of coronavirus infection, as mentioned, lead to serious respiratory complications,

and the text notes the significant mortality of SARS in 2003, mainly in China. the digital and connected world's impact on managing the COVID-19 pandemic, contrasting it with the SARS outbreak in 2003. It mentions the vast amount of data generated today and the potential of AI and ML in tracing and predicting the spread of infectious diseases. The importance of leveraging big data and intelligent analytics for public health is highlighted. detecting COVID-19, including clinical characteristics, RT-PCR, and thermal imaging. It

details the effectiveness of the U-Net++ neural network model in processing CT scans to classify COVID-19 cases, noting its high specificity, sensitivity, NPV, PPV, and overall accuracy. Additionally, it mentions a 3-category classification model for distinguishing COVID-19 cases using feature extraction with the Feature Pyramid Network, reporting an acceptable total accuracy. The early COVID-19 pandemic response, highlighting the technological advancements in diagnostics and data analytics compared to previous outbreaks.



Block diagram of Proposed system

1. Preprocessing

Data preprocessing is a fundamental stage in the machine learning workflow, where engineers dedicate significant time to clean and prepare data ahead of model development. This process involves identifying and correcting anomalies in the data, such as outliers and missing values, and removing irrelevant or noisy information that could detract from model performance. In parallel, image preprocessing represents a series of basic operations applied to images to prepare them for further analysis or processing. Unlike data preprocessing, the aim here is not to enrich the data but to refine it for specific tasks. This might involve reducing the data (in a way, decreasing its entropy) to simplify subsequent operations, ensuring that the transformations do

2. Segmentation

Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image. Typically the first step in any automated computer vision application.

Some applications of image segmentation in medical filed includes:

- ⊙ a. Locate tumors and other pathologies
- ⊙ b. Measure tissue volumes
- ⊙ c. Computer guide surgery

Classification -Classification is a procedure to automatically categories all pixel in an image of a terrain into land use and land cover classes. This concept is dealt under broad subject, namely. "pattern recognition". Spectral pattern recognition

refers to the family of classification procedures that utilizes this pixel-by-pixel spectral information as the basis for automated land cover classification. Spatial pattern recognition of mage on the spatial relationship with pixel surrounding.

CNN-Convolutional Neural Networks (CNNs) are a specialized kind of neural networks particularly effective for processing data with a grid-like topology, such as images. A CNN typically consists of three distinct types of layers: convolutional layers, pooling layers, and fully-connected layers. These layers are stacked to form a CNN architecture, which can be exemplified by a model designed for classifying images from the MNIST dataset. The operation of a CNN can be understood through its four main functional stages:

1. Input Layer-Similar to traditional Artificial Neural Networks (ANNs), the input layer of a CNN holds the pixel values of the input image. This layer serves as the entry point for data into the network.

2. Convolutional Layer-This layer computes the output of neurons that are connected to local regions in the input, using the dot product of their weights and the region in the input volume they are connected to. To introduce non-linearity, an activation function, such as the Rectified Linear Unit (ReLU), is applied element-wise to the output of this layer. This process allows the network to learn complex patterns in the data.

3. Pooling Layer-Following the convolutional layer, the pooling layer performs downsampling along the spatial dimensions (i.e., width and height) of the input. This reduction in spatial dimensions results in a decrease in the number of parameters and computation in the

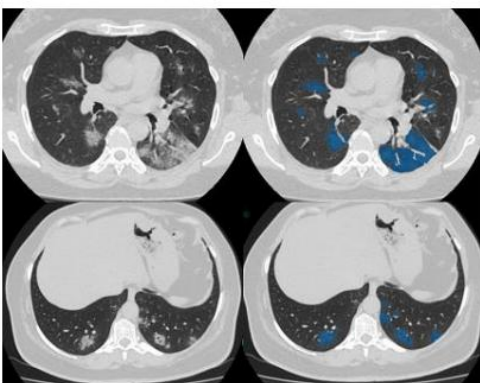
network. Pooling helps the network to be more robust to variations in the position of features in the input image.

4. Fully-Connected Layers These layers function similarly to the layers in standard ANNs. They aim to produce class scores from the activations received from the previous layers, which are then used for classification purposes. It is common to use ReLU activation functions between these layers as well to enhance network performance. Through sequential processing -

starting with convolutional operations to extract features, applying non-linear activation functions to introduce non-linearity, reducing dimensionality with pooling, and classifying with fully-connected layers - CNNs efficiently transform the initial input into outputs for classification or regression tasks. This structured transformation process enables CNNs to perform exceptionally well on tasks involving image recognition and analysis.

2. IMPLEMENTATION METHODOLOGY
DATASET CHARACTERISTICS

In total, 40 contrast-enhanced lungs CT with 512x512 in-plane resolution. 15 normal lung images are from the Montgomery County CT Set, while other 25 images are infected lungs with COVID-19 from covid-chestxray dataset-master. In total, 20 lungs CT with 512x512 in-plane resolution. Among them, 10 normal lung images are from the Montgomery County CT Set, and 10 are from the public database covid-chestxray-datasetmaster . Figure shows sample of the images we have used in this study



Sample CT images of the normal and covid-19 Affected patients

MULTI-LEVEL THRESHOLDING (MT)

A multi-level threshold is a process that splits grey image into several distinct areas. This technique specifies various image thresholds and splits the image into areas with certain intensities that correspond to one background and several objects . The method works very well with coloured objects or complex backgrounds, cases in which a binary level threshold fails to realize satisfactory results is given in Eq. (1). Where p is one of the pixels $m \times n$ that can be represented as one of the grayscale levels $L1 = \{0, 1, 2, \dots, L-1\}$, C_1 and C_2 are classes with pixels p, representing $\{th_1, th_2, \dots, th_i, th_i + 1, th_k\}$ different t

$$C_1 \leftarrow p \text{ if } 0 \leq p \leq th_1$$

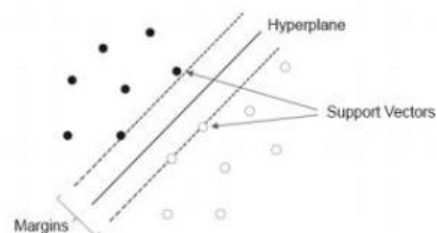
$$C_2 \leftarrow p \text{ if } th_1 \leq p \leq th_2$$

$$C_{i+1} \leftarrow p \text{ if } th_i \leq p \leq th_{i+1}$$

$$C_n \leftarrow p \text{ if } th_k \leq p \leq L-1$$

SUPPORT VECTOR MACHINES

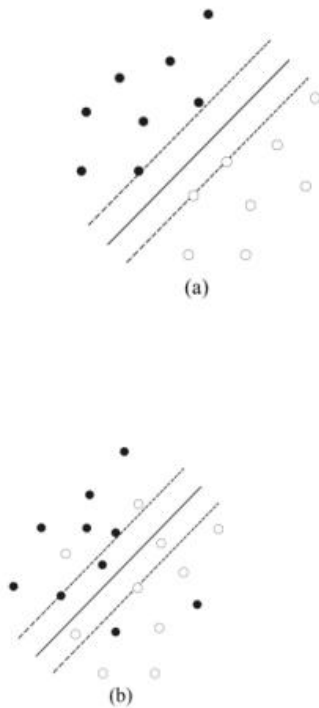
Cortes and Vapnik proposed a new class of models known as support vector machines (SVM), which are based on the theory of Statistical Learning . Basically, this class of models is based on the construction of optimal hyperplanes to classify labels—usually into two distinct classes—as well as on the use of kernelization to increase the model’s flexibility [48]. SVM is used to solve classification problems, finding the hyperplane with the greatest separation space (delimited by its margins) between the categories of the chosen variable. Figure 5 displays the general representation of an SVM. The 14 hyperplane is the equation separating the two classes, while the support vectors are the data points closest to the margin boundaries



Vector machine (SVM)

Support Vector Classifier

The simplest form of SVM is called support vector classifier (or linear SVM), which is a technique that allows for linear separation of data whenever possible. This technique can be used with rigid (hard) margins, which do not allow for misclassification but hinders generalization; or soft margins, which allow the entry of wrongly classified data but improve the model generalization. Figure 6a depicts an SVM using rigid margins, with no data points between the margins; while Figure 6b depicts a soft margin SVM.



In SVM with soft margins, with n the sample size of training data and $i = 1, \dots, n$, the ϵ_i are slack variables (errors) which are added to relax the restrictions and turn the model more receptive to new data. Therefore, the function to be maximized or minimized can be expressed as:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \epsilon_i,$$

since C is a regularization constant that imposes a weight to minimize errors, there is no limit to the number of wrong classifications. If $C \rightarrow \infty$, a soft margin SVM will act as a rigid margin SVM. After applying the maximization process via Lagrange multipliers of the Equation (1) with margins constraints, the SVM classifier with soft margins is given by

$$f(x) = \text{sign} \left(\sum_{i=1}^n \alpha_i y_i (x_i \cdot x) + b \right).$$

where b is a constant parameter also called bias. The α_i are the

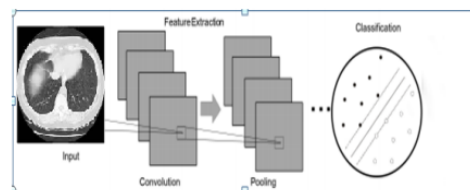
Lagrangian parameters that identify the support vectors when $\alpha_i > 0$ implies that

x_i is a support vector and an important observation in the entire model. The

function sign is responsible to verify if any observation is upper or lower of the estimated hyperplan

3. CONVOLUTION PROCEDURES AND SVM

Analogously to a convolutional neural network (CNN), convolutional support vector machine uses the convolutional image processing to reduce the complexity of the image, through operations as the convolution and pooling as shown in Section 2.2. Initially a convolutional model is generated to select and optimize the filter's that will be used. Afterwards, with fixed hyper parameters of the convolutional layers, the traditional support vector machine (SVM) is employed, and the model can be applied using any of the kernels discussed above, with their due parameters that can be estimated through a tuning process. Figure 8 shows the convolution support vector machine (CSVM) architecture.

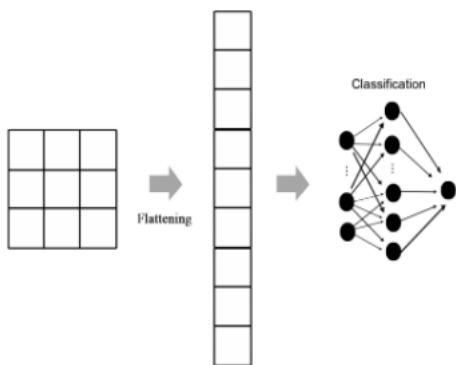


CVSM architecture

Among the existing layers in the pre-processing of data through convolution, we can mention two that have extremely important roles, pooling layer and flattening layer. Directly, the pooling layer seeks to somehow reduce the size of the data, so that information is not lost, but to keep those that have a greater amount of significant information, reducing noise in order to reduce processing time. In the final stage of the pre-processing stage, the flattening layer seeks to convert

the data into a single observation in order to facilitate insertion into the model sequence. In general, the convolution acts as a complex processing filter which can be applied in other model instead of the classic Multilayer Perceptron (MLP). After the flattening step, SVM replaces MLP.

Figure 4.5 shows flattening layer after convolution filter processing



Representation of a flattening layer.

4. REAL DATA STUDY

Data and CT Image Acquisition Images used in this analysis came from anteroposterior radiographs (CTs).

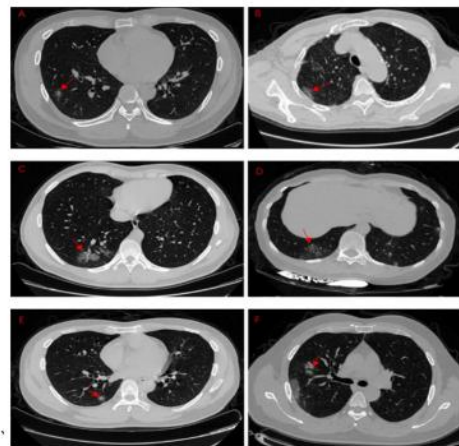
The section describes the data collection and preprocessing steps for a study involving CT images for COVID-19 research. The images were obtained from two primary sources: COVID-19 and other lung disease cases from the COVID-19 Image Data Collection, and healthy CT images from the Open Access Biomedical Search Engine (Open-i). The study aimed to assist in understanding and treating COVID-19 by using these images.

Initially, 105 COVID-19 patient images were collected from the GitHub repository dedicated to COVID-19 CXR data collection. A subsequent update on July 27, 2020, added 142 new images, including AP and AP Supine views, after confirming SARS-CoV-2 infection in different patients. A hashing technique was employed to identify and remove 30 duplicate images, resulting in a final count of 217 COVID-19 positive CT images.

For preprocessing, the images underwent grayscale or color conversion via the OpenCV Python library, with no additional noise reduction techniques applied. Another set of 108 images featuring other lung diseases (such as

SARS, Pneumocystis, or Legionella) was also gathered from the same GitHub repository. For healthy chest CT images, a search was conducted through the Open-i service, initially retrieving 105 images, with seven new images labeled “no Finding” from the COVID-19 Image Data Collection later added to the dataset, totaling 112 images. No duplicates were found in this set.

The final dataset comprised 437 images, with sizes ranging from a minimum of 235x256 pixels to a maximum of 4757x5623 pixels. This comprehensive dataset aims to support the analysis and development of diagnostic tools for COVID-19 and other lung diseases.



Example of CTs: (a,b) coronavirus (COVID-19); (c,d) other diseases; and (e,f) healthy

An exploratory analysis of the metadata provided by the COVID-19

Image Data Collection resulted in the data shown by Such metadata

information was used in our predictive models since it is only available for

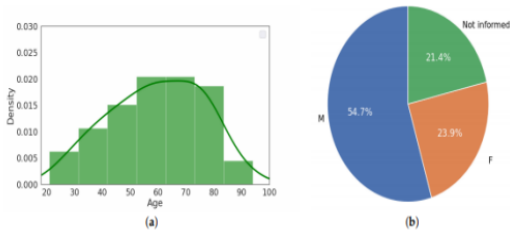
COVID-19 cases. In addition, statistics metrics from patients’ ages and sex

information were calculated: mean $\bar{X} = 59.45$, median Med = 61, and standard

deviation $\sigma = 16.88$. Figure 12 displays an example of a healthy chest CT and an

CT of a patient diagnosed with COVID-19. These images correspond to the

dataset used to train our models. No data augmentation process was used.



Metadata provided by the COVID-19 reference dataset. Age histogram (a) and gender pie chart (b).

Overview of the image dataset used in this work

	COVID-19	Other Diseases	Healthy	Total
Quantity	217	108	112	437

Predictive Models

The evaluation of all methods was realized through a repeated holdout validation technique, with a split ratio of the data of 90–10% in the training-test set and a total of 100 repetitions. Despite the most common validation technique used in image classification tasks being K-Fold, a high number of repetitions in the hold-out achieve lower bias and variance [61]. The classification performance of the CSVM models was compared with those from other methods in cluing traditional neural network (MLP1 and MLP2), in which all of them converged after 20 epochs. Table 6 presents the performance measure from ten methods ran 100 times each.

	Cell No	hyperechoic Level	microcalcificatio
1	1	494.3083	2.19
2	2	475.2076	1.62
3			
4			

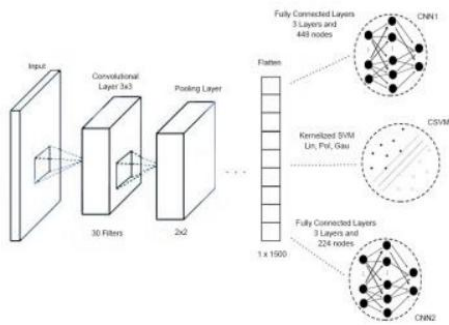
CONVOLUTIONAL NEURAL NETWORK

The provided text outlines a comprehensive study on the application of Convolutional Neural Networks (CNNs) in image recognition, specifically using CT images for COVID-19 detection among other lung diseases. The text delves into several aspects of CNNs, including data acquisition, CNN architecture, and the application of CNN models for image classification. CT images were sourced from anteroposterior radiographs, with COVID-19 and other lung disease cases collected from the COVID-19 Image Data Collection and healthy images from the Open Access Biomedical Search Engine. A total of 217 COVID-19 detected CTs were

finalized after removing duplicates, with a subsequent dataset including images of other lung diseases and healthy lungs, totaling 437 images. These images were preprocessed using grayscale/color conversion via OpenCV, with no noise reduction methods applied. CNNs are highlighted as a subset of deep learning techniques, well-suited for image recognition due to their ability to automatically extract features. A CNN architecture typically comprises two main blocks: feature extraction (consisting of convolutional and pooling layers) and classification (fully connected layers with activation functions). Feature extraction involves applying filters to input images to generate feature maps, which are then simplified through pooling. The classification block predicts output based on these features. The text emphasizes the challenges of overfitting in neural networks and the importance of reducing complexity to prevent it, ensuring better model generalization and predictive performance. For COVID-19 Detection Two CNN architectures, CNN1 and CNN2, are described, differing in their layer configurations and filter arrangements, used for comparative analysis in detecting COVID-19 from CT images. The architectures vary in the number of nodes per layer and the sequence of convolutional and pooling layers, with specific details on the activation functions and optimizers used.

- A dataset comprising 420 images (half with COVID-19 and half without) was employed to train the models over 25 epochs, using specific parameters like batch size, objective function, and metrics for performance evaluation. The study also mentions the use of pre-trained weights from CNN as filters for classification in a convolutional SVM model, adjusting the support vector component with different kernel functions.

This summary encapsulates the key points from the detailed exposition on using CNNs for CT image classification in the context of COVID-19 detection, highlighting the data acquisition process, CNN architecture and functionality, and specific configurations and outcomes of applying CNN models to real-world data.



CNN architecture

SOFTWARE REQUIREMENT

1. OPERATING SYSTEM : WINDOWS XP and Above
2. FRONT END: MATLAB 2013 a and above

INTRODUCTION TO MATLAB

Description about MATLAB

MATLAB is the ideal, timesaving solution for understanding complex data through interactive visualization and analysis and software application development.. From “quick-look” analysis and display to commercial programming projects, MATLAB combines the technology and tools need with a modern, extensible environment. MATLAB requires less code and programming expertise to produce dynamic visualizations and analyses, or to develop powerful software

applications with user-friendly interfaces

Access Virtually Any Type of Data

MATLAB provides built-in support for the data sources, data types, file formats, and file sizes use. Use MATLAB to access common formats like TIFF, JPEG, DICOM, and DXF, hierarchical scientific data formats like HDF, HDF EOS, CDF, and net CDF, as well as custom binary and ASCII formats. And, since MATLAB is designed for large, multi-dimensional data, no job is too complex. Receive data from remote servers or write files to a disk at a remote

location using common protocols. MATLAB acts as a client to HTTP and FTP 29 servers, and provides access to common Open Geospatial Consortium (OGC) servers such as Web Map Service (WMS) and Web Coverage Service (WCS).

Analyzer Data

From ad-hoc data analysis to commercially distributed applications, MATLAB has the data analysis capabilities, including: Simple syntax, dynamic data typing, and

array-oriented operations, Built-in functionality suitable for

many data trends, with tools for 2- and 3-dimensional gridding and interpolation, routines for curve and surface fitting, and the ability to perform multithreaded computations. A rich library of high-performance routines to analyze data. The ability to add own specialized routines to the library by writing procedures more quickly than in C or FORTRAN

Image Processing

If need to convert raw images to information, MATLAB contains an extensive library of image processing and analysis routines. MATLAB allows extracting useful information from nearly any type of digital image with tools for transforming image geometry, mapping, masking and statistics, warping, analyzing regions of interest, and contrasting and filtering.

Signal Processing

MATLAB has advanced tools for processing signal data, including transforms for signal decomposition, windowing algorithms, routines for smoothing, convolving and applying digital filters to remove noise, and correlation and covariance techniques to analyze signals with random components. MATLAB also has a built in wavelet toolkit that provides GUI and programmatic wavelet analysis of multi-dimensional data.³⁰

Math and Statistics Routines

If need to add mathematical and statistical functionality quickly and easily to an MATLAB application, MATLAB Analyst is a premium MATLAB product that combines advanced data visualization and analysis capabilities with

the comprehensive mathematical and statistical routines of the IMSL™ C Numerical Library. This solution allows to access proven algorithms and routines directly from the MATLAB environment without using additional software.

Visualize Data

MATLAB is designed for visualizing large and complex datasets. Easily and quickly create powerful visualizations, ranging from simple 2D plots and image displays to complex, interactive 3D graphics designed to

leverage OpenGL hardware acceleration and take advantage of multiple processors for rapid rendering.

Pre-Built Functionality

Whether're an experienced MATLAB programmer or an ad-hoc user, the iTools will reduce r programming time and effort. iTools provide interactive, pre-built applets for graphing, plotting, displaying, mapping, and more. Use them to open, analyze and visualize data and produce quality, publication-ready graphics or to build customized visualization environments.

Custom Visualization Building Blocks

While MATLAB provides a rich set of pre-built visualization tools, it also provides complete access to a full programmatic graphics GUI. Create own customized, interactive visualization application with the graphics and graphical user interface (GUI) widgets creation capabilities. With MATLAB, can do 31 everything from creating a general application for distribution to developing a solution optimized for a specific hardware configuration.

Develop an Application

MATLAB's easy-to-use development environment - the MATLAB Workbench - is based on the popular Eclipse development platform. Providing a modern, familiar environment for users to develop and deploy commercial applications, it combines the advantages of a full-featured, cross-platform development framework with the tools including: cross-platform user interface, Code editing tools, Code debugging capabilities, Plug-in architecture for extensibility, User assistance, such as hover help and content assist, Internationalization of user interface menus and text.

Analysis Enhancements

The enhancement has been made to MATLAB's data-analysis

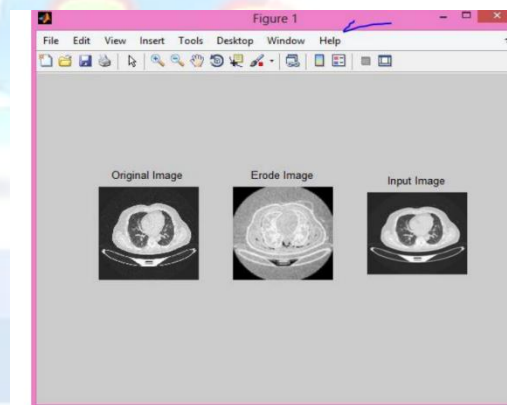
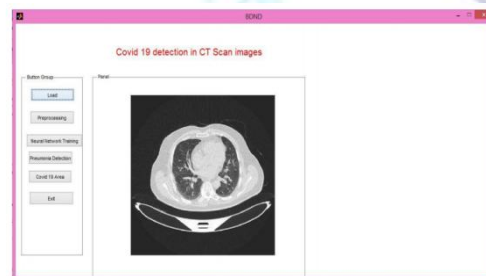
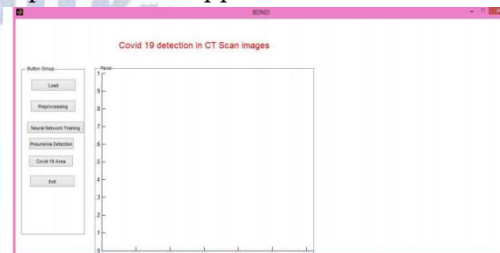
functionality for the 6.3 release are Butterworth Filter, Canny Edge Filter, Impulse Response Filters, and Functional.

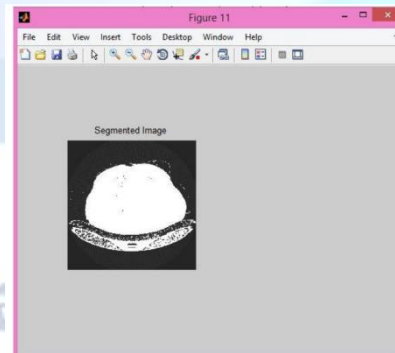
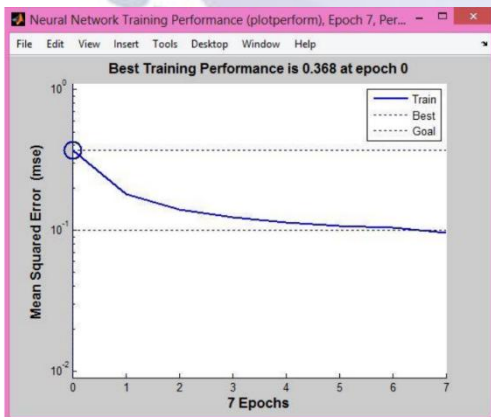
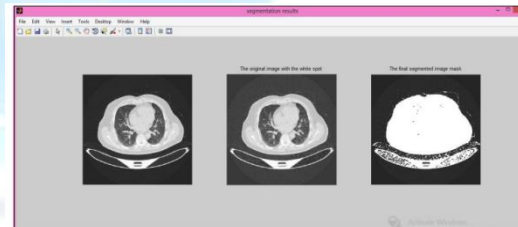
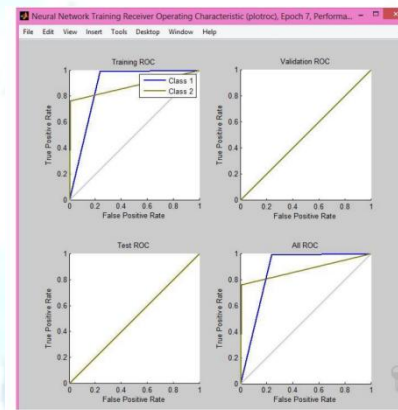
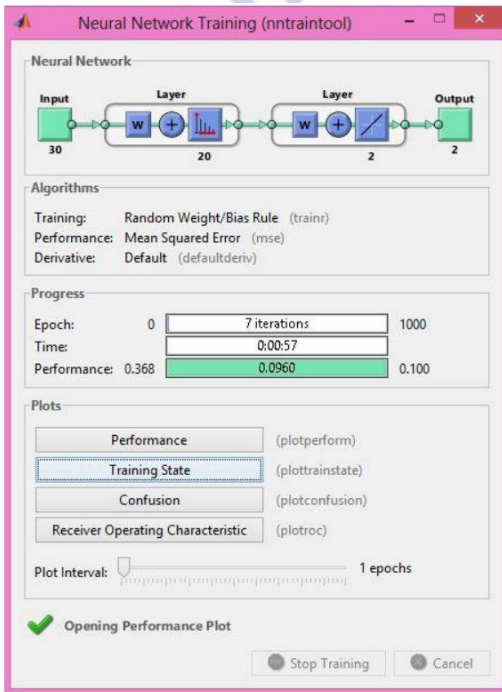
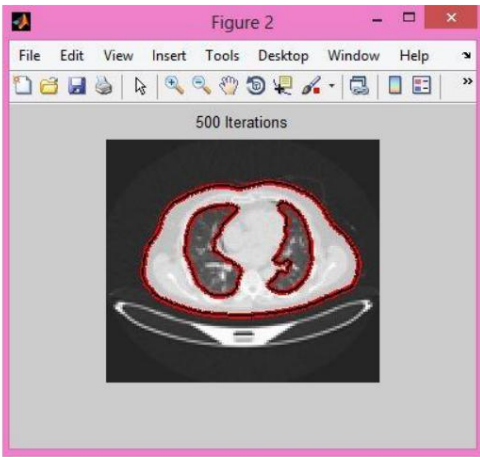
List of MATLAB Routines

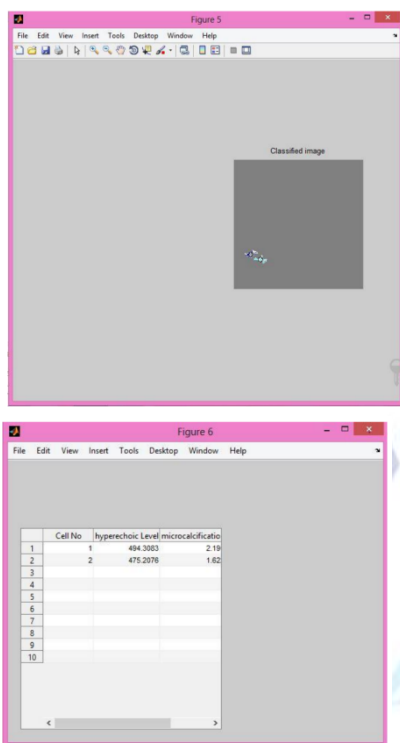
The list of all routines included in MATLAB, categorized by functionality are 3D Visualization, Animation, Direct Graphics, Widget Routines, Window Routines, Compound,

Integrate Applications

MATLAB is a flexible and extensible visualization, analysis, and development environment that easily integrate with the other software tools use. MATLAB provides a host of ways to import code and functionality from other programming languages and advanced methods for exporting MATLAB capabilities to r applications written in other languages







5. FINAL CONSIDERATIONS

This project presented a novel study and application of the convolutional support vector machines to classify patients infected with COVID-19 using CT data. The result was compared with the use of a CNN approach for this task, which is considered the state-of-art in many image classification tasks [68–71]. In addition, the CSVM showed up as being computationally cheaper since it can be one-hundred times faster, on average, when compared with CNN in this situation with a medium sample size. In comparison with other works that presented analysis on detecting SARS-CoV-2 from CT images, we presented a novel database with the most recent data that is composed of more 26 instances, and with a greater diversity of diseases in the group of no-COVID, improving the capacity of the model to generalize and predict new observations. Additionally, even by boosting research on COVID-19, data on such patients are still small or medium-scale, a fact that makes CSVM even more effective for these cases. For future works, the number of kernel functions, as well as its hyper parameters can be explored to achieve even stronger results in CSVM. In addition, the architectures in both cases, CSVM and CNN, can be diversified. Reverse transcription polymerase chain reaction (RT-PCR) is a laboratory technique that is used for testing COVID-19 using a blood sample of the case. Using real time RT-PCR test confirms the actual number

of confirmed cases globally. The gold standard test to detect the COVID-19 confirmed cases is quantitative reverse transcriptase PCR (qRT-PCR). However, this approach needs a CDC guide's sample collection, qualified microbiology expertise, time from 4hrs up to 6hrs. Therefore, medical imaging is very important candidate in screening for the COVID-19 cases. Radiologist's diagnosis involves computed tomography (CT) scans, chest CT (CXR) radiographs. The model produced 93.55% and 100% for specificity and sensitivity, respectively. In addition, the model presented negative positive value (NPV) of 100%, positive value (PPV) of 84.62%, and a total accuracy of 95.24%. The

model greatly helps radiologists by decreasing the reading time of CT scans by 65%. On the other hand, 3-category classification model is designed to distinguish the COVID-19 cases where both Song et al. (2020) and Xu et al. (2020) have applied feature extraction using Feature Pyramid Network and several fullyconnected layers for cases classifications. The feature extraction using Feature Pyramid Network model produces an acceptable total accuracy of 86.7% using CT scans. The fully-connected layers model can accurately detect the COVID-19 cases with AUC of 0.99 and sensitivity of 0.93.27

As a result of this surveying, Although the diagnosis of COVID-19 is based firstly and mainly on the presence of clinical signs related to respiratory systems and more specifically an pneumonia related signs (e.g., dry cough, fatigue, myalgia, fever, and dyspnea) as well as history of recent (two weeks). Although, CT scans and Xray imaging are time consuming and exhaustive even for expert radiologists, it is characterized by that the highest sensitivity diagnostic method in comparison to the gold standard (qRT-PCR) and other already developed or will be developed rapid diagnostics.

So, relying on radiological imaging in screening and diagnosis of COVID-19 is very medically meaningful and therefore a need to automated system based on Artificial Intelligence (AI) tools could be accurately provide automated, less exhaustive for medical imaging workers, early detection, follow up method of COVID-19 cases and at same time method of differentiation of the radiological images of affected lung due to COVID-19 from other causes of lung affections especially MERS which caused by (MERS-CoV) and

SARS which caused by (SARS-CoV-1) which are another viruses belonging to

Coronaviridae viruses family, the same family to which the COVID-19 causative agent (SARS-CoV-2) is belonging and all three viruses have an overlapping clinical signs and affected lung radiological image patterns but a differences and uniqueness still present. Through detecting and analyzing the previously mentioned early lung affection patterns present in the radiological images. This project presents automatic CT COVID-19 lung image classification system based on multi-level thresholding and supported vector machine. The study is usually recommended for the early detection of COVID-19 infected patients using CT images. The help vector gadget classifies the COVID-19 affected lung CT images from others through usage of the deep features. The technique is useful for the clinical practitioners for early detection of COVID-19 infected patients.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

- [1] Revathi et al. (2016) explored bioinformatics, emphasizing the role of SVM and RST in analyzing protein datasets. They proposed an optimal feature selection method using Ant Colony with Rough Set Theory (ACRST) to enhance system performance by reducing data complexity.
- [2] Nayak et al. (2015) reviewed the application of SVM in data mining from 2001 to 2014, discussing the optimization challenges and the integration of prior knowledge into SVMs. The study highlighted the scalability issues with SVM as data size increases.
- [3] Khan et al. (2015) described the use of SVM for sentiment analysis of user opinions on social media, aiming to classify comments as positive or negative to facilitate text categorization.
- [4] Zhou et al. (2015) introduced a practical reduction from Elastic Net and Lasso to SVM, showcasing a parallel solver for Elastic Net and Lasso that significantly speeds up computation, leveraging GPU and multi-core CPUs.
- [5] Jagtap et al. (2014) discussed the challenges in data mining within healthcare, focusing on the sensitive and voluminous nature of medical data. They proposed a unified learning framework incorporating LS-SVM, PSVM, and ELM for efficient data handling.
- [6] Hanaa et al. (2012) presented a classification model based on RST and decision rules to increase classification efficiency, employing Boolean reasoning for data discretization and genetic algorithms for finding reducts.
- [7] El-Naqa (2012) explored an SVM-based method for detecting microcalcifications in mammograms, achieving high sensitivity and suggesting SVM's promise in medical imaging applications.
- [8] Gogoi et al. (2011) proposed an RST-based approach for mining concise rules from inconsistent data, achieving satisfactory accuracy on various datasets.
- [9] Lee et al. (2011) introduced a Rough-Neuro-Fuzzy classifier and ARFIS, a rough-fuzzy hybrid approach for Alzheimer's disease data classification, emphasizing the efficiency of rough set-based feature reduction.
- [10] Chen et al. (2009) developed a method for pattern recognition and multi-class classification using SVM, focusing on vehicle identification using color and type vectors from roadside camera images.
- [11] These studies collectively showcase the versatility of SVM and RST in addressing complex problems across different domains, from bioinformatics and sentiment analysis to healthcare and pattern recognition.