Soft Computing (2023) 27:10555–10570
https://doi.org/10.1007/s00500-023-08508-y

DATA ANALYTICS AND MACHINE LEARNING

Diagnosis of COVID-19 from blood parameters using convolutional neural network

Gizemnur Erol Doğan[1] · Betül Uzbaş[2]

Accepted: 10 May 2023 / Published online: 28 May 2023

Abstract
Asymptomatically presenting COVID-19 complicates the detection of infected individuals. Additionally, the virus changes too many genomic variants, which increases the virus's ability to spread. Because there isn't a specific treatment for COVID-19 in a short time, the essential goal is to reduce the virulence of the disease. Blood parameters, which contain essential clinical information about infectious diseases and are easy to access, have an important place in COVID-19 detection. The convolutional neural network (CNN) architecture, which is popular in image processing, produces highly successful results for COVID-19 detection models. When the literature is examined, it is seen that COVID-19 studies with CNN are generally done using lung images. In this study, one-dimensional (1D) blood parameters data were converted into two-dimensional (2D) image data after preprocessing, and COVID-19 detection was made with CNN. The t-distributed stochastic neighbor embedding method was applied to transfer the feature vectors to the 2D plane. All data were framed with convex hull and minimum bounding rectangle algorithms to obtain image data. The image data obtained by pixel mapping was presented to the developed 3-line CNN architecture. This study proposes an effective and successful model by providing a combination of low-cost and rapidly-accessible blood parameters and CNN architecture making image data processing highly successful for COVID-19 detection. Ultimately, COVID-19 detection was made with a success rate of 94.85%. This study has brought a new perspective to COVID-19 detection studies by obtaining 2D image data from 1D COVID-19 blood parameters and using CNN.

Keywords COVID-19 · Deep learning · 1D to 2D conversion · Convolutional neural network (CNN) · Image processing

1 Introduction

COVID-19 is an infectious respiratory disease that emerged in Wuhan, China in December 2019 (Brinati et al. 2020), caused by the SARS-CoV-2 virus. People with chronic diseases such as diabetes, cancer, respiratory tract, and heart disease are especially vulnerable to the catastrophic consequences of these diseases, which can result in permanent discomfort or even death (World Health Organization (WHO) 2020). According to the World Health Organization's November 2022 report (World Health Organization (WHO) 2022), there have been approximately 640 million confirmed cases and more than 6 million deaths since the pandemic outbreak. In most reported cases, the virus incubates for 1–14 days before symptoms of infection appear (World Health Organization (WHO) 2019). In COVID-19 patients, these symptoms manifest as cough, shortness of breath, fever, and other acute respiratory distress syndromes (Paules et al. 2020; Menni et al. 2020). These symptoms, which are also symptoms of other respiratory diseases, cannot be decisive in COVID-19 diagnosis (Day 2020). Therefore, the spread of the disease cannot be stopped. Today, the biggest advancement in the fight against COVID-19 is the development of vaccine formulations with the sharing of the first full genome sequence of COVID-19 and the production of many vaccines for the disease (Alexandridi et al. 2022).

✉ Gizemnur Erol Doğan
gerol@ktun.edu.tr

Betül Uzbaş
buzbas@ktun.edu.tr

[1] Software Engineering Department, Konya Technical University, Konya, Turkey

[2] Computer Engineering Department, Konya Technical University, Konya, Turkey

However, the virus's ability to change too many genetic variants facilitates the disease's ability to spread, which is still another major reason why the spread cannot be stopped (Alkhodari and Khandoker 2022). And although the effects of the disease seem to be alleviated in infected individuals thanks to the vaccines developed, too many genomic variants of the virus change the effectiveness of the vaccines, and thus the spread of the disease cannot be prevented (Barton et al. 2021; Harvey et al. 2021). This disease, whose rate of spread cannot be stopped, has quite affected the health infrastructure systems, economies, industries, and socialities of the countries (Dey et al. 2020; Fong et al. 2020). Although there is no definitive treatment for this disease, whose spread cannot be stopped, promising results have been obtained for the treatment of COVID-19 by conducting various studies on the virus and the human genome (Zannella et al. 2022). Since there is no certain treatment for COVID-19 in a short time, the essential goal has been to reduce the transmission of the disease. The first attempt to reduce the virulence of the disease is the rapid COVID-19 detection of individuals. In addition, contact tracing of detected COVID-19 patients is an important factor in controlling the virulence of the disease. In the literature, a successful COVID-19 contact tracing method has been proposed to improve the performance of COVID-19 monitoring applications by obtaining human activity recognition (HAR) images (D'Angelo and Palmieri 2021).

RT-PCR is the most commonly used diagnostic test in COVID-19 diagnosis from respiratory samples in a laboratory setting. Due to the extraordinarily rapid spread of the virus, there has been a high demand for RT-PCR tests worldwide. This increase in demand has revealed many limitations of the RT-PCR method, such as the need for certified laboratories, the average diagnosis that can be made in 2–3 h, trained personnel, and expensive equipment (Li et al. 2020). Other most preferred diagnostic methods are computed tomography (CT) and X-ray imaging. These methods are costly, difficult to reach, and require expert knowledge. Finally, blood parameters, which are less used than RT-PCR, CT, and X-ray imaging and are a COVID-19 diagnostic method, have significant clinical values for infectious diseases (Demirdal and Sen 2018; Jan et al. 2019) and are easy to access and low cost. There is an urgent need for rapid and easily accessible alternative diagnostic methods for COVID-19, which continues to be contagious with variant change. Therefore, the development of artificial intelligence-based COVID-19 detection systems has become an emergency.

While CNN can work on 2D data such as CT and X-ray images, it cannot work with 1D data such as blood parameters. It is a disadvantage that such a successful neural network model cannot be used for 1-dimensional data. To eliminate this disadvantage, Sharma et al. developed the DeepInsight (Sharma et al. 2019) method converting 1D data to 2D data. The method developed showed good performance in classification processes with many datasets such as genomics, voice, and text. In this study, the DeepInsight (Sharma et al. 2019) method was used to obtain image data from the blood parameters of individuals, and the developed CNN architecture and the COVID-19 detection model were presented. The study has two main aims:

1. Generalizing the conversion of 1D data to 2D data in serious and large-scale diseases such as COVID-19 in the healthcare industry. In this way, increasing the usability of blood parameters, which are cheaper than CT and X-ray imaging, with successful artificial intelligence models.
2. To provide an effective and successful model for COVID-19 detection by providing the combination of blood parameters, which is low-cost and rapidly accessible, with CNN architecture, which brings high success in image data.

Within the scope of this study, a COVID-19 detection model was developed with the CNN architecture developed by obtaining image data from the blood parameters of COVID-19 positive and negative patients. In this way, the combination of low-cost and fast-accessible blood parameters for COVID-19 detection and the CNN architecture, which brings high success in image data, has been achieved. This study has brought a new perspective to the detection studies of positive and negative patients from 1D COVID-19 blood parameters using CNN architecture.

The remainder of this article is organized as follows: The literature review is presented in chapter 2. Furthermore, a base explanation of the data preprocessing applied to the dataset, the method of converting 1D data to 2D data, the basic structure of CNN, and the success evaluation criteria of the model are provided in chapter 3 of the study. Explanation of the dataset used in the study, experimental work steps performed in the study, the CNN model developed for COVID-19 detection, and the experimental results are presented in the 4th chapter. Additionally, the success achieved with the proposed CNN model is compared with popular machine learning (ML) models and studies carried out with the same dataset used in this study in the literature in the 4th chapter. Consequently, in chapter 5, the study is summarized, and the contribution of the study to this field is presented and concluded with recommendations.

2 Related work

Various studies have been carried out on COVID-19 detection using CT and X-ray images in the literature. Jiang et al. conducted a study (Jiang et al. 2020) comparing RT-PCR with CT and X-ray images for COVID-19 detection in 51 patients with severe respiratory and fever symptoms. In this study, the importance of developing automatic detection systems with image data that will be an alternative to the RT-PCR method to prevent the spread of COVID-19 was emphasized by obtaining a sensitivity of 98% of chest CT scans despite the sensitivity of 71% of the RT-PCR test. DarkCovidNet (Ozturk et al. 2020) deep learning (DL) model, which performs automatic COVID-19 detection with X-ray images by Öztürk et al., classifies infected and non-infected individuals and also included individuals with lung inflammation in the classification. As a result, 98.08% success was achieved in binary classification between COVID-19 positive and negative individuals, and 87.02% success in multi-classification between COVID-19 positive and pneumonia disease patients. Xu et al. reported that the RT-PCR test has low accuracy in the early stages of COVID-19 and developed a DL model which distinguishes COVID-19 from flu cases using lung CT scans (Xu et al. 2020). This model has 86.7% accuracy and is based on ResNet. Bassi et al. presented two transfer learning approaches by fine-tuning pre-trained neural networks on ImageNet and using a dataset containing chest X-ray images (Bassi and Attux 2022). With this study, a different perspective was presented by using the learning transfer technique on COVID-19 detection. Haghanifar et al. applied lung segmentation and different image enhancement methods to X-ray image data developed a 431-layered, COVID-CXNet (Haghanifar et al. 2022) DL model using the learning transfer technique and achieved 96.10% success. Kavya et al. realized a study on VGG16 and ResNet50 models using X-ray images of individuals to analyze COVID-19 patients from pneumonia patients, and the success of the models was 89.34% and 91.39%, respectively (Kavya et al. 2022).

When the studies using COVID-19 blood parameters were examined, Brinati et al. applied specific data preprocessing to the dataset with 279 samples and designed two ML models (Brinati et al. 2020) that included all the features and excluded the gender feature. The most successful results for the designed models were obtained with random forest (RF) classifier and 82% and 86%, respectively. In the study performed by Erol et al., various data preprocessing combinations were tried on five different datasets containing COVID-19 blood parameters (Erol et al. 2022). These combinations tried were classified by ensemble ML methods and the most successful data preprocessing-COVID-19 detection models were presented. Using the SPSS program, Sun et al. statistically examined the discrimination of important blood values such as Platelets, Monocytes, and Leukocytes for the disease in COVID-19 diagnosis (Sun et al. 2020). The ERLX (Aljame et al. 2020), a two-level ensemble learning model, was developed for COVID-19 detection by Aljame et al.. After various data preprocessing was applied to the dataset containing 5644 group blood data, first-level classification was made with Extra Trees, RF, and Logistic Regression algorithms. Finally, 99.88% success was achieved by performing second-level classification with XGBoost. Guan et al. conducted an analytical study with clinical blood data of thousand of COVID-19 patients and identified abnormalities in the blood parameters of sick individuals (Guan et al. 2020). In addition, many blood parameter values such as C-reactive proteins, platelets, and white blood cells of infected and uninfected individuals were analyzed comparatively (Ferrari et al. 2020). In the study of Göreke et al., a COVID-19 detection model was developed by taking into account the genetic differences in blood values (Göreke et al. 2021). For this purpose, an algorithm that considers genetic differences and extracts a new feature group based on blood parameters has been developed. Then, a new hybrid classifier model based on DL was designed using this feature set, and COVID-19 detection with 94.95% success was carried out. In the study realized by Rikan et al., eight ML and four DL models developed for COVID-19 detection were evaluated comparatively over three datasets (Rikan et al. 2022). On the other hand, in the study conducted by Abayomi-Alli et al., by using CNN models as first-stage classifiers and 15 ML algorithms as second-stage classifiers, they achieved an average of 99.28% success with the ensemble learning model based on CNN and ExtraTrees (Abayomi-Alli et al. 2022). The ML model named QCovSML (Rahman et al. 2022) was developed by Rahman et al. For COVID-19 detection was run with six different datasets obtained from three countries, and the highest success rate of 92.02% was achieved. Finally, Alakuş and Türkoğlu compared six COVID-19 detection models on a dataset containing 600 sample blood data, and the CNNLSTM (Alakus and Turkoglu 2020) model was chosen as the most successful model with 92.30% success.

3 Proposed approach

In this study, an effective and successful model for COVID-19 detection is proposed by providing a combination of low-cost and rapidly accessible blood parameters and CNN architecture, which is highly successful in image processing. The applied steps in the study are presented in

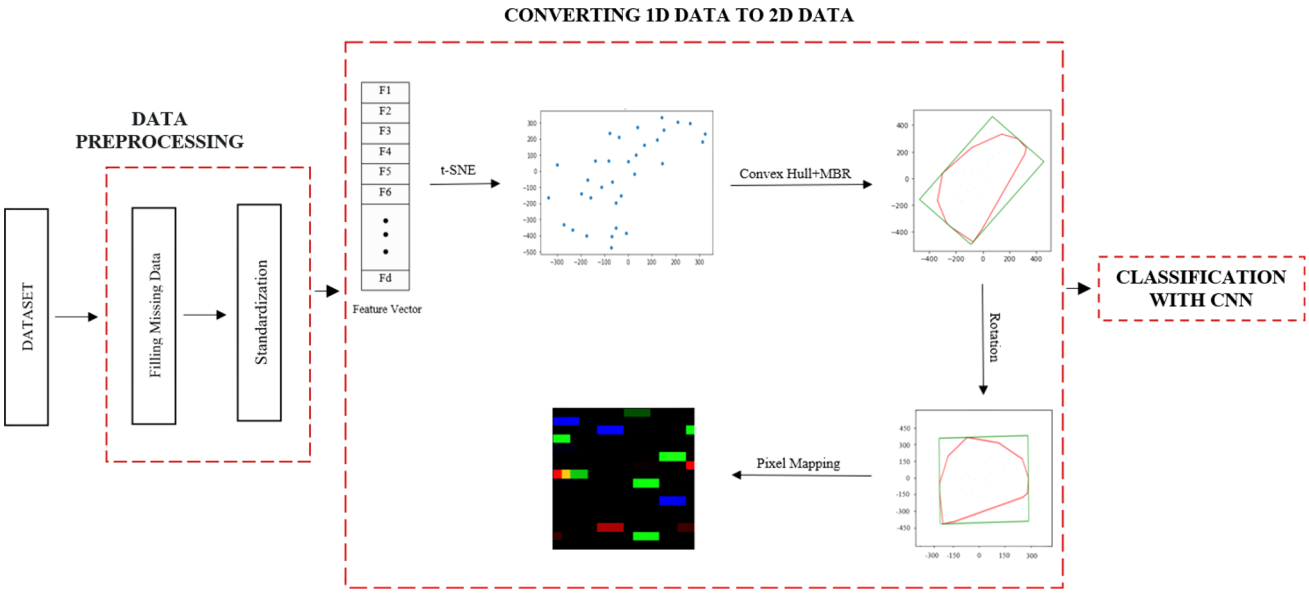


Fig. 1 Applied steps in the study

Fig. 1. If more than 75% of the dataset containing the blood parameters of 1736 COVID-19 positive and negative individuals contains missing data (Cabitza et al. 2020), those samples were deleted then various data preprocessing methods were applied. These methods are: complementing missing data with the K-nearest neighbors (KNN) method (Erol et al. 2022) and standardization in the range of [− 1, 1] with the standard scaler method. Subsequently, the steps of converting 1D data to 2D data (Sharma et al. 2019) were applied: firstly, the t-SNE dimensionality reduction method was used to transfer each feature vector to the 2D plane. Since the image should have frames horizontally or vertically for CNN architecture, the rotation process was applied then image matrices were obtained by pixel mapping. Image data was obtained by converting the obtained image matrices to RGB format. Finally, the image data obtained were classified using CNN.

3.1 Data preprocessing

In this study, the OSR (Cabitza et al. 2020) dataset, which contains information about 1736 patients whose blood samples were taken at the San Raffaele Hospital in Milan, Italy, was used. If more than 75% of the OSR dataset containing the blood parameters of COVID-19 positive and negative 1736 individuals contains missing data, those samples were deleted (Cabitza et al. 2020). Then, the missing data were completed using the KNN method (Erol et al. 2022), and standardization data preprocessing was performed with the standard scaler method in the range [− 1, 1]. Finally, feature vectors were obtained by taking the transpose of the dataset.

3.1.1 Missing data completing with KNN

While completing the missing data with the KNN algorithm, a k value is determined first. The k parameter represents the number of nearest neighbors to the missing data. KNN measures the distance between missing data and non-missing observations around it. With this measurement, the k observations closest to the missing data are found and the missing data is completed with the average of these k variables. In this study, the k value was chosen as 5.

3.1.2 Standardization with StandardScaler

Standardization is the rescaling of value distributions with a mean of 0 and a standard deviation of 1, assuming that the observed values fit the Gaussian distribution (Karlis 2002). In this study, the standardization process was applied to the data in the range of − 1 to 1 with the StandardScaler method.

3.2 Methodology of converting 1D data to 2D data

3.2.1 t-SNE algorithm

t-SNE (Maaten and Hinton 2008) is a nonlinear dimensionality reduction technique used to visualize high-dimensional datasets in the cartesian coordinate system. t-SNE finds low-dimensional representations of multidimensional data by placing similar features together and dissimilar ones further apart in the 2D plane. In the study, 2D plane representations were obtained by applying t-SNE

to the feature vectors in the dataset containing blood parameters.

3.2.2 Convex hull and MBR algorithms

The Convex Hull Algorithm, developed for the problem of constructing the convex polygon containing all points in the cartesian plane and covering the smallest area, obtains the smallest convex polygon covering all the points in the cartesian plane (Barber et al. 1996a; Hershberger and Suri 1992). The convex hull algorithm works in the logic of sorting algorithms and its working principle is briefly as follows: for a dataset P consisting of n data points in the cartesian plane, the cartesian coordinates of these n points are represented as a circularly linked vertex list of data points in convex hull counterclockwise order. So in dataset P, if point i is a data point of convex hull, next[i] is the index of the next data point counterclockwise, and previous[i] is an index of the next data point clockwise. It doesn't matter which corner to keep at the top of the list, and the rule for the connection to be clockwise or counterclockwise is optional. Examining Fig. 2, the data point $q1$ is the first element of the vertex list, next[$q1$] = $q2$, and previous[$q1$] = $q7$.

MBR is the limitation of a given dataset to the smallest rectangle by using the minimum and maximum values of the x and y coordinates (Chaudhuri and Samal 2007). For this rectangle, firstly, the edge list of the convex body is obtained from the corner lists of the data obtained in the coordinate system with the convex hull algorithm. The MBR is obtained for each of the resulting convex edges. In this study, the framing process was carried out by obtaining the MBR with the convex obtained from the feature vector by the convex hull algorithm.

3.2.3 Pixel mapping

Pixel mapping (Sharma et al. 2019) was applied so that the feature vector transferred to the 2D cartesian coordinate could be processed by CNN after framing. The Pixel mapping process consists of 2 steps:

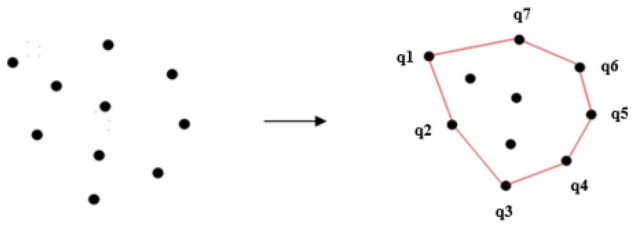


Fig. 2 Convex hull algorithm

1. Coordinate calculation of feature vectors since coordinate values need to be converted to pixel form for image data.
2. Obtaining image data by mapping each feature vector to pixel points obtained by coordinate calculation.

Before the coordinate calculation is made, the feature vector applied t-SNE, convex hull, and MBR is rotated to make it parallel to the x and y axes in the cartesian plane since the image for the CNN architecture must have values horizontally or vertically.

3.3 Classification with CNN

The network model that performs the extraction of the features representing the data within the network is called deep learning. CNN one of the DL architectures developed, is a special artificial neural network architecture developed for visual data processing and brings successful results in a wide range of study areas (LeCun et al. 1989). CNN is a DL architecture that takes the input image and assigns importance (weight and bias) to various objects in the image and enables the objects to be distinguished from each other, as well as inferring the relations of objects with each other. Thanks to its convolution, pooling, fully connected, and classification layers (Jauro et al. 2020), CNN detects distinctive features in images, performs compatibility calculations, and performs artificial learning through a neural network. A basic CNN structure is shown in Fig. 3.

As seen in Fig. 3, CNN has two basic parts: first, convolution and pooling layers, which extract features from the image and analyze these features, second, fully connected layer which takes the data produced by convolution and pooling layers as input and predicts the best class label and classifies them. Although CNN has high success in 2D data like image data (Hakak et al. 2020; Iwendi et al. 2020; Narin et al. 2021), it needs at least two dimensions for convolutional processing so it cannot work with 1D data (Albawi et al. 2017; Jamro and Wiatr 2001; Sharma 2020).

4 Experimental framework and results

4.1 Dataset

The OSR (Cabitza et al. 2020) dataset used in the study contains information on 1736 patients whose blood samples were taken at the San Raffaele Hospital in Milan, Italy. The dataset, which consists of 816 positive and 920 negative patients, includes the age and gender values of each patient and certain blood parameters in routine blood tests. The dataset includes both specific blood parameters of

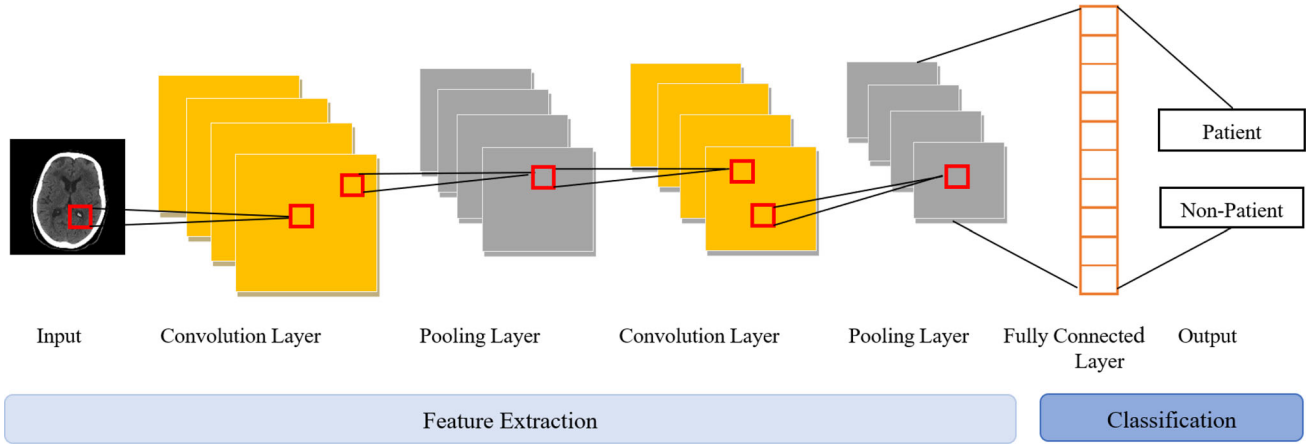


Fig. 3 Basic structure of CNN

COVID-19 and complete blood count (CBC) blood parameters, which are used as a screening test control of many diseases. The features found in the dataset are given in Table 1.

4.2 Data preprocessing

If more than 75% of the OSR dataset containing the blood parameters of COVID-19 positive and negative 1736 individuals contains missing data, those samples were deleted (Cabitza et al. 2020). Then, the missing data were completed using the KNN method (Erol et al. 2022), and standardization data preprocessing was performed with the standard scaler method in the range [− 1, 1]. Finally, as seen in Fig. 4, feature vectors were obtained by taking the transpose of the dataset.

4.3 Converting 1D data to 2D data

4.3.1 Transferring the feature vector to 2D cartesian plane

To convert the dataset to image data, firstly, the feature vectors in the dataset were transferred to the 2D cartesian plane. In the study, this process was carried out by finding representations in the 2D Cartesian plane, with similar features close to each other and dissimilar features far from each other with the t-SNE algorithm. The complexity parameter, which is the basic parameter of t-SNE, represents the estimated number of close neighbors that each feature vector has in the 2D Cartesian plane. The inventors of this method, Maaten and Hinton (2008) suggested that the performance of t-SNE is directly dependent on this parameter and that different complexity values should be analyzed between 5 and 50, with optimum complexity between 5 and 50. In this study, optimum complexity parameter value analysis between 5 and 50 for the t-SNE method was performed and the complexity parameter was chosen as 8. The feature vector was transferred to the 2D cartesian plane with the t-SNE algorithm as in Fig. 5.

4.3.2 Framing the feature vector

Various methods have been developed for the convex hull algorithm, which creates a convex polygon containing all points in the cartesian plane and occupies the smallest area. In this study, the method used for the convex hull algorithm is quickhull (Barber et al. 1996b). quickhull performs the framing process by obtaining the convexity in the cartesian plane of the feature vectors transferred to the 2D plane with t-SNE. The DBG algorithm applied to the feature vectors transferred to the cartesian plane with the t-SNE algorithm is presented in Fig. 6.

Transferred to the 2D Plane with t-SNE.

To obtain image data from feature vectors transferred to 2D cartesian plane with t-SNE, a framing process including all feature points is required. As presented in Fig. 7, the framing process was performed by obtaining the MBR with the convex obtained from the feature vector through the convex hull algorithm in the study.

4.3.3 Converting cartesian coordinates to pixels: pixel mapping

For the feature vector transferred to 2D cartesian coordinate to be processed by CNN after framing and then rotation operations, cartesian coordinate values need to be converted to pixel forms. This conversion process is called the mapping of pixels. Pixel mapping is performed in two steps: coordinate calculation with normalization of feature vectors and generating image data with input matrices. The coordinate calculation was performed as shown in Fig. 8 as follows:

Table 1 Features found in the OSR dataset

Feature	Acronym	Covıd-19 blood parameters	CBC blood parameters
Age	–	X	X
Gender	–	X	X
Leukocytes	WBC	X	X
Red blood cells	RBC	X	X
Hemoglobin	HGB	X	X
Hematocrit	HCT	X	X
Average erythrocyte volume	MCV	X	X
Mean erythrocyte hemoglobin	MCH	X	X
Mean erythrocyte hemoglobin concentration	MCHC	X	X
Platelets	PLT	X	X
Neutrophil count (%)	NE	X	X
Lymphocyte count (%)	LY	X	X
Monocyte count (%)	MO	X	X
Eosinophil count (%)	EO	X	X
Basophil count (%)	BA	X	X
Neutrophil count	NET	X	X
Lymphocyte count	LYT	X	X
Monocyte count	MOT	X	X
Eosinophil count	EOT	X	X
Basophil count	BAT	X	X
Glucose	GLU	X	
Creatinine	CREA	X	
Urea	UREA	X	
Alanine aminotransferase	ALT	X	
Aspartate aminotransferase	AST	X	
Alkaline phosphatase	ALP	X	
Gamma glutamyl transferase	GGT	X	
Lactate dehydrogenase	LDH	X	
Creatine kinase	CK	X	
Calcium	CA	X	
C-reactive protein	CRP	X	
Kell antigen typing	KAT	X	
Nucleic acid amplification test	NAAT	X	

(a) Performing min–max normalization for both the x and y coordinates of each feature vector.
(b) Ensuring the coordinate values of the features take values in the appropriate range for the pixel domain by multiplying the normalized x and y coordinate values with the determined image pixel size. In the study, the pixel size was determined as 16×16.
(c) Assigning the exact parts of the obtained comma coordinate values as pixel values.

By coordinate calculation, pixel_x and pixel_y values were obtained corresponding to x and y cartesian coordinate values for each feature vector. After the coordinate calculation, an input matrix is created with the pixel_x and pixel_y values of the feature vectors, as seen in Fig. 9 performed with feature vector 1. Using this input matrix, the values of each feature vector are mapped to the *pixel_x* and *pixel_y* locations where the feature vector is located, and the image matrix is obtained. The process of obtaining the image matrix is presented schematically in Fig. 9 over the feature vector 1. With pixel mapping, the pixels of the image matrices have a brightness value in the range of 0–255.

4.4 Proposed 3-line CNN model

In this study, COVID-19 was detected in blood parameters. Initially, if more than 75% of the OSR dataset containing blood parameters in COVID-19 positive and negative 1736 cells contains missing data, those images were deleted (Cabitza et al. 2020), missing data were completed using the KNN method (Erol et al. 2022), and the standard scaler method [− 1, 1] comprehensive standardization data preprocessing was done. Then, the dataset is 80% training set and 20% test set; 10% of the training set is also divided as the validation set. Image matrices were created by applying t-SNE, Convex Hull, MBR algorithms, and pixel mapping methods to each set, and image data was obtained by converting these image matrices to RGB format. The obtained image data were classified with the proposed 3-line CNN model. The training, validation, and test sets that were preprocessed but not converted into image data were subjected to classification with ML classifiers.

CNN is widely used in image processing, usually extracting features from 2D data (Albawi et al. 2017; Astanin et al. 2016). In this study, 1673 2D image data were obtained by applying t-SNE, convex hull, MBR, pixel mapping, and RGB format conversion to 1D COVID-19 blood parameters. Of these 1673 image data, 1204 are of the training set, 124 of the validation set, and 335 of the test set. And a 3-line CNN architecture presented in Fig. 10 was developed to classify these image data. The first line of the proposed 3-line CNN architecture performs intensive feature extraction from images thanks to multiple convolution and pooling layers. The second line obtains the final features of the dataset by making the extracted features from the first line more stable through various layers.

	Sex (Feature 1)	Age (Feature 2)	WBC (Feature 3)	RBC (Feature 4)	HGB (Feature 5)	...	NAT (Feature 33)
Sample 1	0.8661	1.0931	-0.2662	-0.2155	-0.0119		-0.4656
Sample 2	0.8661	-0.5225	-0.5198	0.2815	-0.1970		-0.1928
⋮							⋮
Sample 1673	0.8661	-0.4287	-0.1605	0.8321	-0.1250	...	0.0772

(a)

	Sample 1	Sample 2	...	Sample 1673
Sex (Feature 1)	0.8661	0.8661		0.8661
Age (Feature 2)	1.0931	-0.5225		-0.4287
WBC (Feature 3)	-0.2662	-0.5198		-0.1605
RBC (Feature 4)	-0.2155	0.2815		0.8321
HGB (Feature 5)	-0.0119	-0.1970		-0.1250
⋮				⋮
NAT (Feature 33)	-0.4656	-0.1928	...	0.0772
	Feature Vector 1			Feature Vector 1673

(b)

Fig. 4 **a** Dataset **b** feature vectors obtained by transposing the dataset

Finally, the classification process is realized thanks to the fully connected and sigmoid layers in the third line.

Convolution, batch normalization, and ReLU block structure used in the proposed model and presented in Fig. 11 are inspired by the Identity block structure developed in the ResNet (He et al. 2015) architecture. This block structure is designed to normalize and activate the extracted features in the convolution layer. The normalization process is provided by the batch normalization (Loffe and Szegedy 2015) layer, and the activation process is provided by the ReLu layer. As in Fig. 11, using batch normalization after each convolution operation, the inputs to be activated are rescaled. This process made the model more stable. The normalized model was activated with the ReLu activation function. This block structure forms the basic structure of feature extraction, which is the first line of the proposed 3-line CNN model.

The first line of the proposed 3-line CNN model performs intensive feature extraction. As shown in Fig. 12, a max pooling layer comes after convolution, batch normalization, and ReLU block structure. The max pooling layer halves the width and height dimensions of the feature map with a 2×2 filter. After pooling, the feature map is again provided as input to the convolution, batch normalization, and ReLU block structure. At this stage of the model, the Dropout (Srivastava et al. 2014) layer is added in addition to the block structure. Dropout is a layer that prevents overfitting by reducing some connections within the network. In the proposed model, the disadvantages arising from the limited number of data are prevented by adding the dropout layer to the basic block structure, which is used iteratively. Finally, the features of images transformed from 1D data are determined by applying an activated convolution operation. The basic model steps given in Fig. 12 were continued with various combinations to create the first line with the highest success.

The second line of the proposed 3-line CNN model performs the final feature extraction by applying

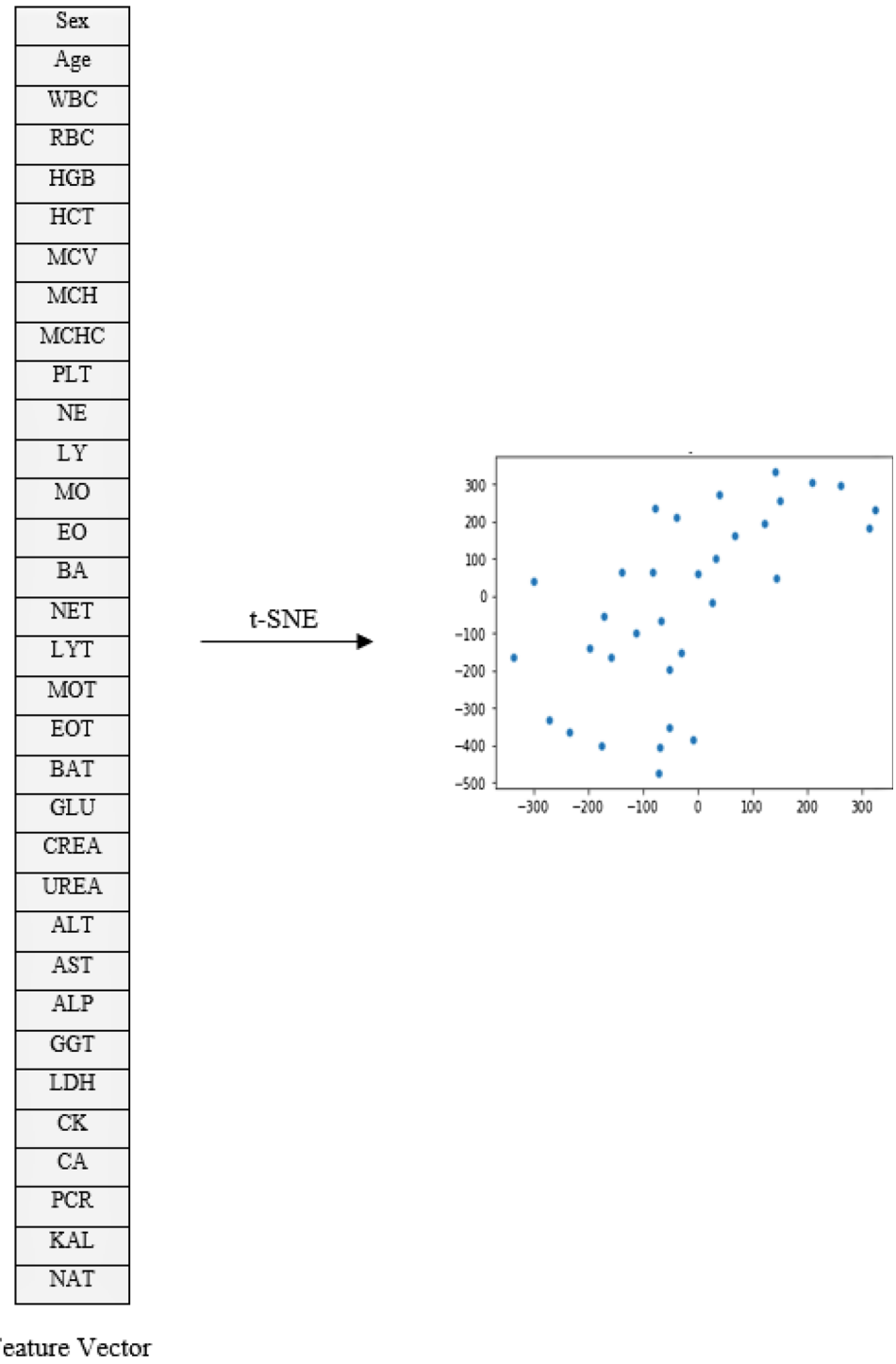


Fig. 5 Transfer to the 2D plane by applying t-SNE to feature vector in OSR dataset

convolution to the feature map it takes as input from the first line. And the second line of the proposed 3-line CNN model is terminated by the ReLu layer, after the adaptive average pooling (AAP) layer. AAP is a pooling operation that calculates the correct kernel size required to produce a given size output from a given input and has been applied to the second line of the proposed model, inspired by its use in the GoogLeNet (Szegedy et al. 2014) architecture. The third and last line of the proposed 3-line CNN model performs classification with a fully connected layer and sigmoid activation function.

With the proposed 3-line CNN, a training model was created using 1204 data in the train set, and the performance of this training model was improved with 124 data in the validation set. The performance obtained as a result of 100 epochs gave the success of the training set. After the

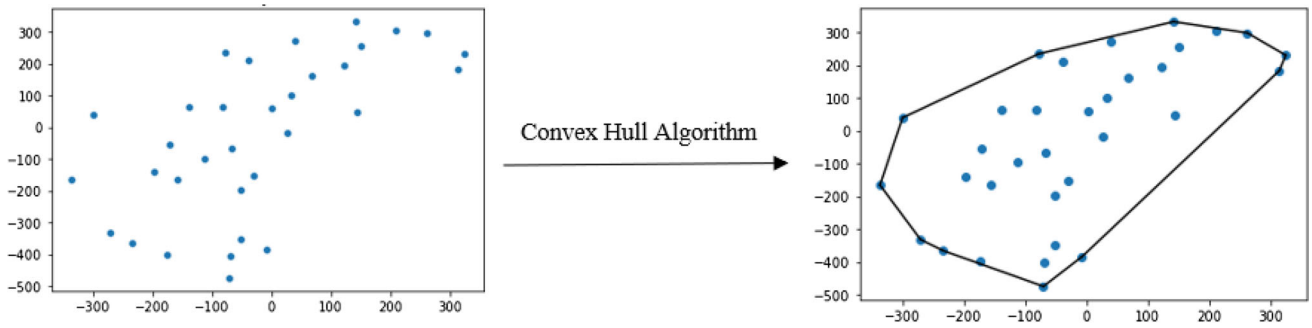


Fig. 6 Obtaining the convex with convex hull algorithm of the feature vector

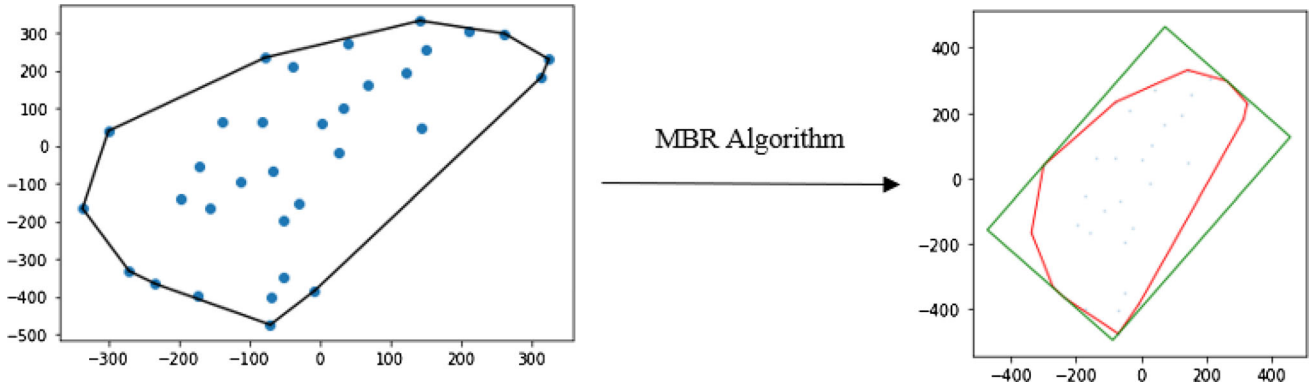
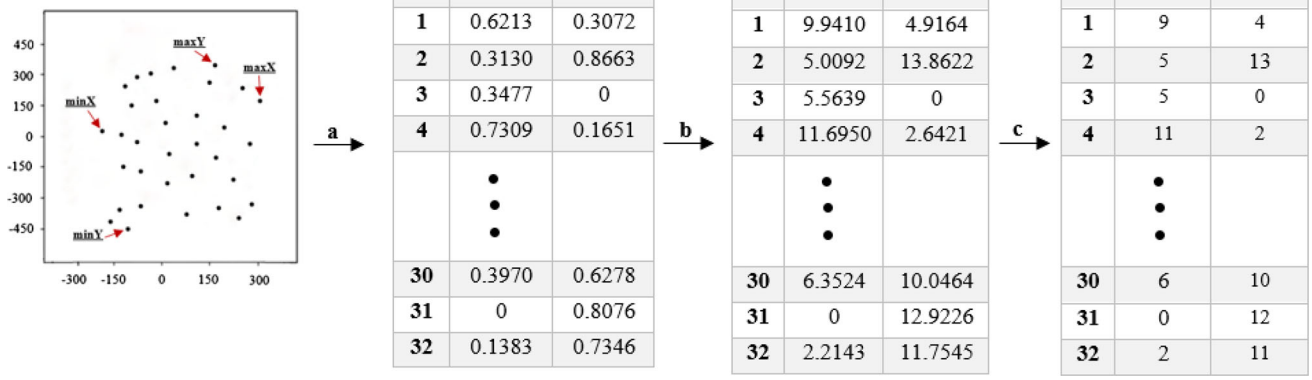


Fig. 7 Framing with MBR of the feature vector obtained the convex with convex hull algorithm



	x	y
0	0.8975	0.0207
1	0.6213	0.3072
2	0.3130	0.8663
3	0.3477	0
4	0.7309	0.1651
⋮	⋮	⋮
30	0.3970	0.6278
31	0	0.8076
32	0.1383	0.7346

	x	y
0	14.3615	0.3323
1	9.9410	4.9164
2	5.0092	13.8622
3	5.5639	0
4	11.6950	2.6421
⋮	⋮	⋮
30	6.3524	10.0464
31	0	12.9226
32	2.2143	11.7545

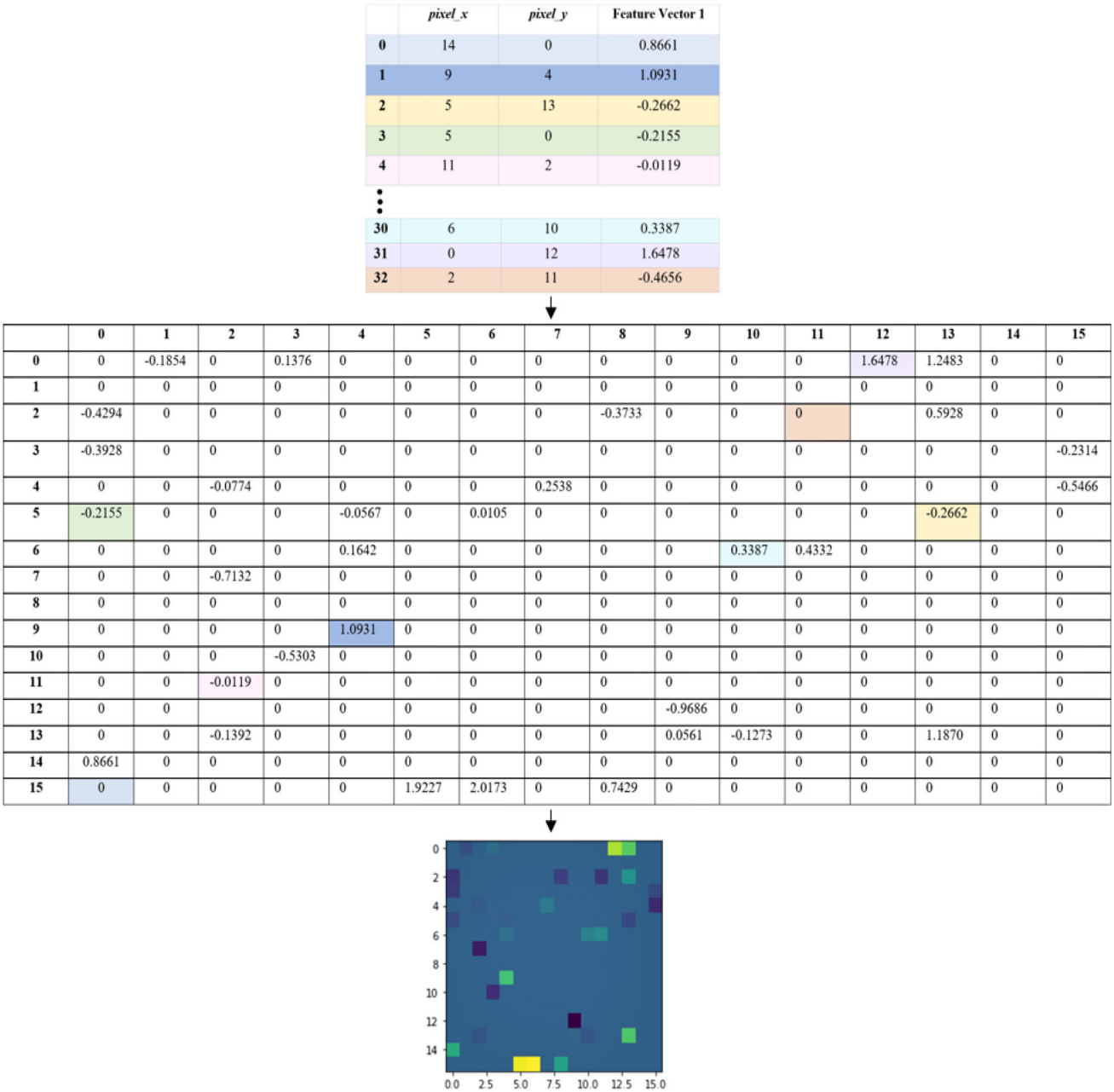
	x	y
0	14	0
1	9	4
2	5	13
3	5	0
4	11	2
⋮	⋮	⋮
30	6	10
31	0	12
32	2	11

Fig. 8 Normalization and coordinate calculation of feature vector in 2D plane

success of the training set was achieved, 335 images in the test set were presented to the proposed 3-line CNN model on this training model, and the test processes were applied. As a result of the test procedures, 94.85% success was achieved.

4.5 Experimental results

The classification performance of the CNN model developed in this study was evaluated with accuracy, precision, recall, specificity, F1-score, and AUC (Alazab 2020; Zhao et al. 2017) metrics. These metrics are calculated with true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) scores. TP and TN represent cases where the model correctly predicts positive and negative classes, respectively. In contrast, FP and FN represent cases where the model incorrectly predicted positive and negative classes, respectively. Accordingly, the formulations of accuracy, precision, recall, and F1-score are given by Eqs. 1–5. AUC is the probability measure of how



	pixel_x	*pixel_y*	Feature Vector 1
0	14	0	0.8661
1	9	4	1.0931
2	5	13	-0.2662
3	5	0	-0.2155
4	11	2	-0.0119
⋮			
30	6	10	0.3387
31	0	12	1.6478
32	2	11	-0.4656

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	-0.1854	0	0.1376	0	0	0	0	0	0	0	0	1.6478	1.2483	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-0.4294	0	0	0	0	0	0	0	-0.3733	0	0	0		0.5928	0	0
3	-0.3928	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.2314
4	0	0	-0.0774	0	0	0	0	0.2538	0	0	0	0	0	0	0	-0.5466
5	-0.2155	0	0	0	-0.0567	0	0.0105	0	0	0	0	0	0	-0.2662	0	0
6	0	0	0	0	0.1642	0	0	0	0	0	0.3387	0.4332	0	0	0	0
7	0	0	-0.7132	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1.0931	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	-0.5303	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	-0.0119	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0		0	0	0	0	0	0	-0.9686	0	0	0	0	0	0
13	0	0	-0.1392	0	0	0	0	0	0	0.0561	-0.1273	0	0	1.1870	0	0
14	0.8661	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	1.9227	2.0173	0	0.7429	0	0	0	0	0	0	0

Fig. 9 Obtaining image data with input matrices

successfully the two classes can be separated and represents the area under the ROC curve. The *x*-axis of the ROC curve represents the FP ratio, and the *y*-axis represents the TP ratio. As the one under the curve increases, the FP ratio decreases, and the TP ratio increases, so the discrimination performance between classes increases.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

$$F1 - \text{Score} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{5}$$

In Table 2, the success of the proposed 3-line CNN model, which takes the OSR dataset converted to image

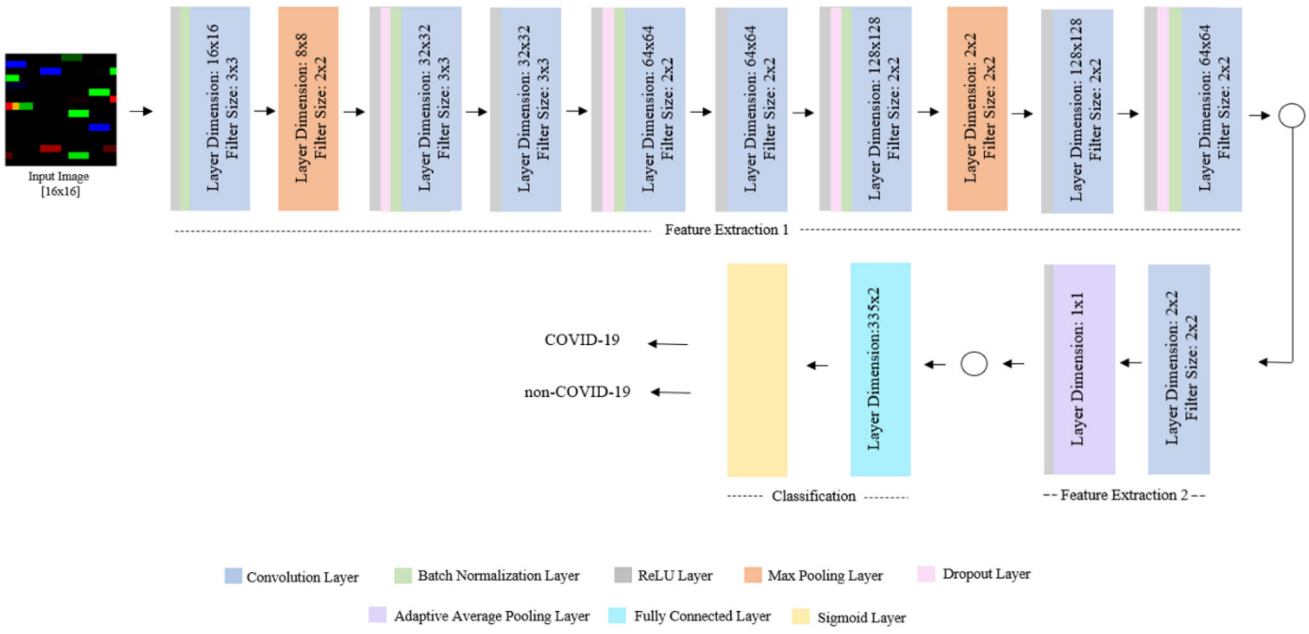


Fig. 10 Proposed 3-line CNN model

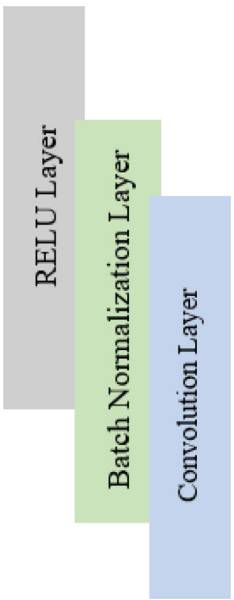


Fig. 11 Convolution, batch normalization, and ReLU block structure of the proposed 3-line CNN model

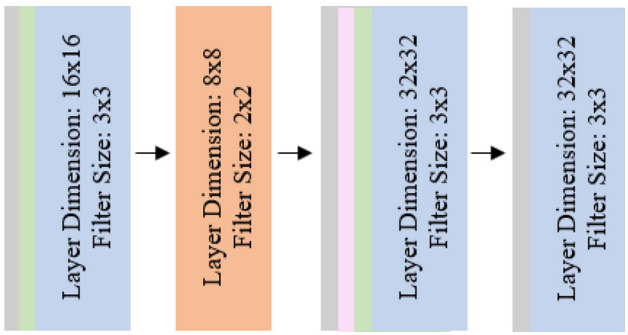


Fig. 12 Basic steps of the first line of the proposed 3-line CNN model

data as input, is compared with the success of the ANN, bagging, adaboost, and RF ML classifiers, which takes the original version of the OSR dataset with data preprocessing as input. The hyperparameters of these classifiers were determined as the values that reached the most successful result in the training model. With these values, a test model was created over the training model created for each classifier.

The ML methods with the most successful classification with the OSR dataset used in this study were compared with the 3-line CNN method proposed in Table 2. As can be seen in Table 2, the ANN model, which took the original version of the OSR dataset with data preprocessing as input, achieved 75.52% success, whereas the 3-line CNN model, which converted the OSR dataset into image data, achieved 94.85% success. This demonstrated the success of converting 1D blood parameter data to 2D image data and using it in CNN models. Ensemble learning methods produce more successful results than other ML techniques (Wiysobunri et al. 2020). 81.49%, 78.50%, and 81.19% success was achieved with bagging, adaboost, and RF classifiers, which are the ensemble methods that take the original form of the OSR dataset as input, respectively. As a result of these achievements, the success of the proposed 3-line CNN model in the study was found to be quite successful than the ensemble methods. Precision shows how many of the values we predicted as positive are positive. Recall shows how much of what we should have predicted positively was predicted positively. Specificity is the metric that shows how much of what we should have predicted negatively was predicted negatively. As can be

seen in Table 3, the proposed 3-line CNN model was more successfully identified than the ML models that predicted positively, should be predicted positively, and should be predicted negatively. When the AUC values of the classifiers are examined, it is seen that the AUC value of the proposed 3-line CNN model is the highest. This again proved that the most successful model distinguishing between positive and negative classes is the proposed 3-line CNN model. In addition, although the other highest AUC value was obtained with RF, the most successful model among ML models was bagging. This situation has revealed the necessity of examining many performance criteria in the detection models developed. TP, TN, FP, and FN values of the proposed 3-line CNN model, and these ML models are presented in Table 3.

The most important and primary case in COVID-19 detection is the correct identification of infected patients. This case is represented by the TP in the confusion matrix. In other words, it is the positive detection of the infected patient in the detection system. On the contrary, the situation that makes it difficult to control the epidemic is that the infected patient is detected as negative in the detection system. This is represented as FN in the confusion matrix. These two evaluation criteria should be considered in the developed COVID-19 systems. As can be seen in Table 3, 162 of the 335 test data forming the classifier models have positive class labels and 173 have negative class labels. The proposed 3-line CNN model from 162 positive patients correctly labeled 151 and mislabeled 11 of them. This is highly accurate when compared to the labeling of ML models. On the other hand, when an evaluation is made between ML models in terms of high TP and low FN ratio; Adaboost was found to be more accurate than the others. But looking at Table 2, adaboost's success is 78.50, which is lower than bagging and RF. This situation has revealed the necessity of mutual examination of all evaluation criteria in the detection models developed.

The success of the proposed 3-line CNN model and the success of other studies in the literature conducted with the OSR dataset used in the study are presented in Table 4.

Table 4 shows the results of other studies done with the dataset used in this study, and when it is examined, Cabitza et al., two separate datasets were obtained from the OSR dataset, the dataset containing the blood parameters specific to COVID-19 and the dataset containing the CBC blood parameters. Then, various ML models were evaluated with these two datasets and the OSR dataset. As a result, it achieved success between 88.00 and 93.00% with the best RF model. In the study performed by Rikan et al., eight ML and four DL models were developed for COVID-19 detection with three datasets containing blood parameters (Rikan et al. 2022). Among the developed models, the highest successes were achieved with Extra Tree, which is an ML model, at 84.98%, and with DNN, a DL model, at 93.16%. When these achievements are examined, it has been proven that the DL method for the OSR dataset brings more successful results than the ML method. The ML model named QCovSML (Rahman et al. 2022) developed by Rahman et al. for COVID-19 detection is an ML model containing Gradient Boosting, RF, and XGBoost classifiers. This stack model, developed with three classifiers, achieved 91.45% success and it was found more successful than other ML models developed for COVID-19 detection. Finally, in the study of Kistenev et al., 88.00% success was obtained from the OSR dataset with RF and SVM classifiers (Kistenev et al. 2022). As a result, the success of the 3-line CNN model, which classifies the OSR dataset by transforming it into image data, produced more successful results than the results of other studies using the OSR dataset. The fact that the proposed 3-line CNN model was more successful against other COVID-19 detection studies using the same dataset revealed the applicability of the DeepInsight (Sharma et al. 2019) method to blood parameter data and its usability in the COVID-19 detection with blood parameters. The 3-line CNN model developed in this study has proven effective in artificial intelligence-

Table 3 TP, TN, FP, and FN values of the proposed 3-line CNN model and popular ML models

Classifier	TP	TN	FP	FN
Proposed 3-line CNN	151	167	6	11
ANN	122	131	42	40
Bagging	129	144	29	33
Adaboost	132	131	42	30
RF	131	141	32	31

Table 2 Comparison of the success of the proposed 3-line CNN model with popular ML models

Classifier	Accuracy	Precision	Recall	Specificity	F1-score	AUC
Proposed 3-line CNN	94.85	95.92	93.27	95.10	94.58	0.9570
ANN	75.52	75.60	75.50	75.60	75.50	0.8190
Bagging	81.49	81.50	81.50	81.40	81.50	0.8780
Adaboost	78.50	78.70	78.50	78.70	78.50	0.8570
RF	81.19	81.20	81.20	81.20	81.20	0.8800

Table 4 Comparison of proposed 3-line CNN success with other COVID-19 studies using OSR dataset

Study	Classifier	Accuracy
Proposed 3-line CNN	CNN	94.58%
Cabitza et al. (2020)	RF	88.00–93.00%
Rikan et al. (2022)	Extra Tree	84.98%
	DNN	93.16%
Rahman et al. (2022)	Stacking Model (Gradient Boosting + RF + XGBoost)	91.45%
Kistenev et al. (2022)	RF and SVM	88.00%

based COVID-19 detection studies by achieving high success against many ML models and CNN models.

When Table 4 is examined, it is seen that the most successful results in classification studies using the OSR dataset are obtained when the data is classified using ensemble learning methods and deep neural network algorithms after various preprocessing. The CNN model, which provides high success in image data by converting the recommended 1D blood parameters to 2D image data, has achieved more successful results than the methods in the literature. Standard ML methods require additional efforts, such as feature selection, to identify features that contribute to success in the dataset. In DL methods, according to the structure of the data, which feature in the dataset should be given weight is discovered in the network. This allows us to achieve more successful results than ML methods without additional effort.

5 Conclusion

In this study, the proposed data preprocessing for blood parameters consisting of 1D data, the conversion of 1D data to 2D, and the success of the combination of 3-line CNN in COVID-19 detection are presented. It has been observed that high success rates can be achieved in COVID-19 detection with an appropriate CNN model by converting blood test results to image data for COVID-19 diagnosis. In this study, if more than 75% of the OSR dataset containing the blood parameters of COVID-19 positive and negative 1736 individuals had missing data, those samples were deleted (Cabitza et al. 2020), missing data were completed with the KNN method (Erol et al. 2022), and standardization was performed with the standard scaler method in the range [− 1, 1]. Afterward, the steps of converting 1D data to 2D data (Sharma et al. 2019) were applied: first, each feature vector was transferred to the 2D plane with the t-SNE algorithm. The feature vectors transferred to the 2D plane to form the boundaries of the image data are framed by convex hull and MBR algorithms. Since the image should have frames horizontally or vertically for CNN architecture, pixel mapping was done after the rotation process. As a result of pixel mapping, image matrices were obtained, and image data was obtained by converting the obtained image matrices to RGB format. Experimental studies were carried out by presenting the obtained images to the proposed 3-line CNN architecture and 94.85% success was achieved. This achievement was analyzed in comparison with ensemble ML methods that took the original form of the OSR dataset as input and other COVID-19 studies in the literature carried out with the OSR dataset. As a result of the analysis, the applicability of the proposed method to blood parameter data and its usability in COVID-19 detection with blood parameters were revealed. This study provides a guide for COVID-19 detection by presenting a combination of blood parameters that are easy to access and low-cost and a high-success CNN architecture. In future studies, blood parameters converted to image data and CT images can be comparatively evaluated with large datasets obtained by collecting both CT images and blood parameters of patients. More successful COVID-19 detections can be made. In addition, the presented method can be run with many popular combinations of CNN architectures, resulting in improved systems that are more powerful and adaptable to many disease detections. Finally, various special epidemic detection models can be developed by making necessary updates for other epidemic diseases similar to COVID-19.

Funding No source of funding was used in this study

Data availability The data that support the findings of this study are available in reference (Cabitza et al. 2020).

Declarations

Conflict of interest No conflict of interest in this study.

References

Abayomi-Alli OO, Damaševicius R, Maskeliunas R, Misra S (2022) An ensemble learning model for COVID-19 detection from

blood test samples. Sensors 22(6):2224. https://doi.org/10.3390/s22062224

Alakus TB, Turkoglu I (2020) Comparison of deep learning approaches to predict COVID-19 infection. Chaos Solitons Fractals 140(11):110120. https://doi.org/10.1016/j.chaos.2020.110120

Alazab M (2020) Automated malware detection in mobile app stores based on robust feature generation. Electronics 9(3):435. https://doi.org/10.3390/electronics9030435

Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology (ICET). https://ieeexplore.ieee.org/document/8308186

Alexandridi M, Mazej J, Palermo E, Hiscott J (2022) The coronavirus pandemic-2022: viruses, variants and vaccines. Cytokine Growth Factor Rev 63:1–9. https://doi.org/10.1016/j.cytogfr.2022.02.002

Aljame M, Ahmad I, Imtiaz A, Mohammed A (2020) Ensemble learning model for diagnosing COVID-19 from routine blood tests. Inform Med Unlocked 21:100449. https://doi.org/10.1016/j.imu.2020.100449

Alkhodari M, Khandoker AH (2022) Detection of COVID-19 in smartphone-based breathing recordings: a pre-screening deep learning tool. PLoS ONE 17(1):e0262448. https://doi.org/10.1371/journal.pone.0262448

Astanin S, Antonelli D, Chiabert P (2016) Optimal selection of the workpiece recognition parameters by minimizing the total error cost. IFAC-PapersOnLine 49(12):1424–1429. https://doi.org/10.1016/j.ifacol.2016.07.770

Barber CB, Dobkin DP, Hundanpaa H (1996) The quickhull algorithm for convex hulls. ACM Trans Math Softw 22(4):469–483. https://doi.org/10.1145/235815.235821

Barton MI, MacGowan SA, Kutuzov MA et al (2021) Effects of common mutations in the SARS-CoV-2 Spike RBD and its ligand, the human ACE2 receptor on binding affinity and kinetics. Elife. https://doi.org/10.7554/eLife.70658

Bassi PRAS, Attux R (2022) A deep convolutional neural network for COVID-19 detection using chest X-rays. Res Biomed Eng 38:139–148. https://doi.org/10.1007/s42600-021-00132-9

Brinati D, Campagner A, Ferrari D, Locatelli M, Banfi G, Cabitza F (2020) Detection of COVID-19 infection from routine blood exams with machine learning : a feasibility study. J Med Syst 44:135. https://doi.org/10.1007/s10916-020-01597-4

Cabitza F, Campagner A, Ferrari D et al (2020) Development, evaluation and validation of machine learning models for COVID-19 detection based on routine blood tests. Clin Chem Lab Med 59(2):421–431. https://doi.org/10.1515/cclm-2020-1294

Chaudhuri D, Samal A (2007) A simple method for fitting of bounding rectangle to closed regions. Pattern Recognit 40(7):1981–1989. https://doi.org/10.1016/j.patcog.2006.08.003

D'Angelo G, Palmieri F (2021) Enhancing COVID-19 tracking apps with human activity recognition using a deep convolutional neural network and HAR-images. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05913-y

Day M (2020) Covid-19: identifying and isolating asymptomatic people helped eliminate virus in Italian village. BMJ. https://doi.org/10.1136/bmj.m1165

Demirdal T, Sen P (2018) The significance of neutrophil-lymphocyte ratio, platelet-lymphocyte ratio and lymphocyte-monocyte ratio in predicting peripheral arterial disease, peripheral neuropathy, osteomyelitis and amputation in diabetic foot infection. Diabetes Res Clin Pract 144:118–125. https://doi.org/10.1016/j.diabres.2018.08.009

Dey N, Mishra R, Fong SJ, Santosh KC, Tan S, Crespo RG (2020) COVID-19: psychological and psychosocial impact, fear, and passion. Digit Gov Res Pract 1:1–4. https://doi.org/10.1145/3428088

Erol G, Uzbaş B, Yücelbaş C, Yücelbaş Ş (2022) Analyzıng the effect of data pre-processing techniques using machine learning algorithms on the diagnosis Of COVID-19. Concurr Comput Pract Exp. https://doi.org/10.1002/cpe.7393

Ferrari D, Motta A, Strollo M, Banfi G, Locatelli M (2020) Routine blood tests as a potential diagnostic tool for COVID-19. Clin Chem Lab Med 58(7):1095–1099. https://doi.org/10.1515/cclm-2020-0398

Fong S, Li G, Dey N, Crespo RG, Herrera-Viedma E (2020) Finding an accurate early forecasting model from small dataset: A case of 2019-nCoV novel coronavirus outbreak. Int J Interact Multimed Artif Intell 6(1):1–10. https://doi.org/10.9781/ijimai.2020.02.002

Göreke V, Sarı V, Kockanat S (2021) A novel classifier architecture based on deep neural network for COVID-19 detection using laboratory findings. Appl Soft Comput 106(1):107329. https://doi.org/10.1016/j.asoc.2021.107329

Guan W, Ni Z, Hu Y, Liang W, Ou C, He GJ et al (2020) Clinical characteristics of coronavirus disease 2019 in China. Ann Palliat Med 9(4):1404–1412. https://doi.org/10.21037/apm-20-887

Haghanifar A, Majdabadi MM, Choi Y, Deivalakshmi S, Ko S (2022) COVID-CXNet: detecting COVID-19 in frontal chest X-ray images using deep learning. Multimed Tools Appl 7:1–31. https://doi.org/10.1007/s11042-022-12156-z

Hakak S, Khan WZ, Imran M, Choo KKR, Shoaib M (2020) Have you been a victim of COVID-19-related cyber incidents? Survey, taxonomy, and mitigation strategies. IEEE Access 30(8):124134–124144. https://doi.org/10.1109/ACCESS.2020.3006172

Harvey WT, Carabelli AM, Jackson B et al (2021) SARS-CoV-2 variants, spike mutations and immune escape. Nat Rev Microbiol 19(7):409–424. https://doi.org/10.1038/s41579-021-00573-0

He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. Comput vis Pattern Recognit. https://doi.org/10.48550/arXiv.1512.03385

Hershberger J, Suri S (1992) Applications of a semi-dynamic convex hull algorithm. BIT Numer Math 32:249–267. https://doi.org/10.1007/3-540-52846-6_106

Iwendi C, Bashir AK, Peshkar A, Sujatha R, Chatterjee JM et al (2020) COVID-19 patient health prediction using boosted random forest algorithm. Front Public Health 3(8):357. https://doi.org/10.3389/fpubh.2020.00357

Jamro E, Wiatr K (2001) FPGA implementation of addition as a part of the convolution. In: Proceedings euromicro symposium on digital systems design. https://doi.org/10.1109/DSD.2001.952368

Jan HC, Yang WH, Ou CH (2019) Combination of the preoperative systemic immune-inflammation index and monocyte-lymphocyte ratio as a novel prognostic factor in patients with upper-tract urothelial carcinoma. Ann Surg Oncol 26(2):669–684. https://doi.org/10.1245/s10434-018-6942-3

Jauro F, Chiroma H, Gital AY, Almutairi M, Abdulhamid SM, Abawajy JH (2020) Deep learning architectures in emerging cloud computing architectures: recent development, challenges and next research trend. Appl Soft Comput 96:106582. https://doi.org/10.1016/j.asoc.2020.106582

Jiang F, Deng L, Zhang L, Cai Y, Cheung CW, Xia Z (2020) Review of the clinical characteristics of coronavirus disease 2019 (COVID-19). J Gen Intern Med 35(5):1545–1549. https://doi.org/10.1007/s11606-020-05762-w

Karlis D (2002) An EM type algorithm for maximum likelihood estimation of the normal-inverse gaussian distributions. Stat Probab Lett 57(1):43–52. https://doi.org/10.1016/S0167-7152(02)00040-8

Kavya NS, Shilpa T, Veeranjaneyulu N, Priya DD (2022) Detecting Covid19 and pneumonia from chest X-ray images using deep convolutional neural networks. Mater Today Proc 64:737–743. https://doi.org/10.1016/j.matpr.2022.05.199

Kistenev YV, Vrazhnov DA, Shnaider EE, Zuhayri H (2022) Predictive models for COVID-19 detection using routine blood tests and machine learning. Heliyon 8(10):e11185. https://doi.org/10.1016/j.heliyon.2022.e11185

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hub-bard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. https://doi.org/10.1162/neco.1989.1.4.541

Li Z, Yi Y, Luo X, Xiong N et al (2020) Development and clinical application of a rapid IgM-IgG combined antibody test for SARS-CoV-2 infection diagnosis. J Med Virol 92(9):1518–1524. https://doi.org/10.1002/jmv.25727

Loffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing ınternal covariate shift. Mach Learn. https://doi.org/10.48550/arXiv.1502.03167

Menni C, Valdes AM, Freidin MB et al (2020) Real-time tracking of selfreported symptoms to predict potential COVID-19. Nat Med 26(7):1037–1040. https://doi.org/10.1038/s41591-020-0916-2

Narin A, Kaya C, Pamuk Z (2021) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Anal Appl 24(3):1207–1220. https://doi.org/10.1007/s10044-021-00984-y

Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med 121:103792. https://doi.org/10.1016/j.compbiomed.2020.103792

Paules CI, Marston HD, Fauci AS (2020) Coronavirus infections more than just the common cold. JAMA 323(8):707–708. https://doi.org/10.1001/jama.2020.0757

Rahman T, Khandakar A, Abir FF, Faisal MAA et al (2022) QCovSML: a reliable COVID-19 detection system using CBC biomarkers by a stacking machine learning mod. Comput Biol Med 143:105284. https://doi.org/10.1016/j.compbiomed.2022.105284

Rikan SB, Azar AS, Ghafari A, Mohasefi JB, Pirnejad H (2022) COVID-19 diagnosis from routine blood tests using artificial intelligence techniques. Biomed Signal Process Control 72(A):103263. https://doi.org/10.1016/j.bspc.2021.103263

Sharma A, Vans E, Shigemizu D, Boroevich KA, Tsunoda T (2019) DeepInsight: a methodology to transform a non-image data to an image for convolution neural network architecture. Nature Sci Rep 9(1):1–7. https://doi.org/10.1038/s41598-019-47765-6

Sharma A (2020) Non-image data classification with convolutional neural networks. https://arxiv.org/pdf/2007.03218.pdf

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Sun S, Cai X, Wang H, He G, Lin Y, Lu B et al (2020) Abnormalities of peripheral blood system in patients with COVID-19 in Wenzhou, China. Clin Chim Acta 507:174–180. https://doi.org/10.1016/j.cca.2020.04.024

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. Comput vis Pattern Recognit. https://doi.org/10.48550/arXiv.1409.4842

Van der Maaten L, Hinton G (2008) Viualizing data using t-SNE. J Mach Learn Res 9(86):2579–2605

Wiysobunri B, Erden H, Toreyin B (2020) An ensemble deep learning system for the automatic detection of COVID-19 in X-ray images. https://spacing.itu.edu.tr/pdf/beltus-ytb.pdf

World Health Organization (WHO) (2019) Report of the WHO-China joint mission on coronavirus disease 2019 (COVID-19). https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf. Accesed 26 June 2022

World Health Organization (WHO) (2020) Health topics, coronavirus. https://www.who.int/health-topics/coronavirus#tab=tab_1. Accessed 26 June 2022

World Health Organization (WHO) (2022) WHO coronavirus (COVID-19) dashboard. https://covid19.who.int/. Accessed 26 June 2022

Xu X, Jiang X, Ma C, Du P, Li X, Lv S et al (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. https://arxiv.org/abs/2002.09334

Zannella C, Chianese A, Greco G, Santella B, Squillaci G, Monti A, Doti N, Sanna G, Manzin A, Morana A, De Filippis A, D'Angelo G, Palmieri F, Franci G, Galdiero M (2022) Design of three residues peptides against SARS-CoV-2 infection. Viruses 14(10):2103. https://doi.org/10.3390/v14102103

Zhao B, Lu H, Chen S, Liu J, Wu DJ (2017) Convolutional neural networks for time series classification. Syst Eng Electron 28(1):162–169. https://doi.org/10.21629/JSEE.2017.01.18