

RESEARCH ARTICLE

COVID-19 diagnosis from CT scans and chest X-ray images using low-cost Raspberry Pi

Khalid M. Hosny^{1*}, Mohamed M. Darwish², Kenli Li^{3*}, Ahmad Salah^{1,3}

1 Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt, **2** Faculty of Computers and Informatics, Assiut University, Assiut, Egypt, **3** College of Computer Science and Electrical Engineering, Hunan University, Changsha, China

* k_hosny@yahoo.com (KMH); lkl@hnu.edu.cn (KL)



Abstract

The diagnosis of COVID-19 is of vital demand. Several studies have been conducted to decide whether the chest X-ray and computed tomography (CT) scans of patients indicate COVID-19. While these efforts resulted in successful classification systems, the design of a portable and cost-effective COVID-19 diagnosis system has not been addressed yet. The memory requirements of the current state-of-the-art COVID-19 diagnosis systems are not suitable for embedded systems due to the required large memory size of these systems (e.g., hundreds of megabytes). Thus, the current work is motivated to design a similar system with minimal memory requirements. In this paper, we propose a diagnosis system using a Raspberry Pi Linux embedded system. First, local features are extracted using local binary pattern (LBP) algorithm. Second, the global features are extracted from the chest X-ray or CT scans using multi-channel fractional-order Legendre-Fourier moments (MFrLFMs). Finally, the most significant features (local and global) are selected. The proposed system steps are integrated to fit the low computational and memory capacities of the embedded system. The proposed method has the smallest computational and memory resources, less than the state-of-the-art methods by two to three orders of magnitude, among existing state-of-the-art deep learning (DL)-based methods.

OPEN ACCESS

Citation: Hosny KM, Darwish MM, Li K, Salah A (2021) COVID-19 diagnosis from CT scans and chest X-ray images using low-cost Raspberry Pi. PLoS ONE 16(5): e0250688. <https://doi.org/10.1371/journal.pone.0250688>

Editor: Gulistan Raja, University of Engineering & Technology, Taxila, PAKISTAN

Received: February 9, 2021

Accepted: April 13, 2021

Published: May 11, 2021

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pone.0250688>

Copyright: © 2021 Hosny et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The data underlying this study are available on Kaggle (<https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>).

1 Introduction

COVID-19 pandemic affects the lifestyle of the entire world. New challenges are raised for human beings to use the existing knowledge to fight COVID-19 disease. One of these challenges is COVID-19 disease diagnosis using images of chest X-ray [1]. COVID-19 chest radiographs outline bilateral air-space consolidation, as described in the disease characteristics [2].

As DL-based methods are successfully utilized to solve different problems [3, 4], there are several attempts to use chest X-ray and CT scan images to detect COVID-19 cases [5, 6]. For instance, Apostolopoulos and Mpesiana [7] utilized a DL model to classify the X-ray images of patients into one of three classes: bacterial pneumonia, COVID-19 disease, and normal cases. Apostolopoulos and his coauthor used the deep transfer learning approach with four

Funding: Kenli Li was supported in part by the National Natural Science Foundation of China under Grant 61702170.

Competing interests: The author(s) received no specific funding for this work.

architectures, namely, VGG-19 [8], MobileNet [9], Inception [10], and Xception [11]. Their proposed method has the highest accuracy (98.75%) using the VGG-19 model.

Generally, the DL-based classification methods achieve the highest reported accuracy rates. Despite their classification high accuracy rates, running the deep learning models require very expensive computational resources with high specifications. This high-cost processing process might be affordable for large hospital in first-world countries, but hospitals in developing countries and rural areas do not have such expensive computational resources. To reduce the computational cost, Howard et al. [9] build a deep learning model that consumes fewer resources while sacrificing the accuracy rates. Despite the trial of Howard and his colleagues, successful DL-based classification models still require extremely expensive computational machines with high configuration.

Recently, orthogonal moments were utilized to extract features form color images and successfully used in various applications such as recognition of bacterial species [12]. Since medical images have fine details; thus, the task of extracting these features requires highly accurate descriptors. The recent fractional-order descriptors [13] enable the proposed system to extract high-accurate global features from the input CT scan or X-ray images.

These fractional-order descriptors have many characteristics as follows:

1. Their orthogonality enables the representation of medical images without information redundancy.
2. These descriptors are invariant with rotated, scaled and translated images, which improves the classification rates.
3. There is significant robustness against common noise, such as speckle.
4. The MFrLFM descriptors have much faster computation times than other moments.

The computational challenges of DL-based methods and the success of orthogonal moments in classification problems motivate the authors to develop a cost-effective diagnosis system of COVID-19 cases (i.e., less than 100 USD), which classifies input chest scan images such as X-ray and CT into COVID-19 or other lung disease with high classification accuracy rates. The main contributions of this work are as follows.

1. The proposed work is the first system to utilize the Linux embedded system to diagnose COVID-19 cases from the CT scan or X-ray images to the best of the authors' knowledge. Besides, the proposed system can run on any embedded system that supports running the Python code. The proposed system consists of two separate classifiers, one used for classifying the chest X-ray images and another model to classify the CT scan images.
2. The proposed system is designed to be a memory-efficient classification model, as state-of-the-art methods are DL-based methods with huge memory requirements. Thus, the proposed classifier model's main impact is that it becomes possible to obtain a high accuracy rates under a limited memory condition for predicting COVID-19 cases from chest CT and X-ray images.
3. The proposed system is the first system to utilize MFrLFMs moments for global features extraction from chest CT scans or X-ray images. Besides, the Local Binary Pattern (LBP) is utilized for extracting the local features of the input images.

The remainder of the paper is organized as follows. Section 2 exposes the required background of the used techniques and platform. Section 3 discusses the related work. Section 4

describes the proposed system. Section 5 evaluates the proposed system performance. Finally, the work is concluded in Section 6.

2 Preliminaries

2.1 Local binary patterns

The LBPs algorithm has two advantages: robustness to monotonic grayscale changes and low computational cost [14]. The effective performance of the LBPs operator is thoroughly discussed in [15]. LBPs have been utilized in various application domains including medical images classification, texture classification, and facial micro-expression recognition [16].

The LBP algorithm’s basic idea is to assign a certain value, called a code, to each pixel. This pixel’s value (i.e., code) encodes the local features of the 3 × 3 neighborhood window of the eight neighbor cells, as explained in [15]. In a 3 × 3 window, the value of the central pixel is considered the threshold. If the value of any neighbor pixel is less than the threshold, then this neighbor pixel is set to zero; otherwise, the neighbor pixel value is set to one. For example, the threshold of the 3 × 3 window, which is a portion of the image, in Fig 1(a), is 131. In Fig 1(b), each neighbor pixel is set to zero or one depending on the threshold value (i.e., 131). Then, the weight of each pixel is multiplied by the pixel value (i.e., zero or one). The LBP code/value, which is assigned to the window central pixel, is the summation of the multiplied value of all of the eight neighbors.

2.2 Multichannel fractional-order Legendre-Fourier moments

The RGB color image defined using the $f(r, \theta)$ intensity function is represented in three primary channels as $f(r, \theta) = (f_R(r, \theta), f_G(r, \theta), f_B(r, \theta))$ [17].

The MFrLFMs are:

$$FrM_{pq}(f_C) = (2p + 1)/2\pi \int_0^{2\pi} \int_0^1 f_C(r, \theta) [W_{pq}(r, \theta)]^* r dr d\theta \tag{1}$$

where C denotes each primary channel (R-, G- or B-); p and q are the moment order and repetition, respectively; $|p| = 0, 1, 2, 3, \dots, \infty$, $|q| = 0, 1, 2, 3, \dots, \infty$.

The function is

$$W_{pq}(r, \theta) = L_p(\alpha, r) e^{iq\theta} \tag{2}$$

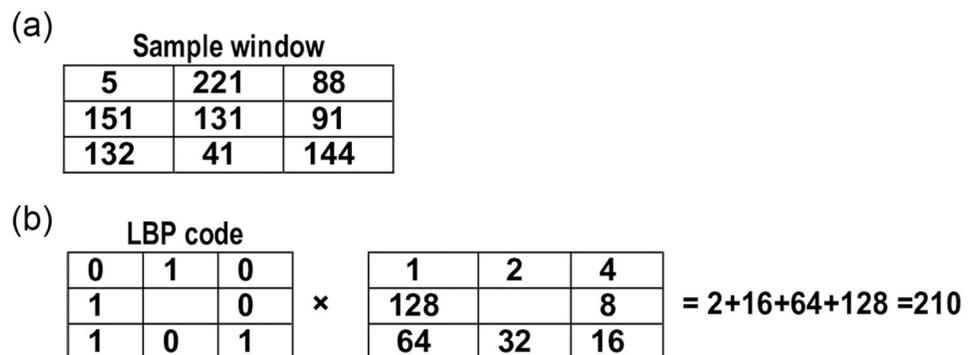


Fig 1. An example of LBP code calculation of a single window. ((a)) 3 × 3 Sample window, ((b)) The calculation of the LBP code of the input windows.

<https://doi.org/10.1371/journal.pone.0250688.g001>

where the fractional-order Legendre polynomials $L_p(\alpha, r)$ are:

$$L_p(\alpha, r) = \sqrt{\alpha} \sum_{k=0}^p (-1)^{(p+k)} \frac{(p+k)! r^{\alpha k + \frac{(\alpha-2)}{2}}}{(p-k)!(k!)^2} \tag{3}$$

Because direct computation using Eq 3 is time-consuming, the three-term recurrence relation is utilized as an alternative.

Since,

$$|M_{(p,q)}(f_C^{rot})| = |M_{(p,q)}f_C|, C \in R, G, B \tag{4}$$

Eq 4 shows that the rotation does not affect the magnitude values of MFrLFMs. The MFrLFMs scale invariants forms are:

$$\varphi_{pq} = \sum_{k=0}^p \frac{2p+1}{2k+1} \left(\sum_{i=k}^p (FrM_{00}(f_C)^{-(2i+3)/3}) C_{pi} d_{ik} \right) FrM_{kq}(f_C) \tag{5}$$

where coefficients C_{pi} and d_{ik} are [18]:

$$C_{pi} = (-1)^{(p+i)} \frac{(p+i)!}{(p-i)!(i!)^2} \tag{6}$$

$$d_{ik} = \frac{(2k+1)(i!)^2}{(i+k+1)!(i-k)!} \tag{7}$$

A highly accurate kernel-based computational framework [19] is used to compute MFrLFMs as follows:

$$FrM_{pq} = \frac{(2p+1)}{2\pi} \sum_i \sum_j I_p(r_i) J_q(\theta_{ij}) \hat{f}_c(r_i, \theta_{ij}) \tag{8}$$

The interpolated function $\hat{f}_c(r_i, \theta_{ij})$ is calculated from the intensity functions of the original image. This task can be achieved using the cubic interpolation, as explained in [20].

Based on Eq 8, the radial and polar kernels are:

$$J_q(\theta_{ij}) = \int_{V_{ij}}^{V_{i,j+1}} e^{-iq\theta} d\theta \tag{9}$$

$$I_p(r_i) = \int_{U_i}^{U_{i+1}} L_p(\alpha, r) r dr = \int_{U_i}^{U_{i+1}} R(r) dr \tag{10}$$

Eq 9 shows that kernel $J_q(\theta_{ij})$ is exactly evaluated as follows:

$$J_q(\theta_{ij}) = \begin{cases} \frac{1}{q} (e^{-iqV_{i,j+1}} - e^{-iqV_{ij}}) & q \neq 0 \\ V_{i,j+1} - V_{ij} & q = 0 \end{cases} \tag{11}$$

An accurate numerical integration approach is used for calculating the integration in Eq 11.

2.3 Raspberry Pi: A Linux embedded system

Raspberry Pi is a single-board computer or a Linux embedded system; it is an open-source ecosystem. It is a cost-effective, lightweight, and portable computer. Raspberry Pi has been utilized in several machine learning applications such as computer vision and image classification [21].

Because Raspberry Pi hardware supports Linux OS, we can benefit from the Python programming language and its powerful packages, especially the Scikit-learn package [22]. Thus, Raspberry Pi hardware can run many machine learning tasks. Another advantage of the Raspberry Pi model is that one of its versions has a multi-core CPU, which enables the acceleration of the running programs by providing parallel implementations of the utilized algorithms.

In [23], the authors discussed the methodology of task division on a Raspberry Pi hardware using OpenMP [24] and MPI [25]. Then, the authors utilized this parallel implementation over a cluster of Raspberry Pi devices to address the problem of edge detection. Another example of the parallel implementation of Raspberry Pi devices is reported in [26]. The authors proposed using a cluster of Raspberry Pi 2 to accelerate the 3D wavelet transform and make it portable.

A user can realize the overall performance of the Raspberry Pi 4 model B as the performance of an entry-level x86 PC, as shown in Fig 2. Raspberry Pi 4 model B utilizes a 64-bit CPU with four cores. The Raspberry Pi 4 model comes with three different options of main memory (i.e., RAM), namely, 1 GB, 2 GB, and 4 GB. For the display options, Raspberry Pi 4 Model B supports a dual-display option (i.e., two micro-HDMI ports). The quality of the display is as high as 4K video resolution.

In addition, Raspberry Pi 4 model B supports different connectivity methods: wireless connection via a dual-band 2.4/5.0 GHz wireless LAN port, Gigabit Ethernet, and Bluetooth 5.0. These features allow one to connect the Raspberry Pi model to any other device, which supports IoT applications. In addition, the USB 3.0, Raspberry Pi model has three USB ports: one port for power connection and two ports for attaching four different peripherals (e.g., mouse and keyboard).

3 Related work

There is much-conducted research that addressed COVID-19 diagnosis from chest CT scans or X-ray images with machine learning techniques. These efforts can be classified based on which deep architecture was utilized by the proposed work. In the following, we discuss representative research works based on the utilized deep architectures.

The DL-based models of COVID-19 diagnosis from chest CT scans or X-ray images are considered the mainstream. Several classification models are proposed, while the main difference is the utilized deep architecture (e.g., Residual Network (ResNet), VGG, Dense Convolutional Network (DenseNet), etc.).

Convolutional Neural Networks (CNNs) [27] is considered the most used deep architecture for image classification. In [28], the authors proposed a CNN-based model for detecting COVID-19 cases from chest X-ray images. They proposed two models; the first one is a binary classifier with two possible outcomes, COVID-19 and Non-COVID-19. The second proposed model is a multi-class classifier model with three possible outcomes, Pneumonia, COVID-19, and Non-COVID-19. Their proposed model classification accuracy rates are 98% and 87%, respectively. In [29], Abd Elaziz et al. utilized two classifier models and two different chest X-ray image datasets to detect COVID-19 cases. Then, they proposed several CNN-based methods for the purpose of comparison. The proposed model accuracy rates were 96% and 98% for

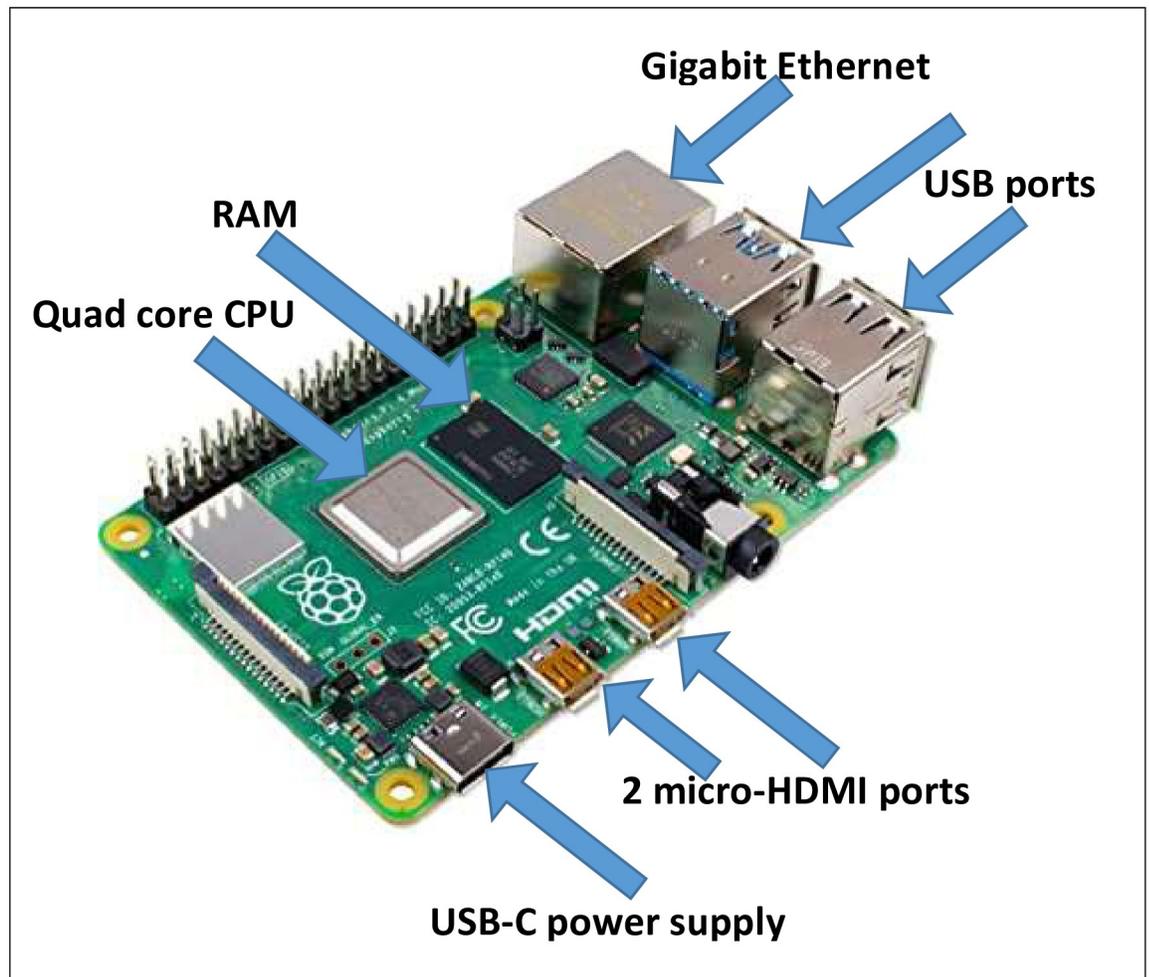


Fig 2. Raspberry Pi 4 model B hardware design.

<https://doi.org/10.1371/journal.pone.0250688.g002>

the first and second datasets, respectively. In addition, the authors in [30] compared several CNN-based COVID-19 detection models.

The ResNet architecture [31] has a significant performance in image classification on several image datasets. In [32], the authors utilized the deep transfer learning approach to train a ResNet architecture for the sake of automatic COVID-19 detection from chest X-ray images. The authors utilized a dataset of 350 normal, 350 Pneumonia, and 210 COVID-19 chest X-ray images. The classification accuracy rate is 94.28%.

The DenseNet architecture is proposed in [33]. In DenseNet, a layer receives inputs from all previous layers; meanwhile, the same layer passes on its feature-maps to all of the following layers. As CT scan images play a vital role in COVID-19 cases automatic detection, several works utilized CT scan images [34–36]. The authors in [37] proposed using deep transfer learning on DenseNet-201 architecture to classify the suspected case as COVID-19 or normal using the patient's CT scan image. They trained the proposed classifier model using a dataset consisting of 2,492 CT scans. The achieved classification accuracy rate is 96%. In [38], the authors proposed using a portable on-device system to detect COVID-19 patients based on the chest X-ray images automatically. The proposed system can follow-up on the case progression as well. The authors utilized the DenseNet-121 architecture with the help of deep transfer

learning to build the classifier model. The highest reported classification accuracy by the proposed system is 88%.

In [39], the authors proposed a 3D deep CNN-based model to recognize COVID-19 cases using CT volumes automatically. The authors proposed generating 3D lung masks using the pre-trained UNet [40], and then these generated masks are classified. The obtained classification accuracy is 90%. In the same context, several research works proposed different tasks on COVID-19 CT scan images. For instance, the authors in [41, 42] proposed two segmentation methods for removing the noise data from the input image, as a pre-processing step for the classification task. These proposed segmentation methods eased the classification task and resulted in improving the classification accuracy rates.

The VGG deep architecture achieves high classification accuracy rates despite its huge memory requirements. In [43], the authors utilized a dataset of 592 CT scan images with two classes COVID-19 and normal. Then, they proposed the CTnet-10 model, a binary classifier model. This proposed model's classification accuracy rate is 82.1% while utilizing the pre-trained VGG-19 model for the classification task yields an accuracy of 94.5%. Another VGG-based model is proposed in [44]. The authors proposed a multi-class classifier to classify a chest X-ray image as COVID-19, pneumonia, or normal. The utilized dataset consists of 360 images. They proposed creating feature maps from the X-ray images, and then the vectorized version of these feature maps are classified using the VGG-16 architecture. They utilized the deep transfer learning approach by using the saved VGG-16 weights as trained on the ImageNet dataset. Besides, they proposed adding an output layer for the three possible classification outcomes. The classification accuracy rate is 91%.

4 Proposed system

The proposed system consists of four main phases, as shown in Fig 3. The first phase includes extracting the local features using the LBPs algorithm. The second phase includes global features using MFrLFMs. In the third phase, the local and global features are combined and then a feature selection method is applied to select the most significant features. Finally, the fourth phase includes a binary classifier that takes the selected local and global features as an input to classify the input image as COVID-19 disease or other diseases.

4.1 Local features using the LBPs

The LBP feature vector is computed as a $1 \times N$ vector, where N is the number of extracted local features. The LBPs algorithm partitions the input image into non-overlapping windows. A wider window size corresponds to less computational complexity and fewer details of the collected local features. In the proposed system, the number of neighbors P is set to 8. Thus, the total number of extracted local features is $N = (P \times P - 1) + 3 = (8 \times 8) + 3 = 67$ features.

4.2 Global feature extraction using MFrLFMs

The sequential computations of MFrLFMs are inconvenient for multicore CPU without loop fusion, since the computations consist of four nested loops. Thus, we utilized the loop fusion technique to the outermost two loops to parallelize the sequential computations of MFrLFMs. Thus, the iterations of this fused loop can be independently computed.

Since Raspberry Pi has at most four cores, the loop fusion of the outermost two loops provides a sufficient number of independent iterations. The iteration number of the two fused loops of MFrLFM computation is mapped to the original loop iteration numbers as shown in

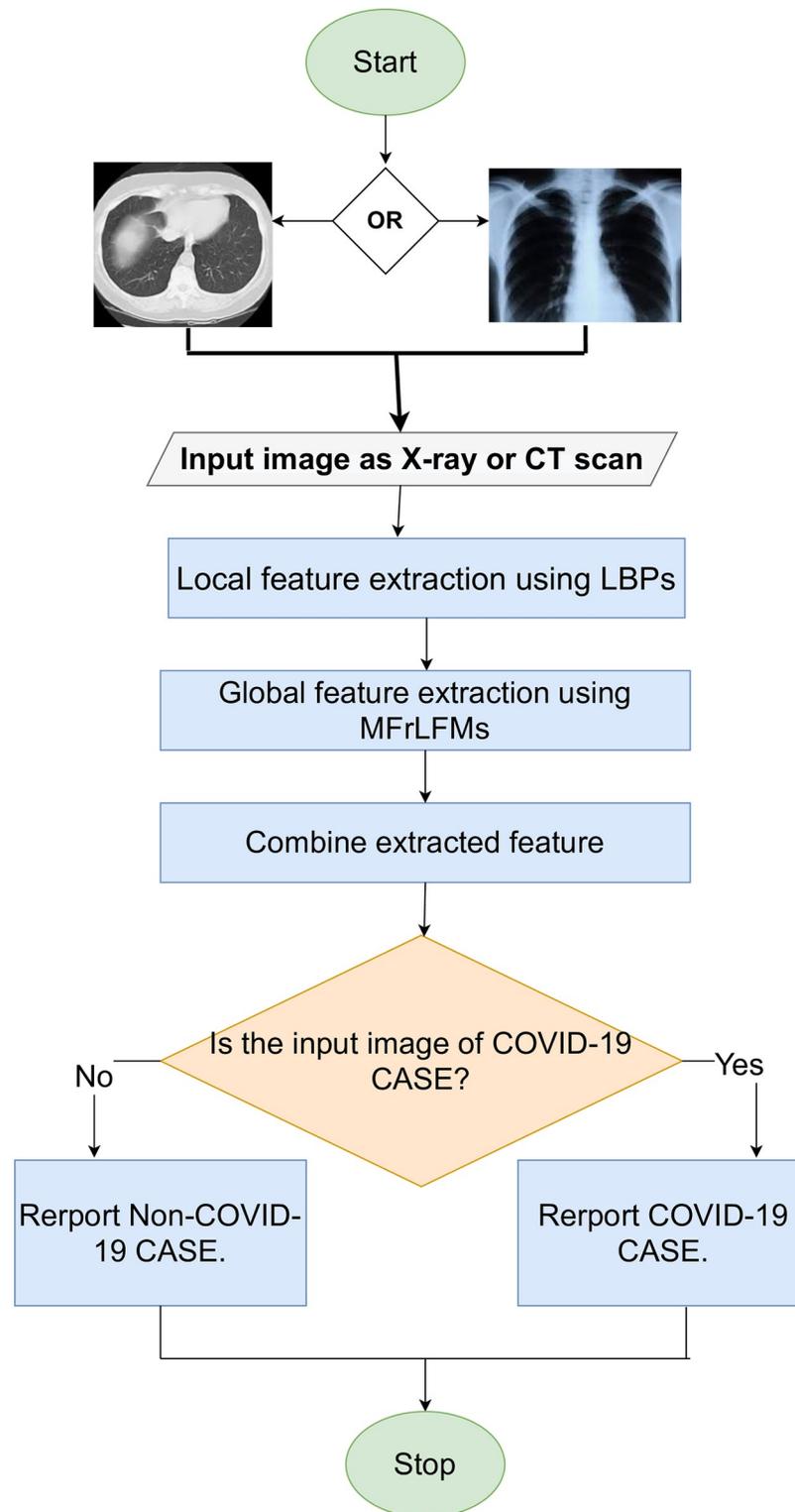


Fig 3. Flowchart of x-ray image and CT scan classification models.

<https://doi.org/10.1371/journal.pone.0250688.g003>

Eq 12 for two loops.

$$\begin{aligned} p &= i_{fused} / (qmax + 1) \\ q &= i_{fused} \bmod (qmax + 1) \end{aligned} \tag{12}$$

where i_{fused} is the iteration number of the four fused loops, $i_{fused} \in [0, (pmax + 1) \times (qmax + 1)]$ for a two-loop fusion.

Algorithm 1 lists the parallel implementation of the MFrLFM computations. In Algorithm 1 line 1, the algorithm divides the iterations of the outer loop over only p parallel resources, i.e., Raspberry Pi CPU cores. This task can easily be accomplished using the OpenMP directive `#pragma omp parallel for num_threads(p)`. This OpenMP evenly divides $(pmax + 1) \times (qmax + 1)$ iterations over the available p threads/cores.

In Algorithm 1 line 2, the **for** loop represents two fused loops of kernels p and q . Iterator i_{fused} goes through $(pmax + 1) \times (qmax + 1)$ iterations; each iteration represents unique p and q values. Thus, the variable i_{fused} should be mapped to the corresponding p and q values, as listed in lines 4 and 5, using Eq 12. Line 6 resets the accumulative variables of each $M_{p,q}$ moment. The **for** loop in Line 7 goes through all of the M image rings. Similarly, the **for** loop in Line 8 goes through each sector r . Line 9 computes the kernel value. To compute the kernel value, two terms should be multiplied; the radial kernel value is accessed using the p and $ring$ values, and the repeating kernel is accessed using the $ring$, sec , and q values. Lines 10-12 compute the moment of the three channels, i.e., red, green, and blue. At each of these three lines, the image pixel is accessed by the term $r_image[ring][sec]$ using the $ring$ and sec values and multiplied by the kernel value, as computed in Line 9. Finally, the $M_{p,q}$ moment is computed by multiplying the value computed within the loop by a constant.

Algorithm 1 consists of three nested loops. The time complexity of the first loop is $O\left(\frac{pmax \times qmax}{p}\right)$. The second and third loops iterate over each pixel of the $N \times N$ pixels of the input image. The time complexity of the second and third inner loops is $O(N^2)$. Thus, the time complexity of Algorithm 1 is the multiplication of time complexity of these three loops. Using p parallel resources, the time complexity of Algorithm 1 (i.e., MFrLFMs) is $O\left(\frac{pmax \times qmax \times N^2}{p}\right)$.

The time complexity of computing the LBP algorithm is N^2 . There are N^2 pixels per the input image; for each pixel, a binary patten of size eight is generated, where each neighbor contributes by one bit. Thus, the time complexity of computing the N^2 LBP codes is $O(8 \times N^2) = O(N^2)$. As the LBPs can be calculated independently, the LBPs algorithms can easily be parallelized by dividing the N^2 LBP codes computation over p parallel resources. Thus, the final time complexity of the local feature extraction phase is $O(N^2/p)$.

The proposed systems' overall time complexity equals the time complexity of the summation of local and global feature extraction time complexities. Thus, the proposed system overall time complexity is $O\left(\frac{pmax \times qmax \times N^2}{p} + \frac{N^2}{p}\right) = O\left(\frac{pmax \times qmax \times N^2}{p}\right)$.

The space complexity of the MFrLFMs algorithm is $O(pmax \times qmax)$, as the algorithm stores the computed moments in a 2D matrix of $pmax$ rows and $qmax$ columns regardless the image size, i.e., the value of N . On the other hand, the space complexity of the local feature extraction phase by the LBP algorithm is $O(N^2)$. This is because the LBPs algorithm stores a code for each pixel of the N^2 image pixels. Thus, the overall space complexity of the proposed method is $O(N^2 + pmax \times qmax) = O(N^2)$, as $pmax \ll N$ and $qmax \ll N$.

Algorithm 1 The parallel algorithm of MFrLFMs computations.

- 1 Divide the following for iterations over p cores
- 2 **for** $i_{fused} = 0$:
- 3 $(pmax + 1) \times (qmax + 1)$ **do**

```

4  $p = i_{used} / (q_{max} + 1)$ 
5  $q = i_{used} \bmod (q_{max} + 1)$ 
6  $r = g = b = 0$ 
7 for  $ring = 1: M$  do
8   for  $sec = 1: S \times (2 \times i + 1)$  do
9      $kernel\_val = I_p[p][ring] \times I_q[ring][q][sec]$ 
10     $r += r\_image[ring][sec] \times kernel\_val$ 
11     $g += g\_image[ring][sec] \times kernel\_val$ 
12     $b += b\_image[ring][sec] \times kernel\_val$ 
13  end for
14   $Red_{M_p,q}[p][q] = r \times ((2 \times p) + 1) / (2 \times PI)$ 
15   $Green_{M_p,q}[p][q] = g \times ((2 \times p) + 1) / (2 \times PI)$ 
16   $Blue_{M_p,q}[p][q] = b \times ((2 \times p) + 1) / (2 \times PI)$ 
end for

```

4.3 Feature selection and classification

The last step to prepare the data for the classification task is to select the most significant features to classify the images. The number of extracted local features of each input image is 59, as discussed in 4.1. The number of extracted global features of each input image is $(p_{max} + 1) \times (q_{max} + 1)$ features. For example, if $p_{max} = q_{max} = 30$, then each image is represented by 961 global features. Thus, the total number of local and global features for an image when $p_{max} = q_{max} = 30$ is 1,020 features. In other words, each image is represented by 1,020 decimal values.

We proposed applying a feature selection technique to remove any irrelevant, redundant, and noise features. The feature selection reduces the classifier training time and classifier prediction time, since extracting fewer features reduces the feature extraction phase time. To achieve this goal, we proposed using Sequential Feature Selector (SFS) greedy search technique. This greedy approach has k iterations. The SFS method adds one feature at each iteration, which is the most significant feature; finally, The SFS algorithm selects the most k significant features. If the SFS algorithm finds a subset of features with fewer than k features, this feature subset is reported. Thus, the number of selected features can be smaller than or equal to k . The SFS is executed one time; thus, it has no effect on the proposed method run-time nor the proposed method time complexity.

Once we selected the most significant k features for all input images using the SFS technique, the dataset is ready to train the classifier. The proposed method is applicable for any binary classifier, COVID-19 or non-COVID-19 input image, where the input image can be a chest X-ray or a CT scan image. We proposed two separate classifier models for each of the two image types.

5 Experimental results

5.1 Dataset

The utilized dataset consists of eight lung diseases from eight chest X-ray dataset [45]: 1) Atelectasis; 2) Cardiomegaly; 3) Effusion; 4) Infiltration; 5) Mass; 6) Nodule; 7) Pneumonia; 8) Pneumothorax. Each lung disease has 212 images. These images of the eight lung diseases are collected in one image class, which is called Non-COVID-19 diseases. The second class consists of 212 images of chest X-ray of COVID-19 patients [46]. This data collection approach results in unbalanced classes, as the first class has $212 \times 8 = 1,696$ images and the second class has 212 images. In addition, we used a dataset of CT scans of COVID-19 patients and other lung diseases. The second dataset consists of 2,842 images classified in two classes: COVID-19

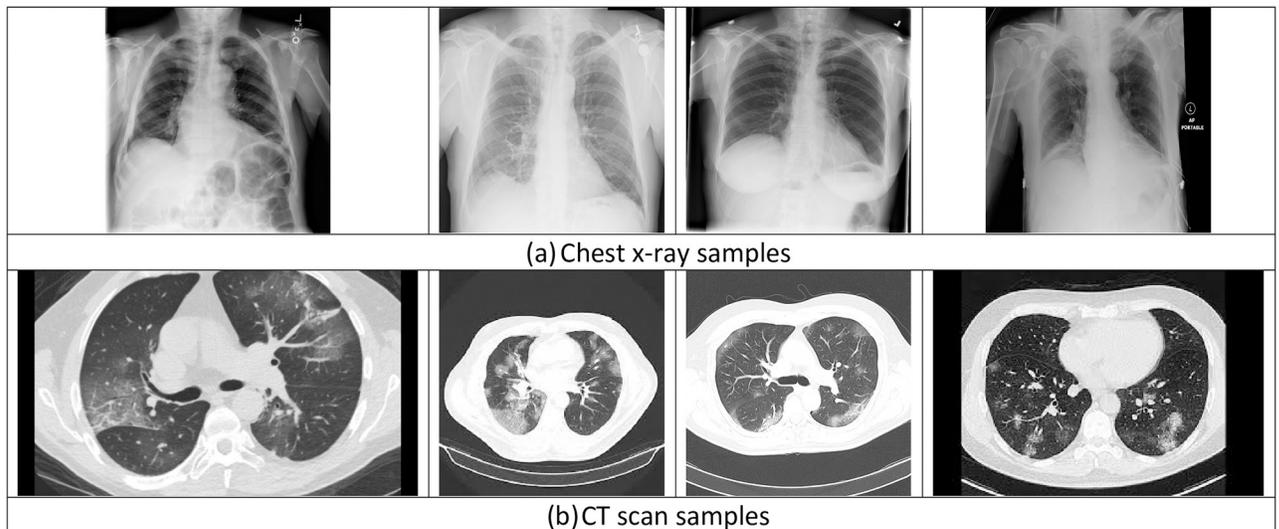


Fig 4. Sample of the two datasets: (a) chest X-ray images [45]; (b) CT scans [47].

<https://doi.org/10.1371/journal.pone.0250688.g004>

class with 1,252 images and non-COVID-19 class with 1,230 images [47]. Fig 4 shows samples of these two datasets.

The utilized datasets are split into 80% and 15% as training and test sets, respectively. These two sets are used to train and test the proposed models and the comparison method. Besides, the cross-validation technique was utilized, where the number of folds was five. In other words, the dataset was divided into five different ways. Finally, the hyperparameters of these methods were set to the default values during the training phase for the methods of comparison.

5.2 Setup

Experiments are performed on a Raspberry Pi 4 Model B with 4-cores CPU. The utilized OS is 64-bit Linux. The implementations were written in the C++ and the Python programming languages. C++ is used to implement the local- and global-feature extraction algorithms, and Python is used to implement the image classifier. We used the standard OpenMP thread library [24] for CPU multi-core implementation. The reported results are the average of running each experiment three times. Fig 5 shows the result of the proposed system on Raspberry Pi 4 Model B, where the input image is classified as a chest X-ray of a COVID-19 patient.

The MFrLFM radial and repeating kernel order is set to 31. Thus, the number of global features is 961. The number of local features is 59. The total number of utilized features is 1,020. After the SFS feature selection method has been applied, the chest X-ray image classifier and CT scan classifier are trained on 26 global plus 15 local features (i.e., 41 features).

5.3 Results

Table 1 lists three different accuracy metrics to evaluate the proposed methods, including the accuracy, AUC, and F1-score. The proposed method achieved comparable results with comparison to other deep learning methods in terms of accuracy, AUC, and F1-score metrics.

Table 2 lists the required memory and prediction time of the proposed trained classifier.

Table 3 lists memory requirements of the proposed method and state-of-the-art models. As

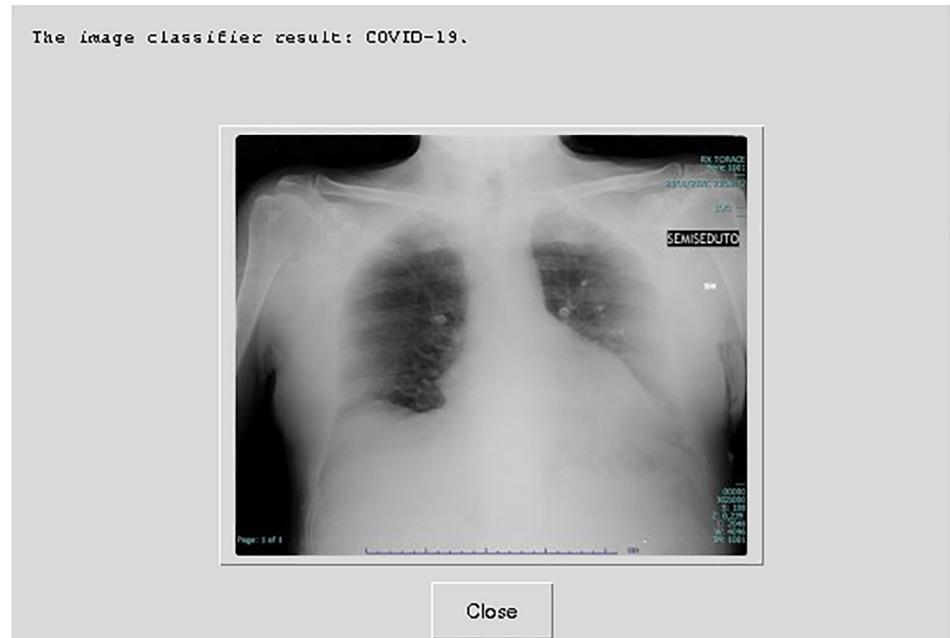


Fig 5. Proposed system on Raspberry Pi 4 Model B.

<https://doi.org/10.1371/journal.pone.0250688.g005>

listed in Table 3, the proposed system has the least memory requirements and prediction time in comparison to the other deep-learning-based methods.

To evaluate the ROC AUC values, Fig 6 depicts the receiver operating characteristic (ROC) curve, and Fig 7 depicts the precision curve of the proposed method. Figs 6 and 7 show the efficiency of the proposed system to classify the input chest X-ray CT scan as a COVID-19 disease or other disease.

Table 1. Accuracy of the proposed models.

data	Accuracy(%)	F1-score	AUC
chest X-ray	99.3±0.2%	93.1±0.2%	94.9±0.1%
CT scans	93.2±0.3%	92.1±0.3%	93.2±0.3%

<https://doi.org/10.1371/journal.pone.0250688.t001>

Table 2. Required memory and the prediction time in seconds of the two proposed models on Raspberry Pi.

data	Memory	Prediction time (s)
chest X-ray	3MB	10
CT scans	3MB	10

<https://doi.org/10.1371/journal.pone.0250688.t002>

Table 3. The required memory of the proposed model and state-of-the-art models on Raspberry Pi.

Method	Memory
The proposed system	1MB
VGG19 [8]	1,406MB
MobileNet v2 [9]	13MB
Inception-3 [10]	232MB
ResNet-50 [48]	101MB

<https://doi.org/10.1371/journal.pone.0250688.t003>

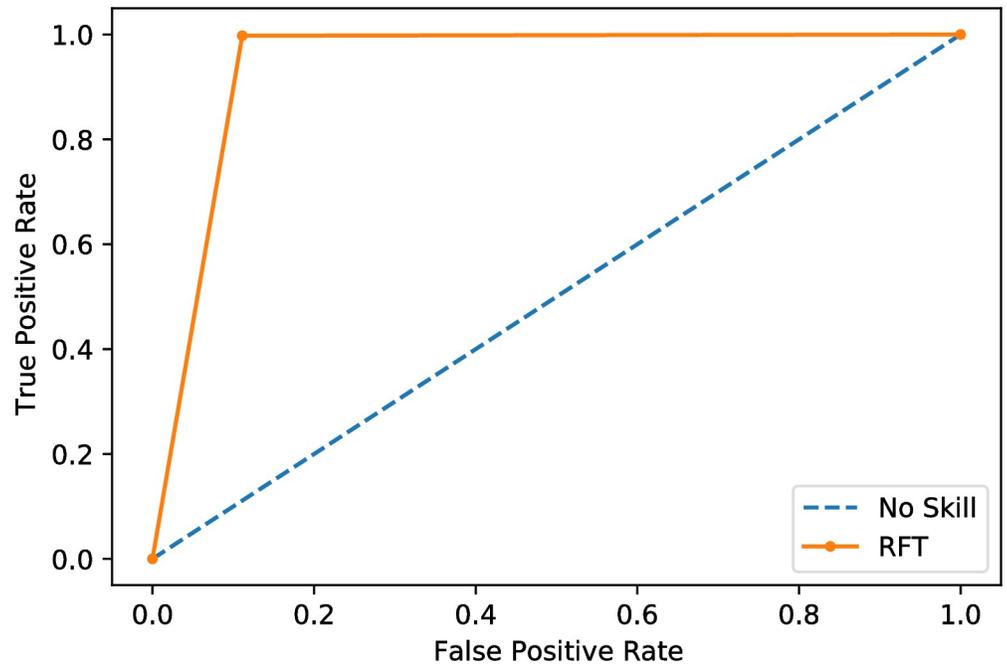


Fig 6. ROC curve of the proposed X-ray image classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g006>

Finally, the precision, recall, and confusion matrices are examined for the two proposed models. Figs 8 and 9 show the precision-recall scores of the proposed two models. Besides, the confusion matrices are depicted in Figs 10 and 11 for the two proposed classifiers.

The results above outline the memory requirements gap between of the proposed classifiers and state-of-the-art methods; the proposed models require memory spaces less the existing

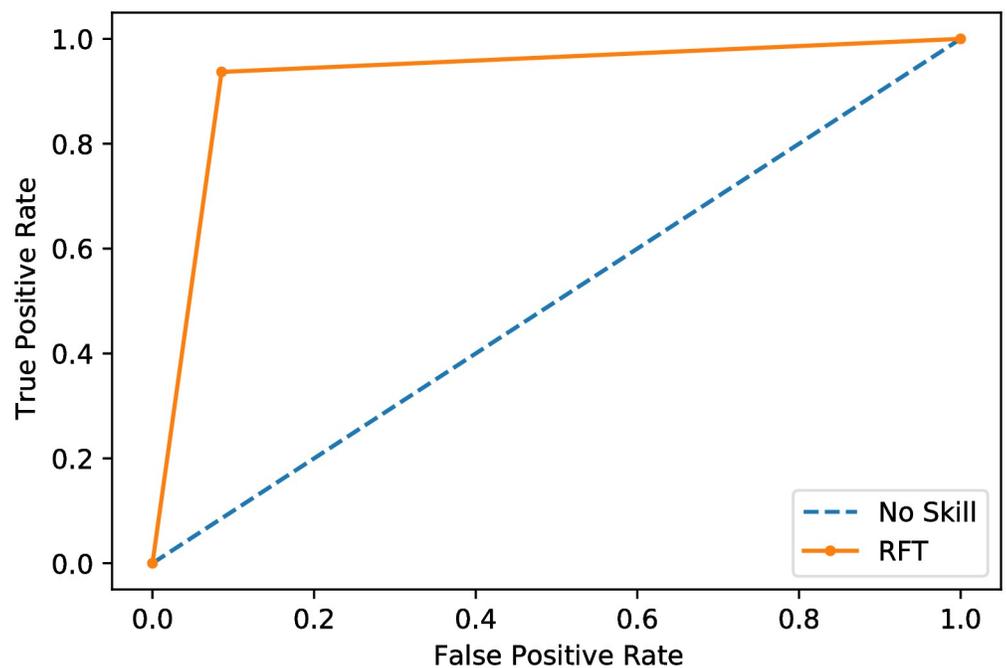


Fig 7. ROC curve of the proposed CT scan classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g007>

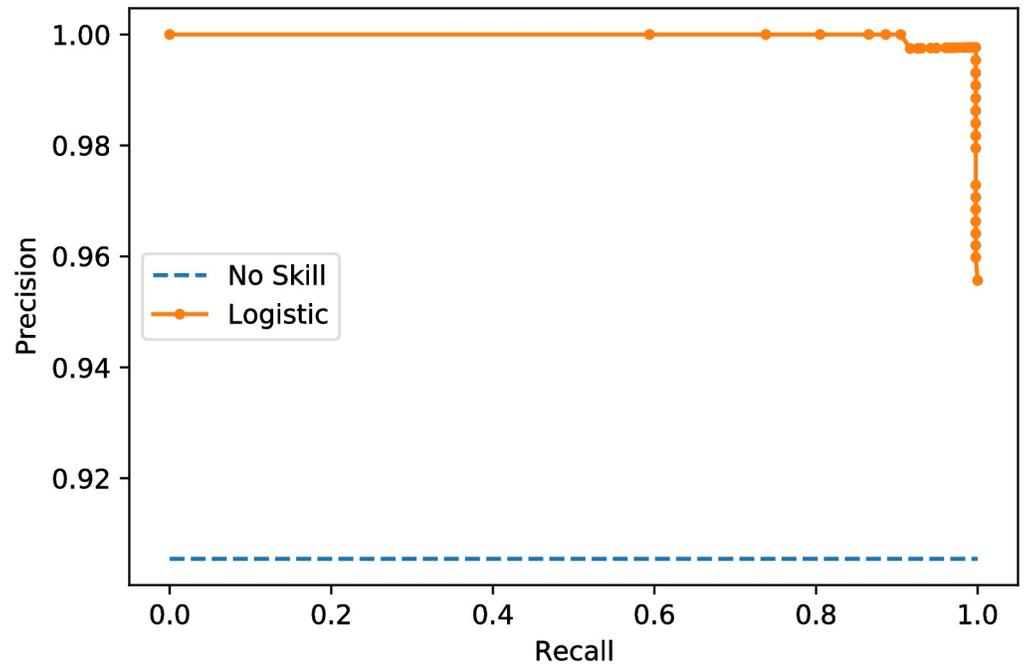


Fig 8. Precision-recall score curve of the proposed X-ray image classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g008>

methods by 2-3 orders of magnitude, as listed in Table 3. Thus, the main goal of this research is achieved, as a small-sized (i.e., 3 MB) model for COVID-19 diagnosis can fit the low-memory embedded system. In addition, the proposed models maintain the accuracy rates of state-of-the-art methods.

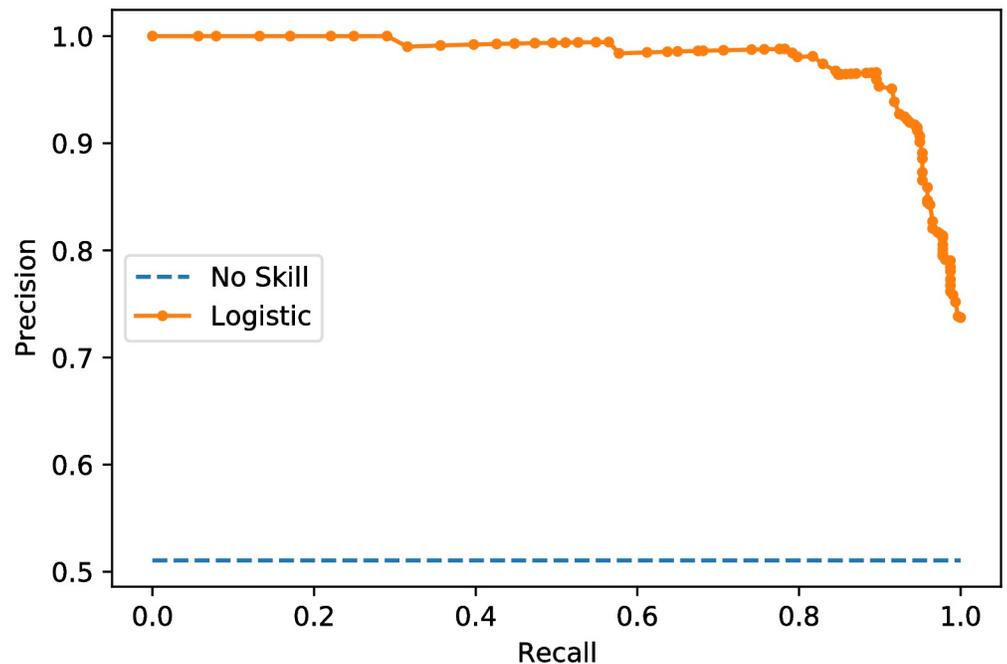


Fig 9. Precision-recall score curve of the proposed CT scan classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g009>

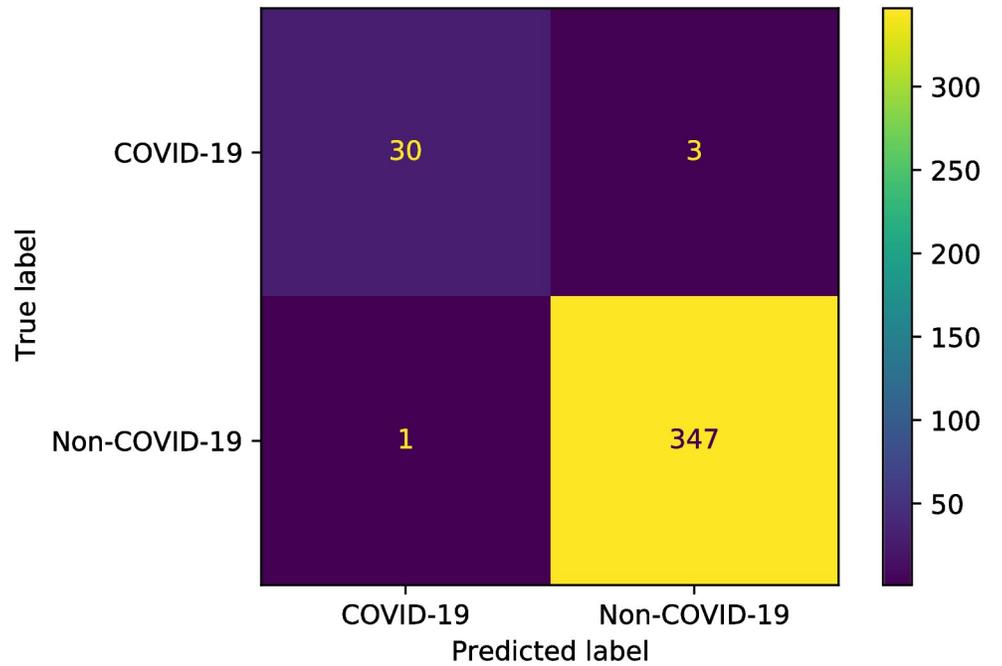


Fig 10. Confusion matrix of the proposed X-ray classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g010>

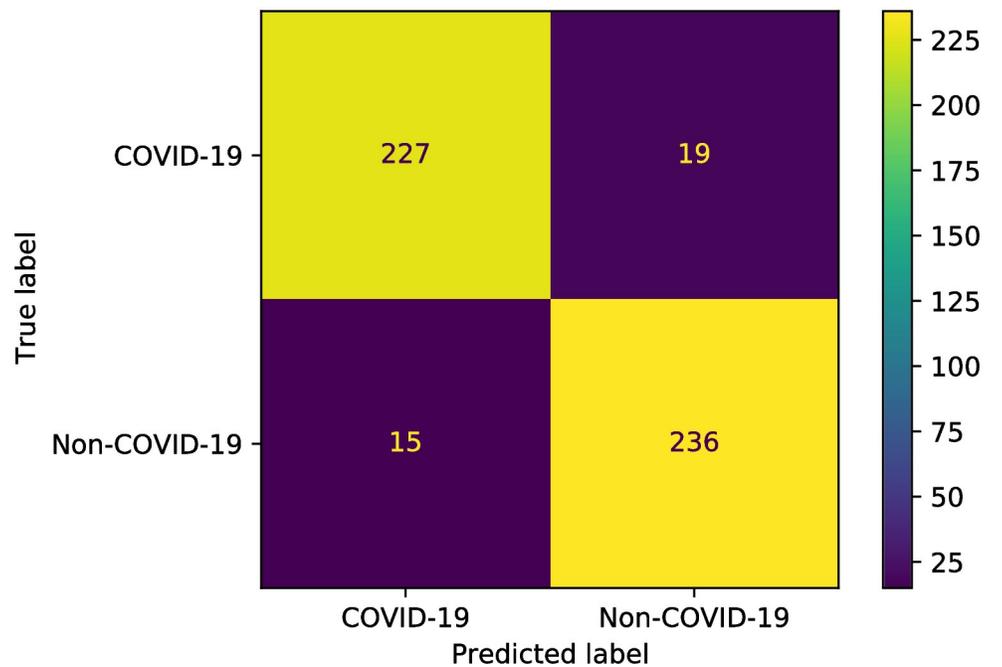


Fig 11. Confusion matrix of the proposed CT scan images classifier model.

<https://doi.org/10.1371/journal.pone.0250688.g011>

6 Conclusion

In this work, we proposed two low-cost image classifier models that can operate on a Linux-embedded system, i.e., Raspberry Pi, to automatically detect COVID-19 cases on two types of imagery data, namely, chest X-ray and CT scan images. To our knowledge, this is the first time to achieve this task. The proposed system consists of several steps. First, the proposed methods extract the local features using LBP and then extract the global features using MFrLFMs moments from the input image. Second, the combined local and global features represent the input chest X-ray or CT scan image's final features. Finally, a classifier is trained to distinguish COVID-19 cases from the chest X-ray or CT scan images of other lung diseases. The proposed classification models require the smallest amount of memory (approximately 3 MB), which makes these models suitable for computationally limited hardware. The two proposed models are evaluated on a chest X-ray dataset of 1,926 images and a CT scan dataset of 2,482 images; each dataset has two classes (i.e., COVID-19 and other lung diseases). The proposed system has comparable scores on the evaluation metrics with state-of-the-art methods. At the same time, their computational and memory requirements are less than those of state-of-the-art DL-based methods by 2-3 orders of magnitude. As future work, the proposed system can be extended to classify more lung diseases. This can be achieved by proposing a multi-class classifier and utilizing the proper dataset.

Author Contributions

Conceptualization: Khalid M. Hosny, Kenli Li, Ahmad Salah.

Data curation: Khalid M. Hosny.

Investigation: Khalid M. Hosny, Mohamed M. Darwish, Kenli Li.

Methodology: Khalid M. Hosny, Ahmad Salah.

Resources: Khalid M. Hosny, Mohamed M. Darwish, Ahmad Salah.

Supervision: Khalid M. Hosny, Kenli Li, Ahmad Salah.

Visualization: Khalid M. Hosny, Mohamed M. Darwish.

Writing – original draft: Khalid M. Hosny, Ahmad Salah.

Writing – review & editing: Khalid M. Hosny, Mohamed M. Darwish.

References

1. Jacobi A, Chung M, Bernheim A, Eber C. Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review. *Clinical Imaging*. 2020; 64:35–42. <https://doi.org/10.1016/j.clinimag.2020.04.001> PMID: 32302927
2. Guan Wj, Ni Zy, Hu Y, Liang Wh, Ou Cq, He Jx, et al. Clinical characteristics of coronavirus disease 2019 in China. *New England journal of medicine*. 2020; 382(18):1708–1720. <https://doi.org/10.1056/NEJMoa2002032>
3. Fathalla A, Salah A, Li K, Li K, Francesco P. Deep end-to-end learning for price prediction of second-hand items. *Knowledge and Information Systems*. 2020; 62(12):4541–4568. <https://doi.org/10.1007/s10115-020-01495-8>
4. Duan M, Li K, Ouyang A, Win KN, Li K, Tian Q. EGroupNet: A Feature-enhanced Network for Age Estimation with Novel Age Group Schemes. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2020; 16(2):1–23.
5. Heidari M, Mirniaharikandehi S, Khuzani AZ, Danala G, Qiu Y, Zheng B. Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms. *International journal of medical informatics*. 2020; 144:104284. <https://doi.org/10.1016/j.ijmedinf.2020.104284> PMID: 32992136

6. Gupta M, Bansal A, Jain B, Rochelle J, Oak A, Jalali MS. Whether the weather will help us weather the COVID-19 pandemic: Using machine learning to measure twitter users perceptions. *International journal of medical informatics*. 2020; 145:104340. <https://doi.org/10.1016/j.ijmedinf.2020.104340> PMID: [33242762](https://pubmed.ncbi.nlm.nih.gov/33242762/)
7. Apostolopoulos I, Mpesiana T. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*. 2020; 43(2):635–640. <https://doi.org/10.1007/s13246-020-00865-4> PMID: [32524445](https://pubmed.ncbi.nlm.nih.gov/32524445/)
8. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. 2014;.
9. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:170404861*. 2017;.
10. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:160207261*. 2016;.
11. Chollet F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2017. p. 1251–1258.
12. Abd Elaziz M, Hosny KM, Hemedan AA, Darwish MM. Improved recognition of bacterial species using novel fractional-order orthogonal descriptors. *Applied Soft Computing*. 2020; 95:106504. <https://doi.org/10.1016/j.asoc.2020.106504>
13. Hosny KM, Darwish MM, Aboelenen T. New fractional-order Legendre-Fourier moments for pattern recognition applications. *Pattern Recognition*. 2020; p. 107324. <https://doi.org/10.1016/j.patcog.2020.107324>
14. Pietikäinen M, Zhao G. Two decades of local binary patterns: A survey. In: *Advances in independent component analysis and learning machines*. Elsevier; 2015. p. 175–210.
15. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*. 2002; 24(7):971–987. <https://doi.org/10.1109/TPAMI.2002.1017623>
16. Huang X, Wang SJ, Liu X, Zhao G, Feng X, Pietikäinen M. Discriminative spatiotemporal local binary pattern with revisited integral projection for spontaneous facial micro-expression recognition. *IEEE Transactions on Affective Computing*. 2017; 10(1):32–47. <https://doi.org/10.1109/TAFFC.2017.2713359>
17. Hosny KM, Darwish MM. New set of multi-channel orthogonal moments for color image representation and recognition. *Pattern Recognition*. 2019; 88:153–173. <https://doi.org/10.1016/j.patcog.2018.11.014>
18. Hosny KM, Darwish MM. Invariant color images representation using accurate quaternion Legendre–Fourier moments. *Pattern Analysis and Applications*. 2019; 22(3):1105–1122. <https://doi.org/10.1007/s10044-018-0740-1>
19. Hosny KM, Darwish MM. A Kernel-Based method for Fast and accurate computation of PHT in polar coordinates. *Journal of Real-Time Image Processing*. 2019; 16(4):1235–1247. <https://doi.org/10.1007/s11554-016-0622-y>
20. Hosny KM, Shouman MA, Salam HMA. Fast computation of orthogonal Fourier–Mellin moments in polar coordinates. *Journal of Real-Time Image Processing*. 2011; 6(2):73–80. <https://doi.org/10.1007/s11554-009-0135-z>
21. John N, Surya R, Ashwini R, Kumar SS, Soman K. A low cost implementation of multi-label classification algorithm using Mathematica on Raspberry Pi. *Procedia computer science*. 2015; 46:306–313. <https://doi.org/10.1016/j.procs.2015.02.025>
22. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*. 2011; 12:2825–2830.
23. Govindaraj V. Parallel programming in Raspberry Pi cluster. A Design Project Report, School of Electrical and Computer Engineering, Cornell University. 2016;.
24. Chandra R, Dagum L, Kohr D, Menon R, Maydan D, McDonald J. *Parallel programming in OpenMP*. Morgan kaufmann; 2001.
25. Gropp W, Thakur R, Lusk E. *Using MPI-2: Advanced features of the message passing interface*. MIT press; 1999.
26. Bernabé G, Hernández R, Acacio ME. Parallel implementations of the 3D fast wavelet transform on a Raspberry Pi 2 cluster. *The Journal of Supercomputing*. 2018; 74(4):1765–1778. <https://doi.org/10.1007/s11227-016-1933-2>
27. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*. 2018; 9(4):611–629. <https://doi.org/10.1007/s13244-018-0639-9> PMID: [29934920](https://pubmed.ncbi.nlm.nih.gov/29934920/)

28. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in biology and medicine*. 2020; 121:103792. <https://doi.org/10.1016/j.compbiomed.2020.103792> PMID: 32568675
29. Elaziz MA, Hosny KM, Salah A, Darwish MM, Lu S, Sahlol AT. New machine learning method for image-based diagnosis of COVID-19. *PloS one*. 2020; 15(6):e0235187. <https://doi.org/10.1371/journal.pone.0235187> PMID: 32589673
30. Shi F, Wang J, Shi J, Wu Z, Wang Q, Tang Z, et al. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE reviews in biomedical engineering*. 2020;.
31. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: *European conference on computer vision*. Springer; 2016. p. 630–645.
32. Keles A, Keles MB, Keles A. COV19-CNNnet and COV19-ResNet: diagnostic inference Engines for early detection of COVID-19. *Cognitive Computation*. 2021; p. 1–11. <https://doi.org/10.1007/s12559-020-09795-5> PMID: 33425046
33. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2017. p. 4700–4708.
34. Zhu X, Song B, Shi F, Chen Y, Hu R, Gan J, et al. Joint prediction and time estimation of COVID-19 developing severe symptoms using chest CT scan. *Medical image analysis*. 2021; 67:101824. <https://doi.org/10.1016/j.media.2020.101824> PMID: 33091741
35. Tang Z, Zhao W, Xie X, Zhong Z, Shi F, Ma T, et al. Severity assessment of COVID-19 using CT image features and laboratory indices. *Physics in Medicine & Biology*. 2021; 66(3):035015. <https://doi.org/10.1088/1361-6560/abbf9e> PMID: 33032267
36. Sun L, Mo Z, Yan F, Xia L, Shan F, Ding Z, et al. Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*. 2020; 24(10):2798–2805. <https://doi.org/10.1109/JBHI.2020.3019505> PMID: 32845849
37. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M. Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*. 2020; p. 1–8. <https://doi.org/10.1080/07391102.2020.1788642> PMID: 32619398
38. Li X, Li C, Zhu D. COVID-MobileXpert: On-device COVID-19 patient triage and follow-up using chest X-rays. In: *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2020. p. 1063–1067.
39. Zheng C, Deng X, Fu Q, Zhou Q, Feng J, Ma H, et al. Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*. 2020;.
40. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer; 2015. p. 234–241.
41. Fan DP, Zhou T, Ji GP, Zhou Y, Chen G, Fu H, et al. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*. 2020; 39(8):2626–2637. <https://doi.org/10.1109/TMI.2020.2996645> PMID: 32730213
42. Wu YH, Gao SH, Mei J, Xu J, Fan DP, Zhang RG, et al. Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. *IEEE Transactions on Image Processing*. 2021;. <https://doi.org/10.1109/TIP.2021.3058783> PMID: 33600316
43. Shah V, Keniya R, Shridharani A, Punjabi M, Shah J, Mehendale N. Diagnosis of COVID-19 using CT scan images and deep learning techniques. *Emergency radiology*. 2021; p. 1–9. <https://doi.org/10.1007/s10140-020-01886-y> PMID: 33523309
44. Dansana D, Kumar R, Bhattacharjee A, Hemanth DJ, Gupta D, Khanna A, et al. Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm. *Soft Computing*. 2020; p. 1–9. <https://doi.org/10.1007/s00500-020-05275-y> PMID: 32904395
45. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2017. p. 2097–2106.
46. Cohen JP, Morrison P, Dao L. COVID-19 image data collection. *arXiv 200311597*. 2020;.
47. Soares E, Angelov P, Biaso S, Froes MH, Abe DK. SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. *medRxiv*. 2020; <https://doi.org/10.1101/2020.04.24.20078584>
48. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016. p. 770–778.