



Classifier Fusion for Detection of COVID-19 from CT Scans

Taranjit Kaur¹ · Tapan Kumar Gandhi¹

Received: 30 March 2021 / Revised: 10 December 2021 / Accepted: 11 December 2021 /
Published online: 3 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. COVID-19 is found to be the most infectious disease in last few decades. This disease has infected millions of people worldwide. The inadequate availability and the limited sensitivity of the testing kits have motivated the clinicians and the scientist to use Computer Tomography (CT) scans to screen COVID-19. Recent advances in technology and the availability of deep learning approaches have proved to be very promising in detecting COVID-19 with increased accuracy. However, deep learning approaches require a huge labeled training dataset, and the current availability of benchmark COVID-19 data is still small. For the limited training data scenario, the CNN usually overfits after several iterations. Hence, in this work, we have investigated different pre-trained network architectures with transfer learning for COVID-19 detection that can work even on a small medical imaging dataset. Various variants of the pre-trained ResNet model, namely ResNet18, ResNet50, and ResNet101, are investigated in the current paper for the detection of COVID-19. The experimental results reveal that transfer learned ResNet50 model outperformed other models by achieving a recall of 98.80% and an F1-score of 98.41%. To further improvise the results, the activations from different layers of best performing model are also explored for the detection using the support vector machine, logistic regression and K-nearest neighbor classifiers. Moreover, a classifier fusion strategy is also proposed that fuses the predictions from the different classifiers via majority voting. Experimental results reveal that via using learned image features and classification fusion strategy, the recall, and F1-score have improvised to 99.20% and 99.40%.

Keywords CT images · COVID-19 · Diagnosis · Transfer learning · Activations · Classifier fusion

✉ Taranjit Kaur
Taranjit.Kaur@ee.iitd.ac.in
Tapan Kumar Gandhi
tgandhi@ee.iitd.ac.in

¹ Department of Electrical Engineering, Indian Institute of Technology, Delhi (IIT Delhi), Hauz Khas, New Delhi 110016, India

1 Introduction

The emergence of COVID-19 infection has critically affected the social and economic structures of both the developing and the developed countries since December 2019 [12]. Researchers and health care workers around the globe are trying to apprehend the COVID-19 etiology and its effect on the quality of life [30].

Nowadays, Computer Tomography (CT) scans are emerging as an alternative for screening in contrast to the conventional reverse transcription-polymerase chain reaction (RT-PCR). This is owing to the limitations associated with the availability, reproducibility, and significant false-negative outcomes produced by RT-PCR kits.

Distinctive CT scans having patchy ground-glass opacities are important biomarkers that can aid in speedy detection and isolation of the subject [1, 16, 31].

As the cases are increasing at an alarming rate, an automated detection system for COVID-19 is the need of the hour that can assist in faster virus detection at different stages thereby relieving the healthcare professionals from the manual annotation task.

Several artificial intelligence (AI) based methods have evolved that automatically provide a diagnosis whether a CT image is COVID +ve or not [3, 4, 8, 19–21, 29, 32, 34].

Soares et al. [28] build a publically available CT scan data for severe acute respiratory syndrome coronavirus 2 (SARS-COV-2) comprising a total of 2482 images. The authors proposed a non-iterative algorithm named eXplainable deep network (x-DNN) based on recursive calculation for the categorization of data as non-COVID and COVID +ve. The authors attained a promising F1 value of 97.31% over the validation data.

Nour et al. [21] developed a diagnosis model for SARS-COV-2 infection detection using deep discriminative features and a Bayesian algorithm. The authors used a five convolutional layered model as a deep feature extractor; thereafter, the features were fed to the standard K-nearest neighbor (KNN), support vector machine (SVM), and decision trees whose hyperparameters were fine-tuned with Bayesian optimization procedure. The authors validated their method on the publically available X-ray image dataset and concluded that the SVM algorithm provided the best prediction resulting in an accuracy of 98.97% under the data partitioning ratio of 7:3.

He et al. [11] introduced a dataset of CT scan images with hundreds of COVID +ve images freely available for research. The authors also developed the Self-Trans method integrating self-supervised contrastive learning with the transfer learning-based mechanism for the separation of infected scans from normal ones. The researchers attained an F1-value of 0.85 using a split proportion of 0.6, 0.25, and 0.15 for training, testing, and validation.

Saygılı [25] proposed an automated system for separating COVID +ve CT scans from the non-infected scan images. The authors followed the pipeline of data set acquisition, image pre-processing that includes rgb2gray transformation, image resizing & image sharpening, Feature extraction that includes Local Binary Patterns & histograms of the Oriented Gradients, Feature reduction using Principal Component Analysis, and classification using standard machine learning (ML) algorithms. They achieved an accuracy of 98.11% on the dataset provided by Soares et al. [28] using the handcrafted features with a tenfold data partitioning scheme.

Kaur and Gandhi [13] designed a method for COVID detection based upon concatenation of deep features from the learned ResNet50 and learned MobileNetv2 model. The deep features were deduced by taking activations from the ‘avg_pool’ and the ‘Conv1’ layer of the learned models. Thereafter, the features were concatenated and given as input to the SVM for classification. The feature fusion approach, although with the high dimensionality of 63,232 yielded a validation accuracy of 98.35% on the benchmark COVID CT dataset.

In another work via Kaur et al. [15], they developed a diagnosis scheme for COVID-19 signature detection using deep features and Parameter Free-BAT (PF-BAT) enhanced Fuzzy-KNN (FKNN). Firstly, the pre-trained MobileNetv2 model was fine-tuned on CT chest radiographs. Thereafter, the features were extracted by performing activations onto the fully connected layer of the fine-tuned model. The features along with the corresponding label were fed to the FKNN classifier whose hyperparameters, i.e., nearest neighbor (‘ k ’) and fuzzy strength measure (‘ m ’) were fine-tuned via PF-BAT. Experimenting on the dataset by Soares et al. [28] reveals that the system achieved an average accuracy of 99.18%.

Goel et al. [7] designed an automated method for SARS-COV-2 detection using a framework that employs Generative Adversarial Network (GAN) for augmentation, Whale optimization for hyperparameter tuning of GAN network, and classification using transfer learned Inception V3 model. The researchers achieved a prediction accuracy value of 99.22% on the benchmark CT scan dataset using train test splits as 7:3.

Sen et al. [26] developed a feature selection approach for COVID-19 signature detection from lung CT scans. The detection framework uses a Convolutional Neural Network (CNN) architecture as a deep feature extractor. Thereafter, feature selection was done in two stages, i.e., firstly filter-based method is used then the Dragonfly optimization algorithm was applied over the ranked features. The selected features were then used by SVM for classification. The prediction rate was 90% and 98.39% on the benchmark CT scan image datasets.

Surveying the recent literature divulges that the use of CT images is evolving rapidly for COVID-19 detection owing to the shortage and the limited sensitivity of the RT-PCR detection kits [17]. Furthermore, to automate and expedite the screening of distinctive CT scans, the concept of transfer learning has proven to be quite advantageous, especially if the size of the dataset is limited [13, 15]. Additionally, to generalize the performance of the different algorithms on the benchmark CT scan dataset, a public database was provided by Soares et al. [28]. The survey also highlights that the predicted accuracy over this dataset is still limited, and there is a scope to improvise this rate further via effective mathematical models.

With the inspiration to improvise the prediction results, variants of the ResNet model are explored for detection. ResNet model variants are selected because of their proven better performance for the SARS-COV-2 detection task using CT scans [19].

Additionally, the potential of the learned image features from the transfer learned model is also investigated for COVID-19 detection when combined with the established ML algorithms (SVM, KNN, Logistic Regression (LR)). A classification fusion

Table 1 Train validation splits

| Data split | Non-COVID | COVID | Cumulative |
|------------|-----------|-------|------------|
| Training | 983 | 1003 | 1986 |
| Validation | 246 | 250 | 496 |

mechanism is also proposed that combines the predictions from the different ML algorithms via majority voting. Summarizing the vital contributions of the proposed work are:

1. A comparative study is performed with various architectures of Deep CNN's (DCNN) like ResNet18, ResNet50, and ResNet101 for COVID-19 detection using the transfer learning concept.
2. Results revealed that the transfer learning-based deep ResNet50 model exhibited the best performance w.r.t the other residual network variants.
3. The potential of the image attributes extracted from the various layers of the transfer learned model is also investigated by combining them with well-established ML algorithms like SVM, KNN, and LR.
4. A classification fusion scheme is also proposed that combines the predictions from the different classifiers via majority voting to further boost the classification performance.

In this present work, Sect. 2 provides the materials and methods, followed by Sect. 3, which describes the experimental results. Discussion and conclusions are illustrated in Sects. 4 and 5, respectively.

2 Materials and Methods

2.1 COVID Dataset Depiction

The proposed method has been analyzed on the publically available benchmark COVID CT scan dataset provided by Soares et al. [28]. The statistics relating to the data are provided in Table 1. Sample CT scans are shown in Fig. 1.

2.2 Performance Measures

In the present paper, Accuracy, Recall, Precision, F-score, and Area under the Curve (AUC) are selected to quantitatively validate the competence of the designed method.

2.3 Transfer Learning

One of the effective mechanisms making use of pre-trained deep models for image classification under limited data scenarios is transfer learning. In the present work, variants of a pre-trained ResNet model are used [11]. ResNet models follow a forward neural network architecture with “shortcut connections” to train CNN [10, 24].

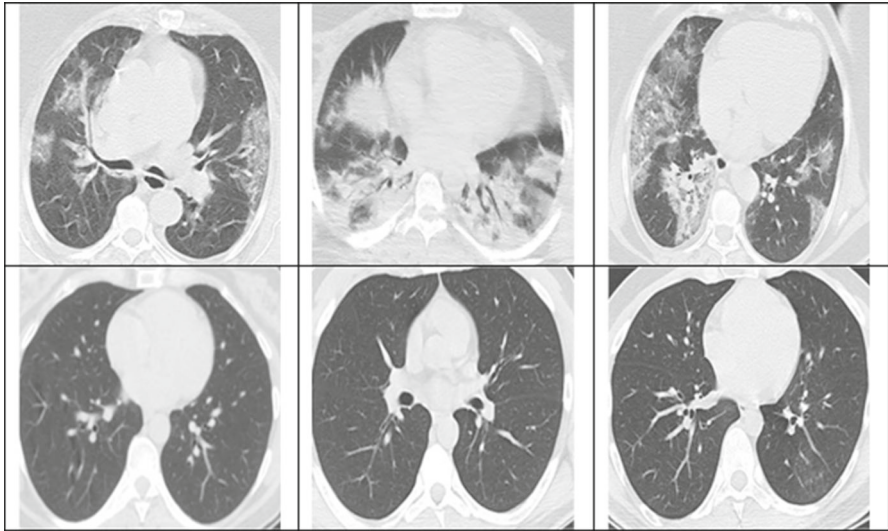


Fig. 1 CT scans from the dataset COVID + ve (upper row) and non-infected by SARS-COV-2 (lower row)

Through these “shortcut connections” gradients can easily propagate, which makes the training faster. Primarily, ResNet18 [24], ResNet50 [10], and ResNet101 [10] with transfer learning are used in this work as they have exhibited better performance than other computational models [19]. Only the last three layers of the model variants are fine-tuned to accommodate new image categories, as shown in Fig. 2 [33].

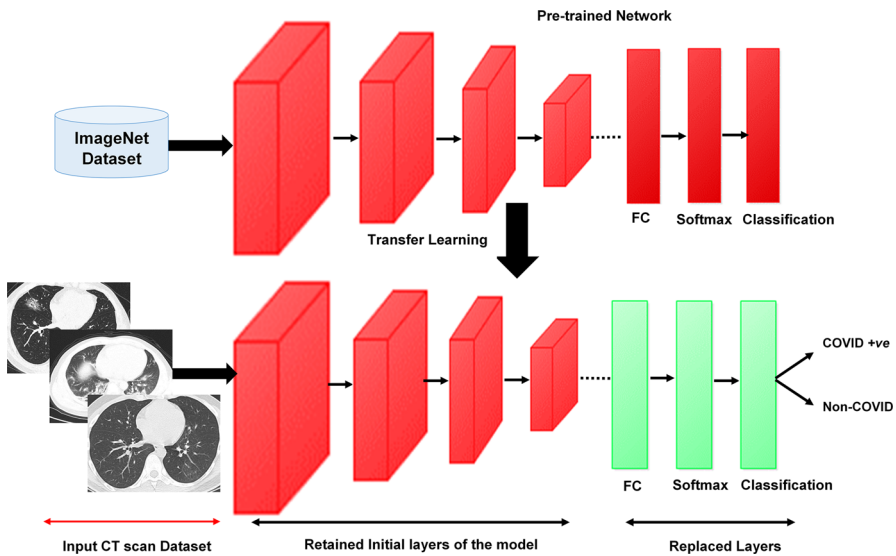


Fig. 2 Visualization for transfer learning using pre-trained models [14]

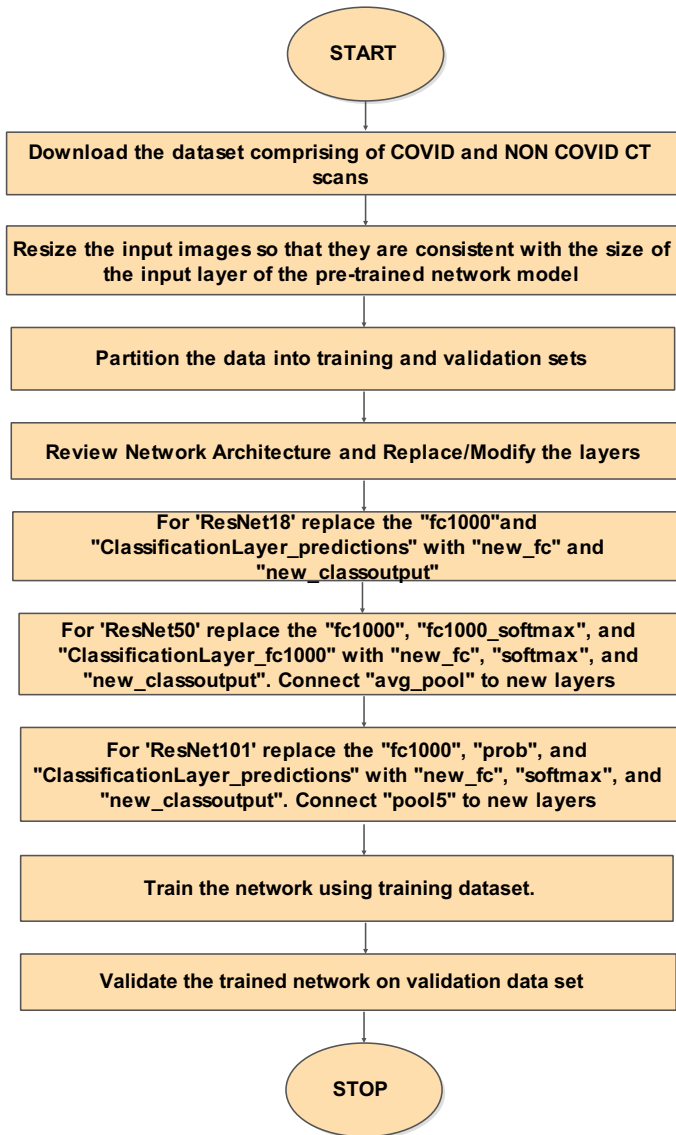


Fig. 3 Proposed methodology

2.4 Feature Classification Using Conventional Machine Learning Algorithms

Apart from using the transfer learned model for classification, the potential of learned image features is also explored for COVID-19 detection. The network activations from different layers of the learned model are taken, and then, they are fed to the proven state-of-the-art classifiers like SVM, KNN, and the LR. The following section briefly outlines the mathematical formulation for SVM, KNN, and LR.

2.4.1 Support Vector Machine (SVM)

SVM is among the most popular supervised ML algorithms that function by constructing an optimal hyperplane [9]. Let M training examples (p_i, z_i) be represented in an N -dimensional sample space, where p_i is an example pattern and $z_i \in \{-1, 1\}$ is the label. Let the kernel value matrix be represented by K and α_i be the Lagrange coefficients to be calculated via optimization procedure. Solving the quadratic equation given below results in an optimal separating hyperplane

$$\max W(\alpha) = -\frac{1}{2} \sum_i^M \sum_j^M \alpha_i \alpha_j z_i z_j K(p_i; p_j) + \sum_i^M \alpha_i \quad (1)$$

The equation given above is subject to the constrain $0 \leq \alpha_i \leq C, \forall i$ and $\sum_i^M \alpha_i z_i = 0$.

2.4.2 K-Nearest Neighbor (KNN)

KNN is one of the most simplistic nonparametric pattern recognition technique [5, 18]. In the KNN algorithm, a label is assigned according to the most common labels from its k -nearest neighbors. The main advantages of the KNN classifier are its simple implementation and fewer parameters to tune, i.e., distance metric and k .

Step by Step procedure for KNN algorithm:

- (i) Initialize the number of nearest neighbors (k).
- (ii) Compute the distance between the test image and all the training images. Any distance criteria could be used. E.g. Euclidean distance is primarily used and it is governed by the equation

$$\text{Distance}(a, b) = \|a - b\| \quad (2)$$

where (a, b) are two different samples in the feature space.

- (iii) Sort the distances and calculate the nearest neighbor based on the k^{th} minimum distance.
- (iv) Get the corresponding labels of the training data which falls under k for the sorted condition.
- (v) Take the majority of k -nearest neighbors as the output label.

2.4.3 Logistic Regression (LR)

LR method uses a sigmoid or a logistic function. Considering a 2 class problem with $f_0(X), f_1(X)$ as the class conditional densities, $q_0(X), q_1(X)$ as posterior probabilities, and $p_0(X), p_1(X)$ as prior probabilities, then according to Bayes rule [2]

$$q_0(X) = \frac{f_0(X)p_0}{f_0(X)p_0 + f_1(X)p_1} = \frac{1}{1 + \exp(-\xi)} \quad (3)$$

where ξ is defined as

$$\xi = -\ln\left(\frac{f_1(X)p_1}{f_0(X)p_0}\right) = \ln\left(\frac{f_0(X)p_0}{f_1(X)p_1}\right) \quad (4)$$

Alternatively,

$$\ln\left(\frac{f_0(X)p_0}{f_1(X)p_1}\right) = W^T X + w_0 \quad (5)$$

The above equation holds if f_0 and f_1 are Gaussian with similar covariance matrix. This is the case of logistic regression where it would result in an optimal classifier. In logistic regression, the goal is to find W and w_0 that minimizes $\frac{1}{2} \sum_{i=1}^n (h(W^T X_i + w_0) - y_i)^2$ where $h(a) = (1 + \exp(-a))^{-1}$ is the sigmoid or logistic function and $y_i \in \{0, 1\}$ are the targets.

3 Experimental Results

3.1 Experimental Setup

The experimental settings for the proposed work include using an ‘Adam’ optimizer with the initial learning rate of 0.0006 to minimize the cross-entropy loss. The variant models were fitted for 20-epochs with 150 as the batch size. The image data were processed on an Intel Core i7-4500U CPU having 8 GB RAM, with a 1.8 GHz processor in a MATLAB 19a platform. Other training options include shuffling the data before each epoch using L2-regularizer with a weight decay value of 0.05 to circumvent overfitting. Moreover, the data splits, i.e., train/validation were used as per the base paper by Soares et al. [28]. The researchers in Soares et al. [28] have given the .mat files for train-validation splits that do not contain subject-related information, i.e., without any metadata on the images. The images are only designated by numbers without any indication of whether they are subject-independent or not.

3.2 Results

The experimental results via deploying the learned ResNet model variants for the COVID-19 detection task are given in Table 2. The tabulated entries show that the best

Table 2 Performance comparison of the ResNet model variants over validation data

| Sr. no | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC |
|-----------|--------------|---------------|------------|--------------|--------|
| ResNet18 | 97.12 | 96.83 | 97.60 | 97.21 | 0.9973 |
| ResNet50 | 98.35 | 98.02 | 98.80 | 98.41 | 0.9994 |
| ResNet101 | 96.71 | 96.43 | 97.20 | 96.81 | 0.9944 |

| | | | | | | | | | |
|-----|-----------|-----------|-----------|-----|-----------|-----------|-----------|-----------|-----------|
| (a) | Actual | | | (b) | Actual | | | | |
| | Predicted | | COVID +ve | | NON COVID | Predicted | | COVID +ve | NON COVID |
| | | COVID +ve | 244 | | 8 | | COVID +ve | 247 | 5 |
| | | Non COVID | 6 | | 228 | | Non COVID | 3 | 231 |
| (c) | Actual | | | (d) | Actual | | | | |
| | Predicted | | COVID +ve | | NON COVID | Predicted | | COVID +ve | NON COVID |
| | | COVID +ve | 243 | | 9 | | COVID +ve | 248 | 1 |
| | | Non COVID | 7 | | 227 | | Non COVID | 2 | 235 |

Fig. 4 Confusion matrix a ResNet18 b ResNet50 c ResNet101 d classification fusion

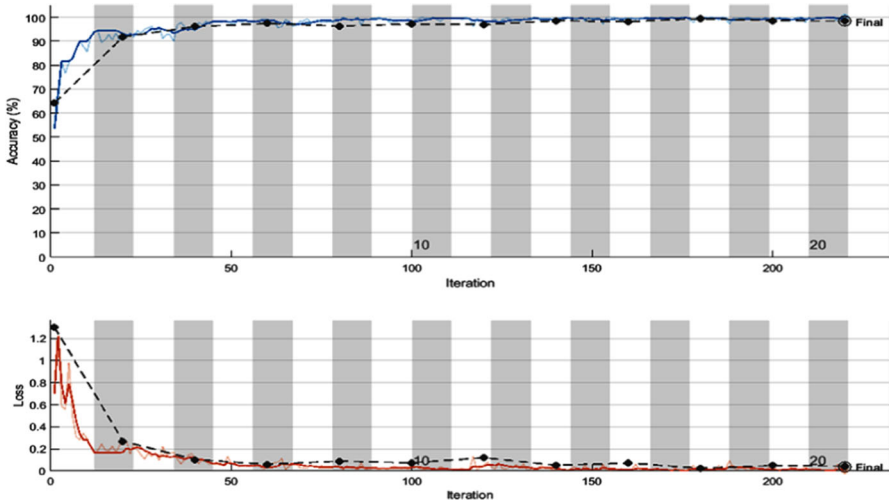


Fig. 5 Training versus epoch and loss versus epoch plot for best performing transfer learned model (ResNet50)

results are attained by the ResNet50 model. It has achieved a precision of 98.02%, recall of 98.80%, AUC of 0.9994, F1-score of 98.41%, and a validation accuracy of 98.35%. Other models that are taken for comparison, i.e., ResNet18 and ResNet101 achieved a validation accuracy of 97.12% and 96.71%, respectively. The confusion matrix for all the three learned variants is given in Fig. 4. The smallest misclassification error is achieved by ResNet50 followed by ResNet18, and then ResNet101. The accuracy/loss versus epoch plot for the best performing model is given in Fig. 5, which indicates that validation closely follows the training. The AUC plot for ResNet50 is shown in Fig. 6, which specifies that the model attains a value of 0.9994 for True Positive Rate Vs False Positive fraction.

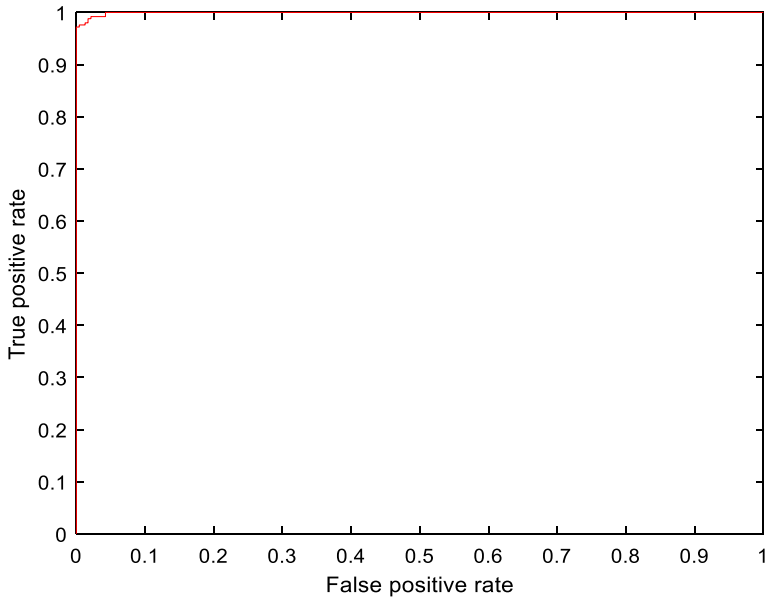


Fig. 6 AUC curve for best performing transfer learned model (ResNet50)

Figure 7 shows the occlusion sensitivity maps for the learned ResNet50 model. It gives us an idea about which area of the scan is most decisive for classification, i.e., occluding which results in a maximum drop in the probability score. The regions that are positively contributing to the probability score are shown in red color.

In the present work, we also tried to investigate the efficacy of the activations from the several layers of the transfer learned ResNet50 model for the COVID-19 detection task. These activations from the various layers of the trained model are used in conjunction with well-established classifiers such as SVM, KNN, and LR.

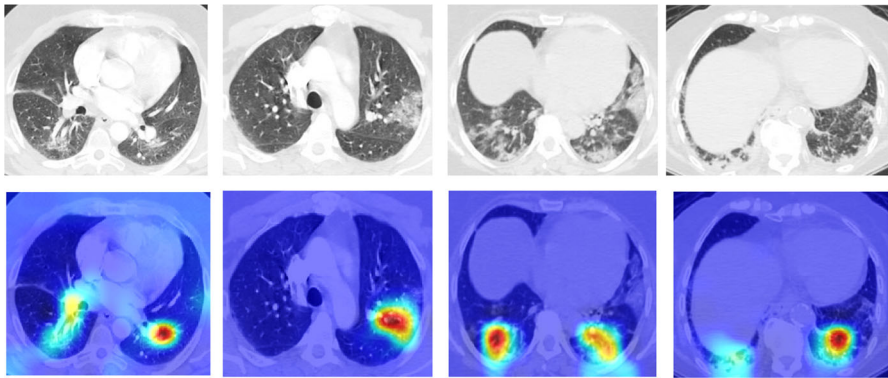


Fig. 7 Activation maps obtained via transfer learned ResNet50 model

Beginning from the fully connected layer, activations from specific layers are taken and the detection results are reported in Table 3. Interestingly, e.g., the activations extracted from the layer ‘res5b_branch2c’ proved to be decisive for the classification using SVM and the LR classifier. It rendered a value of 99.20% for precision, recall, F1-score, and a value of 99.18%, 0.9997 for accuracy, and AUC.

To further improve the classification performance, a fusion strategy is also proposed as outlined in Fig. 8, where predictions from the three different classifiers are fused according to the majority voting rule. The confusion matrix resulting from the classification fusion is shown in Fig. 4d indicating that the misclassifications are reduced to merely three samples.

On fusing the predictions from SVM, KNN, and LR using features from the ‘res5b_branch2c’ layer, a validation accuracy of 99.38% and F1-score of 99.40% is achieved. The advantage of classification fusion is also highlighted for the activations from ‘avg_pool’ layer where a validation accuracy of 99.38% is achieved using a feature dimension of 2048 rather than using a feature space of 100,352.

4 Discussion

The proposed fusion approach has also been compared with other model architectures that have used the same dataset. As apparent from Table 4, the proposed prediction fusion mechanism yields a precision of 99.60%, F-score of 99.40%, Recall of 99.20%, Accuracy of 99.38% that is higher than the existing network architectures (detection accuracy of 97.38%, 98.35%, 98.39%, 98.37%, 98.99%, 94.04%, 92% reported in Kaur and Gandhi [13], Panwar et al. [22], Pathak et al. [23], Silva et al. [27], Soares et al. [28], Fouladi et al. [6], Sen et al. [26]). In contrast to the reported works in literature, the proposed prediction fusion mechanism offers some benefits in addition to achieving a promising level of detection performance: (1) It employs a single pre-trained network architecture for COVID-19 classification differing from the usage of both VGG-16 and x DNN proposed in Soares et al. [28]. (2) It does not employ any optimization procedure for fine-tuning the model hyperparameters as in Kaur et al. [15], Pathak et al. [23] where the authors have employed PF-BAT and Memetic Adaptive Differential Evolution optimization for FKNN and deep bidirectional long short-term memory network with a Mixture Density model (DBM) hyperparameter tuning.

Table 3 Performance metrics for transfer learned ResNet50 using activations from specific layers

| Layer | Classifier | Precision (%) | Recall (%) | Accuracy (%) | F1-score (%) | AUC | Dimension of training feature space | Dimension of validation feature space |
|------------------|-------------------|---------------|--------------|--------------|--------------|---------------|-------------------------------------|---------------------------------------|
| 'fc' | SVM | 98.02 | 98.80 | 98.35 | 98.41 | 0.9932 | 1986 × 2 | 486 × 2 |
| | KNN | 97.24 | 98.80 | 97.94 | 98.02 | 0.9792 | | |
| | LR | 98.02 | 98.80 | 98.35 | 98.41 | 0.9994 | | |
| 'avg_pool' | Classifier fusion | 98.02 | 98.80 | 98.35 | 98.41 | 0.9994 | | |
| | SVM | 98.41 | 99.20 | 98.77 | 98.80 | 0.9995 | 1986 × 2048 | 486 × 2048 |
| | KNN | 98.80 | 99.20 | 98.97 | 99.00 | 0.9896 | | |
| 'res5c_branch2c' | LR | 99.20 | 99.20 | 99.18 | 99.20 | 0.9993 | | |
| | Classifier fusion | 99.60 | 99.20 | 99.38 | 99.40 | 0.9997 | 1986 × 100,352 | 486 × 100,352 |
| | SVM | 98.02 | 98.80 | 98.35 | 98.41 | 0.9975 | | |
| 'res5c_branch2b' | KNN | 99.59 | 97.60 | 98.56 | 98.59 | 0.9859 | | |
| | LR | 98.41 | 99.20 | 98.77 | 98.80 | 0.9993 | | |
| | Classifier fusion | 98.41 | 99.20 | 98.77 | 98.80 | 0.9993 | | |
| 'res5c_branch2a' | SVM | 98.42 | 99.60 | 98.97 | 99.01 | 0.9997 | 1986 × 25,088 | 486 × 25,088 |
| | KNN | 98.81 | 99.60 | 99.18 | 99.20 | 0.9916 | | |
| | LR | 99.20 | 99.20 | 99.18 | 99.20 | 0.9995 | | |
| 'res5c_branch2a' | Classifier fusion | 99.60 | 99.20 | 99.38 | 99.40 | 0.9997 | 1986 × 25,088 | 486 × 25,088 |
| | SVM | 98.41 | 98.80 | 98.56 | 98.60 | 0.9997 | | |
| | KNN | 98.02 | 99.20 | 98.56 | 98.61 | 0.9854 | | |

Table 3 (continued)

| Layer | Classifier | Precision (%) | Recall (%) | Accuracy (%) | F1-score (%) | AUC | Dimension of training feature space | Dimension of validation feature space |
|------------------|-------------------|---------------|--------------|--------------|--------------|---------------|-------------------------------------|---------------------------------------|
| 'res5b_branch2c' | LR | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |
| | Classifier fusion | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |
| | SVM | 99.20 | 99.20 | 99.18 | 99.20 | 0.9997 | 1986 × 100,352 | 486 × 100,352 |
| | KNN | 98.02 | 99.20 | 98.56 | 98.61 | 0.9854 | | |
| | LR | 99.60 | 98.80 | 99.18 | 99.20 | 0.9997 | | |
| 'res5b_branch2b' | Classifier fusion | 99.60 | 99.20 | 99.38 | 99.40 | 0.9997 | | |
| | SVM | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | 1986 × 25,088 | 486 × 25,088 |
| | KNN | 97.66 | 100 | 98.77 | 98.81 | 0.9873 | | |
| 'res5b_branch2a' | LR | 97.64 | 99.20 | 98.35 | 98.41 | 0.9996 | | |
| | Classifier fusion | 97.64 | 99.20 | 98.35 | 98.41 | 0.9996 | | |
| | SVM | 98.02 | 99.20 | 98.56 | 98.61 | 0.9997 | 1986 × 25,088 | 486 × 25,088 |
| | KNN | 99.20 | 99.20 | 99.18 | 99.20 | 0.9997 | | |
| | LR | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |
| 'res5a_branch1' | Classifier fusion | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |
| | SVM | 98.02 | 99.20 | 98.56 | 98.61 | 0.9994 | 1986 × 100,352 | 486 × 100,352 |
| | KNN | 97.27 | 99.60 | 98.35 | 98.42 | 0.9832 | | |
| | LR | 96.88 | 99.20 | 97.94 | 98.02 | 0.9997 | | |
| 'res5a_branch2c' | Classifier fusion | 98.02 | 99.20 | 98.56 | 98.61 | 0.9994 | | |
| | SVM | 98.02 | 99.20 | 98.56 | 98.61 | 0.9996 | 1986 × 100,352 | 486 × 100,352 |
| | KNN | 98.80 | 99.20 | 98.97 | 99.00 | 0.9896 | | |
| | LR | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |

Table 3 (continued)

| Layer | Classifier | Precision (%) | Recall (%) | Accuracy (%) | F1-score (%) | AUC | Dimension of training feature space | Dimension of validation feature space |
|------------------|-------------------|---------------|--------------|--------------|--------------|---------------|-------------------------------------|---------------------------------------|
| 'res5a_branch2b' | Classifier fusion | 98.41 | 99.20 | 98.77 | 98.80 | 0.9997 | | |
| | SVM | 98.80 | 99.20 | 98.97 | 99.00 | 0.9994 | 1986 × 25,088 | 486 × 25,088 |
| | KNN | 97.65 | 99.60 | 98.56 | 98.61 | 0.9853 | | |
| | LR | 98.80 | 98.80 | 98.77 | 98.80 | 0.9992 | | |
| 'res5a_branch2a' | Classifier fusion | 98.80 | 99.20 | 98.97 | 99.00 | 0.9896 | | |
| | SVM | 98.80 | 99.20 | 98.97 | 99.00 | 0.9994 | 1986 × 25,088 | 486 × 25,088 |
| | KNN | 96.88 | 99.20 | 97.94 | 98.02 | 0.9791 | | |
| | LR | 99.19 | 98.40 | 98.77 | 98.80 | 0.9994 | | |
| 'res4f_branch2c' | Classifier fusion | 99.20 | 99.20 | 99.18 | 99.20 | 0.9997 | | |
| | SVM | 98.80 | 98.80 | 98.77 | 98.80 | 0.9991 | 1986 × 200,704 | 486 × 200,704 |
| | KNN | 93.80 | 90.80 | 92.18 | 92.28 | 0.9222 | | |
| | LR | 98.41 | 98.80 | 98.56 | 98.60 | 0.9992 | | |
| 'res4f_branch2b' | Classifier fusion | 98.80 | 98.80 | 98.77 | 98.80 | 0.9991 | | |
| | SVM | 98.02 | 98.80 | 98.35 | 98.41 | 0.9989 | 1986 × 50,176 | 486 × 50,176 |
| | KNN | 92.28 | 95.60 | 93.62 | 93.91 | 0.9356 | | |
| | LR | 98.40 | 98.40 | 98.35 | 98.40 | 0.9988 | | |
| 'res4f_branch2a' | Classifier fusion | 98.40 | 98.40 | 98.35 | 98.40 | 0.9988 | | |
| | SVM | 94.30 | 99.20 | 96.50 | 96.69 | 0.9977 | 1986 × 50,176 | 486 × 50,176 |
| | KNN | 84.96 | 90.40 | 86.83 | 87.60 | 0.8673 | | |
| | LR | 94.64 | 98.80 | 96.50 | 96.67 | 0.9974 | | |
| | Classifier fusion | 88.55 | 87.22 | 86.83 | 87.88 | 0.8669 | | |

Bold signifies the best results

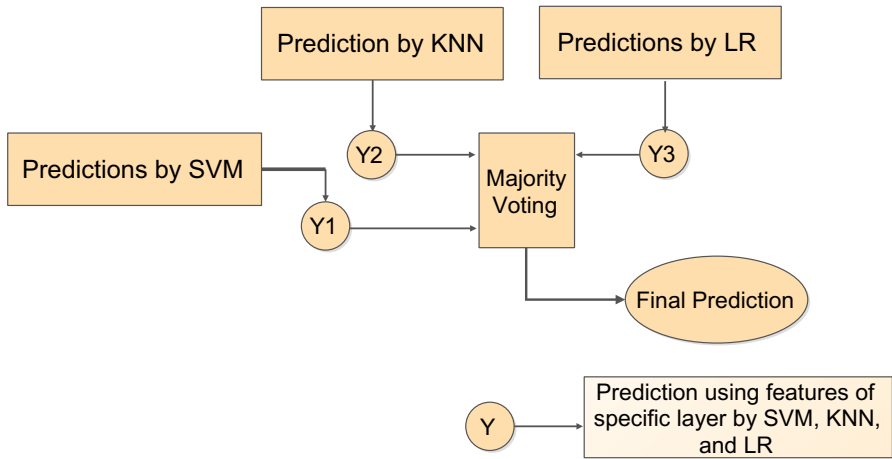


Fig. 8 Classification fusion strategy

Table 4 Comparison of the proposed prediction fusion scheme with the recent state-of-the-art works

| Method | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) | AUC |
|--|--------------|---------------|------------|--------------|--------|
| xDNN [28] | 97.38 | 99.16 | 95.53 | 97.31 | 0.9736 |
| Alexnet [28] | 93.75 | 94.98 | 92.28 | 93.61 | 0.9368 |
| VGG16 [28] | 94.96 | 94.02 | 95.43 | 94.97 | 0.9496 |
| GoogleNet [28] | 91.73 | 90.20 | 93.50 | 91.82 | 0.9179 |
| AdaBoost [28] | 95.16 | 93.63 | 96.71 | 95.14 | 0.9519 |
| Decision Tree [28] | 79.44 | 76.81 | 83.13 | 79.84 | 0.7951 |
| EfficientNet [27] | 98.99 | 99.20 | 98.80 | – | – |
| DBM [23] | 97.23 | 98.14 | 97.68 | 97.89 | 0.9771 |
| DBM + MADE [23] | 98.37 | 98.74 | 98.87 | 98.14 | 0.9832 |
| MobileNetv2 + SVM [13] | 98.35 | 97.64 | 99.20 | 98.41 | 0.9912 |
| MobileNetv2 + PF-BAT enhanced FKNN [15] | 99.38 | 99.20 | 99.60 | 99.40 | 0.9958 |
| CNN+ bi-stage feature selection + SVM [26] | 98.39 | 98.21 | 97.78 | 98 | 0.9952 |
| VGG19 [22] | 94.04 | 95.00 | 94.00 | 94.50 | – |

Table 4 (continued)

| Method | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) | AUC |
|--|--------------|---------------|------------|--------------|--------|
| U-Net++ [6] | 92 | – | 100 | – | – |
| Proposed | 98.35 | 98.02 | 98.80 | 98.41 | 0.9994 |
| Proposed (with features from ‘res5b branch2c’) | 99.18 | 99.20 | 99.20 | 99.20 | 0.9997 |
| Proposed (with classification fusion) | 99.38 | 99.60 | 99.20 | 99.40 | 0.9997 |

5 Conclusion

In the present paper, we have investigated the efficacy of different pre-trained network architectures with transfer learning for COVID-19 detection using a limited CT scan dataset. Investigation reveals that the transfer learned ResNet50 model turned out to be the finest by achieving an accuracy value of 98.35% that is superior to the considered models and the existing state-of-the-art works in the literature. Moreover, the potential of the activations from different layers of the learned ResNet50 network is also explored for detection using the established ML algorithms. The exploration reveals that the activations from some of the specific layers of the learned ResNet50 model are quite decisive for classification yielding an accuracy of 99.18% using SVM and LR classifiers. A classification fusion strategy is also proposed that further improvised the accuracy to 99.38% by combining the predictions from the different classifiers via majority voting.

The proposed automated system can assist the healthcare professionals in rapid detection of the virus at different stages.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability The datasets generated during and/or analyzed during the current study are available in the repository available at the following link: <https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. A. Bernheim, X. Mei, M. Huang, Y. Yang, Z.A. Fayad, N. Zhang, K. Diao, B. Lin, X. Zhu, K. Li et al., Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection. *Radiology* **66**, 200463 (2020)

2. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006)
3. J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao, S. Hu, Y. Wang, X. Hu, B. Zheng et al., Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: a prospective study. *Sci. Rep.* **10**(1), 1–11 (2020)
4. J. Choe, S.M. Lee, K.-H. Do, G. Lee, J.-G. Lee, S.M. Lee, J.B. Seo, Deep learning–based image conversion of CT Reconstruction Kernels Improves Radiomics Reproducibility For Pulmonary Nodules Or Masses. *Radiology* **292**(2), 365–373 (2019)
5. T. Cover, P. Hart, Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**(1), 21–27 (1967)
6. S. Fouladi, M.J. Ebadi, A.A. Safaei, M.Y. Bajuri, A. Ahmadian, Efficient deep neural networks for classification of COVID-19 based on CT images: virtualization via software defined radio. *Comput. Commun.* **176**, 234–248 (2021)
7. T. Goel, R. Murugan, S. Mirjalili, D.K. Chakrabarty, Automatic screening of covid-19 using an optimized generative adversarial network. *Cognit. Comput.* **66**, 1–16 (2021)
8. O. Gozes, M. Frid-Adar, H. Greenspan, P.D. Browning, H. Zhang, W. Ji, A. Bernheim, E. Siegel, Rapid ai development cycle for the coronavirus (covid-19) pandemic: initial results for automated detection & patient monitoring using deep learning ct image analysis (2020). arXiv Prepr [arXiv:200305037](https://arxiv.org/abs/200305037)
9. H.P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, V. Vapnik, Parallel support vector machines: the cascade SVM, in *Advances in Neural Information Processing Systems* (2005), pp. 521–528
10. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778
11. X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, P. Xie, Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. *medRxiv* **66**, 1–10 (2020)
12. C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu et al., Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* **395**(10223), 497–506 (2020)
13. T. Kaur, T.K. Gandhi, Automated diagnosis of COVID-19 from CT scans based on concatenation of Mobilenetv2 and ResNet50 features, in *CVIP* (No. 1) (2020), pp. 149–160
14. T. Kaur, T.K. Gandhi, Deep convolutional neural networks with transfer learning for automated brain image classification. *Mach. Vis. Appl.* **31**(3), 1–16 (2020)
15. T. Kaur, T.K. Gandhi, B.K. Panigrahi, Automated diagnosis of COVID-19 using deep features and parameter free BAT optimization. *IEEE J. Transl. Eng. Heal. Med.* **9**, 1–9 (2021)
16. H.J. Koo, S. Lim, J. Choe, S.-H. Choi, H. Sung, K.-H. Do, Radiographic and CT features of viral pneumonia. *Radiographics* **38**(3), 719–739 (2018)
17. R. Kundu, H. Basak, P.K. Singh, A. Ahmadian, M. Ferrara, R. Sarkar, Fuzzy rank-based fusion of CNN models using Gompertz function for screening COVID-19 CT-scans. *Sci. Rep.* **11**(1), 1–12 (2021)
18. D.Y. Liu, H.L. Chen, B. Yang, X.E. Lv, L.N. Li, J. Liu, Design of an enhanced fuzzy k-nearest neighbor classifier based computer aided diagnostic system for thyroid disease. *J. Med. Syst.* **36**(5), 3243–3254 (2012)
19. M. Loey, F. Smarandache, N.E.M. Khalifa, A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images. *Neural Comput. Appl.* **66**, 1–13 (2020)
20. H. Mohammad-Rahimi, M. Nadimi, A. Ghalyanchi-Langeroudi, M. Taheri, S. Ghafouri-Fard, Application of machine learning in diagnosis of COVID-19 through X-ray and CT images: a scoping review. *Front. Cardiovasc. Med.* **8**, 185 (2021)
21. M. Nour, Z. Cömert, K. Polat, A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization. *Appl. Soft. Comput.* **97**, 106580 (2020)
22. H. Panwar, P.K. Gupta, M.K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, V. Singh, A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos Solitons Fract.* **140**, 110190 (2020)
23. Y. Pathak, P.K. Shukla, K.V. Arya, Deep bidirectional classification model for COVID-19 disease infected patients. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **6**, 66 (2020)
24. O.A.B. Penatti, K. Nogueira, J.A. Dos Santos, Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2015) pp. 44–51
25. A. Saygılı, A new approach for computer-aided detection of coronavirus (COVID-19) from CT and X-ray images using machine learning methods. *Appl. Soft Comput.* **105**, 107323 (2021)
26. S. Sen, S. Saha, S. Chatterjee, S. Mirjalili, R. Sarkar, A bi-stage feature selection approach for COVID-19 prediction using chest CT images. *Appl. Intell.* **66**, 1–16 (2021)

27. P. Silva, E. Luz, G. Silva, G. Moreira, R. Silva, D. Lucio, D. Menotti, COVID-19 detection in CT images with deep learning: a voting-based scheme and cross-datasets analysis. *Inform. Med. Unlocked* **66**, 100427 (2020)
28. E. Soares, P. Angelov, S. Biaso, M.H. Froes, D.K. Abe, SARS-CoV-2 CT-scan dataset: a large dataset of real patients CT scans for SARS-CoV-2 identification. *medRxiv* **66**, 1–8 (2020)
29. Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, H. Zhao, Y. Jie, R. Wang et al., Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *medRxiv* **6**, 66 (2020)
30. Z. Tan, X. Li, M. Gao, L. Jiang, The environmental story during the COVID-19 lockdown: how human activities affect PM2.5 concentration in China? *IEEE Geosci. Remote. Sens. Lett.* **6**, 66 (2020)
31. T.P. Velavan, C.G. Meyer, The COVID-19 epidemic. *Trop. Med. Int. Heal.* **25**(3), 278 (2020)
32. S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng et al., A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *Eur. Radiol.* **66**, 1–9 (2021)
33. J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in *Advances in Neural Information Processing Systems* (2014), pp. 3320–3328
34. J. Zhao, Y. Zhang, X. He, P. Xie, COVID-CT-Dataset: a CT scan dataset about COVID-19 (2020). *arXiv Prepr* [arXiv:200313865](https://arxiv.org/abs/200313865)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.