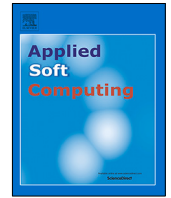




Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



# InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray



Anunay Gupta <sup>a</sup>, Anjum <sup>b</sup>, Shreyansh Gupta <sup>c</sup>, Rahul Katarya <sup>b,\*</sup>

<sup>a</sup> Department of Electrical Engineering, Delhi Technological University, New Delhi, India

<sup>b</sup> Department of Computer Science, Delhi Technological University, New Delhi, India

<sup>c</sup> Department of Civil Engineering, Delhi Technological University, New Delhi, India

## ARTICLE INFO

### Article history:

Received 8 August 2020

Received in revised form 11 October 2020

Accepted 27 October 2020

Available online 29 October 2020

### Keywords:

InstaCovNet-19

COVID-19

Integrated stacking

Pneumonia

Convolution network

## ABSTRACT

Recently, the whole world became infected by the newly discovered coronavirus (COVID-19). SARS-CoV-2, or widely known as COVID-19, has proved to be a hazardous virus severely affecting the health of people. It causes respiratory illness, especially in people who already suffer from other diseases. Limited availability of test kits as well as symptoms similar to other diseases such as pneumonia has made this disease deadly, claiming the lives of millions of people. Artificial intelligence models are found to be very successful in the diagnosis of various diseases in the biomedical field. In this paper, an integrated stacked deep convolution network InstaCovNet-19 is proposed. The proposed model makes use of various pre-trained models such as ResNet101, Xception, InceptionV3, MobileNet, and NASNet to compensate for a relatively small amount of training data. The proposed model detects COVID-19 and pneumonia by identifying the abnormalities caused by such diseases in Chest X-ray images of the person infected. The proposed model achieves an accuracy of 99.08% on 3 class (COVID-19, Pneumonia, Normal) classification while achieving an accuracy of 99.53% on 2 class (COVID, NON-COVID) classification. The proposed model achieves an average recall, F1 score, and precision of 99%, 99%, and 99%, respectively on ternary classification, while achieving a 100% precision and a recall of 99% on the binary class, while achieving a 100% precision and a recall of 99% on the COVID class. InstaCovNet-19's ability to detect COVID-19 without any human intervention at an economical cost with high accuracy can benefit humankind greatly in this age of Quarantine.

© 2020 Elsevier B.V. All rights reserved.

## 1. Introduction

Novel Coronavirus (COVID-19) is another strain of the infection SARS-Cov-2. It is a zoonotic, positive-stranded RNA virus that causes severe acute respiratory distress and pneumonia-like symptoms in humans. It is a highly infectious virus that gets transmitted by aerosol and respiratory droplets. It has been declared as a pandemic by WHO affecting more than 227 countries [1]. Since its origin, the number of cases rose exponentially and arrived at in excess of 14 million cases around the world [2]. The worst-hit countries include — USA, Brazil, and India [2]. The typical manifestations of COVID-19 are — fever, dry hack, sleepiness, and loss of taste. The less regular indications are — loose bowels, conjunctivitis and headache. In extreme cases, the disease may cause pneumonia, trouble in breathing, multi-organ failure, and the demise of the patient. Due to the rapid growth in

the number of cases every day, medical facilities of even the most advanced countries are at a brink of collapsing. Aggressive testing is the most efficient way of controlling this pandemic. Right now, Reverse Transcription Polymerase Chain Reaction (RT-PCR) is being utilized for the identification of COVID-19 [3]. This method requires Respiratory specimens acquired from the subject's body. This method is Time-consuming, as well as costly. COVID-19 test cost more than \$500, and some even as high as \$2315 in the USA [4] and in India, the cost for a COVID-19 test varies from ₹1000–₹6500 [5]. Further, these tests have a low detection rate and hence need to be repeated for confirmation. A rapid diagnostic method is crucial for winning the fight against COVID-19. Another method for detection of COVID-19 is through Chest X-rays and Computed Tomography (CT). Studies have shown that abnormalities in the chest are caused due to COVID-19, which are visible in Chest X-rays in the form of ground-glass opacities. These opacities can further be used to detect COVID-19. Although this method has some benefits over the current RT-PCR test in terms of early detection, this method requires an expert to comprehend the X-ray images.

\* Corresponding author.

E-mail addresses: [aganunay@outlook.com](mailto:aganunay@outlook.com) (A. Gupta),

[anjum\\_2792@yahoo.com](mailto:anjum_2792@yahoo.com) (Anjum), [shreyansh.gupta1299@gmail.com](mailto:shreyansh.gupta1299@gmail.com) (S. Gupta),

[rahulkatarya@dtu.ac.in](mailto:rahulkatarya@dtu.ac.in) (R. Katarya).

In the following study, a deep learning-based Computer-aided diagnosis System (InstaCovNet-19) is presented to detect COVID-19 efficiently, economically and with low misclassification rates. The computer-aided diagnosis system presented in this paper makes use of fine-tuned pre-trained deep learning models to extract distinct features present in chest X-rays of COVID-19 patients. These extracted features are then made the basis of the detection of COVID-19. This COVID Diagnosis System (InstaCovNet-19) is internally a deep learning model which consist of various pre-trained models such as ResNet-101 [6], Inception v3 [7], Xception [8], MobileNetv2 [9] and NASNet [10]. These pre-trained models are then combined using the Integrated stacking technique, where the features are combined to produce the most accurate results. This technique constitutes InstaCovNet-19. InstaCovNet-19 is first trained and tested to classify X-ray images into three classes viz. COVID infected, Pneumonia infected and Normal. InstaCovNet-19 is then trained to perform binary classification that is classified X-ray classes into COVID and Non-COVID classes. The classification performance achieved by our proposed deep learning model InstaCovNet-19 makes it evident that X-ray images can be used for the detection of COVID-19 in a real-world scenario.

The major contributions of this paper are summarized below:

1. Proposed an Integrated Stacking InstaCovNet-19 Classification Model, which is used to classify patients affected by COVID-19 by considering their chest X-ray images.
2. InstaCovNet-19 was benchmarked against other state of the art models.
3. Various pre-processing and training techniques were employed to boost classification performance.
4. The proposed model was developed to help the medic to identify COVID-19 more effectively and efficiently.

The paper is organized as follows. Section 2 discusses the previous state of the art models designed for the detection of COVID-19, Section 3 clarifies about fundamental information required to study and implement the proposed model. Section 4 deeply explains the strategy employed and leveraged to create the proposed InstaCovNet-19 model, the architectural design and the implementational details of InstaCovNet-19. The experimental results, the dataset, image pre-processing techniques, and the comparative study of our proposed model are discussed in Section 5. In the last section, i.e., Section 6, conclusion and areas of study in the future are discussed in detail.

## 2. Related work

Artificial Intelligence has come a long way since the dawn of its era from basic digit recognition models [11] to recognize human activities [12] [13]. Nowadays, artificial intelligence is used for almost everything from Sentiment Analysis [14] [15] to Violence Detection [16], to genre classification of movies [17]. Recently there is a significant increase in the use of AI in health care, especially in medical imaging. There has been a significant rise in the amount of work done in medical imaging. Medical imaging has been used to detect cardiovascular diseases [18], brain tumors [19], and now medical imaging is also being used for detection of COVID-19. Artificial Intelligence-based diagnosis systems can help ease the burden on health professionals while increasing the detection rate of COVID-19; with this motivation, several studies have been conducted on Machine learning-based diagnosis systems. The major problems faced by machine learning-based models are the lack of efficient feature extraction and pre-processing of input images. In [20,21] the author showed that pre-processing through the Fuzzy color image enhancement technique can significantly improve feature extraction

of computer vision models and hence improve the classification performance.

The second major issue of feature extraction can be tackled by utilizing sophisticated deep learning techniques. Squeeze net, along with Bayesian optimization of parameters such as learning rate and momentum were utilized in [22]. Transfer learning on pre-trained Xception CNN architecture is utilized by authors of [23] to classify X-ray images into 4 classes, namely COVID-19, Pneumonia bacterial, pneumonia viral, Normal; the proposed model achieved an accuracy of 89.6 % on 4 class classification and 95% on 3 class classification. In [24], the authors utilized Resnet for feature extraction upon which a classification model is used to classify an image as COVID and Non-COVID. The cost-sensitive top-2 smooth loss function is used to improve the outcomes further. This model achieves an accuracy of 93.01%. 121 layered pre-trained DenseNet architecture (Chexnet) was utilized in [25] to detect pneumonia in 112,120 X-ray images of 30,805 unique patients; this model is then extended to detect 14 diseases in X-ray images. In [26], the Authors used a pre-trained inceptionV3 for the extraction of image embeddings and an artificial neural network for classification. The said structure was able to classify and segregate distinctive aspiratory diseases proficiently and achieved an extraordinarily high accuracy of 99.01%. A Deep domain adaption algorithm was proposed in [27], the authors in this algorithm extended a pneumonia classifier to detect COVID-19 disease by making use of both shared and distinct features of COVID-19 and pneumonia. This deep domain adaption algorithm achieved an AUC of 0.985 and an F1-Score of 92.98%. In [28], the authors utilized a pre-trained ResNet50 for feature extraction and SVM for classification and achieved an accuracy of 95.38% on binary classification. In [29], the authors used DarkNet and different filtering on each layer and achieved an accuracy of 98.08% on binary classification and 87.02% in the classification of x-ray images as Pneumonia, Covid-19 and Normal. A stacked model consisting of pre-trained VGG19 model and a new 30 layered COVID detection model is proposed in [30] for feature extraction, while the Logistic regression algorithm is used for classification of X-ray images and detection of COVID-19. The authors in [31] made a comparison between various pre-trained models and concluded that Resnet50 achieves the highest accuracy of 98% in the detection of COVID-19 between InceptionV3 and Inception-ResNetV2. In [32], the authors proposed a deep learning model for early screening of Covid-19 utilizing pre-processing image methods based on HU values and 3D CNN models for feature extraction from X-ray images. In [33], the authors utilized a pre-trained Chexnet model, which was proposed for detecting abnormalities in chest X-rays to classify between normal, pneumonia, Covid-19. A pre-trained SE-ResNext101 encoder along with SSD RetinaNet is utilized in [34]. This model was further tweaked on a database containing chest X-rays of 26,684 unique patients. Each image is labeled with one of three different classes from the associated radiological [35] reports No Lung Opacity/Not Normal, Normal, Lung Opacity, since lung opacity is a significant indication of pneumonia. The authors in [35] used CNN architectures, including InceptionV3, InceptionResnetV2, and Xception for classification of Chest X-ray images while statistical algorithms, like, Markov chain Monte Carlo (MCMC) and genetic algorithms are used to tune the hyperparameters of the models. In [36] the authors used pre-trained CNN models such as AlexNet, Vgg16, Vgg19 for feature extraction. The features obtained from the models mentioned above were then reduced with the help of minimum redundancy maximum relevance algorithm. The resultant feature set was then used as input to classification algorithms such as Linear Regression, K Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Decision Tree for classification of images as pneumonia and non-Pneumonia.

As it is evident from the explanations given above, most of the models proposed till date do not offer classification performance that can be deployed in real-world scenarios. Furthermore, most of the models were trained on an unbalanced dataset and thus may lack in robustness. We believe that by using efficient image processing techniques and efficient feature extraction, a scalable deep learning model can be built. This motivated us to study further and develop a deep learning model that can be deployed to help healthcare care professionals detect COVID-19 in these tough times of global pandemic. In the next section, we discuss the preliminary technologies and algorithms required to implement our proposed algorithm.

### 3. Preliminary

This section explains basic concepts such as deep learning, convolution neural networks, integrated stacking technique, transfer learning, and some pre-trained Convolution neural network along with their respective advantages and disadvantages. Basic knowledge of these concepts is required to understand and implement our proposed InstaCovNet-19 model.

#### 3.1. Deep learning and neural networks

The structure of the cerebral cortex inspires the neural networks. At the fundamental level, the perceptron is the scientific portrayal of a natural neuron. A single neuron behaves like a logistic regression model, as every neuron has two functions, one where it combines the input feature to a single number and an activation function, which transforms the output of the neuron into a usable number. The fundamental information goes through this network of hidden layers consisting of one or many layers of neurons. The output layer predicts certain values, or in the case of classification, it predicts the probability of a certain event occurring. Hidden layers adjust the information weightings until the neural network error is insignificant. To discover what these ideal weightings should be, we ordinarily use the backpropagation algorithm, which adjusts the weights using an optimizer (example Adam, RMSprop, gradient descent) and a loss function (e.g., Categorical Cross-Entropy for Multiclass classification and Mean Squared Error primarily used for regression models). Furthermore, the loss function defines the loss of the prediction; this loss is to be minimized.

#### 3.2. Convolutional Neural Network

Robust deep networks that are used in tasks such as object detection, image segmentation, image recognition, and other computer vision-related tasks are all Convolutional Neural Networks or CNN's. CNN's work on template matching techniques to complete a given task. A convolutional network tries to extract essential features from the input image through a progression of convoluting layers with channels (Kernels), pooling layers, fully connected layers (FC), and afterward, SoftMax function is then applied to classify the image with probabilistic characteristics in the scope of 0 and 1 [37]. Convolution Neural Network as the name suggests, contains Convolution Layers. Convolution Layers are made up of various filters, and each filter extracts different kinds of features and one activation map. Multiple activation maps are combined by stacking to form output volume. CNN layers take input of specific volume and output a volume of different shapes. A convolution layer takes in a set of parameters, which includes — Number of filters, stride, and activation. A convolution layer is often combined with a pooling layer, either max pooling or average pooling, to make it more efficient. Convolution operation was introduced in neural networks to prevent overfitting in neural networks as well as making the neural net concentrate on essential features [38].

#### 3.3. Transfer learning

In Transfer Learning, the weights of a particular model pre-trained on some dataset are used to enhance the results of classification on the dataset at hand. Transfer learning can be done in 2 ways:

- a. **Feature Extraction:** In this technique, a model that is pre-trained on some standard dataset such ImageNet is taken. Then the classification part of the model is removed. The remaining network is then treated as a Feature extractor on which any classification algorithm can run [39].
- b. **Fine Tuning:** In this strategy, we not only supplant and train the classifier, i.e., the head of the system on the dataset, but additionally adjust the pre-trained model weights by progressing on with the training process on all layers present [40].

Transfer Learning is used when there is a deficiency of data available, and it helps in preventing overfitting and randomization effects on the weights and better training overall.

#### 3.4. Integrated stacking

Integrated stacking is a variation of stacking ensemble technique used on neural networks. In this, we generally use neural networks as sub-models for first classifying on the given dataset and then use these predictions as features for another neural network known as estimate-learner. The estimate-learner figures out how to join the forecasts received from each input sub-model. This technique permits the stacked model to be treated as one significant model. The advantages of this method are that the projections of the sub-models are directly fed to the estimate-learner. Further, it is also possible to fine-tune the weights of the sub-models in aggregation with the estimate-learner model [41].

### 4. InstaCovNet-19 model implementation

The primary motivation behind the development of our proposed model is to automatically differentiate between a person suffering from COVID-19, A person suffering from pneumonia and a healthy person while decreasing the time required for detection as well as increasing the efficiency concerning the current methods. In this section, we explained our proposed algorithm and methodology for our proposed InstaCovNet-19 model.

InstaCovNet-19 is a deep convolutional architecture (DCNN) used for the detection of patients with COVID-19 using chest X-ray images. As there is a shortage of data consisting of COVID-19 X-rays, training models from scratch using randomly initialized weights are not very efficient and may lead to lousy variance versus bias trade-off. Therefore, to avoid these problems, Transfer Learning and multiple pre-trained DCNNs were used in this study. For the fine-tuning process, we used Inception v3 [7], MobileNetV2 [42], ResNet101 [6], NASNet [10] and Xception [8]. The above-mentioned pre-trained models were chosen after rigorous experimentation, the results of which concluded that each of the above-mentioned pre-trained models contributes towards the Improvement in classification performance because of the unique feature extraction techniques employed by each of these models, explained in detail in later subsections. These models were first imported with their pre-trained weights matrix (on ImageNet). Then these models were fine-tuned for our dataset. The fine-tuned models were then combined using the Integrated Stacking [41] technique, making the stacked model a larger and more robust model.

Fig. 1 gives a pictorial overview of our proposed model, InstaCovNet-19. Though the sub models are not drawn to scale

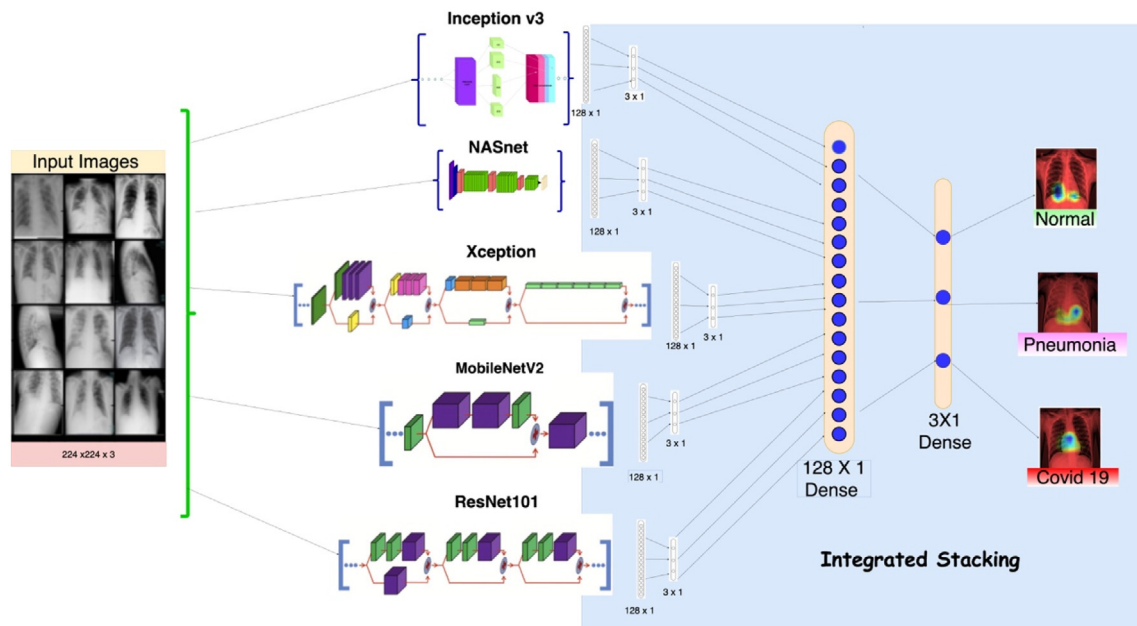


Fig. 1. InstaCovNet-19 Integrated stacked model.

owing to the large size of the model, It is suitable for giving the readers a basic understanding of our proposed model. The algorithm used to operate or train this integrated stacking model is given below as Algorithm 1. The proposed architecture is employed for both binary and 3-class classification. In the process of classification, images are first passed through the five heads of the input layer, i.e., the five copies of the image are given as input. The images then pass through the different models and get processed accordingly. The last convolution layers of each pre-trained model are fine-tuned again, and hence, we obtain forecasts from each model, which are then combined by a stacking layer. Now the combined output layer is passed through a dense layer of 128 nodes from where the stacked model learns how to use the predictions and what changes are to be made in the sub-models last convolution layer. Then results obtained from the last dense layer are passed through a dense layer of 3 nodes (1 node in binary classification), SoftMax activation (sigmoid activation for binary classification) is used to make predictions in form probabilities. This section discusses the process mentioned above in detail

**Input:** We take images of size 224x224x3 as input and due to limitations of data, we augment our dataset on the fly by randomly generating images by modifying original images by changing their zoom and shear parameters. Five copies of all images are made. These copies are then fed into different pre-trained models explained below.

#### 4.1. Deep learning model: Inception v3

Inception v3 [7] improves upon the previous inception architectures by being more computationally inexpensive. The basic building blocks of an inception model are Inception modules. An Inception Module allows for efficient computation and deeper networks through a dimensionality reduction with stacked  $1 \times 1$  convolution. The modules were intended to tackle the issue of computational cost, overfitting, among different issues. The basic concept behind the inception module is to make various filters of different sizes run in parallel rather than in series [43]. The networks in Inception modules have an extra  $1 \times 1$  convolution layer before the  $3 \times 3$  convolution layer and  $5 \times 5$  convolution layer, which makes the process computationally inexpensive and robust [43].

In our study, a pre-trained inceptionv3 model (trained on the Imagenet dataset) is imported. Dense layers of  $128 \times 1$  then replace the classification part of the model, i.e., the head of the model,  $3 \times 1$  and  $128 \times 1$  for binary and ternary classification, respectively. The model is then fine-tuned on COVID X-ray Images for better feature extraction. For training Inception v3 is provided with an input image of  $224 \times 224 \times 3$ , the input then goes through various inception modules, which help prevent overfitting while reducing the computational expense. After passing through the inception modules, the input is passed to a dense layer of dimensions  $128 \times 1$  and  $3 \times 1$  or  $2 \times 1$  for classification. Then after various iterations of forward propagation and backpropagation using Adam optimizer, InceptionV3 is ready to classify images and be integrated into InstaCovNet-19.

#### 4.2. Deep learning model: NASNet

NASNet [10] is an architecture that was created using a neural architectural search algorithm. The search method called Neural Architecture Search (NAS) makes uses of a control neural net to propose the best CNN architecture for a given dataset. The version of NASNet that was used in InstaCovNet-19 was designed for a dataset called ImageNet. Two sorts of convolutional cells are utilized in this design, i.e., the Reduction cell and the Standard cell. Wherein the Reduction cell reduces the area of the feature map by a factor of 2. NASNet is specially optimized for the ImageNet dataset, which contains images from all walks of life excels in feature extraction.

In our study, a pre-trained NASNet model (trained on the ImageNet dataset) is imported. The lack of a Large scale dataset necessitates the use of a pre-trained model. Dense layers of  $128 \times 1$  then replace the classification part of the model, i.e., the head of the model,  $3 \times 1$  and  $128 \times 1$ ,  $2 \times 1$  for binary and ternary classification, respectively.

The obtained pre-trained model is then fine-tuned on COVID-19 X-ray images. In the process of fine-tuning, NASNet is given an input image of dimensions  $224 \times 224 \times 3$ . The input then goes through various Normal and reduction layers that extract the best of the features. Finally, the features obtained are fed into 2 Dense layers of dimensions  $128 \times 1$  and  $3 \times 1$  for classification. The above-described process is carried out repeatedly during various

iteration backpropagation. The use of Adam further optimizes the process of fine-tuning.

#### 4.3. Deep learning model: Xception

Xception [8] stands for “extreme inception”. Xception was presented in 2016. The Xception model is 36 layers deep, excluding the fully connected layers in the end. Xception contains depth wise separable layers like MobileNet, and it also contains “shortcuts,” where the output of specific layers is summed with the output from previous layers. Unlike inceptionV3, Xception parcels input record into a few compacted lumps, it maps the spatial connections for each yield channel autonomously, then  $1 \times 1$  depth wise convolution is performed to catch cross channel relationships. Xception overtakes inception v3 on the classification of the ImageNet dataset.

In our study, a pre-trained Xception model (trained on the ImageNet dataset) is imported. A Pre-trained model is used owing to the lack of large scale datasets for COVID-19 detection. The classification part of the model, i.e., the head of the model, is then replaced by dense layers of  $128 \times 1$  and  $3 \times 1$  for binary classification and  $128 \times 1$  and  $2 \times 1$  for ternary classification, respectively. The model is then fine-tuned on COVID X-ray images during the various epochs. In the Fine-tuning process, Xception is given an input image of  $224 \times 224 \times 3$ , which then goes through various depth wise separable layers and shortcuts. The features thus obtained are fed into two dense layers of  $128 \times 1$  and  $3 \times 1$  or  $2 \times 1$  for classification.

#### 4.4. Deep learning model: MobileNet

MobileNet [42] is an architecture that is indented to run on mobiles and embedded systems or devices which lack computational power. This architecture was proposed by Google. Depth wise separable convolutions are used in MobileNet architecture to drastically reduces the number of trainable parameters in comparison to regular CNNs having comparable depth. The depth wise separable convolution deals with both spatial dimensions along with depth dimension (no of channels). Depth wise separable convolution splits the kernel into two small kernels one for depth wise convolution and other for pointwise convolution. This splitting of kernels reduces computational cost significantly. MobileNet gives results that are comparable to AlexNet while reducing the trainable parameters considerably.

In our study, a pre-trained MobileNet model (trained on the ImageNet dataset) is imported. A Pre-trained model is used owing to the lack of large scale datasets for COVID-19 detection. Dense layers of  $128 \times 1$  then replace the classification part of the model, i.e., the head of the model,  $3 \times 1$  and  $128 \times 1$ ,  $2 \times 1$  for binary and ternary classification, respectively. The model is then fine-tuned on COVID-19 X-ray images for better performance. In the process of fine-tuning, MobileNetV2 is given an input image of dimensions  $224 \times 224 \times 3$ . The input then undergoes depth wise and pointwise convolution various times. Lastly, the features obtained from the above process are fed into two dense layers of dimension  $128 \times 1$  and  $3 \times 1$  or  $2 \times 1$  for classification. The above process is repeated in various iterations of forward propagation and backward propagation using Adam optimizer. The various iterations of forward and backward propagation make the model optimized for detection COVID-19.

#### 4.5. Deep learning model: Resnet 101

ResNet101 [6] is a short name for Residual Network. Deep convolution networks have proved to be excellent in image classification tasks, so in order to improve classification accuracy, CNNs are usually made deep. Nevertheless, as we go deeper, the training of the neural network becomes difficult as the gradient used in backpropagation starts to vanish. Residual learning attempts to take care of this issue. Traditionally, a deep CNN consists of several stacked layers. In residual learning, instead of trying to extract some features, we extract the residual. Residual can be comprehended as the deduction of highlights gained from the contribution of that layer. ResNet makes use of skip-connection to avoid vanishing gradients. It has verified that training through this type of architectures is more effective than training regular deep CNN's.

In our study, a pre-trained Resnet101 (trained on the ImageNet dataset) is imported. The classification part of the model, i.e., the head of the model, is then replaced by dense layers of  $128 \times 1$ ,  $3 \times 1$  and  $128 \times 1$ ,  $2 \times 1$  for binary and ternary classification, respectively. The model is then fine-tuned on COVID-19 X-ray images. In the process of fine-tuning ResNet given in an input image of  $224 \times 224 \times 3$  [44]. The input then passes through a very deep convolutional network consisting of 101 layers. The problem of vanishing gradient is resolved through the use of residual learning. Finally, the obtained features are fed into two dense layers, which are  $128 \times 1$  and  $3 \times 1$  or  $128 \times 1$  and  $2 \times 1$  in dimensions, respectively. The process explained above is repeated many times during forward and backward propagation optimized by Adam.

After obtaining the prediction from the different pre-trained models, The predictions are combined and then classified by a network of the dense layer, as explained in the next section.

#### 4.6. Integrated stacking InstaCovNet-19 classification model

**Mathematical concepts and equations used in the explanation below:**

$$\text{Relu Function} = y(\xi) = \max(0, \xi) \quad (1)$$

$$\text{Softmax Function } f(a^i) = \frac{e^{a_i}}{\sum e^{a_i}} \quad (2)$$

$$\text{Categorical Cross entropy} = \text{Loss}(y_i) = - \sum y^j * \log y^j \quad (3)$$

$$\text{Adam Optimizer} : \Theta = \Theta - \frac{(\alpha * V_{d\Theta}^{\text{corrected}})}{\sqrt{S_{d\Theta}^{\text{corrected}} + \psi}} \quad (4)$$

$$\Phi = \Phi - \frac{(\alpha * V_{d\Phi}^{\text{corrected}})}{\sqrt{S_{d\Phi}^{\text{corrected}} + \psi}} \quad (5)$$

Note:  $\Theta$  and  $\theta$  are weights and bias,  $\alpha$  is the learning rate,  $V$  is the momentum, and  $S$  is the squared gradient, and  $\psi$  is the stability factor prevents the denominator becoming 0.

After fine-tuning all the above models, the larger stacked model is defined where these pre-trained models are utilized as different inputs to the larger stacked model. Usually, in integrated stacking, all the layers of the sub-models are to be marked as untrainable. However, after experimentation, we found that training the last convolution layers of all sub-models is beneficial for classification. Therefore, we set all the layers of the sub-models as non-trainable except the last convolution layer of all sub-models. For the training process first, we pass an input of dimensions (224,224,3) to the model. The input layer for every

one of these sub-models is utilized as a different contribution to the new stacked model. This infers five duplicates of the information are given to the model. The primary highlights are extricated from the contribution as it goes through the sub-models. In this way, flattened image features are acquired in the form of  $(\xi_1, \xi_2, \xi_3, \xi_4, \dots, \xi_n)^T$  from the global average pooling layer. The outputs of every one of the models are then combined. For this situation, concatenation merge was utilized, and a solitary 15-component vector is obtained. This vector was made from the three class-probabilities anticipated by every one of the five models.

The feature vector obtained from the above-explained process is then passed through a dense layer with 128 nodes having Relu function as an activation function Eq. (1), where these features are combined with the dense layer weights to obtain a matrix of shape (Input\_size,128), the feature matrix (matrix-1) is as given in Box 1.

Inputs from matrix one are then passed through a Dense layer with three nodes and SoftMax (sigmoid in binary classification) activation function from Eq. (2), and according to we get forecasts in the form of probabilities.

After this loss on the predictions is calculated through Categorical Cross-Entropy function (Binary Cross entropy in binary classification) using Eq. (3), and it is optimized on Adam optimizer, here Adam is used as an optimizer since it combines the advantages of both SGD optimizer and RMS prop optimizer. Adam uses momentum term like SGD and uses squared gradients like RMS prop. It also has one distinctive feature of learning rate decay, i.e., when model training reaches it ends, through Adam, the learning rate of the optimizer is decayed or decreased automatically in order to prevent overshooting. The equations for Adam is given by Eqs. (4) and (5). We proposed an integrated stacking InstaCovNet-19 classification algorithm for the detection of COVID 19 patients.

Output: Prediction of the probability of a class to which the image belongs. In the form of (matrix-2);

$$y = [P(y_{class\ 1}) \quad P(y_{class\ 2}) \quad P(y_{class\ 3})] \quad (\text{matrix-2})$$

(where  $P(\alpha) \rightarrow$  Probability of class)

## 5. Experiments and results

The 5th section of this paper discusses the datasets used and various image pre-processing techniques harnessed to optimize feature extraction, the hyperparameters, and the system architecture used in InstaCovNet-19 results achieved by InstaCovNet-19 and Comparative analysis of InstaCovNet-19 with other state of the art models. Hence this section is divided into five sub-sections. The first sub-section discusses the methods used to balance the dataset; various techniques were used to improve feature extraction and hence achieve better classification performance. The second section explains the hyperparameter tuning methods, and the hyperparameters used to execute the proposed InstaCovNet-19 model, this section also explains the system architecture used for executing the proposed model. The third section illustrates the results obtained by the proposed model using various evaluation metrics such as accuracy, F1-Score, precision and recall. The fourth section compares our proposed model with other, state of the art models (discussed in Section 2) on various standard evaluation metrics. While the 5th section illustrates the qualitative results achieved by InstaCovNet-19 using the grad-cam technique.

For Implementing the proposed algorithm and for obtaining results, we used Python 3.7 Programming language, NumPy,

### Raw vs Balanced Dataset

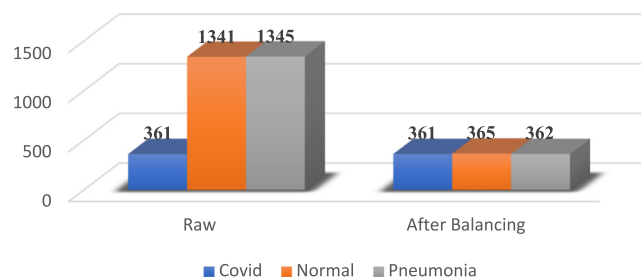


Fig. 2. Balanced dataset.

Scikit-Learn, and TensorFlow 2.0 libraries; for visualizations, matplotlib and seaborn libraries were used. System specifications: 12.6 GB Ram, Intel(R) Xeon(R) CPU @ 2.30 GHz, and, 12GB GDDR5 VRAM, GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores. In the upcoming sections, we described the dataset used and the results and observations obtained from our experiment. We also showed a comparative analysis with other traditional research paper methodologies.

### 5.1. Chest X-ray dataset

The proposed InstaCovNet-19 model was trained and tested on a combined dataset consisting of images obtained from two publicly available repositories. This combination makes our model more robust and less prone to variance. Our combined dataset had three classes of images, namely: COVID-19, Pneumonia and Normal. The sources from which the images were acquired are updated consistently by the respective authors, and hence the number of images available in these repositories is prone to change in the future. The repositories from which our X-ray images were obtained are as follows-

1. COVID-19 Radiography Database by [45] from Kaggle<sup>1</sup>: This dataset contains 219 chest X-ray images of COVID-19 class, 1345 Pneumonia chest X-ray images and 1341, Normal chest X-ray images [45].
2. Chest X-ray dataset<sup>2</sup> by [46]. This dataset contained 142 COVID-19 chest X-rays when the dataset was compiled for the experiment [46].

As shown below in Fig. 2, the combined dataset had 361, 1341 and 1345 images of COVID, Normal and Pneumonia class, respectively. Since the combined dataset was initially imbalanced; therefore, we used random sampling to make it balanced. In the process of random-sampling 361 COVID-19 Images, 365 Normal class Images and 362 Pneumonia class Images were selected for the experiment, as shown in Fig. 2.

In order to obtain better features for our model to train on, two image pre-processing or reconstruction techniques were applied:

1. Fuzzy Color image enhancement Technique: This technique is applied for image enhancement and image noise reduction. We enhance the colors in an image by tuning brightness and contrast. This algorithm divides the image into fuzzy windows and every pixel has a specific weight to every window, these weights are proportional to the separation between the window and the pixel. The weights are normalized; the final image is

<sup>1</sup> <https://www.kaggle.com/tawsifurrahman/COVID19-radiography-database?rvi=1>.

<sup>2</sup> <https://github.com/ieee8023/COVID-chestxray-dataset>.

$$[(\theta_1^1 \cdot \xi_1 + \theta_2^1 \cdot \xi_2 + \dots + \theta_{15}^1 \cdot \xi_{15}) \quad (\theta_1^2 \cdot \xi_1 + \theta_2^2 \cdot \xi_2 + \dots + \theta_{15}^2 \cdot \xi_{15}) \quad \dots \quad (\theta_1^{128} \cdot \xi_1 + \theta_2^{128} \cdot \xi_2 + \dots + \theta_{15}^{128} \cdot \xi_{15})] \quad (\text{matrix-1})$$

**Box I.****Algorithm 1: InstaCovNet-19 Classification Model****Input:** Image Dataset shape = 224x224x3**Output:**  $y = [P(y_{class1}), P(y_{class2}), P(y_{class3})]$  where  $P(\alpha) \rightarrow$  Probability of a class**Steps:**

1. Read Images in batches of 16 batch of size 224x224x3
2. Pass image data through all 5 models trained above and get predictions from it
3. Stack the predictions made from the 5 models to obtain a feature matrix of shape (None,15)
4. Initialize:  $\theta_c =$  Xavier initialization, epoch = 0
5. While epochs  $\leq 5$  Do
6. Pass this feature matrix to a dense layer of 128 nodes
7. DenseLayer(nodes = 128, data = feature\_vector)  $\rightarrow$  here the features\_vector is converted to  $\rightarrow$  a vector of shape = (None, 128)  $\rightarrow$  feature\_vector\_dence =  $[(\theta_1^1 \cdot \xi_1 + \theta_2^1 \cdot \xi_2 + \dots + \theta_{15}^1 \cdot \xi_{15}) \quad (\theta_1^2 \cdot \xi_1 + \theta_2^2 \cdot \xi_2 + \dots + \theta_{15}^2 \cdot \xi_{15}) \quad \dots \quad (\theta_1^{128} \cdot \xi_1 + \theta_2^{128} \cdot \xi_2 + \dots + \theta_{15}^{128} \cdot \xi_{15})]$
8. Relu\_activation(feature\_vector\_dence)  $\rightarrow$  features\_to\_predict\_on = max(feature\_vector\_dence, 0)
9. Dense\_2(features\_to\_predict\_on)  $\rightarrow$  vector of shape = (None, 3)  
features\_to\_predict\_on =  $[(\theta_1^1 \cdot \xi_1 + \theta_2^1 \cdot \xi_2 + \dots + \theta_{128}^1 \cdot \xi_{128}) \quad (\theta_1^2 \cdot \xi_1 + \theta_2^2 \cdot \xi_2 + \dots + \theta_{128}^2 \cdot \xi_{128}) \quad (\theta_1^3 \cdot \xi_1 + \theta_2^3 \cdot \xi_2 + \dots + \theta_{128}^3 \cdot \xi_{128})]$
10. Softmax(features\_to\_predict\_on)  $\rightarrow$  predictions =  $f(\text{features\_to\_predict\_on}[i]) = f(a^i) = \frac{e^{a_i}}{\sum e^{a_i}}$
11. Find loss on the prediction according to categorical\_crossentropy loss  $\rightarrow$  Loss( $y_i$ ) =  $-\sum y^i * \log y^i$
12. Update weight of hidden layers as well as input weight  $W$ s according to Adam optimizer  $\rightarrow$

$$\theta = \theta - \frac{(\alpha * V_{d\theta}^{corrected})}{\sqrt{S_{d\theta}^{corrected} + \psi}}$$

**output**  $y = [P(y_{class1}) \quad P(y_{class2}) \quad P(y_{class3})]$  (where  $P(\alpha) \rightarrow$  Probability of class)

obtained by summing up the images of every fuzzy window in a weighted way. [47]. Fig. 3 demonstrates the improvements made by fuzzy color techniques.

2. Stacking: Here, we took both (fuzzy color image & original image), image stacking technique used to enhance the quality of an image. It is otherwise known as focus blending or z-stacking and central plane consolidating. This method means to destroy the irregularity from the first picture by joining two pictures in progression, and the image is split into two segments, overlay, and foundation. The primary picture (non-pre-prepared picture) is utilized as a foundation, and the following picture (pre-handled picture) is overlaid on it [21]. Fig. 4 shows the improvements in a dataset by stacking.

After obtaining a balanced dataset of pre-processed images, we split the dataset into training images and testing images. 80% of the total images were kept for the purpose of training, while 20% of the images were kept for the purpose of testing.

**5.2. Hyperparameters used**

The models explained in Section 4, i.e., ResNet101, Inception v3, MobileNetV2, NASNet and Xception were trained and optimized using Adam and a learning reduction algorithm was implemented which reduced the learning rates of the models every time validation loss increased. The loss functions used were categorical cross-entropy and binary cross-entropy for 3-class classification and binary classification, respectively. Due to constraints imposed due to computational resources, images were passed in a batch of 16. While fine-tuning the pre-trained models, the 5 last layers of all the models were kept trainable while all other layers were un-trainable. In ResNet101 a total 42,889,219 parameters were trained of 266,755 parameters were trainable, similarly Inception v3, MobileNetV2, NASNet and Xception had a total of 43,151,875; 2,422,339; 85,433,429; 21,123,881 parameters respectively of which 262,401; 884,097; 516,353; 3,429,121 were trainable respectively.



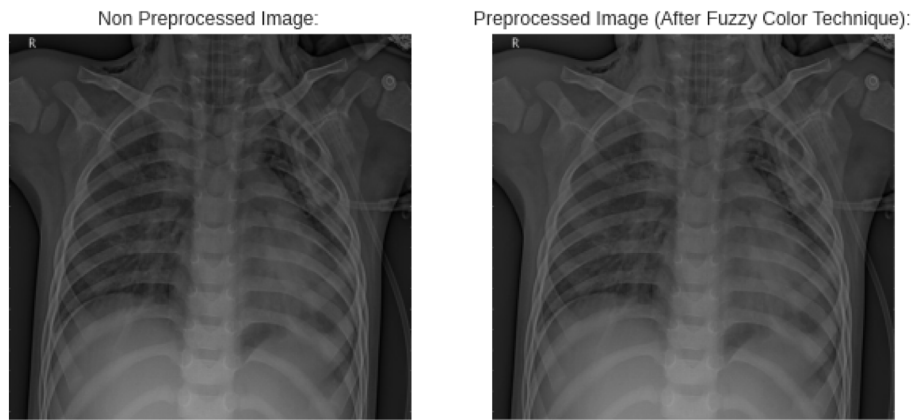


Fig. 3. Image after fuzzy color technique.



Fig. 4. Image after fuzzy color technique and stacking.

### 5.3. Result discussion

In this section, we explain the evaluation metrics that were used to quantify the model’s classification performance. We used Confusion matrix-based metrics for this purpose. These metrics include accuracy, precision (class-wise and macro average), recall (class-wise and macro average), F1-Score (class-wise and macro average). For evaluating these measures, we needed the count of the following quantities – True Positive, False Negative, True Negative, and False Positive.

1. Accuracy: Ratio of all predictions predicted correctly to the total number of predictions

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

2. Recall: Ratio of true positive to the total observation made by the proposed model

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

3. Precision: Ratio of true positive to total positive predictions

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

4. F1Score: It is the harmonic mean of precision and recall

$$\text{F1 Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{9}$$

5. Confusion matrix: It is the measurement of the performance of the model. It compares the actual and predicted

values in the form of True Positive, False Negative, True Negative and False Positive (matrix-3).

$$\begin{bmatrix} \text{True Positive (TP)} & \text{False Positive (FP)} \\ \text{False Negative (FN)} & \text{True Negative (TN)} \end{bmatrix} \tag{matrix-3}$$

- True Positive (TP): True positive are the forecasts which were at first positive and, additionally, anticipated by the AI model as positive.
- False Positive (FP): False positives are the forecasts which were initially negative and anticipated by the AI model as positive.
- True Negative (TN): True negatives are the forecasts which were initially negative and anticipated by the AI model as unfavorable.
- False Negative (FN): False-negative are the forecasts which were initially positive and anticipated by the model as negative

#### 5.3.1. Image pre-processing results

We first trained and tested our proposed integrated stacking model on un-pre-processed images, and from that, we obtained an accuracy of 97.705%, average precision of 0.9766, average recall of 0.9766 and an average F1-Score of 0.9766 on the testing set. Then after de-noising and enhancement of images using the fuzzy-color technique, our proposed model achieved an accuracy of 98.01%, average precision of 0.972, an average recall of 0.98, and an average F1-Score of 0.975 on the testing set. Table 1 explains the classification report of all the pre-processing techniques in detail. From Table 1 below, it can be perceived that even though there is a small increment in accuracy using images enhanced by the Fuzzy color image enhancement technique, precision and recall of Normal class decrease by small amounts. A

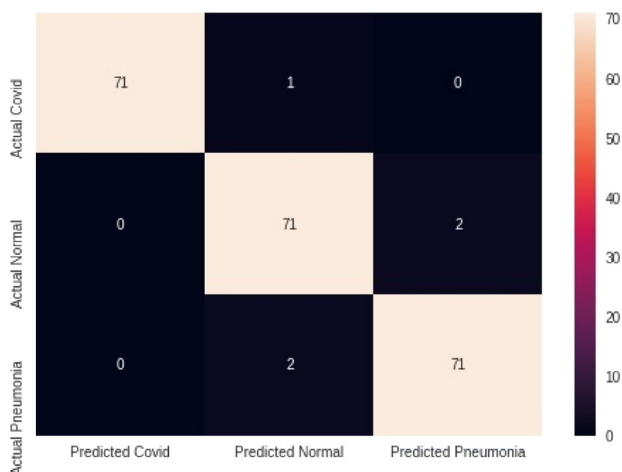


Fig. 5. Confusion Matrix un pre-processed images.

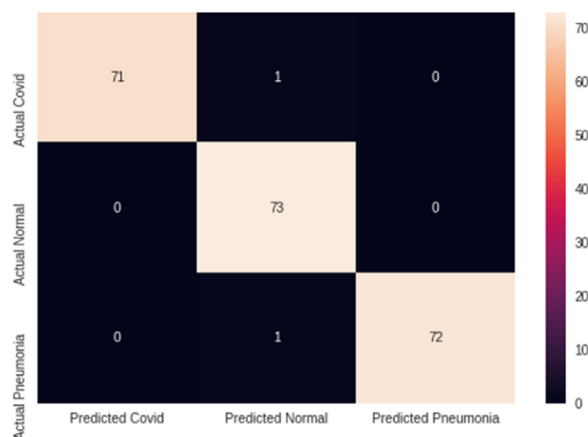


Fig. 7. Confusion Matrix for stacked images.

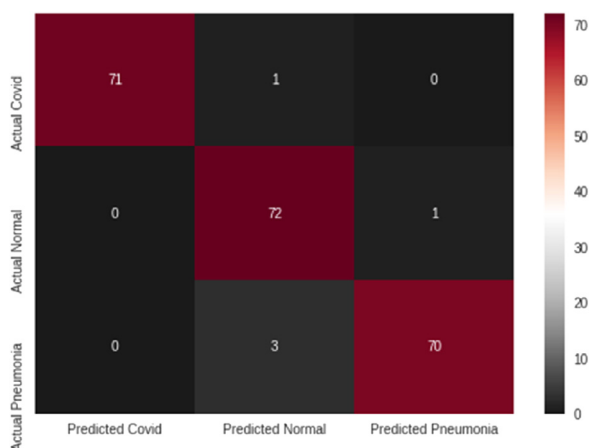


Fig. 6. Confusion Matrix Fuzzy pre-processed images.

Decrement in the precision of Normal class implied that our model was classifying people infected with diseases as Healthy. Such misclassification can lead to disastrous implications. The cause for such ordinary improvements obtained from using the Fuzzy color technique can be attributed to the fact that the Subtle logarithmic changes made by Fuzzy color techniques were going unnoticed by our model. Hence to decrease the rate of misclassification of the disease classes, i.e., COVID and Pneumonia and to make our model aware of the changes made by the Fuzzy color technique, we applied the stacking technique explained in Section 5.1 to enhance the input images further and aid our proposed model in feature extraction. Using Images enhanced by stacking techniques, an astounding accuracy of 99.08%, an F1-score of 0.99 and an improvement of 1% in the precision of Normal class was obtained on the testing set.

Figs. 5–7 describe the Confusion Matrices for each image pre-processing explained in the above section. This gives us an overview of how all the images are classified and where most of the misclassification is happening. From the above figure, we see that un-pre-processed data made five misclassifications while the same is with fuzzy color technique. However, the impact of misclassification would be less in the case of fuzzy-color technique as in un-pre-processed images, because in fuzzy-color technique, misclassification is only in 2 classes while in un-pre-processed images there is misclassification in all the classes. While on the other hand, misclassification in the stacked images is very less as only two images are misclassified.

### 5.3.2. Three-class classification

As shown in Table 2, after fine-tuning the five pre-trained models on a combined dataset consisting of stacked X-ray images, ResNet-101 achieves an accuracy of 98% on the test set while achieving an accuracy of 97% on the training set; It also achieves a precision, Recall and F1-score of 0.98, 0.983 and 0.980 respectively. InceptionV3 attains an accuracy of 97% on the test set while attaining accuracy of 97% on the training set, InceptionV3 attains a precision, recall and F1-score of .966, .966 and .97, respectively. MobileNetV2 though very compact, accomplishes a remarkable validation set accuracy and training set accuracy of 98.08% and 98.09% while accomplishing .976, .976 and .9766 as precision–recall and F1-score respectively. NASNet, A neural net that was optimized for feature extraction on the Imagenet dataset, is not able to translate its performance on our dataset by accomplishing an accuracy of 95.6% on the test set and 94% on the training set. Xception performs reasonably well, reaching an accuracy of 97% on both test and training set while achieving a precision, recall and F1-score of .973, .973 and .976, respectively. While all models used in our study performed reasonably well and achieved an average accuracy of 97% on the test set, an accuracy of 97% cannot be considered acceptable in machine learning models to be used in Healthcare, since these models will be the difference between life and death for the patients. Since all the model achieves their respective accuracy using different methodologies. ResNet101 uses residual learning, Inception v3 uses inception modules, MobileNetV2 uses separable convolution to extract features, Xception uses extreme inception and skip connections. We thought of harnessing the benefits from all methodologies mentioned above by using Integrated stacking to stack these models, as explained in Section 4. InstaCovNet-19 was found to achieve remarkable high results on all evaluation metrics. InstaCovNet-19 achieves an accuracy of 99.08% on the test set while achieving an accuracy of 99% on the training set.

As shown in Table 3, InstaCovNet-19 achieves an average precision of .99, an average recall of .99 and an average F1-score of .99. On studying InstaCovNet-19 under a fine lens, it is found that it achieves perfect precision of 1 on both COVID and pneumonia class while achieving a precision of .97 on the Normal class. A perfect recall of 1 is achieved in Normal class while recall of .99 is achieved in the other two classes. InstaCovNet-19. As it is evident by the confusion matrix, out of all the 218 images used in the test set InstaCovNet-19 misclassifies only two images, one from the COVID class and one from the pneumonia class. From the evaluation metrics achieved by InstaCovNet-19, thus it can be concluded that InstaCovNet-19 performs better than previously

**Table 1**  
Classification comparison pre-processing techniques.

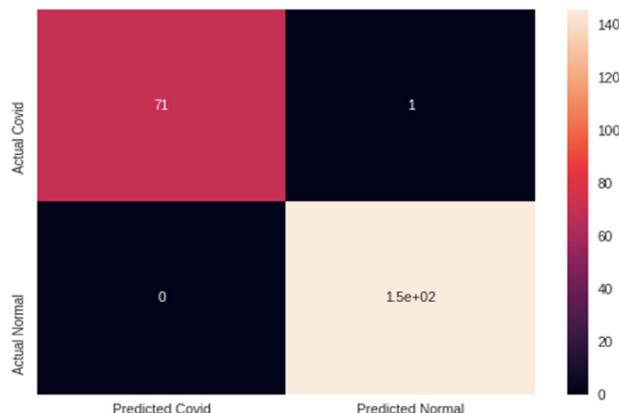
Image-pre-processing	Classes	Accuracy	Precision	Recall	F1-Score
Un-pre-processed	COVID	97.705%	1.00	0.99	0.99
	Normal		0.96	0.97	0.97
	Pneumonia		0.97	0.97	0.97
	Average		0.9766	0.9766	0.9766
Fuzzy color pre-processing	COVID	97.8%	0.99	0.99	0.99
	Normal		0.945	0.99	0.97
	Pneumonia		0.99	0.96	0.97
	Average		0.9733	0.98	0.976
Stacked images	COVID	<b>99.08%</b>	<b>1.00</b>	<b>0.99</b>	<b>0.99</b>
	Normal		<b>0.97</b>	<b>1.00</b>	<b>0.99</b>
	Pneumonia		<b>1.00</b>	<b>0.99</b>	<b>0.99</b>
	Average		<b>0.99</b>	<b>0.993</b>	<b>0.99</b>

**Table 2**  
Classification of fine-tuned models.

Model	Accuracy	Precision	Recall	F1-score
ResNet-101	0.98	0.98	0.983	0.98
Inception-v3	0.97	0.966	0.966	0.97
MobileNetV2	0.98	0.976	0.976	0.9766
NASNet	0.95	0.956	0.956	0.956
Xception	0.97	0.973	0.973	0.976

**Table 3**  
Three class classification report.

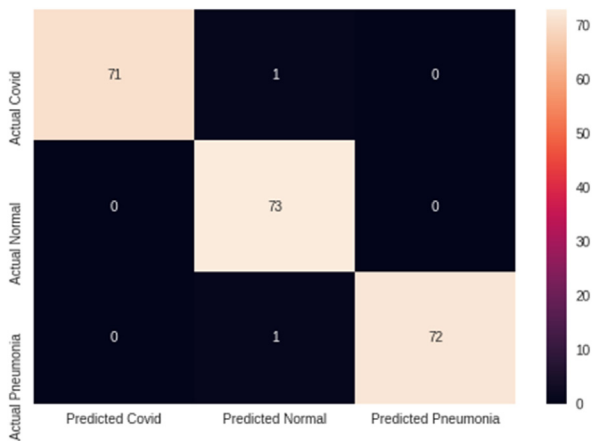
Class	Accuracy	Precision	Recall	F1-Score
COVID	99.08%	1.00	0.99	0.99
Normal		0.97	1.00	0.99
Pneumonia		1.00	0.99	0.99



**Fig. 9.** Classification report for binary classification.

**Table 4**  
Classification report binary classification.

Model	Accuracy	Precision	Recall	F1-Score
ResNet-101	0.99	1.00	0.99	0.99
Inception-v3	0.97	0.97	0.97	0.97
MobileNetV2	0.97	0.97	0.95	0.96
NASNet	0.99	1.00	0.99	0.99
Xception	0.99	1.00	0.99	0.99



**Fig. 8.** Confusion Matrix for three-class classification.

discussed pre-trained models in all respects. An average increase of .1 in all evaluation parameters such as precision, recall and F1-score can be seen in InstaCovNet-19.

Fig. 8 explains the Confusion-Matrix for 3-class classification. From the matrix, we can conclude that our proposed model made only 2 misclassifications on the testing dataset. These misclassifications occurred in the COVID-19 class and the Pneumonia class. We see that out of 218 tests, and our model is correct on 216 tests.

5.3.3. Binary classification

Similar to three class classification, for fine-tuning of models on x-images, the dataset was modified to contain two classes COVID and Non-COVID, for creating the Non-COVID classes, pneumonia and Normal class were combined. Resnet101 achieves an accuracy of 99.08% on both the test set and training set.

Inceptionv3 achieves an accuracy of 97.12% on the test set while achieving an accuracy of 92% on the training set. In contrast to three class classification NASNet performed very well, accomplishing an accuracy of 99% on the test set while achieving a training set accuracy of 98.89%. The compactness of mobilenetv2 became a disadvantage as it achieved a relatively low accuracy of 96% on binary classification. Xception also achieves an accuracy of 99% on the test as well as the training set. Though these models had achieved excellent accuracies, our experience with three-class classification prompted us that InstaCovNet-19 accomplishes remarkable results on evaluation metrics discussed in section InstaCovNet-19 achieves an accuracy of 99.54% on the test set on a training set accuracy of 98.89%. The average precision and recall of InstaCovNet-19 are one and .99, respectively. On investigating the results of InstaCovNet-19 under a fine lens, it is found that InstaCovNet-19 accomplishes 1.00, 0.99, 0.99 as precision, recall and F1 score in COVID class while achieving 1.00,1.00,1.00 as precision, recall and F1-score of Non-COVID class. Out of the 218 images, InstaCovNet-19 only misclassifies one COVID class image as shown by Fig. 9 the proposed model InstaCovNet-19 archives and average accuracy of 99.53%, while the recall and precision are 0.995 and 0.995, respectively, and the

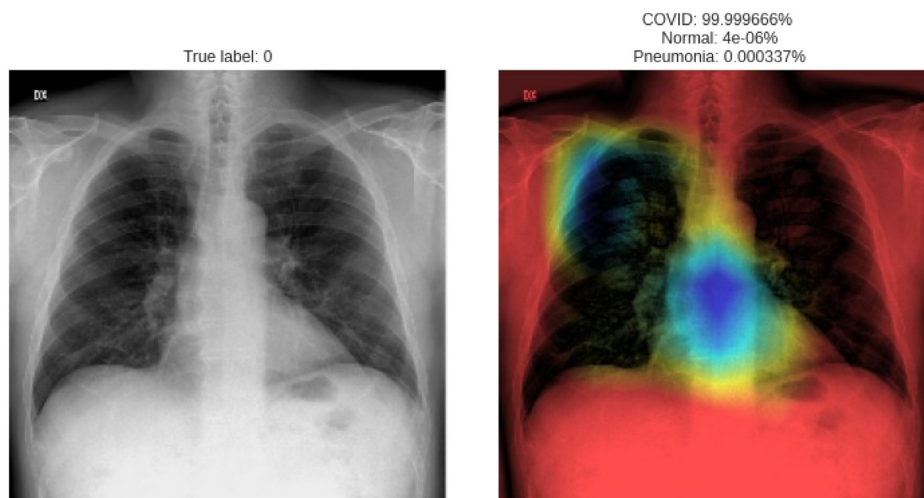


Fig. 10. Class activation map of a COVID-19 positive chest X-ray.

Table 5  
Comparison with other, state of the art models.

Reference	Model name	3 class accuracy	3 class F1 score	2 class accuracy
[22]	COVIDiagnosis-Net	0.9833	0.9833	N/A
[23]	CoroNet	0.896	0.896	.99
[24]	ResNet-50 + DCNN	N/A	N/A	.93
[48]	COVID-Net	0.933	0.90	N/A
[29]	DarkCovidNet	0.8702	0.8737	.98
[49]	MobileNet v2	0.9472	N/A	96.78
[33]	CovidAID	0.923	0.905	N/A
<b>Proposed Model</b>	<b>InstaCovNet-19</b>	<b>0.9908</b>	<b>0.99</b>	<b>.9952</b>

\*\*N/A: Authors did not perform the specified classification.

average F1 score is 0.995. The performance of COVID and Non-COVID classes is discussed in Table 4, and the confusion matrix for binary classification is shown in Fig. 9.

#### 5.4. Comparative analysis

Table 5, shown below, illustrates and compares the results of other state of the art models presents in the domain of COVID-19 detection with our proposed model InstaCovNet-19. A brief introduction about the methodology of these state of the art models is given in Section 1. As it is evident from the table given below, our model outperforms other state of the art models by a significant margin in the evaluation metrics such as accuracy and F1-Score. The next most accurate model, i.e., COVIDiagnosis-Net, accomplishes an accuracy of 98.33% while InstaCovNet-19 achieves an accuracy of 99.08%, displaying a difference of 1% approximately. Similar trends can be seen while comparing F1-Scores and binary classification, where InstaCovNet-19 surpasses all other models in terms of accuracy. Further, InstaCovNet-19 was trained on a well-balanced dataset, having an equal number of images from all 3 classes, thereby making it more robust and readily deployable in real-world scenarios. The best accuracy is marked in bold.

#### 5.5. Visualizations

In order to demonstrate the specific regions of the image where our proposed model concentrated, a class activation map using the grad cam technique was generated. Grad-Cam is an explainable machine learning technique that makes use of gradients from the last convolution layer of the model to generate a heat map that explains the area the proposed model is looking at to

make predictions. From Fig. 10, it is evident that our model is concentrating on lung opacities, which is a significant indication of COVID-19 and pneumonia. Thus, the predictions made by our proposed model are in line with current medical diagnosis techniques

## 6. Conclusion and future work

Current methods of detecting COVID-19 cost approximately 4500Rs and take at least 5–6 h. Though faster methods are available, the accuracy of such methods remains an issue. We aimed to propose a robust model that detects COVID-19 with high accuracy with a low rate of false negatives, which would not require expert supervision. We investigated the effects of various image processing techniques and used a multi-headed ensemble model for the classification of COVID-19. Our model was trained on a balanced dataset containing 290 images [29] of each class. Our model accomplishes an accuracy of 99.08% in three-class classification (Pneumonia, COVID-19, and Normal) while accomplishing an accuracy of 99.53% in Binary classification (COVID-19 and Non-COVID-19). In further studies for the detection of COVID-19, deep learning techniques can also be applied to other symptoms of COVID-19, which introduce abnormalities in organs of the Human body. Features such as age, gender, patient’s history, geolocation data, genetics can be considered to improve the efficiency of Computer-aided diagnosis in line with views of COVID-19 specialists/experts. Hence, as the world grips with the deadly Coronavirus, Computer-aided Diagnosis systems can prove to be highly effective in humanity’s fight against coronavirus.

## CRediT authorship contribution statement

**Anunay Gupta:** Conceived and designed the analysis, Collected the data, Contributed data or analysis tools, Performed analysis, Wrote the paper, Other necessary contributions related to this paper. **Anjum:** Conceived and designed the analysis, Collected the data, Contributed data or analysis tools, Performed analysis, Wrote the paper, Other necessary contributions related to this paper. **Shreyansh Gupta:** Conceived and designed the analysis, Collected the data, Contributed data or analysis tools, Performed analysis, Wrote the paper, Other necessary contributions related to this paper. **Rahul Katarya:** Conceived and designed the analysis, Collected the data, Contributed data or analysis tools, Performed analysis, Wrote the paper, Other necessary contributions related to this paper.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] WHO, WHO Director-general's opening remarks at the media briefing on COVID-19-27 July 2020, 2020, <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19-27-july-2020>, (Accessed 2 August 2020).
- [2] The COVID-19 Pandemic: Over 14 Million Cases Worldwide; US Once Again Shatters New Infection Record; and More - Docwire News, (n.d.), <https://www.docwirenews.com/docwire-pick/hem-onc-picks/the-covid-19-pandemic-over-14-million-cases-worldwide-despite-grim-numbers-fauci-remains-cautiously-optimistic-and-more/>, (Accessed 2 August 2020).
- [3] How is the COVID-19 Virus Detected using Real Time RT-PCR? | IAEA, (n.d.), <https://www.iaea.org/newscenter/news/how-is-the-covid-19-virus-detected-using-real-time-rt-pcr>, (Accessed 2 August 2020).
- [4] Most Coronavirus Tests Cost About \$100. Why Did One Cost \$2, 315? - The New York Times, Nytimes.Com. (n.d.), <https://www.nytimes.com/2020/06/16/upshot/coronavirus-test-cost-varies-widely.html>, (Accessed 2 August 2020).
- [5] Coronavirus testing in India: Check out testing cost, COVID-19 testing centre near you, other details, Bus. Today (2020) <https://www.businesstoday.in/current/economy-politics/coronavirus-testing-in-india-check-out-testing-cost-covid-19-testing-centre-near-you-other-details/story/407889.htm> l, (Accessed 2 August 2020).
- [6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., IEEE Computer Society, 2016, pp. 770–778, <http://dx.doi.org/10.1109/CVPR.2016.90>.
- [7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., IEEE Computer Society, 2016, pp. 2818–2826, <http://dx.doi.org/10.1109/CVPR.2016.308>.
- [8] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, 2017, pp. 1800–1807, <http://dx.doi.org/10.1109/CVPR.2017.195>.
- [9] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks, Comput. Biol. Med. 121 (2020) 103795, <http://dx.doi.org/10.1016/j.combiomed.2020.103795>.
- [10] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2018) 8697–8710, <http://dx.doi.org/10.1109/CVPR.2018.00907>.
- [11] A. Carruthers, J. Carruthers, Introduction, Dermatol. Surg. 39 (2013) 149, <http://dx.doi.org/10.1111/dsu.12130>.
- [12] C. Dhiman, D.K. Vishwakarma, View-invariant deep architecture for human action recognition using two-stream motion and shape temporal dynamics, IEEE Trans. Image Process. 29 (2020) 3835–3844, <http://dx.doi.org/10.1109/TIP.2020.2965299>.
- [13] T. Singh, D.K. Vishwakarma, A deeply coupled convnet for human activity recognition using dynamic and RGB images, Neural Comput. Appl. 0123456789 (2020) <http://dx.doi.org/10.1007/s00521-020-05018-y>.
- [14] A. Yadav, D.K. Vishwakarma, Sentiment analysis using deep learning architectures: a review, Artif. Intell. Rev 53 (2020) 4335–4385, <http://dx.doi.org/10.1007/s10462-019-09794-5>.
- [15] A. Yadav, D.K. Vishwakarma, A deep learning architecture of RA-DLNet for visual sentiment analysis, Multimed. Syst. 26 (2020) 431–451, <http://dx.doi.org/10.1007/s00530-020-00656-7>.
- [16] A. Jain, D.K. Vishwakarma, State-of-the-arts violence detection using ConvNets, in: Proc. 2020 IEEE Int. Conf. Commun. Signal Process, ICCSP 2020, 2020, pp. 813–817, <http://dx.doi.org/10.1109/ICCSP48568.2020.9182433>.
- [17] A. Yadav, D.K. Vishwakarma, A unified framework of deep networks for genre classification using movie trailer, Appl. Soft Comput. J. 96 (2020) 106624, <http://dx.doi.org/10.1016/j.asoc.2020.106624>.
- [18] J. Wang, H. Ding, F.A. Bidgoli, B. Zhou, C. Iribarren, S. Molloy, P. Baldi, Detecting cardiovascular disease from mammograms with deep learning, IEEE Trans. Med. Imag. 36 (2017) 1172–1181, <http://dx.doi.org/10.1109/TMI.2017.2655486>.
- [19] S. Deepak, P.M. Ameer, Brain tumor classification using deep CNN features via transfer learning, Comput. Biol. Med. 111 (2019) 103345, <http://dx.doi.org/10.1016/j.combiomed.2019.103345>.
- [20] J. Arnal, L. Súcar, Hybrid filter based on fuzzy techniques for mixed noise reduction in color images, Appl. Sci. 10 (2020) <http://dx.doi.org/10.3390/app10010243>.
- [21] M. Toğaçar, B. Ergen, Z. Cömert, COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches, Comput. Biol. Med. 121 (2020) <http://dx.doi.org/10.1016/j.combiomed.2020.103805>.
- [22] F. Ucar, D. Korkmaz, COVIDiagnosis-Net: Deep Bayes-SqueezeNet Based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses 140 (2020) 109761, <http://dx.doi.org/10.1016/j.mehy.2020.109761>.
- [23] A.I. Khan, J.L. Shah, M.M. Bhat, CoroNet: A Deep neural network for detection and diagnosis of COVID-19 from chest x-ray images, Comput. Methods Programs Biomed. 196 (2020) 105581, <http://dx.doi.org/10.1016/j.cmpb.2020.105581>.
- [24] Y. Pathak, P.K. Shukla, A. Tiwari, S. Stalin, S. Singh, Deep transfer learning based classification model for COVID-19 disease, Irbm. 1 (2020) 1–6, <http://dx.doi.org/10.1016/j.irbm.2020.05.003>.
- [25] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M.P. Lungren, A.Y. Ng, CheXNet: RADIologist-level pneumonia detection on chest X-rays with deep learning, ArXiv (2017) 3–9, <http://arxiv.org/abs/1711.05225>.
- [26] D. Verma, C. Bose, N. Tufchi, K. Pant, V. Tripathi, A. Thapliyal, An efficient framework for identification of tuberculosis and pneumonia in chest X-ray images using neural network, Procedia Comput. Sci. 171 (2020) 217–224, <http://dx.doi.org/10.1016/j.procs.2020.04.023>.
- [27] Y. Zhang, S. Niu, Z. Qiu, Y. Wei, P. Zhao, J. Yao, J. Huang, Q. Wu, M. Tan, COVID-DA: Deep Domain Adaptation from Typical Pneumonia to COVID-19, XX, 2020, pp. 1–8.
- [28] P.K. Sethy, S.K. Behera, P.K. Ratha, P. Biswas, Detection of coronavirus disease (COVID-19) based on deep features and support vector machine, Int. J. Math. Eng. Manag. Sci. 5 (2020) 643–651, <http://dx.doi.org/10.33889/IJMMS.2020.5.4.052>.
- [29] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U. Rajendra Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020) 103792, <http://dx.doi.org/10.1016/j.combiomed.2020.103792>.
- [30] M. Gour, S. Jain, Stacked Convolutional Neural Network for Diagnosis of COVID-19 Disease from X-Ray Images, 2020, <http://arxiv.org/abs/2006.13817>.
- [31] A. Narin, C. Kaya, Z. Pamuk, Department of Biomedical Engineering, Zonguldak Bulent Ecevit University, 67100, Zonguldak, Turkey, ArXiv Prepr. ArXiv2003.10849, 2020, <https://arxiv.org/abs/2003.10849>.
- [32] C. Butt, J. Gill, D. Chun, B.A. Babu, Deep learning system to screen coronavirus disease 2019 pneumonia, Appl. Intell. (2020) (2019) 1–29, <http://dx.doi.org/10.1007/s10489-020-01714-3>.
- [33] A. Mangal, S. Kalia, H. Rajgopal, K. Rangarajan, V. Nambodiri, S. Banerjee, C. Arora, CovidAID: COVID-19 detection using chest X-ray, ArXiv (2020) <http://arxiv.org/abs/2004.09803>.
- [34] T. Gabruseva, D. Poplavskiy, A.A. Kalinin, Deep Learning for Automatic Pneumonia Detection, 2019, 2020, <http://arxiv.org/abs/2005.13899>.
- [35] I. Mohammed, N. Singh, B. Area, L.B. National, Computer-Assisted Detection and Diagnosis of Pediatric Pneumonia in Chest X-ray Images, n.d., pp. 1–9.
- [36] M. Toğaçar, B. Ergen, Z. Cömert, A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models, Irbm. 1 (2019) 1–11, <http://dx.doi.org/10.1016/j.irbm.2019.10.006>.
- [37] Prabhu, Understanding of Convolutional Neural Network (CNN) – Deep Learning | Medium, Medium, 2018, <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>, (Accessed 22 July 2020).

- [38] K. O'Shea, R. Nash, An Introduction To Convolutional Neural Networks, 2015, <http://arxiv.org/abs/1511.08458>, (Accessed 2 August 2020).
- [39] E.C. Orenstein, O. Beijbom, Transfer learning & deep feature extraction for planktonic image data sets, in: Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 1082–1088, <http://dx.doi.org/10.1109/WACV.2017.125>.
- [40] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, R. Feris, SpotTune: Transfer Learning through Adaptive Fine-tuning, n.d..
- [41] C. Ju, A. Bibaut, M.J. Van Der Laan, The relative performance of ensemble methods with deep convolutional neural networks for image classification, *ArXiv* (2017).
- [42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2018) 4510–4520, <http://arxiv.org/abs/1801.04381>, (Accessed 27 July 2020).
- [43] Inception Module Definition | DeepAI, DeepAI, 2020, [https://deepai.org/machine-learning-glossary-and-terms/inception-module?source=post\\_page-d76756f2d6e2](https://deepai.org/machine-learning-glossary-and-terms/inception-module?source=post_page-d76756f2d6e2).
- [44] Residual Network (ResNet), OpenGenusIQ, n.d., <https://iq.opengenus.org/resnet/>.
- [45] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z. Bin Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, T.I. Islam, Can AI help in screening viral and COVID-19 pneumonia? IEEE Access (2020) <http://arxiv.org/abs/2003.13145>, (Accessed 22 July 2020).
- [46] G. Maguolo, L. Nanni, A Critic Evaluation of Methods for COVID-19 Automatic Detection from X-Ray Images, 2020, <http://arxiv.org/abs/2004.12823>, (Accessed 22 July 2020).
- [47] V. Patrascu, Color Image Enhancement Using the Support Fuzzification in the Framework of the Logarithmic Model Vasile PATRASCU, Vasile BUZULOIU Image Processing and Analysis Laboratory (LAPI), Faculty of Electronics and Telecommunication University POLITEHNICA of B, 2014, <http://dx.doi.org/10.13140/2.1.3014.6562>.
- [48] L. Wang, A. Wong, COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, *ArXiv* (2020) 1–12, <http://arxiv.org/abs/2003.09871>.
- [49] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.* 43 (2020) 635–640, <http://dx.doi.org/10.1007/s13246-020-00865-4>.