The aim of the work is to use deep neural network models for solving the problem of image recognition. These days, every human being is threatened by a harmful coronavirus disease, also called COVID-19 disease. The spread of coronavirus affects the economy of many countries in the world. To find COVID-19 patients early is very essential to avoid the spread and harm to society. Pathological tests and Chromatography (CT) scans are helpful for the diagnosis of COVID-19. However, these tests are having drawbacks such as a large number of false positives, and cost of these tests are so expensive. Hence, it requires finding an easy, accurate, and less expensive way for the detection of the harmful COVID-19 disease. Chest-x-ray can be useful for the detection of this disease. Therefore, in this work chest, x-ray images are used for the diagnosis of suspected COVID-19 patients using modern machine learning techniques. The analysis of the results is carried out and conclusions are made about the effectiveness of deep machine learning algorithms in image recognition problems.

Keywords  machine learning · deep neural networks · image recognition · TensorFlow, Keras.

1 Introduction

The new COVID-19 disease caused by the new strain of corona virus was first detected in Wuhan, China since December 2019 [1]. The disease subsequently spread throughout China and around the world [2][3]. A normal person can become infected if they have close contact with an infected person. Signs and symptoms reported include fever, fatigue, dry cough, shortness of breath, and respiratory failure. Most patients have mild symptoms and have a good prognosis. Cases of death are often the elderly and have underlying medical conditions such as cardiovascular disease and diabetes [1]. As of August 15, 2020, 2,589,208 cases and 63896 deaths were reported in India [4].
flights were also canceled due to the epidemic. To perform testing for the detection of COVID-19 are challenge for all, particularly the developed countries, because the testing device and its testing kits are quite limited and not accessible worldwide [5]. Several investigators and research institutions are currently researching on COVID-19 diagnosis [6, 7, 8, 9, 10]. Investigating early COVID-19 signs is not a reliable screening technique because there are certain instances in which patients have the signs but not diagnosed with COVID-19 as verified by the clinical examination or CT scan. While one of the important methods of diagnosing COVID-19 cases is the pathological examination. Nonetheless, this procedure has certain drawbacks because it should be performed in clinical labs that are only located in city centers and need time-consuming results. This can cause a problem as the positive patients cannot be isolated earlier, and they can infect more people through the crucial time of unrestricted movement [11, 12]. CT scan is also one of the popular methods for diagnosis of COVID-19 [13]. However, the problem in this radiological imaging is the overlapping with other diseases. When a COVID-19 patient is infected with another lung disease such as pneumonia, then it is difficult for a medical professional or a radiologist to diagnose these both similar looking CT scan images [14]. Also, one of the major disadvantages of COVID-19 diagnosis using a CT scan is its high radiation dose and its high cost, as it is not easy for common people to use this procedure [15]. Traditional radiography or Chest X-Ray (CXR) images can overcome the problem of costly CT scans and pathological tests because CXR is less costly and has minimal harmful consequences. Also, CXR is capable of identifying various lung diseases earlier [16]. The CXR imaging is a non-invasive procedure that takes 2-3 minutes to capture the image, and results can be fetched within thirty minutes. The modern radiographical machines are affordable for average income countries or underdeveloped countries [17]. Hence, in this research work, CXR is used to identify the deadly COVID-19 coronavirus.

Currently, due to the rapid development of digital technologies, the use of automated and robotic systems has spread to many areas, both in industry, science and in everyday life. As a consequence, there is an increasing need for efficient processing of information presented, in particular, in video and image formats [18]. At the current moment, images have closely merged into human life. Therefore, many automated systems use them as the main source of information [19]. Finding, localizing, classifying and analyzing objects in an image by a computer is a complex task in computer vision. Computer vision is a set of software and technical solutions in the field of artificial intelligence (AI), aimed at reading and receiving information from images, in real time and without human intervention [20]. In the process of analysing information received from the eyes, the human brain does a tremendous amount of work. A person can easily describe what is happening in a randomly taken photograph. Images can carry a tremendous amount of detail and differ in many parameters, such as resolution, color, quality, brightness, noise, etc. Objects in images can also have many features: scale, position, color, rotation, tilt, etc. However, in digital format, each image is just an array of numerical data. Teaching a computer to find and classify images in an image taking into account all factors is a very complex algorithmic task. To solve it, machine learning technologies are actively used. A person receives a large amount of information through sight. Images are capable of storing a huge amount of it. As a consequence, their use in computer systems increases the performance of these systems. However, such technologies require complex calculations. The challenge in computer vision is to develop efficient algorithms that extract and analyze data from images or videos. At the moment, similar technologies are used to solve such complex problems as:

- **OCR-Optical character recognition:** Converts text in an image to editable text.
- **Photogrammetry** is a technology for creating a three-dimensional model of an object based on photographs taken from different angles.
- **Motion capture** is a technology widely used in the film industry that allows to transform the movements of real people into computer animation.
- **Augmented reality (AR)** is a technology that allows virtual objects to be projected onto an image of a real environment in real time.
- **Medical diagnostics** - early detection of cancer cells, increasing the quality of MRI images, their analysis, etc.

Recently many scientist have used machine learning algorithms in disease diagnosis of different diseases [21, 22, 23, 24, 25, 26].

In this work, deep machine learning algorithms are used for solving image recognition problems, and also neural networks are designed and trained for diagnosing COVID-19 from chest x-rays.

Remaining of this paper organized as: section 2 gives the basic information about machine learning techniques. Section 2.5 contains a description of the architecture of a convolutional neural network and a description of modern models based on it. In section 3, modern models of convolutional neural networks were trained to solve the problem of diagnosing pneumonia and COVID-19 using X-ray images. Finally conclusions are drawn in the section 5.
2 Technical Background

2.1 The concept of an artificial neural network

Machine learning (ML) is a branch of research in the field of AI, which is based on methods of developing systems capable of learning. ML algorithms show themselves effectively in tasks in which it is required to determine common features from previously trained data and identify new data from them. Artificial neural networks are often used in the design of such learning systems.

Artificial neural network (ANN) is a computer model, which is based on the principles of a biological neural network - a set of interconnected nerve cells - neurons. Each neuron has a set of input connections - synapses, through which it receives information presented in the form of impulses from other neurons. According to the data obtained, the neuron forms its state and, with the help of the axon, communicates it to other neurons, ensuring the functioning of the system. In the process of forming the system, some neural connections are strengthened, while others are weakened, ensuring the learning of the network.

An artificial neuron is a simplified model of a biological neuron. The principle of its operation is shown in the figure. First, the neuron receives an n-dimensional vector of input values \(X = (x_1, \ldots, x_n)\) and a vector of weights \(W = (w_1, \ldots, w_n)\), denoting “strengthening” of interneuronal connections. The sum of the products of the input values and the weights \(s_j = \sum w_{ji} x_i + b_j\) is calculated. Then the activation function \(\varphi\) is applied to the result. Additionally, the amount of offset \(b_j\). Many neurons form layers. The layers, in turn, form a neural network. The input layer receives data, processes and transmits it to the neurons of the hidden layer. Each subsequent layer works in the same way until the output. A neural network with a large number of hidden layers is called deep. The field of machine learning that uses deep neural networks is called deep learning.

2.2 Activation function

The weighted sum of the inputs is a linear combination, which means that regardless of the number of layers, the values of the output layer depend only on the inputs of the first layer. The activation function of the neuron ensures the normalization of the calculated sum and the nonlinearity of the neural network. Many neural network models also require the activation function to be monotonic and continuously differentiable over the entire domain.
There are many activation functions. The most common ones are shown in the table.

Table 1: Popular activation functions

| Name              | Function                                      | Type |
|-------------------|-----------------------------------------------|------|
| sigmoid function  | $\sigma(x) = \frac{1}{1+e^{-x}}$              |      |
| Hyperbolic tangent| $f(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$        |      |
| ReLU              | $f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases}$ |      |

We should also mention the Softmax function. This feature is often used on the last layer of deep neural networks in classification problems. Let the last layer of the network contain $N$ neurons, each of which corresponds to a certain
class. Then the value of the output of the i-th neuron is calculated by the formula:

\[ y_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \]

Thus, the result of each neuron will take values from the range \([0, 1]\), and their sum is 1. As a result, the network will give the probabilities of the ratio of the input data to the given classes.

### 2.3 Training neural networks

The training of neural networks means the selection of the values of the weights of the connections for the effective solution of the task. Initially, weights are set randomly. Then, in the process of running the test data through the network, the weights are adjusted so that in the end the network gives the correct answers. The learning process is cyclical. During one iteration, a packet is fed to the network containing a number of elements from the input data. A single pass through the network of the entire set of test data is called an epoch.

In order to control the learning process, it is necessary to somehow evaluate the work of the network. For this, a loss function (cost function) is introduced, which calculates the difference between correct and obtained results and forms a certain numerical value characterizing the magnitude of the network operation error. Thus, the task of training the network is reduced to the task of finding the global minimum of a given function. The table contains the most frequently used loss functions, where \( y_i \) is the expected value of the i-th neuron, \( x_i \) is the obtained value of the i-th neuron, \( n \) is the number of output neurons. One of the popular methods in training deep neural networks is the backpropagation algorithm.

Let the network have \( L \) layers, \( a^l, w^l, b^l \) - vectors of values, weights and displacements of neurons on the \( l \)-th layer. There are also \( N \) training pairs \((x, y)\). In the learning process, the following iterations occur in cycles:

1. A vector \( x \) from the training set is fed to the network input, for each layer calculate the values:
   \[ z^l = w^l a^{l-1} + b^l \]
   \[ a^l = \sigma(z^l) \]
2. Calculate the value of the cost function:
   \[ C = \frac{1}{2} \sum_j (y_j - a^L_j)^2 \]
3. Calculate the error values of the output layer:
   \[ \delta^L_j = \delta C / \delta a^L_j \sigma'(z^L_j) \]
4. Calculate errors for each previous layer:
   \[ \delta^l_j = \sum_k w^{l+1}_{kj} \delta^{l+1} k \sigma'(z^l_j) \]
5. Calculate the gradient of the cost function:
   \[ \delta C / \delta w^l_{jk} = \delta^l_j \delta^{l+1} k \]
6. Update link weights:
   \[ w^l_{ij} = w^l_{ij} - \mu \delta C / \delta w^l_{ij}, \quad < \mu \leq 1 \]

| Name               | Function |
|--------------------|----------|
| Mean square error  | \( E = \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i)^2 \) |
| Average absolute error | \( E = \frac{1}{N} \sum_{i=1}^{N} |y_i - x_i| \) |
| Upper bound        | \( E = \frac{1}{N} \sum_{i=1}^{N} \max(1 - x_i, y_i, 0) \) |
| Categorical Cross Entropy | \( E = - \sum_{i=1}^{n} (x_i \cdot \log(y_i)) \) |
In addition to the method described above, other algorithms are often used for training, for example, RMSprop and Adam:

**Algorithm 1. RMSProp**

**Input** : Initial value $x_1$, parameter $\beta_2 \in [0, 1)$, step $\alpha$, constant $\epsilon$

**Function** $\text{RMSProp}(x_1, \beta_2, \alpha, \epsilon)$

$\begin{align*}
&v_0 = 0; \\
&\text{for } t = 1, 2, \ldots \text{ do} \\
&\quad g_t = \nabla f(x_t) \\
&\quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
&\quad x_{t+1} = x_t - \frac{\alpha}{\sqrt{v_t + \epsilon}} g_t \\
&\text{end}
\end{align*}$

**Algorithm 2. Adam**

**Input** : Initial value $x_1$, parameter $\beta_1, \beta_2 \in [0, 1)$, step $\alpha$, constant $\epsilon$

**Function** $\text{Adam}(x_1, \beta_1, \beta_2, \alpha, \epsilon)$

$\begin{align*}
&v_0 = 0; m_0 = 0; \\
&\text{for } t = 1, 2, \ldots \text{ do} \\
&\quad g_t = \nabla f(x_t) \\
&\quad m_t = \beta_1 m_{t-1} - (1 - \beta_1) g_t \\
&\quad v_t = \beta_2 v_{t-1} - (1 - \beta_2) g_t^2 \\
&\quad x_{t+1} = x_t - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}} g_t \\
&\text{end}
\end{align*}$

These methods belong to the **supervised learning** algorithms, the most common type of learning, in which the network learns from pre-labeled data where the correct answers are already known.

There are other approaches to training neural networks:

**Reinforcement learning** is a method that assumes the presence of some environment in which the network operates. Such an environment reacts to the actions of the model and gives it certain signals.

**Unsupervised learning** - learning in which the network does not have the correct answers in advance and independently searches for common and distinctive features of the input data.

**Genetic algorithms** - algorithms that mimic the evolutionary mechanisms of the development of a biological population, act as an alternative to the error backpropagation algorithm. The value of an arbitrary weighting factor in a neural network is called a gene. Genes form chromosomes, and chromosomes form a population. Further, within one epoch, with certain probabilities occurs:

- Crossing of chromosomes - the formation of a new chromosome from the genes of the other two
- **mutation** - random change of an arbitrary gene
- adaptation - the chromosomes showing the worst results are eliminated from the population.

**2.4 Problems of training deep neural networks**

In learning algorithms based on the backpropagation method. The error value depends on the derivative of the activation function, so when using the sigmoid activation function, the error value decreases very quickly when propagating from the last layer to the first, thus the weights in the early layers are poorly corrected. Similarly, the explosive gradient problem can occurred when the error value becomes very large. A simple way to solve this problem is to use the ReLU function, whose derivative takes the values either 0 or 1. To solve such problems, preprocessed input data can be resorted, it is often recommended to limit the input data to the range $[0; 1]$. For example, in images, values can range from 0 to 255 and for better learning stability they can be divided by 255. As another optimization method, values can be centered to calculate the average value for the input image and subtract it from each pixel, so the most average
value of the data in the image will be equal to 0. Together with this method, the standard deviation is often normalized, setting its value to 1.

Network retraining is a problem when the network learns to analyze objects well only from the training set and does not work well with new data. One of the methods for solving this problem is Dropout, the essence of which is as follows: At each training iteration, neurons with some probability are turned off. The remaining neurons are trained by the backpropagation method of oshiyuki, after which the neurons return to the network.

![Figure 4: Dropout](image)

### 2.5 Convolutional Neural Networks

#### 2.5.1 Architecture

Most of the modern neural networks aimed at image analysis are based on the architecture of the convolutional neural network. Early neural networks consisted of fully connected layers-layers in which each neuron is connected to each neuron of the next layer, which significantly increased the computational complexity of the system as the number of neurons increased. Typical convolutional neural networks primarily use convolutional layers. Convolutional layers are characterized by the use of weight matrices, called filters or kernels, that are smaller than the original data. Such a kernel with a certain step goes through the set of input data \( (I) \) and calculates the sums of the products of the corresponding values of the cells and weights, forming a feature map \( (I * K) \) One convolutional layer can contain several kernels and, accordingly, several feature maps.

![Figure 5: Convolutional Neural Network](image)

Since the features have already been detected, to simplify further calculations, it can be reduced the granularity of the input data. This provides a downsampling (pooling) layer, reducing the dimension of the input feature maps: from several neighboring neurons, the maximum or average value is taken, thereby forming a neuron of the feature map of a lower dimension. This reduces the number of parameters used in further network calculations.

A convolutional neural network can have multiple pairs of alternating convolutional and downsampling layers. Thus, using the example of images, on the initial layers, the network finds such simple features as borders and corners. Then more and more complicated structures are defined as we go deeper into the network: from the simple forms to the whole categories irrespective of where they are located. The network ends with completely connected standard layers which correspond to the resulting characteristics in the class.
2.5.2 VGG

The VGG architecture was proposed in 2014 by [27]. The main feature of the network is the use of consecutive convolutional layers with 3x3 filters instead of the previously used convolutional layers with large filters 5x5, 7x7, 11x11. This made it possible to reduce the number of network parameters while maintaining efficiency.

The table lists the various VGG configurations, the most famous of which are VGG-16 (D) and VGG-19 (E), named by the number of layers containing weights. Maxpool is a downsampling layer with a 2x2 max function. FC is a fully connected layer. All hidden layers use the ReLU activation function.

|   | A | A-LRN | B | C | D | E |
|---|---|-------|---|---|---|---|
|   | conv3-64 | conv3-64 | conv3-64 | conv3-64 | conv3-64 | conv3-64 |
| (224 x 224 RGB) | conv3-64 | LRN | conv3-64 | conv3-64 | conv3-64 | conv3-64 |
| Maxpool | conv3-128 | conv3-128 | conv3-128 | conv3-128 | conv3-128 | conv3-128 |
| | conv3-256 | conv3-256 | conv3-256 | conv3-256 | conv3-256 | conv3-256 |
| Maxpool | conv3-512 | conv3-512 | conv3-512 | conv3-512 | conv3-512 | conv3-512 |
| Maxpool | FC-4096 | FC-4096 | FC-1000 | Softmax |
2.5.3 Inception

This model [28], developed by Google, in 2014 took 1st place in the annual competition for image classification-ILSVRC. A key innovation of this network was the use of nested modules as layers, which are a set of filters of different dimensions, with the subsequent merging of their results.

![Inception Module Diagram](image)

Figure 7: Inception Module

Also, Inception completely abandoned the use of fully connected layers, instead of them, a global average pooling is used, which converts each feature map to one number, forming a vector of averaged values. This innovation made it possible to significantly reduce the number of parameters and, as a consequence, the computational complexity of the network. Later, improved versions of Inception were developed, in which the 5x5 layer was replaced with two successive 3x3 layers, and all layers with \( N \times N \) filters were replaced with a \( 1 \times N \) and \( N \times 1 \) filter stack, which also reduced the number of parameters.

2.5.4 ResNet

ResNet [29], also known as residual neural network, won the ILSVRC in 2015. Its feature was the presence of transmission connections that transmit information unchanged to deeper parts of the network, this information is summed up with the value calculated on the missing layers and transmitted further. The block shown in Fig. 8 demonstrates the building block of such a network.

2.5.5 DenseNet

DenseNet [30] is a dense convolutional network, similar to ResNet, but with the difference that all blocks of the network are connected by direct connections, so each block receives information from all previous ones.

2.5.6 Testing

All of these models were tested on ImageNet datasets, which include 1000 classes, over 1.3 million training and 50 thousand verification images. Since networks perform the task of classification and use a Softmax layer at the end, the result of the network is the vector \((x_1, x_2, \ldots, x_n)\), where \( n \) is the number of classes, and \( x_i \) is the probability of the ratio of the input image to the i-th class. The accuracy of the networks was measured in two variants Top-1 and Top-5. Top-1 means that the largest \( x_i \) corresponds to the correct class, and Top-5 means that the correct class belongs to the five highest values in the output vector of the network. The table 5 shows the results of the tests carried out using the accuracy metric-the ratio of the proportion of correct answers to their total number. The number of samples are less in
the COVID-19 class, hence confidence interval (CI) of all the performance metrics were calculated by using the following formula:

\[ r = z \sqrt{\frac{\text{acc}(1 - \text{acc})}{N}} \]  

(1)

Where \( z \) is the number of standard deviation also called the CI level, \( \text{acc} \) is the performance metrics used for the evaluation of performance models. And the value of the \( N \) is the number of samples for the class used. We have used 95% CI and standard deviation as 1.

As it can be seen in the table: VGG networks have the largest number of parameters, significantly surpassing other models, while demonstrating the lowest accuracy. DenseNet shows the best ratio of accuracy to parameter count. However, the third version of Inception has relatively slightly more parameters and precision. The best result was shown by ResNet-152 of the second version, but it has a significantly larger number of parameters, which affects the network learning time.
Table 5: Model testing results. Values in brackets are 95% confidence intervals.

| Network      | Parameters Qty | Top-1 (in %) | Top-5 (in %) |
|--------------|----------------|--------------|--------------|
| VGG-16       | 138 357 544    | 71.3 [69.2, 73.3] | 90.1 [90.0, 91.4] |
| VGG-19       | 143 667 240    | 71.3 [69.3, 73.3] | 90.0 [89.9, 91.3] |
| Inception V3 | 23 851 784     | 77.9 [76.0, 79.8] | 93.7 [93.6, 94.7] |
| ResNet-30 V2 | 25 613 800     | 76.0 [74.1, 77.9] | 93.0 [92.9, 94.1] |
| ResNet-101 V2| 44 675 560     | 77.2 [75.3, 79.1] | 93.8 [93.7, 94.8] |
| ResNet-152 V2| 60 380 648     | 78.0 [76.1, 79.8] | 94.2 [94.1, 95.2] |
| DenseNet-121 | 8 062 504      | 75.0 [73.0, 76.9] | 92.3 [92.2, 93.4] |
| DenseNet-169 | 14 307 880     | 76.2 [74.3, 78.1] | 93.2 [93.1, 94.3] |
| DenseNet-201 | 20 242 984     | 77.3 [75.4, 79.1] | 93.6 [93.5, 94.6] |

3 Methodology

3.1 Staging tasks

COVID-19 is a severe respiratory infection caused by the SARS-CoV-2 coronavirus. At the time of this writing, an outbreak of the virus that originated in the Chinese city of Wuhan has escalated into a global pandemic. The main method for diagnosing COVID-19 is reverse transcription polymerase chain reaction (PCR), but this is a laborious and human-intensive process. An alternative is an x-ray, such as a chest x-ray or computed tomography (CT). Research shows that most images of COVID-19 patients contain specific abnormalities, such as bilateral reticular nodular opacification or ground glass syndrome, to visually distinguish the disease from other types of pneumonia [31]. Despite the fact that X-rays are less sensitive than CT or PCR, it is a much more affordable and quick diagnostic method, which is an essential criterion during a pandemic as discussed in the introduction section. Solving the problem of automatic diagnosis of this disease will reduce the burden on doctors and increase the efficiency of their work.

![a. Normal](image1.png) ![b. Pneumonia](image2.png) ![c. COVID-19](image3.png)

Figure 9: Chest X-rays

3.2 Implementation facilities

cuDNN is a deep neural network library from Nvidia that allows to use GPU power for calculations. Python 3 is a flexible and powerful programming language that efficiently performs data analysis and processing tasks. TensorFlow is a feature-rich open source framework developed by Google that allows to design and train various neural network architectures. Keras is a high-level deep machine learning API included with TensorFlow.

3.3 System configuration

All models have been tested on a system with the following specifications:

- Operating system - Ubuntu 20.04
- CPU - Intel Core i5 9400F CPU 2.90 Ghz
• The amount of RAM - 8 GB
• Video card - Nvidia GeForce GTX 1660

3.4 Quality assessment

The following metrics were used to assess the quality of the algorithms:

• Precision:
  \[ P = \frac{TP}{TP + FP} \]

• Recall:
  \[ R = \frac{TP}{TP + FN} \]

• F1-
  \[ F1 = 2 \cdot \frac{P \cdot R}{P + R} \]

• Accuracy:
  \[ A = \frac{TP + TN}{TP + TN + FP + FN} \]

where TP is the number of true positive, FP - false positive, FN - False Negative Answers.

Precision denotes the proportion of correctly identified objects of a class relative to all objects assigned to this class.

Recall shows the proportion of elements of a class found by the network relative to all elements of this class.

F1 combines precision and recall by calculating their harmonic mean. Also, in the process of training the network, the values of the loss function were considered as a characteristic of the quality of training.

3.5 Training set

Collecting data for training neural networks in tasks of this type is a complex and time-consuming process that requires a lot of time and the participation of a large number of people. Therefore, ready-made, already marked datasets were used as a source of training and verification data: [32] A total of 14,197 photographs were collected, of which 8,066 were healthy patients, 5,558 with pneumonia and 573 with COVID-19. 100 images of each class were selected for training and validation.

The total number of shots is shown in the table 6

|         | Training | Validation |
|---------|----------|------------|
| normal  | 7,966    | 885        |
| Pneumonia | 5,458    | 594        |
| COVID-19 | 473      | 100        |
| Total   | 13,897   | 1,579      |

Images from the training set have different resolutions, but neural networks require a predetermined number of input neurons. Therefore, as a preprocessing of the data, all images are scaled to one resolution before being sent to the network: 512x512 px.

3.6 Image preprocessing

When using deep neural networks in image classification problems, additional research is required in order to select the optimal parameters for various parts of the algorithms. Pre-processing the input data can have a significant effect on training models. Within the framework of this part of the study, the following options for image processing were considered:

• Scaling - dividing all values in the image by 255.
• Centering the mean of the image at 0 and normalizing the standard deviation at 1.

The study tested the following neural network models with standard parameters:
• Inception V3, input layer dimension: 299x299
• ResNet-50, input layer dimension: 224x224
• DenseNet-201, input layer dimension: 224x224

All models were trained over 10 epochs, the size of one package was 16 images. Categorical cross entropy was used as the loss function, and Adam was used as the optimizer. Learning was stopped in advance if the value of the loss function on the validation set did not decrease for five epochs. The values of the error function (loss), precision and recall metrics based on training and validation results (prefix "val_") for processed images using scaling and centering methods are indicated in the tables 7 and 8 respectively.

Table 7: Results of training models with standard parameters and image prescaling. Values in brackets are 95% confidence intervals.

| Metrics   | InceptionV3       | ResNet-50 V2       | DenseNet-201      |
|-----------|-------------------|--------------------|-------------------|
| loss      | 0.24 [0.22, 0.25] | 0.32 [0.30, 0.35]  | 0.27 [0.25, 0.29] |
| precision | 0.92 [0.91, 0.92] | 0.89 [0.87, 0.90]  | 0.91 [0.89, 0.92] |
| recall    | 0.91 [0.91, 0.92] | 0.87 [0.86, 0.89]  | 0.90 [0.88, 0.91] |
| val_loss  | 0.27 [0.27, 0.29] | 0.24 [0.22, 0.26]  | 0.32 [0.30, 0.34] |
| val_precision | 0.79 [0.79, 0.81] | 0.74 [0.72, 0.76]  | 0.76 [0.74, 0.78] |
| val_recall | 0.77 [0.77, 0.79] | 0.71 [0.69, 0.73]  | 0.73 [0.71, 0.75] |

Table 8: Results of training models with standard parameters and preliminary centering of images. Values in brackets are 95% confidence intervals.

| Metrics   | InceptionV3       | ResNet-50 V2       | DenseNet-201      |
|-----------|-------------------|--------------------|-------------------|
| loss      | 0.34 [0.32, 0.36] | 0.34 [0.31, 0.36]  | 0.37 [0.34, 0.39] |
| precision | 0.89 [0.87, 0.90] | 0.88 [0.87, 0.90]  | 0.87 [0.86, 0.89] |
| recall    | 0.87 [0.85, 0.88] | 0.87 [0.86, 0.89]  | 0.86 [0.85, 0.88] |
| val_loss  | 0.53 [0.50, 0.55] | 0.93 [0.92, 0.94]  | 0.35 [0.33, 0.38] |
| val_precision | 0.72 [0.70, 0.74] | 0.65 [0.63, 0.67]  | 0.76 [0.75, 0.78] |
| val_recall | 0.67 [0.65, 0.69] | 0.62 [0.59, 0.64]  | 0.71 [0.71, 0.75] |

As it can be seen from the tables, pre-scaling of values gives metric values better than centering during training and validation. At the same time, the Inception V3 network showed the highest results in this test.

3.7 Optimization

The selected algorithm for optimizing the error function can significantly affect the performance of neural networks. This part of the study analyzes various optimizers: Stochastic Gradient Descent (SGD), RMSProp, and Adam. Testing was carried out on InceptionV3, ResNet-50, DenseNet-201 networks and pre-scaled input images of 500x500 pixels.

All models were trained for 10 epochs, the size of one package was 8 images. The tables 9 indicate the values of the Precision, Recall and F1 metrics obtained as a result of validation for each class.

Table 9: Results of training networks with Adam optimizer. Values in brackets are 95% confidence intervals.

| Models       | Metrics   | COVID-19 | Normal | Pneumonia |
|--------------|-----------|----------|--------|-----------|
| InceptionV3  | precision | 0.53 [0.51, 0.55] | 0.51 [0.49, 0.53] | 0.59 [0.57, 0.59] |
|              | recall    | 0.38 [0.37, 0.40] | 0.58 [0.57, 0.60] | 0.68 [0.68, 0.70] |
|              | f1-score  | 0.44 [0.42, 0.46] | 0.54 [0.52, 0.56] | 0.63 [0.62, 0.65] |
| ResNet-50 V2 | precision | 0.55 [0.55, 0.57] | 0.51 [0.49, 0.53] | 0.52 [0.52, 0.54] |
|              | recall    | 0.40 [0.40, 0.42] | 0.53 [0.51, 0.55] | 0.63 [0.61, 0.65] |
|              | f1-score  | 0.46 [0.44, 0.46] | 0.52 [0.50, 0.54] | 0.57 [0.55, 0.59] |
| DenseNet-201 | precision | 0.52 [0.50, 0.54] | 0.53 [0.51, 0.55] | 0.53 [0.51, 0.55] |
|              | recall    | 0.56 [0.54, 0.58] | 0.51 [0.49, 0.53] | 0.51 [0.49, 0.53] |
|              | f1-score  | 0.54 [0.52, 0.56] | 0.52 [0.50, 0.54] | 0.52 [0.50, 0.54] |
Table 10: Results of training networks with the RMSprop optimizer. Values in brackets are 95% confidence intervals.

| Models       | Metrics | COVID-19          | Normal          | Pneumonia       |
|--------------|---------|-------------------|-----------------|-----------------|
|              | precision | 0.57 [0.55, 0.59] | 0.52 [0.50, 0.54] | 0.53 [0.51, 0.55] |
| InceptionV3  | recall   | 0.51 [0.49, 0.53] | 0.70 [0.68, 0.72] | 0.58 [0.57, 0.60] |
|              | f1-score | 0.54 [0.52, 0.56] | 0.60 [0.59, 0.62] | 0.55 [0.55, 0.57] |
| ResNet-50 V2 | precision | 0.56 [0.54, 0.58] | 0.63 [0.61, 0.65] | 0.54 [0.52, 0.56] |
|              | recall   | 0.84 [0.84, 0.85] | 0.36 [0.35, 0.38] | 0.49 [0.48, 0.51] |
|              | f1-score | 0.67 [0.66, 0.69] | 0.46 [0.43, 0.48] | 0.51 [0.49, 0.53] |
| DenseNet-201 | precision | 0.53 [0.51, 0.55] | 0.56 [0.54, 0.58] | 0.53 [0.53, 0.57] |
|              | recall   | 0.23 [0.21, 0.24] | 0.65 [0.62, 0.67] | 0.77 [0.75, 0.79] |
|              | f1-score | 0.32 [0.31, 0.34] | 0.60 [0.59, 0.62] | 0.64 [0.61, 0.66] |

Table 11: Results of training networks with the SGD optimizer. Values in brackets are 95% confidence intervals.

| Models       | Metrics | COVID-19          | Normal          | Pneumonia       |
|--------------|---------|-------------------|-----------------|-----------------|
| InceptionV3  | precision | 0.63 [0.61, 0.65] | 0.60 [0.59, 0.62] | 0.67 [0.66, 0.69] |
|              | recall   | 0.50 [0.47, 0.52] | 0.64 [0.61, 0.66] | 0.64 [0.61, 0.66] |
|              | f1-score | 0.56 [0.54, 0.58] | 0.62 [0.69, 0.64] | 0.65 [0.62, 0.67] |
| ResNet-50 V2 | precision | 0.50 [0.47, 0.52] | 0.56 [0.54, 0.58] | 0.52 [0.50, 0.54] |
|              | recall   | 0.58 [0.57, 0.60] | 0.55 [0.55, 0.57] | 0.43 [0.42, 0.47] |
|              | f1-score | 0.54 [0.52, 0.56] | 0.55 [0.55, 0.57] | 0.48 [0.45, 0.50] |
| DenseNet-201 | precision | 0.33 [0.30, 0.35] | 0.00 [0.00, 0.00] | 0.53 [0.30, 0.35] |
|              | recall   | 0.88 [0.86, 0.89] | 0.00 [0.00, 0.00] | 0.11 [0.09, 0.12] |
|              | f1-score | 0.48 [0.45, 0.50] | 0.00 [0.00, 0.00] | 0.17 [0.15, 0.18] |

4 Results and Discussion

Tables 9 and 10 show that different optimizers are differently effective for different networks, so the ResNet-50 V2 network together with the RMSProp optimization algorithm for the Recall metric was able to identify a larger number of images with COVID-19, while Inception V3 with SGD method, shows the highest results on average across classes, according to the F1 metric.

Thus, in the course of the study, it was revealed that to solve the problem of diagnosing COVID-19 using X-ray images, it is preferable to use the Inception network in conjunction with the SGD optimization algorithm. In this case, the values in the input images are recommended to be reduced to the range [0; 1] by means of scaling.

In the course of the experiments, models were upgraded for the problem, trained and tested based on the following architectures: Inception, ResNet and DenseNet. The accuracy of networks according to the accuracy metric during training and testing is shown in Fig. 10.
As seen from Fig. 10 the accuracy of the networks increases with the number of training epochs and tends to 1, which indicates the high quality of these algorithms in solving image classification problems.

5 Conclusion

In this paper, a study of deep machine learning algorithms in image recognition problems was carried out. Convolutional neural network architectures such as Inception, ResNet, and DenseNet were considered. Models were also trained and tested to solve the problem of diagnosing pneumonia and COVID-19 from chest x-rays. The results of the study showed that the Inception V3 network together with the SGD optimization algorithm is better at solving this problem. Preliminary reduction of values to the range [0; 1] can improve the quality of network training. Expanding training data and more epochs can significantly improve the quality of image recognition.

Compliance with Ethical Standards

Funding

No funding was received for this study.

Conflict of interest

The authors declare no conflict of interest, financial or otherwise.

Research involving human participants and/or animals

This research paper does not contain any studies with human participants or animals performed by any of the authors.

Ethical approval

Not needed.

Informed consent

All authors agreed on the submitted version.

References

[1] W. H. O. (WHO). Coronavirus. [Online]. Available: https://www.who.int/health-topics/coronavirus
[2] ——. Disease outbreak news. [Online]. Available: https://www.who.int/csr/don/archive/disease/novel_coronavirus/en/

[3] ——. Situation reports. [Online]. Available: https://covid19.who.int

[4] “Covid-19 coronavirus pandemic,” https://www.worldometers.info/coronavirus/, accessed: 2020-08-16.

[5] C. Sohrabi, Z. Alsafi, N. O'Neill, M. Khan, A. Kerwan, A. Al-Jabir, C. Iosifidis, and R. Agha, “World health organization declares global emergency: A review of the 2019 novel coronavirus (covid-19),” *International Journal of Surgery*, 2020.

[6] Y. Yang, Q. Lu, M. Liu, Y. Wang, A. Zhang, N. Jalali, N. Dean, I. Longini, M. E. Halloran, B. Xu et al., “Epidemiological and clinical features of the 2019 novel coronavirus outbreak in china,” *medRxiv*, 2020.

[7] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei et al., “Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study,” *The Lancet*, vol. 395, no. 10223, pp. 507–513, 2020.

[8] 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,” *The Lancet*, vol. 395, no. 10223, pp. 497–506, 2020.

[9] D. K. Chu, Y. Pan, S. M. Cheng, K. P. Hui, P. Krishnan, Y. Liu, D. Y. Ng, C. K. Wan, P. Yang, Q. Wang et al., “Molecular diagnosis of a novel coronavirus (2019-ncov) causing an outbreak of pneumonia,” *Clinical chemistry*, vol. 66, no. 4, pp. 549–555, 2020.

[10] S. Yadav, J. K. Sandhu, Y. Pathak, and S. Jadhav, “Chest x-ray scanning based detection of covid-19 using deepconvolutional neural network,” 2020.

[11] 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 coronavirus disease (covid-19),” *medRxiv*, 2020.

[12] V. M. Corman, O. Landt, M. Kaiser, R. Molenkamp, A. Meijer, D. K. Chu, T. Bleicker, S. Brünink, J. Schneider, M. L. Schmidt et al., “Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr,” *Eurosurveillance*, vol. 25, no. 3, p. 2000045, 2020.

[13] M.-Y. Ng, E. Y. Lee, J. Yang, F. Yang, X. Li, H. Wang, M. M.-s. Lui, C. S.-Y. Lo, B. Leung, P.-L. Khong et al., “Imaging profile of the covid-19 infection: radiologic findings and literature review,” *Radiology: Cardiothoracic Imaging*, vol. 2, no. 1, p. e200034, 2020.

[14] J. P. Kanne, B. P. Little, J. H. Chung, B. M. Elicker, and L. H. Ketai, “Essentials for radiologists on covid-19: an updated radiology scientific expert panel,” 2020.

[15] H. L. Fred, “Drawbacks and limitations of computed tomography: views from a medical educator,” *Texas Heart Institute Journal*, vol. 31, no. 4, p. 345, 2004.

[16] A. S. Bhalla, A. Goyal, R. Guleria, and A. K. Gupta, “Chest tuberculosis: Radiological review and imaging recommendations,” *The Indian journal of radiology & imaging*, vol. 25, no. 3, p. 213, 2015.

[17] P. S. Ngoya, W. E. Muhogora, and R. D. Pitcher, “Defining the diagnostic divide: an analysis of registered radiological equipment resources in a low-income african country,” *The Pan African medical journal*, vol. 25, 2016.

[18] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[19] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images and video: a survey,” *Pattern recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[20] P. Remagnino and G. L. Foresti, “Ambient intelligence: A new multidisciplinary paradigm,” *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, vol. 35, no. 1, pp. 1–6, 2004.

[21] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *Journal of Big Data*, vol. 6, no. 1, p. 113, 2019.

[22] ——, “Machine learning algorithms for disease prediction using iot environment,” *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 4303–4307, 2019.

[23] S. S. Yadav, V. J. Kadam, and S. M. Jadhav, “Comparative analysis of ensemble classifier and single base classifier in medical disease diagnosis,” in *International Conference on Communication and Intelligent Systems*. Springer, 2019, pp. 475–489.
[24] S. S. Yadav, S. M. Jadhav, R. G. Bonde, and S. T. Chaudhari, “Automated cardiac disease diagnosis using support vector machine,” in *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)*. IEEE, 2020, pp. 56–61.

[25] S. S. Yadav, S. M. Jadhav, S. Nagrale, and N. Patil, “Application of machine learning for the detection of heart disease,” in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, 2020, pp. 165–172.

[26] S. S. Yadav and S. M. Jadhav, “Detection of common risk factors for diagnosis of cardiac arrhythmia using machine learning algorithm,” *Expert Systems with Applications*, p. 113807, 2020.

[27] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” 2014.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[30] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2016.

[31] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images,” p. 12, 2020.

[32] “Covid-19 radiography database,” [https://www.kaggle.com/tawsifurrahman/covid19-radiography-database](https://www.kaggle.com/tawsifurrahman/covid19-radiography-database), accessed: 2020-04-08.