Processing and Analyzing Images based on a Neural Network

Bohdan Zhurakovskyi[1], Vadym Poltorak[1], Serhii Toliupa[2], Oleksandr Pliushch[2], and Artem Platonenko[3]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 37 Peremogy ave., Kyiv, 03056, Ukraine*
[2] *Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., Kyiv, 01601, Ukraine*
[3] *Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., Kyiv, 04053, Ukraine*

Abstract
Medical image processing technologies allow for automating and improving diagnostic and analysis processes, providing doctors with more accurate and faster results. The use of artificial intelligence, deep learning, and computer vision allows for the creation of efficient and automated systems that can detect pathologies, classify images, and provide valuable decision support to doctors. The description and preliminary processing of the data set, which is a key stage for the preparation of system input data, has been performed. Models for training are also developed, including the selection and tuning of neural network architectures. The introduction of a new method for training a neural network turned out to be very successful. This approach significantly improved the training quality of the model, helping to increase the accuracy and ability of image classification. The application of this method significantly improved the efficiency and reliability of the X-ray image recognition system. The research results indicate that the new learning method, based on the combination of Adam and SGD methods, raised the accuracy of image recognition to the level of 95–97% while increasing the training time by only 1–2%. The developed system can be considered as an initial version that paves the way for further improvement. It was determined that the main driving factor for improving the system is the developed neural network training method.

Keywords
Image recognition, neural network, machine learning, system, classification, model, model training, dataset, accuracy, training, efficiency.

1. Introduction

The topic of creating an information system for processing and analyzing medical images using artificial intelligence and developing a system for processing medical images in real time are extremely relevant and important in the field of medicine [1]. Here are some aspects of the relevance of these topics:

- Improvement of diagnostic accuracy. Using artificial intelligence to analyze medical images can help doctors detect symptoms and pathologies that may be difficult to detect using traditional methods. This can lead to more accurate and earlier diagnoses.
- Reducing the burden on medical personnel. Medical imaging systems can automate the analysis process, helping doctors focus on important tasks and reducing the burden on medical staff.
- Quick access to information. The development of systems for processing medical images in real-time will allow doctors to have instant access to the results of the analysis, which is especially useful in urgent situations, such as injuries or serious diseases [2].
- Reducing the risk of errors. Artificial intelligence can help weed out false

positives and increase the reliability of diagnoses [3].

- Expanding access to medical care. Such systems can help reduce inequalities in access to health care, as they can be used in different health facilities, including remote areas [4].
- Discovery of new research opportunities. Processing and analysis of medical images can help in the development of new research and approaches to the treatment of various diseases.

In this regard, the development of systems for processing and analyzing medical images using artificial intelligence and systems for processing medical images in real-time has great potential for improving the quality of medical care and saving patients' lives.

In the process of developing the system, the peculiarities of the medical field and specific requirements related to the processing and analysis of medical images should be taken into account. This includes taking into account a high standard of confidentiality and data security, as medical information is particularly sensitive. It is also important to take into account the variety of types of medical images, which requires the development of flexible algorithms capable of working with different formats and modalities.

In addition, it is important to consider the specifics of interaction with medical personnel, providing a convenient and efficient user interface for interacting with the system. The possibility of integrating the system with existing medical information systems to ensure interaction and data exchange should also be taken into account [5].

In addition, this study may have practical value for developers, simplifying their work and opening new opportunities for the implementation of advanced technologies in medical practice [6]. The developed algorithms and methods of medical image processing can become the basis for the creation of intelligent decision support systems in the medical field, contributing to the automation and improvement of diagnostic and treatment processes. This approach not only expands the capabilities of developers in the field of medical informatics but also promotes the implementation of modern technologies to achieve maximum accuracy and speed in the analysis of medical images.

The purpose and objectives of the research. The main goal of the development is to create a system for processing and analyzing medical images aimed at improving X-ray diagnostics of chest organs. This system is designed to increase the accuracy and efficiency of detection of pathologies and diseases, as well as to optimize the time and resources of the medical staff.

2. Statement of Research Problem

Modern technologies allow automation of the processing and analysis of medical images, which contributes to the efficiency and accuracy of diagnosis. This is especially important in the conditions of increasing volume of medical data.

The use of innovative technologies in the processing and analysis of medical images opens up new opportunities for accurate diagnosis, treatment, and monitoring of patients. Such studies contribute to the development of automated systems that help specialists in the fast and reliable interpretation of medical images.

In particular, the development and improvement of algorithms for the recognition of pathologies on X-ray images, detection of signs of diseases on computer tomography, and analysis of other medical images help to improve the speed and accuracy of diagnosis.

Given the rapid pace of technology development and the constant replenishment of medical databases, research in this direction is important for ensuring effective and modern medical practice.

The development of systems for image processing helps to improve the accuracy of diagnosis of various diseases. The ability to automatically detect pathologies and abnormalities in images allows for early detection and effective treatment.

The medical image analysis system can serve as an effective tool for monitoring the progress of diseases, their dynamics, and their response to treatment, which is important in conducting medical statistics and optimizing treatment strategies. Pneumonia and COVID-19 require a significant amount of medical imaging, such as X-rays and CT scans of the lungs. An automated system for their processing and analysis can greatly facilitate

the work of medical personnel. In the case of large epidemics such as COVID-19, a medical image processing system can be used to quickly detect and track the spread of diseases in the population [7].

2.1. Formation of System Requirements

Having analyzed the main popular approaches and already available similar solutions for the classification of objects in images, it was decided to develop the technical requirements for the system that is planned to be developed [8].

The development of a system for processing and analyzing medical images and classifying objects on them is a complex task and requires the definition of technical requirements for the successful implementation of the project. Here are some general technical requirements that may be important for this process:

1. Processing and storage of medical images. The system must be able to download and store medical images, in particular DICOM images, which are widely used in medicine.
2. Segmentation and definition of regions of interest. The system should be able to automatically identify and highlight Regions of Interest (ROIs) on medical images for further processing and analysis.
3. Using deep learning. It is recommended to use deep learning, in particular Convolutional Neural Networks (CNN), to recognize and classify objects in images.
4. Ability to track and analyze changes in real-time. The system should support the analysis of changes in medical images in real-time, in particular for patient monitoring.
5. Support for various types of medical images. The system should be universal and support various types of medical images, such as X-rays, CT, MRI, etc.
6. Data protection and confidentiality. Ensuring a high level of security and confidentiality of medical data, including data storage and transfer requirements.
7. Possibility of integration with other systems. The system must be able to integrate with other medical systems, hospital information systems, and data management systems.
8. Training and retraining of models. The ability to train and retrain models to improve the accuracy of medical image classification and analysis.
9. User interface. Development of a convenient user interface for doctors and medical professionals to navigate, visualize, and analyze results.
10. Documentation and Support. Provision of documentation, instructions, and technical support for system users.
11. Compliance with regulatory requirements. Consideration of regulatory requirements and standards in the medical field when developing and operating the system.
12. Evaluation of results and metrics. Establishing metrics to evaluate system accuracy and performance, such as sensitivity, specificity, accuracy, and others.
13. These requirements can be adapted according to the specific needs of further development of the system.

2.2. Design of the Developed System. Dataset Description and Preprocessing

In recent years, the use of artificial intelligence and machine learning in medical diagnostics has shown great promise, particularly in the recognition of lung diseases.

Scientists and researchers use a variety of chest X-ray image datasets to study and train algorithms for the recognition of opacities, pneumonia, and COVID-19. Among these datasets, several popular ones can be singled out, which have become key tools in the study and development of algorithms for the automatic detection and classification of lung diseases [9].

Below is a list of several important datasets used in this area:

1. The COVID-19 Image Data Collection is a dataset of X-ray images to study the impact of COVID-19 on the lungs. It was created by collecting medical images from websites and publications and currently contains 123 frontal radiographs [10].
2. Chest X-ray images (pneumonia) are a dataset of X-ray images for the study of

pneumonia. The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal). There are 5863 x-ray images (JPEG) and 2 categories (pneumonia/normal) [11].

3. The RSNA Pneumonia Detection Challenge is a data set within the RSNA X-ray Pneumonia Detection Challenge. 30,000 frontal chest radiographs from 112,000 publicly available images from the National Institutes of Health. Portable Network Graphics images were converted to Digital Imaging and Communications in Medicine, and patient gender, patient age, and prognosis (anteroposterior or posteroanterior) were added to the Digital Imaging and Communications in Medicine tag [12].

4. NIH Chest X-rays is a dataset of chest X-rays, including pneumonia images. NIH Chest X-ray Dataset—This NIH Chest X-ray Study dataset consists of 112,120 disease-labeled X-ray images from 30,805 unique patients. To create these labels, the authors used natural language processing to retrieve textual disease classifications from the corresponding radiology reports. Labels are expected to be >90% accurate and suitable for weakly supervised learning [13].

5. CheXpert is a dataset containing annotated images of chest X-rays for the diagnosis of various diseases, including pneumonia. The CheXpert dataset contains 224,316 chest radiographs from 65,240 patients with both frontal and lateral views available [14].

2.3. Creating Models for Learning

Deep learning algorithms can be trained on large datasets of chest X-rays to recognize patterns and features that indicate pneumonia. This involves the use of CNNs, a type of deep learning architecture that is particularly well-suited to image recognition tasks [15]. By analyzing the texture, shape, and intensity of pixels in chest X-ray images, CNNs can learn to identify areas of the image that correspond to areas of infection or inflammation in the lungs [16].

Once trained, deep learning models can be used to classify new chest X-rays as showing signs of pneumonia or not. This can be done in real-time, making it a potentially valuable tool for healthcare providers in the diagnosis and management of patients with pneumonia. In addition, deep learning models can be used to help radiologists interpret chest X-rays, reducing the risk of misdiagnosis and improving patient outcomes.

The purpose of the model is to classify X-ray images of the chest into normal and pneumonic classes. The original chest X-rays are used as a basis for data addition procedures. Pre-trained models are used in combination with augmented images to classify pneumonia.

The "Sequential_1" model (Fig. 1) uses a basic and simple approach to solving number classification problems. This model is built on convolution layers, and fully connected layers, and includes maximum pooling and extraction in the middle layers. This architectural approach allows you to effectively cope with the classification of numerical data [17], and its high level of efficiency confirms its success in solving the relevant problem.

The "Sequential_2" model is a deep convolutional neural network that includes maximum pooling and fully connected layers at the end. This model configuration with a fixed number of elements and layers has shown the best results in various studies using different data sets (Fig. 2) [18].

```
Model: "sequential_1"

Layer (type)                 Output Shape              Param #
=================================================================
model_1 (Model)              (None, 100, 100, 384)     11008

conv2d_4 (Conv2D)            (None, 98, 98, 64)        221248

activation_1 (Activation)    (None, 98, 98, 64)        0

max_pooling2d_1 (MaxPooling2 (None, 49, 49, 64)        0

conv2d_5 (Conv2D)            (None, 47, 47, 32)        18464

activation_2 (Activation)    (None, 47, 47, 32)        0

max_pooling2d_2 (MaxPooling2 (None, 23, 23, 32)        0

flatten_1 (Flatten)          (None, 16928)             0

dropout_1 (Dropout)          (None, 16928)             0

dense_1 (Dense)              (None, 128)               2166912

dropout_2 (Dropout)          (None, 128)               0

dense_2 (Dense)              (None, 64)                8256

dropout_3 (Dropout)          (None, 64)                0

dense_3 (Dense)              (None, 2)                 130
=================================================================
Total params: 2,426,018
Trainable params: 2,426,018
Non-trainable params: 0
```

Figure 1: Results of using the model Sequential_1

dense (Dense)	(None, 1024)	525312
dense_1 (Dense)	(None, 512)	524800
dense_2 (Dense)	(None, 256)	131328
dropout (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 2)	258

```
Total params: 15,929,282
Trainable params: 1,214,594
Non-trainable params: 14,714,688
```

Figure 2: Results of using the model Sequential_2

A third modeling approach involved the use of transfer learning using VGG-16 with some modifications in the last three levels (Fig. 3).

Here, the Chest Pneumonia X-ray dataset is used to collect Pneumonia X-ray images that take into account images from various open sources and are reviewed regularly. Here, two datasets are used to train models for pneumonia diagnosis. The first dataset consists of 5856 chest X-ray images, of which 4273 are pneumonia images and 1583 are normal chest X-ray images [19].

```
Model: "vgg16"
```

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808

Figure 3: Results of using the model VGG-16

A total of 80% of the data is used for training, generating 4642 images (3418 pneumonia images and 1224 normal images), 15% of the data is used for testing, generating 919 images (641 pneumonia cases and 278 normal images), and the last 5% of the data is used for validation (214 cases of pneumonia and 81 images without pneumonia).

The second dataset taken is ChestX-ray14, which contains 112,120 chest X-rays of 30,085 individuals. Of these 112,120 images, 1,431 images had signs of pneumonia. To obtain a balanced data set, 1431 normal X-ray images (marked as "No Results") were selected from the data set. Thus, the final collected data set contains 1431 pneumonia images and 1431 normal X-ray images. 80% of the data is used for training, generating 2290 images (1145 pneumonia images and normal images each), 5% of the data is used for validation, generating 142 images (215 pneumonia images and normal images each), and the last 15% of the data is used for testing, yielding 430 (71 images of pneumonia and normal images).

The count plot of the first data set is used to display pneumonia counts and normal images. To represent all images, Fig. 3 shows that the X-axis of the training set contains the values 0 (corresponding to 1224 normal images) and 1 (representing 3418 pneumonia images), while the X-axis of the test set contains the values 0 (representing 278 normal images) and 1 (representing 641 pneumonia images), and the Y-axis of the training set displays the count graph of both pneumonia and normal images (Fig. 4).
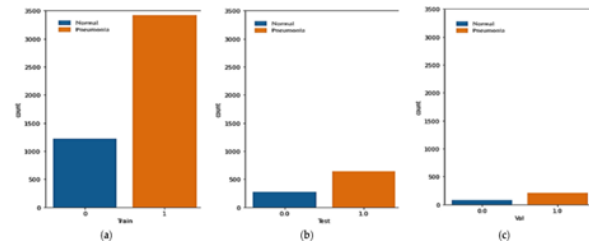

Figure 4: Count plot showing counts for the pneumonia and normal data sets

CNN models require a large number of data sources for optimal training to demonstrate improved performance on larger datasets. Since only a small dataset is used, this is used to artificially expand the dataset. It also helps to avoid overfitting. A data augmentation approach has often been used and increases the number of images by applying a series of changes while preserving the class labels [20]. Data augmentation is applied to the pneumonia class training images to increase image diversity, which also acts as a dataset regulator.

The training and validation loss plot can provide information about how the model learns and how well it generalizes its knowledge to the validation data. Loss

determines how many errors the model makes in forecasting.

Training Loss—training loss reflects how accurately the model predicts the training data during each epoch. Typically, the loss should decrease during training, indicating that the model is learning to identify patterns in the data.

Validation Loss—Validation loss determines how well the model generalizes its skills to data it has not seen during training. A decrease in validation loss indicates that the model effectively recognizes patterns and avoids overfitting. The graph has the form where the epochs (training iterations) are displayed on the X-axis, and the loss value (from 0 to the highest loss value) on the Y-axis is shown in Fig. 5.

A good scenario is that both curves (training and validation loss) will show a decline, indicating the effectiveness of training and generalization. It is important to ensure that the validation loss does not increase, a sign of possible overtraining.
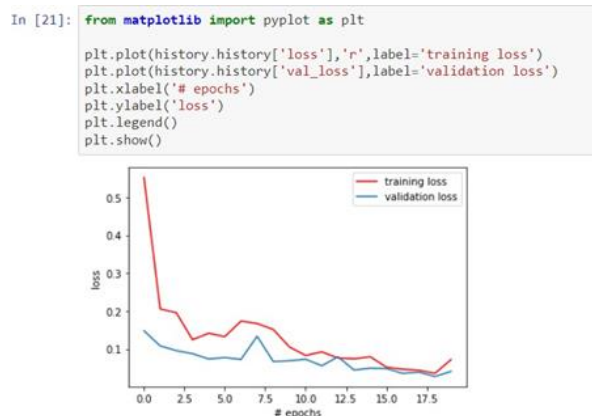
```
In [21]: from matplotlib import pyplot as plt

         plt.plot(history.history['loss'],'r',label='training loss')
         plt.plot(history.history['val_loss'],label='validation loss')
         plt.xlabel('# epochs')
         plt.ylabel('loss')
         plt.legend()
         plt.show()
```


Figure 5: Loss training and validation schedule

The model training schedule in machine learning can include two important metrics: training accuracy and validation accuracy. Both metrics reflect how well the model has learned to recognize patterns in the data during training and how well it generalizes those skills to new data.

Accuracy on the training set (Training Accuracy)—this metric determines the accuracy of the model on the data it used during training. If the model has learned well, the accuracy of the training set will be high. However, it is important to ensure that the model does not overtrain (overfitting) specific examples in the training data and can effectively recognize new data.

Accuracy on the validation set (Validation Accuracy)—this metric determines the accuracy of the model on data that it has not seen during training. It is an indicator of how well the model generalizes its knowledge to new, previously unseen data. If the accuracy of the validation set also increases during training, this may be an indication that the model is effectively learning and detecting common patterns, rather than simply "remembering" the training data.

The graph has the form where epochs (training iterations) are displayed on the X axis, and accuracy (from 0 to 1 or percent) on the Y axis in Fig. 6. Plots for training and validation accuracy can help determine model performance and identify potential problems such as overtraining or undertraining.
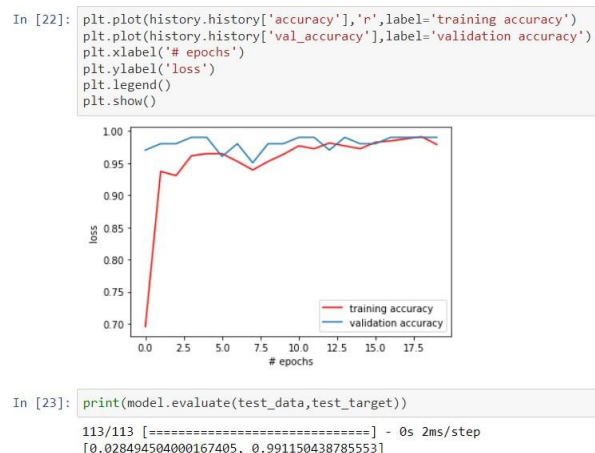
```
In [22]: plt.plot(history.history['accuracy'],'r',label='training accuracy')
         plt.plot(history.history['val_accuracy'],label='validation accuracy')
         plt.xlabel('# epochs')
         plt.ylabel('loss')
         plt.legend()
         plt.show()
```



```
In [23]: print(model.evaluate(test_data,test_target))

         113/113 [==============================] - 0s 2ms/step
         [0.028494504000167405, 0.991150438785553]
```

Figure 6: Model training schedule

A histogram of people's ages on X-ray images (Fig. 7), presented together with the number of training images, gives a visual impression of the age distribution in the dataset.

The x-axis of the histogram shows the age ranges, and the y-axis shows the number of images corresponding to each range. This graphical representation can indicate the age at which X-ray examinations are more common or the age distribution of patients in the sample to train the model.

Such analysis can be useful for understanding whether there is a diversity of age groups in the training dataset and whether particular age groups should be given special attention when training the model.
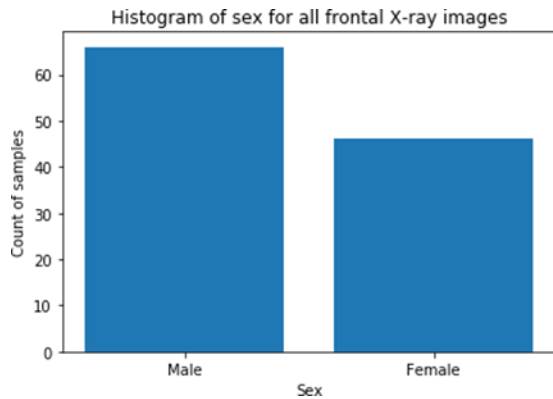
Figure 7: Histogram of the age of people

A histogram of gender on X-ray images can be generated to visualize the distribution of male and female patients in the training dataset. Two bars are marked on the x-axis, representing the number of images for each gender.
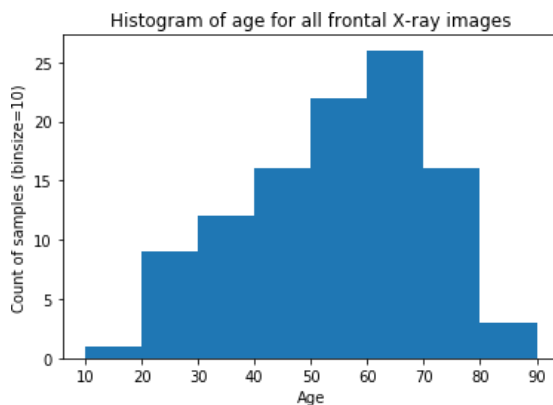

Figure 8: Gender histogram

The information provided in Fig. 9 is metadata for a specific X-ray image and contains some key attributes [21, 22]:

1. patientid (patient identifier)—a unique patient identifier that can be used to track and analyze medical information.
2. offset—this value may indicate certain parameters or displacements associated with the X-ray examination.
3. sex—indicates the gender of the patient.
4. age—shows the age of the patient at the time of the X-ray.
5. finding—describes the result of the examination. In this case, "COVID-19" indicates the detection of the 2019 coronavirus disease.
6. survival—shows whether the patient survived ("Y"—yes).
7. view—indicates how the X-ray examination was carried out. In this

case, "AP Supine" is an anterior-posterior (anteroposterior) view in the supine position.
8. modality—shows what type of examination was used. In this case, "X-ray" is X-ray radiation.
9. date—indicates the date of the X-ray examination.
10. location—shows the place where the survey was conducted.
11. url (URL)—a link to a source or additional information. In this case, a URL is provided where additional information can be obtained.
12. clinical notes—indicates clinical or medical information about the patient's condition.
13. other notes—other additional notes or instructions that may be useful for understanding the context or other details of the examination.
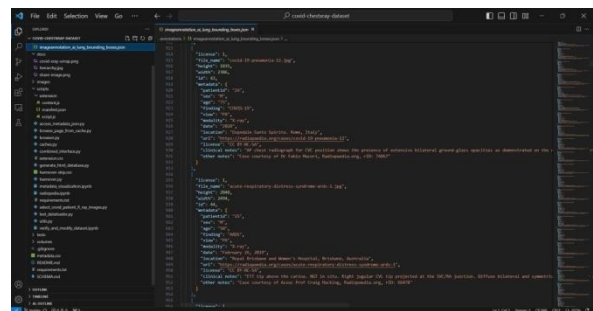

Figure 9: X-ray image metadata

The generated pie chart provides information on the number of images on which different pathogens and conditions are detected (Fig. 10). Each sector of the diagram corresponds to a specific pathology, and its size indicates the percentage or number of images where this pathology was detected [23].
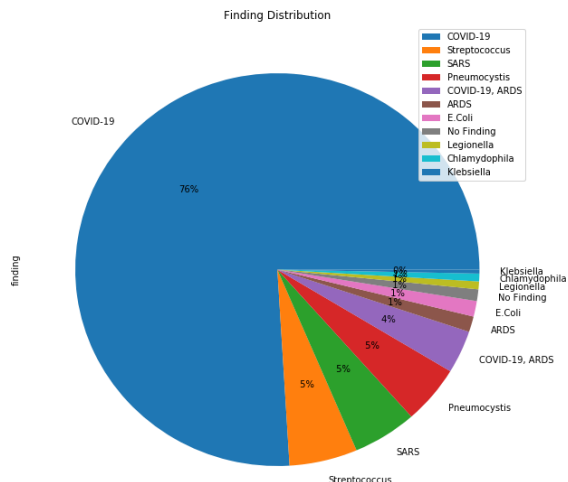

Figure 10: Diagram of the number of images

Implementation of a new method of neural network training

The following algorithms were analyzed:
1. Nesterov Accelerated Gradient
2. Adagrad
3. Adam
4. Adamax
5. RMSProp
6. Adadelta
7. SGD.

The comparison of algorithms made it possible to stop the use of the Adam algorithm. Adam (Adaptive Moment Estimation) is a popular optimization algorithm, especially in deep learning, for several reasons:

1. Learning rate adaptability: Adam uses adaptive learning rates for each parameter, allowing it to efficiently adapt to the geometry of the loss function landscape.
2. Integration with moments: Adam combines the concept of moments (from Nesterov Accelerated Gradient) with the idea of an adaptive penalty on the norm of gradients (from RMSProp). This makes it possible to effectively solve the problem of directed and non-directed gradients [24].
3. Performance in multidimensional spaces: Adam often shows good performance in multidimensional parameter spaces, which is usually characteristic of deep neural networks [25].
4. Defaults: A big advantage of Adam is that it has reasonable defaults that usually work well in many cases without the need for extensive tuning.

Choosing a modified optimization strategy that involves branching at the damping point of the approach to the extremum is an important decision. This approach can have several advantages, such as improving convergence, adapting to the landscape of the loss function, and avoiding local minima more efficiently [26].

The VGG input is set to a 224×244 RGB image. The average RGB value is calculated for all images in the training set image, and then the image is fed as input to the VGG convolutional network. A 3×3 or 1×1 filter is used, and the convolution step is fixed. There are 3 fully connected VGG layers, which can vary from VGG11 to VGG19 according to the total number of convolutional layers+fully connected layers. The minimal VGG11 has 8 convolutional layers and 3 fully connected layers.

The maximum VGG19 has 16 convolutional layers. +3 fully connected layers. In addition, the VGG network is not accompanied by a pooling layer behind each convolutional layer or a total of 5 pooling layers distributed under different convolutional levels.

Each convolution layer (Fig. 11) in AlexNet contains only one convolution, and the size of the convolution kernel is 7×7. In VGGNet, each convolution layer contains 2 to 4 convolution operations. The convolution kernel size is 3×3, the convolution step size is 1, the pooling kernel is 2×2, and the step size is 2. The most obvious improvement of VGGNet is to reduce the size of the convolution kernel and increase the number of convolution layers [27].

Using multiple convolution layers with smaller convolution kernels instead of a larger convolution layer with convolution kernels can reduce parameters on the one hand, and the author believes that this is equivalent to a more nonlinear mapping that increases the ability of the Fit expression [28].
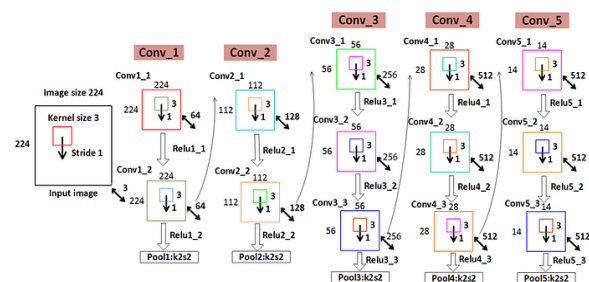

Figure 11: Each layer of convolution

To obtain a 224×224 input image, each scaled image is randomly cropped during each SGD iteration. To improve the dataset, the cropped image is also randomly flipped horizontally and the RGB color shifted.

A 1×1 convolution kernel is introduced into the VGGNet convolution structure. Without affecting the input and output dimensions, a nonlinear transformation is introduced to increase the expressive power of the network and reduce the number of calculations [27].

2.4. Testing and Conducting Experiments

This experiment investigated and compared the performance of different models based on test sets obtained from well-known sources such as PyramidNet, ResNet-32, DenseNet, and SENet. To achieve this goal, 2200 test images

were used, which were analyzed for their classification ability on various models [29]. Each model has been carefully tuned and evaluated to determine which model performs best in object detection on test sets. This experiment provided an opportunity not only to evaluate the effectiveness of each model but also to compare their characteristics to determine the optimal option for further research and applications.

The experiment included several stages:
1. Selection of test sets. Definition of 2200 test sets which were obtained from different sources such as ResNet-32, DenseNet, PyramidNet, and SENet.
2. Data preparation. Processing and preparation of the received test sets for use in the experiment, including standardization and other necessary operations.
3. Application of models. Using different models such as ResNet-32, DenseNet, PyramidNet, and SENet to classify objects on test sets.
4. Collection of results. Evaluate the performance of each model based on its ability to classify objects on the test sets.
5. Analysis of the results. Comparing the performance of different models to determine which one exhibits the best accuracy or other important characteristics.

The conducted study showed that the Adam optimization algorithm, as a rule, turns out to be more effective in terms of learning speed compared to the Stochastic Gradient Descent (SGD) method. This is because Adam uses adaptive learning rates for each parameter separately, which allows you to approach the optimum from different angles and reduces the probability of getting stuck in local minima. This approach can lead to faster convergence of the model during training [30].

In the initial training stages, Adam [31] may be faster because it uses adaptive learning rates and can more closely approximate the optimal parameter values. However, with further training, when the loss function approaches local minima, SGD can become more efficient [32].

For example, consider a situation where the loss function has many local minima. In this case, Adam may exhibit high speed during the initial training phase but may become less efficient when the loss function approaches one of the local minima. On the other hand, SGD, having less computational complexity, may seem simpler and more efficient at later stages of learning, where it is important to avoid getting stuck in local minima [33].

Therefore, the effectiveness of the SGD and Adam methods may vary depending on specific data, model characteristics, and learning parameters. It is recommended to conduct a series of experiments with different optimization methods for a specific task to determine which one works faster and more efficiently under specific conditions.
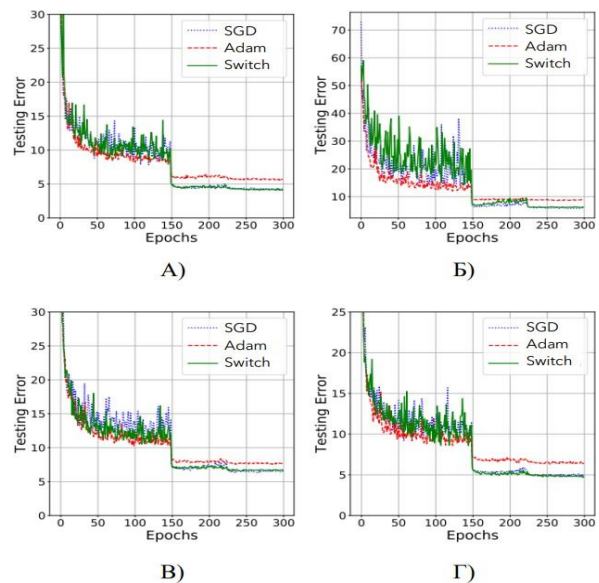

Figure 12: The results of the study of the learning process on different samples: А) ResNet-32, Б) DenseNet, В) PyramidNet, Г) SENet

Optimization methods such as SGD (Stochastic Gradient Descent) and Adam have their advantages and disadvantages, which are important to consider when choosing a specific task.

SGD requires less memory because it uses only one random subset for each weight update. SGD may be less sensitive to large amounts of noise in the data because it uses random subsampling. Convergence can be slower, especially in deep neural networks, because SGD can get stuck in local extrema.

Adam adaptively adjusts the learning rate for each parameter, making it efficient in multidimensional spaces. It usually turns out to be effective in practice for different tasks and datasets. Adam uses more memory to store additional information, such as exponentially

smoothed gradients and their squares.

The average of test results on test images is presented in Table 1.

The combination of SGD and Adam methods leads to the highest accuracy of the model on the test set, but at the same time increases the training time slightly compared to using SGD or Adam alone.

Table 1
Average test results

Optimization algorithm	Accuracy of the model, %	Study time, s
SGD	90.5	172
Adam	92.2	220
Combined SGD and Adam	93	229

3. Conclusion

The application of medical image processing contributes to improving the accuracy of diagnoses, early detection of diseases, and improving the ability of systems to monitor the dynamics of pathologies over time.

The analysis revealed that there is a large and diverse set of methods in the field of pattern recognition in medicine. Such tasks often require the use of non-trivial methodologies, and the development of original models and algorithms. Among the main challenges are the analysis of complex and hybrid images, as well as the improvement of pattern recognition systems.

The analysis of the subject area and existing solutions indicates that image recognition in radiological studies is an actual and promising field of research. Chest X-ray images data set was analyzed.

This dataset includes chest radiographs that can be used to recognize pathologies and detect abnormalities in the images. A detailed review of this dataset allowed us to consider its features and opportunities to improve the performance of our recognition system. Given the diversity of data in this set, we were able to tune our model parameters to more accurately and reliably detect various pathologies in X-ray images.

The description and preliminary processing of the data set, which is a key stage for the preparation of system input data, has been performed. Models for training are also developed, including the selection and tuning of neural network architectures.

The introduction of a new method for training a neural network turned out to be very successful. This approach significantly improved the training quality of the model, helping to increase the accuracy and ability of image classification. The application of this method significantly improved the efficiency and reliability of the X-ray image recognition system. The research results indicate that the new learning method, based on the combination of Adam and SGD methods, raised the accuracy of image recognition to the level of 95–97% while increasing the training time by only 1–2%.

The developed system can be considered as an initial version that paves the way for further improvement. It was determined that the main driving factor for improving the system is the developed neural network training method. This method is based on the efficient use of Adam and GD methods depending on different input parameters.

References

[1] K. Khorolska, et al., Application of a Convolutional Neural Network with a Module of Elementary Graphic Primitive Classifiers in the Problems of Recognition of Drawing Documentation and Transformation of 2D to 3D Models, J. Theor. Appl. Inf. Technol. 100(24) (2022) 7426–7437.

[2] V. Buriachok, V. Sokolov, P. Skladannyi, Security Rating Metrics for Distributed Wireless Systems, in: Workshop of the 8th International Conference on "Mathematics. Information Technologies. Education:" Modern Machine Learning Technologies and Data Science, vol. 2386 (2019) 222–233.

[3] B. Bebeshko, et al., Application of Game Theory, Fuzzy Logic and Neural Networks for Assessing Risks and Forecasting Rates of Digital Currency, J. Theor. Appl. Inf. Technol. 100(24) (2022) 7390–7404.

[4] F. Kipchuk, et al., Assessing Approaches of IT Infrastructure Audit, in: IEEE 8th International Conference on Problems of Infocommunications, Science and

Technology (2021). doi: 10.1109/picst54195.2021.9772181.

[5] O. Shevchenko, et al., Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support, in: Cybersecurity Providing in Information and Telecommunication Systems Vol. 2923 (2021) 277–282.

[6] Z. B. Hu, V. Buriachok, V. Sokolov, Deduplication Method for Ukrainian Last Names, Medicinal Names, and Toponyms Based on Metaphone Phonetic Algorithm, Advances in Computer Science for Engineering and Education III, vol. 1247 (2020) 518–533. doi:10.1007/978-3-030-55506-1_47.

[7] I. Tsochantaridis, et al., Large Margin Methods for Structured and Interdependent Output Variables, J. Mach. Learning Res. 20(9) (2019) 1–41.

[8] L. Deng, D. Yu, Deep Learning: Methods and Applications. Foundations and Trends® in Signal Processing 14(1–2) (2021) 300–370.

[9] Real-World Applications of Convolutional Neural Networks (2021). URL: https://vitalflux.com/real-world-applications-of-convolutional-neural-networks/

[10] Detection and Analysis of COVID-19 in Medical Images Using Deep Learning Techniques (2023). URL: https://www.nature.com/articles/s41598-021-99015-3

[11] Chest X-Ray Images (Pneumonia) (2023). URL: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[12] RSNA Pneumonia Detection Challenge Dataset Description (2018). URL: https://www.rsna.org/rsnai/ai-image-challenge/rsna-pneumonia-detection-challenge-2018

[13] NIH Chest X-rays (2018). URL: https://www.kaggle.com/datasets/nih-chest-xrays/data

[14] CheXpert (2018). URL: https://paperswithcode.com/dataset/chexpert

[15] What is a convolutional neural network (CNN)? (2019). URL: https://www.techtarget.com/searchenterpris

eai/definition/convolutional-neural-network

[16] Convolutional Neural Network (CNN) (2023). URL: https://developer.nvidia.com/discover/convolutional-neural-network

[17] F. Nataliia, et al., Software System for Processing and Visualization of Big Data Arrays, Adv. Comput. Sci f Eng. Educ. (2022) 324–336.

[18] B. Zhurakovskyi, et al., Modifications of the Correlation Method of Face Detection in Biometric Identification Systems, in: Cybersecurity Providing in Information and Telecommunication Vol. 3288 (2022) 55–63.

[19] S. Bandyopadhyay, S. Dutta, Associating Unemployment with PanicAttack Using Stacked-RNN Model During COVID-19 (2020). doi: 10.20944/preprints202006.0242.v1.

[20] B. Zhurakovskyi, et al., Calculation of Quality Indicators of the Future Multiservice Network, Future Intent-Based Networking (2022) 197–209. doi: 10.1007/978-3-030-92435-5_11.

[21] J. Amin, et al., Big Data Analysis for Brain Tumor Detection: Deep Convolutional Neural Networks, Future Gener. Comput.Syst. 87 (2018) 290–297.

[22] S. Obushnyi, et al., Autonomy of Economic Agents in Peer-to-Peer Systems, in:Cybersecurity Providing in Information and Telecommunication Systems Vol. 3288 (2022)125–133.

[23] V. Druzhynin, et al., Features of Processing Signals from Stationary Radiation Sources in Multi-Position Radio Monitoring Systems, CEUR Workshop Proceedings Vol. 2746 (2020) 46–65.

[24] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, International Conference on Learning Representations (2015).

[25] A. Graves, Generating Sequences with Recurrent Neural Networks, arXiv, preprint (2013).

[26] A. Graves, Alex, A.-R. Mohamed, G. Hinton, Speech Recognition with Deep Recurrent Neural Networks. IEEE International Conference on Acoustics, Speech and Signal Processing (2013)

6645–6649. doi: 10.1109/ICASSP.2013. 6638947.

[27] VGG Net Architecture (2023). URL: https://medium.com/@sajals1146/vgg -net-architecture-880df59c37ea

[28] B. Bejnordi, et al., Diagnostic Assessment of Deep Learning Algorithms For Detection of Lymph Node Metastases in Women with Breast Cancer, JAMA 318(22) (2017) 2199–2210. doi: 10.1001/jama.2017.14585.

[29] M. Moshenchenko, et al., Optimization Algorithms of Smart City Wireless Sensor Network Control, in: Cybersecurity Providing in Information and Telecommunication Systems II Vol. 3188 (2021) 32–42.

[30] B. Zhurakovskyi, et al., Secured Remote Update Protocol in IoT Data Exchange System, in: Cybersecurity Providing in Information and Telecommunication Systems Vol. 3421 (2023),67–76.

[31] Adamax (2018). URL: https://keras.io/ api/optimizers/adamax/

[32] Stochastic Gradient Descent (SGD) (2023). URL: https://scikit-learn.org/stable/modules/sgd.html

[33] L. Ziyin, H. Li, M. Ueda, Law of Balance and Stationary Distribution of Stochastic Gradient Descent, arXiv (2023). doi: 10.48550/arXiv.2308.06671