The Intelligent Diagnosis Method of Covid-19 Based on the Lenet-Vit Deep Neural Network

Eugene Fedorov[a], Tetyana Utkina[a], Maryna Leshchenko[a], Olga Nechyporenko[a] and Kostiantyn Rudakov[a]

[a] *Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine*

Abstract

The method for intelligent diagnosis of COVID-19 based on the LeNet-ViT deep neural network was proposed. The LeNet-ViT model was created, it has the following advantages: the input image is not square, which expands the scope; the input image is pre-compressed and the new size depends on the original image size, and it is empirically determined, which increases the model training speed and the model identification accuracy; the number of pairs “convolutional layer - downsampling layer” depends on the image’s size, and it is automatically determined, which increases the model classification accuracy; the number of layer planes is automatically determined, which speeds up the definition of the model structure; the patch size depends on the image size, and it is empirically determined, which increases the model identification accuracy; the number of encoder blocks is empirically determined, which increases the model learning speed; the use of a convolutional neural network allows to efficiently extract features, and the use of a visual transformer allows to effectively analyze these features. The proposed method for intelligent diagnosis of COVID-19 can be used in various intelligent computer systems for medical diagnostics.

Keywords [1]

intelligent diagnostics, COVID-19, deep neural network, convolutional neural network, visual transformer

1. Introduction

The COVID-19 epidemic is rapidly spreading around the world and has already harmed the health and well-being of the people from different countries. By 2022, there are hundreds of millions was infected and millions was dead. Effective diagnosis of COVID-19 is essential to reduce the growth of the disease and promptly respond to those who become ill.

Currently, the COVID-19 diagnosis uses the following methods:

1. The RT-PCR Test [1–6]. The advantages are the ability to extract a DNA fragment that is unique for a particular pathogen (several pathogens can be searched in one sample); sensitivity to the pathogen (a small number of samples is sufficient); efficiency of issuing the result (artificial cultivation of the pathogen is not required); direct determination of the pathogen, and not the reaction to it (diagnosis is possible in the incubation period); universality (it is possible to search for any pathogen). The disadvantages are pain, laboriousness due to the lack of automation, the requirement of the work rules strictest observance in the laboratory, high cost and insufficient accuracy.
2. The computerized tomography (CT) scan [7-12] is used to obtain an image. The advantages are the absence of pain, automation, the absence of the requirement of the work rules strictest observance in the laboratory and high accuracy. The disadvantage is the high cost.

IDDM-2022: 5th International Conference on Informatics & Data-Driven Medicine, November 18–20, 2022, Lyon, France
EMAIL: y.fedorov@chdtu.edu.ua (E. Fedorov); t.utkina@chdtu.edu.ua (T. Utkina); mari.leshchenko@gmail.com (M. Leshchenko); olne@ukr.net (O. Nechyporenko); k.rudakov@chdtu.edu.ua (K. Rudakov)
ORCID: 0000-0003-3841-7373 (E. Fedorov); 0000-0002-6614-4133 (T. Utkina); 0000-0002-0210-9582 (M. Leshchenko); 0000-0002-3954-3796 (O. Nechyporenko); 0000-0003-0000-6077 (K. Rudakov)

3. Radiography [13-17] is used to obtain a CXR image. The advantages are the absence of pain, high speed, the absence of the requirement of the work rules strictest observance in the laboratory, and low cost. The disadvantage is insufficient accuracy.

An ensemble of classifiers can be used to recognize CCT and CXR images (for example, decision tree, artificial neural network, naive Bayes, k-nearest neighbors, etc.) [18].

Currently, classifying deep neural networks have become widespread for intelligent image identification [19].

The following 2D convolutional networks is the first class of such networks.

The LeNet-5 neural network [20] has the simplest architecture and uses two pairs of convolutional and downsampling layers, as well as two fully connected layers. The convolution layer reduces the shear sensitivity of image elements. The downsampling layer reduces the dimensionality of the image. A combination of LeNet-5 (for feature extraction) and long short-term memory (LSTM) (for classification) is currently popular [21].

Dark Net neural networks [22], AlexNet neural networks [23], and VGG neural networks (Visual Geometry Group) [24, 25] are a modification of LeNet. There can be several consecutive convolutional layers in these neural networks.

ResNet neural networks [24, 25, 26] use a Residual block that contains two consecutive convolutional layers. The planes' outputs of the layer preceding of this block are added to the planes' outputs of the second convolutional layer of this block. A combination of ResNet (for feature extraction) and a support vector machine (SVM) (for classification) is currently popular [27].

The DenseNet neural network (Dense Convolutional Network) [25, 28] uses a fully connected (dense) block, which contains a set of Residual blocks. The planes' outputs of the second convolutional layer of the current Residual block of this dense block are concatenated with the planes' outputs of the second convolutional layer of all previous Residual blocks of this dense block and with the planes' outputs of the layer preceding this dense block. In addition, the reduction of the convolutional layers' planes (usually twice) located between dense blocks is used.

The GoogLeNet neural network (Inception V1) [29] uses an Inception block that contains parallel convolutional layers with different sizes of connection regions and one downsampling layer. The planes' outputs of these parallel layers are concatenated. The convolutional layers with a single connection area are connected in series to these parallel layers to reduce the number of operations (in the case of convolutional layers, such a convolutional layer is placed before them, and in the case of a downsampling layer, such a convolutional layer is placed after it). A combination of ResNet (for feature extraction) and support vector machine (SVM) (for classification) [27] used for diagnostics on CXR images is currently popular, which provided a diagnostic probability close to 100%.

The Inception V3 neural network [25, 26, 30] is a modification of GoogLeNet, and its Inception and Reduction blocks are a modification of the GoogLeNet neural network's Inception block.

The Inception-ResNet-v2 neural network [25, 26, 31] is a modification of GoogLeNet and ResNet, its Inception block is a modification of the Residual and Inception block, and the Reduction block is a modification of the Inception block.

The Xception neural network [25, 32] uses a Depthwise separable convolution block that performs first a pointwise convolution and then a depthwise convolution. For both convolutions, the ReLU activation function is usually used.

The MobileNet neural network [33] uses a Depthwise separable convolution block that performs depthwise convolution first and then pointwise convolution. For both convolutions, a linear activation function is usually used.

The MobileNet2 neural network [25, 34] uses an Inverse Residual block that performs pointwise convolution first, then depthwise convolution, and then pointwise again. For both convolutions, the SiLU activation function is usually used.

The MobileNet3 neural network [35] uses a Squeeze-and-Excitation block in some Inverse Residual blocks.

The following transformers is the second class of such networks.

ViT (Visual Transformer) [36] contains an encoder as the main component, consisting of a sequence of blocks. Each block contains the first normalization layer, Multi-Head Attention (weights the image patches), the second normalization layer, a two-layer perceptron.

DeiT (Data-efficient image Transformers) [37] contains an encoder as the main component, consisting of a sequence of blocks. Each block contains the first layer of normalization, Multi-Head Attention, the second layer of normalization, a two-layer perceptron, as in the case of ViT. DeiT, unlike ViT, additionally uses a distillation token in addition to patches.

DeepViT (Deep Visual Transformer) [38] contains an encryptor as the main component, consisting of a sequence of blocks. Each block contains the first normalization layer, Re-Attention (Multi-Head Attention modification), the second normalization layer, a two-layer perceptron

CaiT (Class-Attention in Image Transformers) [39] contains an encoder as the main component, consisting of a sequence of blocks. Each block contains the first normalization layer, Multi-Head Attention or Class-Attention (a modification of Multi-Head Attention that takes into account not only the patch, but also the class), the second normalization layer, a two-layer perceptron.

CrossViT (Cross-Attention Multi-Scale Vision Transformer) [40] contains a multi-scale encoder consisting of a blocks' sequence as the main component. Each block contains two encoders (each encoder is similar to a ViT encoder) for large and small size patches, and a Cross-Attention (a modification of Multi-Head Attention) that allows to share patches of different sizes.

Hybrids of convolutional neural networks and transformers are actively currently used.

The Compact Convolutional Transformer (CCT) [41] contains an encoder, as its main component, consisting of a sequence of blocks. Each block contains the first layer of normalization, Multi-Head Attention, the second layer of normalization, a two-layer perceptron, as in the case of ViT. Patch extraction is used with a sequence of pairs of convolutional and downsampling layers, instead of splitting the image into patches, and there is a pooling sequence layer after the encoder.

The Pooling-based Vision Transformer (PiT) [42] contains an encoder as the main component, consisting of a sequence of blocks. Each block contains the first layer of normalization, Multi-Head Attention, the second layer of normalization, a two-layer perceptron, as in the case of ViT. PiT, unlike ViT, uses depthwise convolution (each plane of the current layer is connected only to the corresponding plane of the next layer) and a downsampling layer before and after the encoder.

LeViT [43] contains a cipher consisting of a blocks' sequence as the main component. Each block contains LeViT Attention (three convolutions with normalization are used) without downsampling and a two-layer perceptron. Between these blocks there are LeViT Attention (using three convolutions with normalization) with downsampling. Patch extraction is used using a sequence of pairs of convolutional and downsampling layers, instead of patching the image. LeViT, unlike ViT, additionally uses a distillation token in addition to patches.

The Convolutional vision Transformer (CvT) [44] contains an encoder as the main component, consisting of a sequence of blocks. Each block contains the first layer of normalization, Multi-Head Attention, the second layer of normalization, a two-layer perceptron, as in the case of ViT. CvT, unlike ViT, uses patch extraction with a pairs' sequence of convolutional and downsampling layers, instead of patching the image.

MobileViT [45] contains Inverse Residual blocks from the MobileViT2 neural network and encoders, as the main components. Each encoder consists of a sequence of blocks. Each block contains the first layer of normalization, Multi-Head Attention, the second layer of normalization, a two-layer perceptron, as in the case of ViT. There are two convolutional layers before and after the encoder.

Deep neural networks have one or more of the following disadvantages:

- classification accuracy is insufficiently high;
- complexity of neural network structure identification (number and size of layers, number of transformer blocks, patch size, etc.);
- speed of parameters identification is insufficiently high.

Parallel algorithms are used to increase the learning rate of deep neural networks [46-47]. The problem of creating an effective deep neural network is relevant.

The goal of the research is to increase the intelligent diagnostics' efficiency of COVID-19 through the use of deep neural networks.

To achieve this goal, the following tasks were set and solved:

1. To create a COVID-19 smart diagnostic model based on a convolutional neural network and a visual transformer.
2. To select the quality criteria for the COVID-19 smart diagnostic method.

3. To determine the structure of the COVID-19 smart diagnostic method.
4. To conduct numerical research of the proposed method for intelligent diagnosis of COVID-19.

2. LeNet-ViT deep neural network

The LeNet-ViT deep neural network for CXR image classification was proposed by the research's authors, and it is a non-recurrent network (fig. 1). LeNet-ViT includes a sequence of alternating convolutional and downsampling layers and a reshape layer, an encoder (consists of blocks; the *k*-th block structure is shown in fig. 2), a flattening block, a multilayer perceptron (MLP).

The encoder, unlike traditional LeNet, was added after the convolutional and downsampling layers, which improves the efficiency of classification.

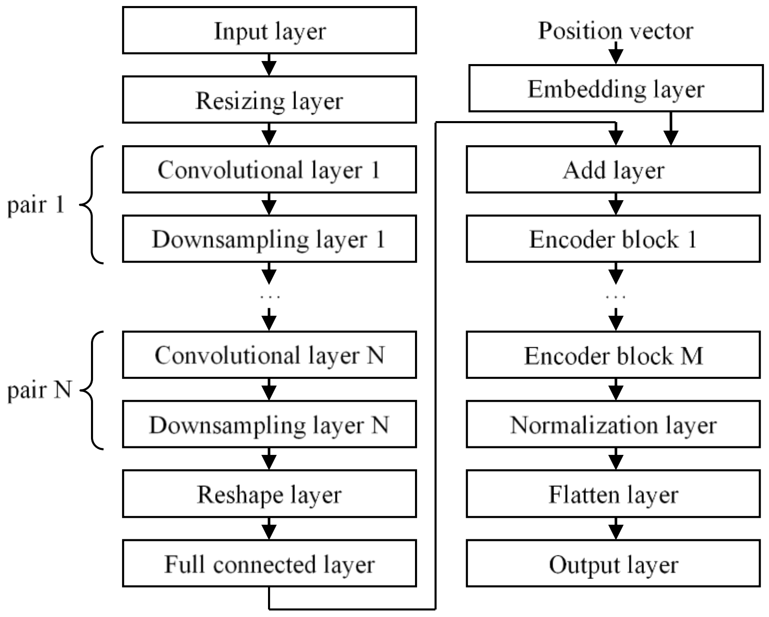

Figure 1: LeNet-ViT for classification

The input CvT image, unlike the traditional one, is not square; the input image is pre-compressed and the new size depends on the original image size and is empirically determined; the number of pairs "convolutional layer - downsampling layer" depends on the size of the image and is empirically determined; the number of planes is automatically determined, as the quotient of dividing the cells' number in the input layer by two to the power (the power is equal to twice number of the pair "convolutional layer - downsampling layer"), which will save the total number of cells in the layer after downsampling; downsampling halves the planes' layer size in height and width by half; the patch size depends on the image size and is empirically determined; the number of encoder blocks is empirically determined; the class number is not added to patches.

3. The ANN functioning.

1. Patch Shaping via convolution and downsampling.

Let ν – position in the connection areas, $\nu = (\nu_x, \nu_y)$, K_I – is the number of cell planes in the I input layer (for RGB images 3), K_{s_l} – is the number of cell planes in the downsampling layer S_l, K_{c_l} – is the number of cell planes in the convolutional layer C_l, A_l – is the connection area of the layer plane S_l, $\widehat{L}$ – is the number of convolutional (or downsampling) layers.

1.1. $l = 1$.

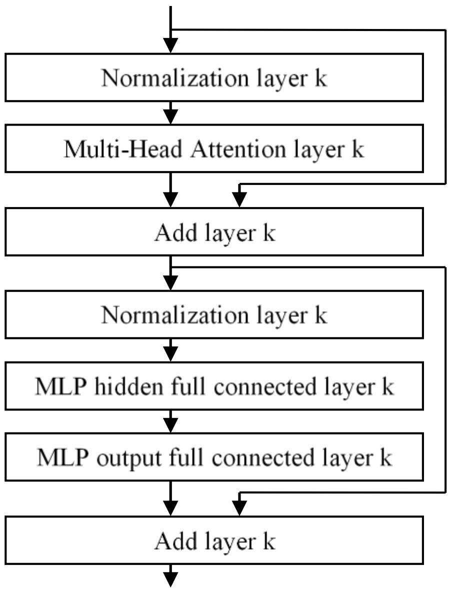
Normalization layer k
Multi-Head Attention layer k
Add layer k
Normalization layer k
MLP hidden full connected layer k
MLP output full connected layer k
Add layer k

Figure 2: Encoder *k* block

1.2. The computation of the output signal for the convolutional layer:

$$u_{c_l}(m,i) = \mathrm{ReLU}(h_{c_l}(m,i)),\ m \in \{1,...,N1_{c_l}\}\{1,...,N2_{c_l}\},\ i \in \overline{1,K_{c_l}},$$

$$K_{c_l} = 2^{2l},\ N1_{c_l} = \begin{cases} N1_I, & l=1 \\ N1_{s_{l-1}}, & l>1 \end{cases},\ N2_{c_l} = \begin{cases} N2_I, & l=1, \\ N2_{s_{l-1}}, & l>1, \end{cases}$$

$$h_{c_l}(m,i) = \begin{cases} b_{c_1}(i) + \sum\limits_{k=1}^{K_I}\sum\limits_{v\in A_l} w_{c_1}(v,k,i)x(m+v,k), & l=1, \\ b_{c_l}(i) + \sum\limits_{k=1}^{K_{s_{l-1}}}\sum\limits_{v\in A_{l-1}} w_{c_l}(v,k,i)u_{s_{l-1}}(m+v,k), & l>1, \end{cases}$$

where $w_{c_1}(v,k,i)$ – is the connection weight from the v-th position in the connection area of the *k*-th input layer's plane of cells I to the *i*-th convolutional layer's plane of cells C_1, $w_{c_l}(v,k,i)$ – is the connection weight of the v-th position in the connection area of the downsampling layer S_{l-1} to the *i*-th convolutional layer's plane of cells C_1, $u_{c_l}(m,i)$ – is the output of the cell in the m-th position in the *i*-th convolutional layer's plane of cells C_1.

1.3. The computation of the output signal of a downsampling cell (scale down by half based on averaging)

$$u_{s_l}(m,k) = \max_{\upsilon\in\{0,1\}^2} u_{c_l}(2m+\upsilon,k),\ m\in\{1,...,N1_{s_l}\}\times\{1,...,N2_{s_l}\},\ k\in\overline{1,K_{s_l}},\ K_{s_l} = 2^{2l},$$

where $w_{s_l}(k,k)$ – is the connection weight of the *k*-th convolutional layer's plane of cells C_l to *k*-th downsampling layer's plane of cells S_l, $u_{s_l}(m,k)$ – is the output of the cell in the m-th position in the *k*-th downsampling layer's plane of cells S_l.

1.4. If $l \le \hat{L}$, then $l = l+1$, go to 1.2.

1.5. The matrix $[y^{(0)}_{ni}]$, $i\in\overline{1,N^{(1)}}$, $n\in\overline{1,P}$ formation, corresponding to a sequence of flat patches (the patch matrix is turned into a vector) of P length.

$$\mathbf{y}^{(1)}_n = (u_{s_{\hat L}}(1,n),...,u_{s_{\hat L}}(N^{(1)},n)),\ n\in\overline{1,P},\ N^{(1)} = N1_{s_{\hat L}}N2_{s_{\hat L}},\ P = K_{s_{\hat L}}.$$

2. The transformation of patch vectors in a fully connected layer.

$$y_{nj}^{(1)} = \mathrm{ReLU}\left(b_{nj}^{(1)} + \sum_{i=1}^{N^{(0)}N^{(0)}} w_{nij}^{(1)} y_{ni}^{(0)} \right),\ j \in \overline{1, N^{(1)}},\ n \in \overline{1, P},$$

where $w_{nij}^{(1)}$ –is the connection weight from the *i*-th element of the image patch to the *j*-th neuron of the fully connected layer, $y_{nj}^{(1)}$ – is the output of the *j*-th neuron of the fully connected layer for the *n*-th patch.

3. The positional encryption (patch position added)

$$y_{ni}^{(2)} = y_{ni}^{(1)} + pos(n,i),\ \ pos(n,i) = \begin{cases} \cos\left(\dfrac{n}{10000^{2i/N^{(1)}}} \right) & i \bmod 2 = 0, \\ \sin\left(\dfrac{n}{10000^{2i/N^{(1)}}} \right) & i \bmod 2 \neq 0, \end{cases}$$

$$i \in \overline{1, N^{(1)}},\ n \in \overline{1, P}.$$

4. The encoder for each block does the following:

4.1. Normalization

$$y_{ni}^{(3)} = \frac{g}{\sigma}\left(y_{ni}^{(2)} - \mu \right),\ i \in \overline{1, N^{(1)}},\ n \in \overline{1, P},\ \mu = \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(2)}),\ \sigma = \sqrt{\frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(2)} - \mu)^2},$$

where *g* – is the amplification parameter, $y_{ni}^{(3)}$ – is the output of the *i*-th normalization layer neuron for the *n*-th patch.

4.2. Multihead attention

There are $N^{(H)}$ attention heads are using.

4.2.1. The computation of the requests for each attention head.

$$q_{\ln j} = \sum_{i=1}^{N^{(1)}} w_{lij}^{(Q)} y_{ni}^{(3)},\ l \in \overline{1, N^{(H)}},\ i \in \overline{1, N^{(K)}},\ n \in \overline{1, P},$$

where $w_{lij}^{(Q)}$ – is the connection weight from the *i*-th image patch element to the *j*-th request for the *l*-th attention head, $q_{\ln j}$ – is the *j*-th request for the *l*-th attention head for the *n*-th patch.

4.2.2. The computation of the keys for each attention head.

$$k_{\ln j} = \sum_{i=1}^{N^{(1)}} w_{lij}^{(K)} y_{ni}^{(3)},\ l \in \overline{1, N^{(H)}},\ j \in \overline{1, N^{(K)}},\ n \in \overline{1, P},$$

where $w_{lij}^{(K)}$ – is the connection weight from the *i*-th image patch element to the *j*-th key for the *l*-th attention head, $k_{\ln j}$ – is the *j*-th key for the *l*-th attention head for the *n*-th patch.

4.2.3. The computation of the key values for each attention head.

$$v_{\ln j} = \sum_{i=1}^{N^{(1)}} w_{lij}^{(V)} y_{ni}^{(3)},\ l \in \overline{1, N^{(H)}},\ j \in \overline{1, N^{(V)}},\ n \in \overline{1, P},$$

where $w_{lij}^{(V)}$ – is the connection weight from the *i*-th image patch element to the *j*-th key's value for the *l*-th attention head, $v_{\ln j}$ – is the *j*-th key for the *l*-th attention head for the *n*-th patch.

Usually, $N^{(V)} = N^{(K)}$

4.2.4. The computation of the attention weights (scores) for each attention head.

Scaled multiplicative attention "dot" is used. The request and the key are matched.

$$e_{lmn} = \frac{1}{N^{(K)}} \sum_{i=1}^{N^{(K)}} q_{\ln i} k_{lmi},\ l \in \overline{1, N^{(H)}},\ m \in \overline{1, P},\ n \in \overline{1, P},$$

$$a_{lmn} = \mathrm{softmax}(e_{lmn}) = \frac{\exp(e_{lmn})}{\sum_{z=1}^{P} \exp(e_{lmz})}, \; n \in \overline{1,P},$$

where a_{lmn} – is the connection weight between the *m*-th and *n*-th patches for the *l*-th attention head.

4.2.5. The computation of the requests for each attention head.

The multiplication of the attention weight (score) by the key value.

$$h_{\ln j} = \sum_{m=1}^{P} a_{lmn} v_{lmj}, \; l \in \overline{1,N^{(H)}}, \; j \in \overline{1,N^{(V)}}, \; n \in \overline{1,P},$$

where $h_{\ln j}$ – is the weighted *j*-th key value for the *j*-th key for the *l*-th attention head for the *n*-th patch.

4.2.6. The formation of the weighted key values matrix for attention heads and patches by concatenation

$$\mathbf{H} = \begin{bmatrix} h_{111} & \dots & h_{11N^{(V)}} & \dots & h_{N^{(H)}11} & \dots & h_{N^{(H)}1N^{(V)}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_{1P1} & \dots & h_{1PN^{(V)}} & \dots & h_{N^{(H)}P1} & \dots & h_{N^{(H)}PN^{(V)}} \end{bmatrix}$$

4.2.7. The computation of the multihead attention.

$$y_{nj}^{(4)} = \sum_{i=1}^{N^{(H)}N^{(V)}} h_{ni} w_{ij}^{(4)}, \; j \in \overline{1,N^{(1)}}, \; n \in \overline{1,P},$$

where $w_{ij}^{(4)}$ – is the connection weight from the *j*-th key value for the *j*-th neuron in the multihead attention layer, $y_{nj}^{(4)}$ – is the output of the *j*-th neuron in the multihead attention layer for *n*-th patch.

4.3. Summation and normalization.

$$y_{ni}^{(5)} = \frac{g}{\sigma}\left(y_{ni}^{(2)} + y_{ni}^{(4)} - \mu\right), \; i \in \overline{1,N^{(1)}}, \; n \in \overline{1,P},$$

$$\mu = \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(2)} + y_{ni}^{(4)}), \; \sigma = \sqrt{\frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(2)} + y_{ni}^{(4)} - \mu)^2},$$

where *g* – is the amplification parameter, $y_{ni}^{(5)}$ – is the output of the *i*-th normalization layer‘s neuron for the *n*-th patch.

4.4. The computation of the output signal for the two-layer perceptron’s hidden and output layers from *P* neural networks for *P* patches.

$$y_{nj}^{(6)} = \mathrm{ReLU}\left(b_{nj}^{(6)} + \sum_{i=1}^{N^{(1)}} w_{nij}^{(6)} y_{ni}^{(5)}\right), \; j \in \overline{1,N^{(6)}}, \; n \in \overline{1,P}, \; j \in \overline{1,N^{(1)}}, \; n \in \overline{1,P},$$

where $w_{zj}^{(k)}$ – is the connection weight of the *z*-th *k-1*-th layer’s neuron to *j*-th neuron of the *k*-th layer, $y_{nj}^{(k)}$ – is the output of the fully connected layer’s *j*-th neuron for *n*-th patch.

5. Summation and normalization.

$$y_{ni}^{(8)} = Norm(y_{ni}^{(4)} + y_{ni}^{(7)}) = \frac{g}{\sigma}\left(y_{i}^{(4)} + y_{ni}^{(7)} - \mu\right), \; i \in \overline{1,N^{(1)}}, \; n \in \overline{1,P},$$

$$\mu = \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(4)} + y_{ni}^{(7)}), \; \sigma = \sqrt{\frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} (y_{ni}^{(4)} + y_{ni}^{(7)} - \mu)^2},$$

where *g* – is the amplification parameter, $y_{ni}^{(8)}$ – is the output of the *i*-th normalization layer‘s neuron for *n*-th patch.

6. Flattening.

$$\mathbf{y}^{(9)} = (y_{11}^{(8)}, \dots, y_{N^{(1)}P}^{(8)})$$

7. The computation of the output signal for the hidden and output layer of a two-layer perceptron.

$$y_j^{(10)} = \mathrm{ReLU}\left(b_j^{(10)} + \sum_{z=1}^{N^{(1)}P} w_{zj}^{(10)} y_z^{(9)} \right),\ j \in \overline{1, N^{(10)}},$$

$$y_j = \mathrm{softmax}\left(b_j^{(11)} + \sum_{z=1}^{N^{(10)}} w_{zj}^{(11)} y_z^{(10)} \right),\ j \in \overline{1, N^{(11)}},$$

where $w_{zj}^{(k)}$ – is the connection weight of the *z*-th *k-1*-th layer's neuron to *j*-th neuron of the *k*-th layer, $y_j^{(k)}$ – is the output of the *j*-th neuron of the *k*-th fully connected layer.

4. The selection of quality criteria for the COVID-19 smart diagnostic method.

The following criteria were chosen to evaluate the learning of the proposed deep neural networks mathematical models in this research:

- accuracy criterion

$$Accuracy = \frac{1}{I}\sum_{i=1}^{I}[\mathbf{d}_i = \hat{\mathbf{y}}_i] \to \max_W,\quad \hat{y}_{ij} = \begin{cases} 1, & j = \arg\max_z y_{iz}, \\ 0, & j \neq \arg\max_z y_{iz}. \end{cases}$$

- categorical cross-entropy criterion

$$CCE = -\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{K} d_{ij} \ln y_{ij} \to \min_W,$$

where $\mathbf{y}_i$ – is the i-th vector by model, $y_{ij} \in [0,1]$, $\mathbf{d}_i$ – is the i-th test vector, $d_{ij} \in \{0,1\}$, I – is the training set power, K – is the classes number (neurons in the output layer), W – is the weight vector.

5. The structure's determination of the COVID-19 smart diagnostic method

The block diagram of the COVID-19 intelligent diagnostic method for the proposed LeNet-ViT deep neural network is shown in fig. 3.

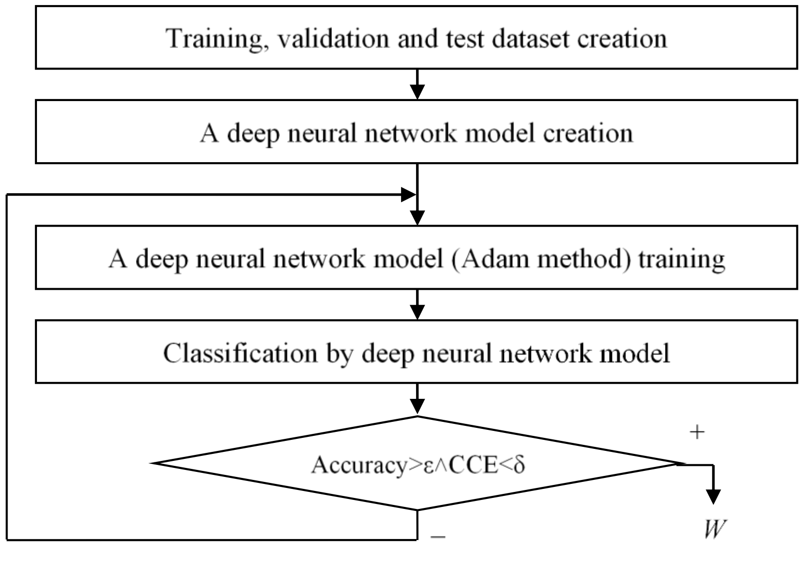
Training, validation and test dataset creation
A deep neural network model creation
A deep neural network model (Adam method) training
Classification by deep neural network model
Accuracy>ε∧CCE<δ
+
–
W

Figure 3: Block diagram of the COVID-19 smart diagnostic method

6. Numerical research

The numerical research was carried out on the basis of data sets [48–49]. The size of the COVIDx CXR-3 dataset was 30386 records. No pre-processing of the dataset was performed.The training data set included 13992 COVID-19 Negative and 15994 COVID-19 Positive. The test data set included 200 COVID-19 Negative and 200 COVID-19 Positive.

The training was performed using the GPU due to the fact that the proposed deep neural networks don't contain recurrent connections. The Tensorflow package was used to implement the proposed deep neural networks, and Google Collaboratory was chosen as the software environment.

The LeNet-ViT model's structure with two encoder blocks is shown in Table 1, where *K* is the number of classes.

Table 1
The LeNet-ViT model structure

Layer type	Input size
A	Text
Input	1024x1024x3
Resizing	32x32x3
Conv2D	32x32x4
MaxPooling2D	16x16x4
Conv2D	16x16x16
MaxPooling2D	8x8x16
Reshape	16x64
Full connect or Dense	16x128
Add	16x128
Normalization	16x128
Multi-Head Attention	16x128
Add	16x128
Normalization	16x128
MLP Hidden (Full connect or Dense)	16x128
MLP Output (Full connect or Dense)	16x128
Add	16x128
Normalization	16x128
Multi-Head Attention	16x128
Add	16x128
Normalization	16x128
MLP Hidden (Full connect or Dense)	16x128
MLP Output (Full connect or Dense)	16x128
Add	16x128
Normalization	16x128
Flatten	2048
MLP Hidden (Full connect or Dense)	2048
MLP Output (Full connect or Dense)	K

The losses dependence (based on categorical entropy) on the image size for the LeNet-ViT model is shown in fig. 4. The losses dependence (based on categorical entropy) on the patch size for the LeNet-ViT model is shown in fig. 5. The losses dependence (based on categorical entropy) on the number of iterations for the LeNet-ViT model is shown in fig. 6. The dependence of accuracy on the number of iterations for the LeNet-ViT model is shown in fig. 7. The losses dependence (based on categorical entropy) on the number of pairs "convolutional layer - downsampling layer" for the LeNet-ViT model is shown in fig. 8.

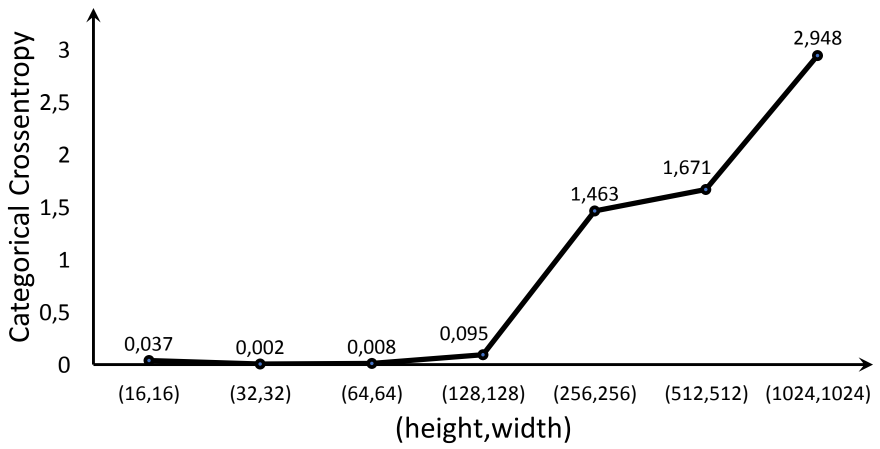

Figure 4: The losses dependence (based on categorical entropy) on the image size for the LeNet-ViT model

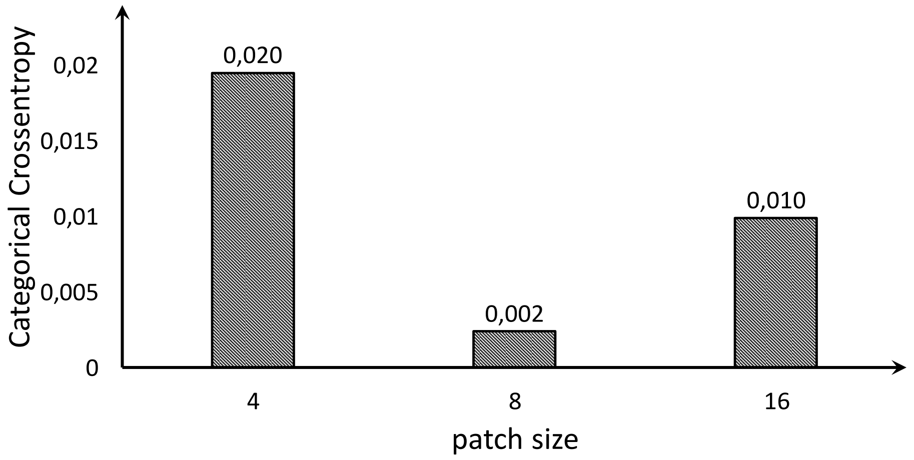

Figure 5: The losses dependence (based on categorical entropy) on the patch size for the LeNet-ViT model

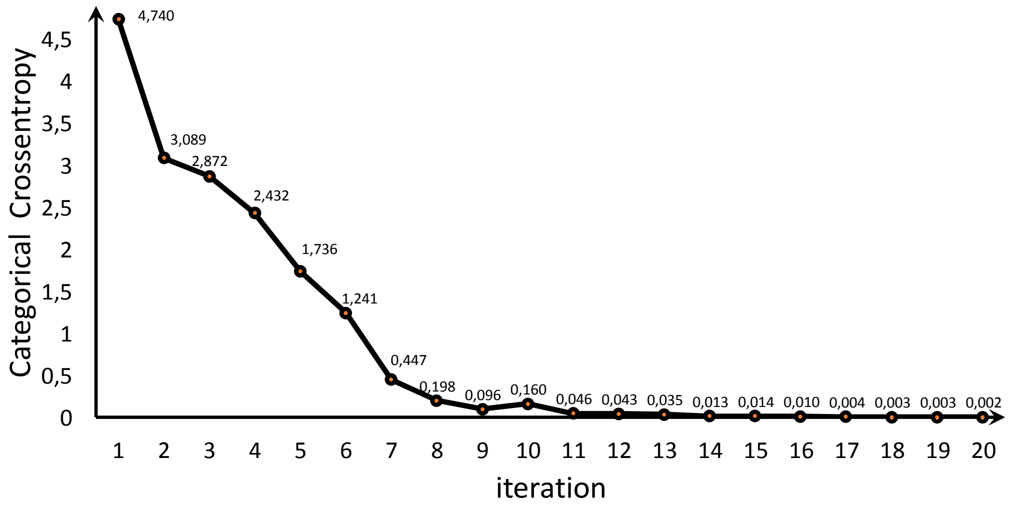

Figure 6: The losses dependence (based on categorical entropy) on the number of iterations for the LeNet-ViT model

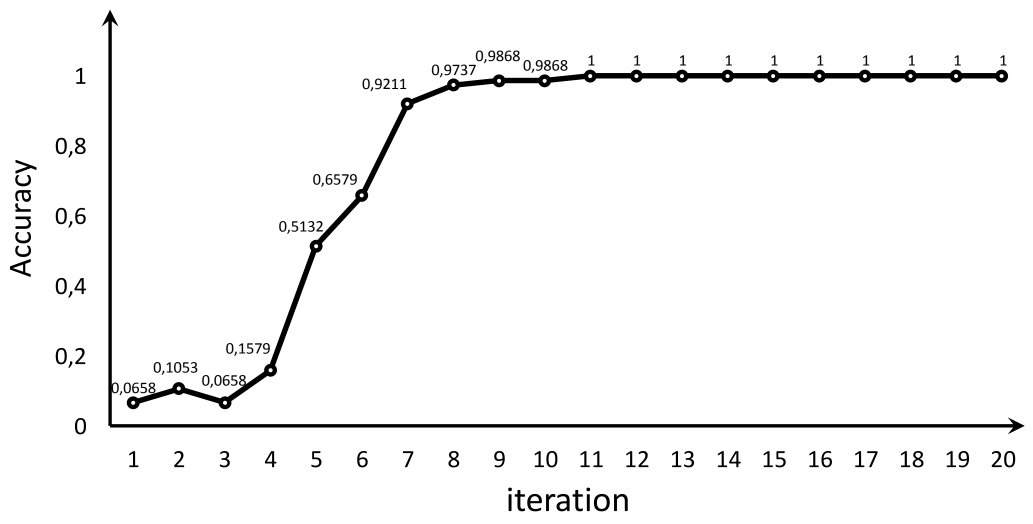

Figure 7: The accuracy dependence (based on categorical entropy) on the number of iterations for the LeNet-ViT model

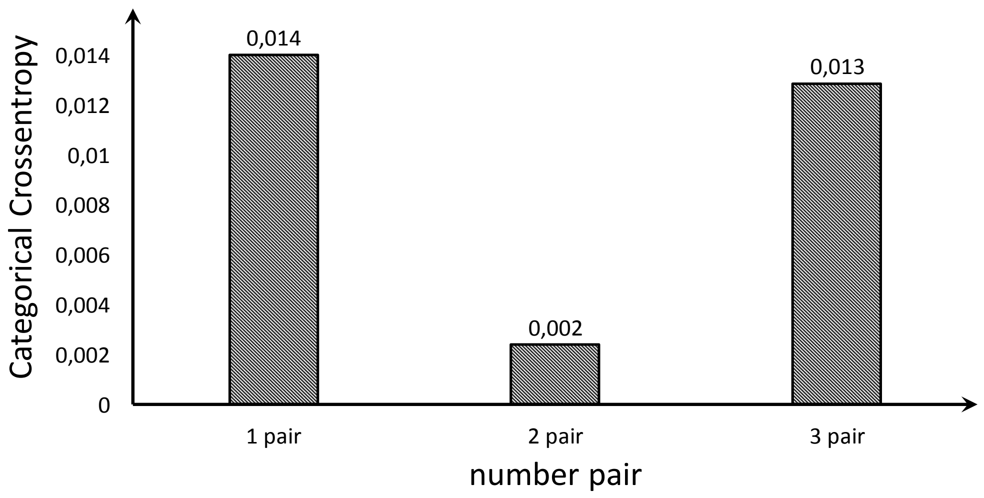

Figure 8: The losses dependence (based on categorical entropy) on the number of pairs "convolutional layer - downsampling layer" for the LeNet-ViT model

The following facts were found as a result of a numerical study:

- the best image size after compression for LeNet-ViT in terms of loss (based on categorical entropy) is 32x32 (fig. 4);
- the best compression patch size for LeNet-ViT in terms of loss (based on categorical entropy) is 8x8 (fig. 5);
- the minimum number of iterations for LeNet-ViT in terms of loss (based on categorical entropy) (fig. 6) and accuracy (fig. 7) is 11;
- the best number of convolutional layer-downsampling layer pairs for LeNet-ViT in terms of loss (based on categorical entropy) is 2 (fig. 8).

The KFold cross-entropy method with 5 folds was used to prevent overfitting.

7. Conclusions

1. The relevant artificial intelligence methods have been researched to solve the problem of improving the efficiency of intelligent diagnosis of COVID-19. The using of deep neural networks is the most effective according to the research's results.

2. The created LeNet-ViT model, unlike traditional CvT, has the following advantages: the input image is not square, which expands the scope; the input image is pre-compressed and the new size depends on the original image size and is empirically determined; this increases the model training speed and the model identification accuracy; the number of pairs "convolutional layer - downsampling layer" depends on the image size and is empirically determined; this increases the model identification accuracy, and the number of planes is automatically determined, as the quotient of dividing the cells' number in the input layer by two to the power (the power is equal to twice number of the pair "convolutional layer - downsampling layer"), which will save the total number of cells in the layer after downsampling; it halves the size of the layer planes in height and width, which automates the structure definition of the model layers; the patch size depends on the image size and is empirically determined, which increases the model identification accuracy; the number of encoder blocks is empirically determined, which increases the model learning speed.
3. Using of the proposed method of intelligent COVID-19 diagnostics for various intelligent medical diagnostic systems is a further research prospect.
The KFold cross-entropy method with 5 folds was used to prevent overfitting.

8. References

[1] Sheridan, Cormac. "Coronavirus and the Race to Distribute Reliable Diagnostics." Nature Biotechnology, vol. 38, no. 4, 2020, pp. 382–384., doi:10.1038/d41587-020-00002-2.
[2] Lippi, Giuseppe, et al. "Potential Preanalytical and Analytical Vulnerabilities in the Laboratory Diagnosis of Coronavirus Disease 2019 (Covid-19)." Clinical Chemistry and Laboratory Medicine (CCLM), vol. 58, no. 7, 2020, pp. 1070–1076., doi:10.1515/cclm-2020-0285.
[3] Lippi, Giuseppe, and Mario Plebani. "A Six-Sigma Approach for Comparing Diagnostic Errors in Healthcare—Where Does Laboratory Medicine Stand?" Annals of Translational Medicine, vol. 6, no. 10, 2018, pp. 180–180., doi:10.21037/atm.2018.04.02.
[4] Oliveira, Beatriz Araujo, et al. "SARS-COV-2 and the COVID-19 Disease: A Mini Review on Diagnostic Methods." Revista Do Instituto De Medicina Tropical De São Paulo, vol. 62, 2020, doi:10.1590/s1678-9946202062044.
[5] Kasotakis, George. "Faculty Opinions Recommendation of Correlation of Chest CT and RT-PCR Testing in Coronavirus Disease 2019 (Covid-19) in China: A Report of 1014 Cases." Faculty Opinions – Post-Publication Peer Review of the Biomedical Literature, 2020, doi:10.3410/f.737441336.793572936.
[6] Wolach, Ofir, and Richard M. Stone. "Mixed-Phenotype Acute Leukemia." Current Opinion in Hematology, vol. 24, no. 2, 2017, pp. 139–145., doi:10.1097/moh.0000000000000322.
[7] Gozes, Ophir, et al. "Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection &amp; Patient Monitoring Using Deep Learning CT Image Analysis." ArXiv, 2020, doi:10.48550/arXiv.2003.05037.
[8] Wang, Shuai, et al. "A Deep Learning Algorithm Using CT Images to Screen for Corona Virus Disease (COVID-19)." 2020, doi:10.1101/2020.02.14.20023028.
[9] Ng, Ming-Yen, et al. "Imaging Profile of the COVID-19 Infection: Radiologic Findings and Literature Review." Radiology: Cardiothoracic Imaging, vol. 2, no. 1, 2020, doi:10.1148/ryct.2020200034.
[10] Wang, Shuo, et al. "A Fully Automatic Deep Learning System for COVID-19 Diagnostic and Prognostic Analysis." European Respiratory Journal, vol. 56, no. 2, 2020, p. 2000775., doi:10.1183/13993003.00775-2020.
[11] Salehi, Sana, et al. "Coronavirus Disease 2019 (COVID-19): A Systematic Review of Imaging Findings in 919 Patients." American Journal of Roentgenology, vol. 215, no. 1, 2020, pp. 87–93., doi:10.2214/ajr.20.23034.
[12] Li, Jingwen, et al. "Radiology Indispensable for Tracking COVID-19." Diagnostic and Interventional Imaging, vol. 102, no. 2, 2021, pp. 69–75., doi:10.1016/j.diii.2020.11.008.
[13] Rubin, Geoffrey D., et al. "The Role of Chest Imaging in Patient Management during the COVID-19 Pandemic: A Multinational Consensus Statement from the Fleischner Society." Radiology, vol. 296, no. 1, 2020, pp. 172–180., doi:10.1148/radiol.2020201365.

[14] Baratella, Elisa, et al. “Severity of Lung Involvement on Chest x-Rays in SARS-Coronavirus-2 Infected Patients as a Possible Tool to Predict Clinical Progression: An Observational Retrospective Analysis of the Relationship between Radiological, Clinical, and Laboratory Data.” Jornal Brasileiro De Pneumologia, vol. 46, no. 5, 2020, doi:10.36416/1806-3756/e20200226.

[15] Amis, E. Stephen, et al. “American College of Radiology White Paper on Radiation Dose in Medicine.” Journal of the American College of Radiology, vol. 4, no. 5, 2007, pp. 272–284., doi:10.1016/j.jacr.2007.03.002.

[16] Tahir, Anas M., et al. “A Systematic Approach to the Design and Characterization of a Smart Insole for Detecting Vertical Ground Reaction Force (Vgrf) in Gait Analysis.” Sensors, vol. 20, no. 4, 2020, p. 957., doi:10.3390/s20040957.

[17] Kallianos, K., et al. “How Far Have We Come? Artificial Intelligence for Chest Radiograph Interpretation.” Clinical Radiology, vol. 74, no. 5, 2019, pp. 338–345., doi:10.1016/j.crad.2018.12.015.

[18] Chandra, Tej Bahadur, et al. “Coronavirus Disease (COVID-19) Detection in Chest X-Ray Images Using Majority Voting Based Classifier Ensemble.” Expert Systems with Applications, vol. 165, 2021, p. 113909., doi:10.1016/j.eswa.2020.113909.

[19] Fedorov, Eugene, et al. “The Method of Intelligent Image Processing Based on a Three-Channel Purely Convolutional Neural Network.” CEUR Workshop Proceedings, vol. 2255, 2018, pp. 336–351., ceur-ws.org/Vol-2255/paper30.pdf. Accessed 25 Sept. 2022.

[20] Wan, Lanjun, et al. “Rolling-Element Bearing Fault Diagnosis Using Improved Lenet-5 Network.” Sensors, vol. 20, no. 6, 2020, p. 1693., doi:10.3390/s20061693.

[21] Islam, Md. Zabirul, et al. “A Combined Deep CNN-LSTM Network for the Detection of Novel Coronavirus (COVID-19) Using X-Ray Images.” Informatics in Medicine Unlocked, vol. 20, 2020, p. 100412., doi:10.1016/j.imu.2020.100412.

[22] Ozturk, Tulin, et al. “Automated Detection of COVID-19 Cases Using Deep Neural Networks with X-Ray Images.” Computers in Biology and Medicine, vol. 121, 2020, p. 103792., doi:10.1016/j.compbiomed.2020.103792.

[23] Krizhevsky, Alex, et al. “ImageNet Classification with Deep Convolutional Neural Networks.” Communications of the ACM, vol. 60, no. 6, 2017, pp. 84–90., doi:10.1145/3065386.

[24] He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.90.

[25] Hemdan, Ezz El-Din, et al. “COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images.” ArXiv, 2020, doi:10.48550/ARXIV.2003.11055.

[26] Narin, Ali, et al. “Automatic Detection of Coronavirus Disease (COVID-19) Using X-Ray Images and Deep Convolutional Neural Networks.” Pattern Analysis and Applications, vol. 24, no. 3, 2021, pp. 1207–1220., doi:10.1007/s10044-021-00984-y.

[27] “Detection of COVID-19 Chest X-Ray Using Support Vector Machine and Convolutional Neural Network.” Communications in Mathematical Biology and Neuroscience, 2020, doi:10.28919/cmbn/4765.

[28] Que, Yue, and Hyo Jong Lee. “Densely Connected Convolutional Networks for Multi-Exposure Fusion.” 2018 International Conference on Computational Science and Computational Intelligence (CSCI), 2018, doi:10.1109/csci46756.2018.00084.

[29] Szegedy, Christian, et al. “Going Deeper with Convolutions.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, doi:10.1109/cvpr.2015.7298594.

[30] Szegedy, Christian, et al. “Rethinking the Inception Architecture for Computer Vision.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.308.

[31] Szegedy, Christian, et al. “Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning.” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017, doi:10.1609/aaai.v31i1.11231.

[32] Chollet, Francois. “Xception: Deep Learning with Depthwise Separable Convolutions.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, doi:10.1109/cvpr.2017.195.

[33] Howard, Andrew G., et al. “Efficient Convolutional Neural Networks for Mobile Vision Applications.” ArXiv, 2017, doi:10.48550/arXiv.1704.04861.

[34] Sandler, Mark, et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, doi:10.1109/cvpr.2018.00474.

[35] Geng, Lei, et al. “Fertility Detection of Hatching Eggs Based on a Convolutional Neural Network.” Applied Sciences, vol. 9, no. 7, 2019, p. 1408., doi:10.3390/app9071408.

[36] Dosovitskiy, Alexey, et al. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.” ArXiv, 2021, doi:10.48550/arXiv.2010.11929.

[37] Touvron, Hugo, et al. “Training Data-Efficient Image Transformers &amp; Distillation through Attention .” ArXiv, 2021, doi:10.48550/arXiv.2012.12877.

[38] Zhou, Daquan, et al. “DeepViT: Towards Deeper Vision Transformer .” ArXiv, 2021, doi:10.48550/arXiv.2103.11886.

[39] Touvron, Hugo, et al. “Going Deeper with Image Transformers.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.00010.

[40] Chen, Chun-Fu Richard, et al. “Crossvit: Cross-Attention Multi-Scale Vision Transformer for Image Classification.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.00041.

[41] Hassani, Ali, et al. “Escaping the Big Data Paradigm with Compact Transformers.” ArXiv, 7 June 2022, arxiv.org/abs/2104.05704.

[42] Heo, Byeongho, et al. “Rethinking Spatial Dimensions of Vision Transformers.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.01172.

[43] Graham, Ben, et al. “Levit: A Vision Transformer in ConvNet’s Clothing for Faster Inference.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.01204.

[44] Wu, Haiping, et al. “CVT: Introducing Convolutions to Vision Transformers.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.00009.

[45] Mehta, Sachin, and Mohammad Rastegari. “Mobilevit: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer.” ArXiv, 4 Mar. 2022, arxiv.org/abs/2110.02178.

[46] Shvachych, G.G., et al. “Parallel Computational Algorithms in Thermal Processes in Metallurgy and Mining.” Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, no. 4, 2018, pp. 129–137., doi:10.29202/nvngu/2018-4/19.

[47] Shlomchak, G, et al. “Automated Control of Temperature Regimes of Alloyed Steel Products Based on Multiprocessors Computing Systems.” Metalurgija, vol. 58, no. 3-4, 2019, pp. 299–302., hrcak.srce.hr/218406. Accessed 25 Sept. 2022.

[48] Zhao, Andy. “Chest x-Ray Images for the Detection of COVID-19.” Kaggle, 2022, www.kaggle.com/datasets/andyczhao/covidx-cxr2.

[49] Gunraj, Hayden. “COVIDx CT: A Large-Scale Chest CT Dataset for COVID-19 Detection.” Kaggle, 2022, www.kaggle.com/datasets/c395fb339f210700ba392d81bf200f766418238c2734e5237b5dd0b6fc724fcb.