Fine-tuning the hyperparameters of pre-trained models for solving multiclass classification problems

D. Kaibassova [1], M. Nurtay [1], A. Tau [1], M. Kissina [1]
[1] Abylkas Saginov Karaganda technical university, 100000, Kazakhstan, Karaganda city, 56 N. Nazarbayev avenue

Abstract

This study is devoted to the application of fine-tuning methods for Transfer Learning models to solve the multiclass image classification problem using the medical X-ray images. To achieve this goal, the structural features of such pre-trained models as VGG-19, ResNet-50, InceptionV3 were studied. For these models, the following fine-tuning methods were used: unfreezing the last convolutional layer and updating its weights, selecting the learning rate and optimizer. As a dataset chest X-Ray images of the Society for Imaging Informatics in Medicine (SIIM), as the leading healthcare organization in its field, in partnership with the Foundation for the Promotion of Health and Biomedical Research of Valencia Region (FISABIO), the Valencian Region Medical ImageBank (BIMCV)) and the Radiological Society of North America (RSNA) were used. Thus, the results of the experiments carried out illustrated that the pre-trained models with their subsequent tuning are excellent for solving the problem of multiclass classification in the field of medical image processing. It should be noted that ResNet-50 based model showed the best result with 82.74 % accuracy. Results obtained for all models are reflected in the corresponding tables.

Keywords: multiclass classification, transfer learning, fine-tuning, CNN, image augmentation, X-ray.

Citation: Kaibassova D, Nurtay M, Tau A, Kissina M. Fine-tuning the hyperparameters of pre-trained models for solving multiclass classification problems. Computer Optics 2022; 46(6): 971-979. DOI: 10.18287/2412-6179-CO-1078.

Introduction

The rapid development of the Convolutional Neural Networks (CNN) has led to the fact that new architectures and approaches to their design were began to invent. Global Imagenet competition stimulated the creation of such networks as AlexNet, DenseNet, VGG, ResNet, etc. [1 – 3]. This made it possible to use the accumulated knowledge of these networks to solve other problems using the Trasfer Learning approach [4 – 5]. Each of these convolutional networks has its own way of achieving maximum accuracy. In this paper, a comparison of the efficiency of networks using the example of the problem of multiclass classification of medical images was proposed.

While exploring the subject area, the authors faced the problem of classifying several classes, which is different from the problem of binary classification. Lack of uniformity makes many traditional Machine Learning algorithms less efficient, especially when the original dataset does not explicitly allow for distinguishing features for each class. One approach used to solve the multiclass classification problem is to divide the dataset into several binary classification datasets and fit a binary classification model for each of them. Two different examples of this approach are One-vs-All and One-vs-One [6]. This study uses an technique based on the use of probabilistic data distribution. This is achieved by applying the Softmax activation function on the last layer of the Deep Convolutional Neural Network (DCNN) which outputs the probability values of belonging to each of the available classes.

Using pre-trained networks is a relatively simple method for applying Deep Learning to Image Analysis, and the comparison results from the present study can be used to select appropriate networks for diagnostic tasks.

In the first work, the authors of [7] conducted a comparative study using pre-trained models such as VGG-19 and ResNet-50. To reduce overfitting, regularization of data increase and dropout were used. Only a binary classification was carried out (pneumonia versus normal). As a result, the authors received 92.03 % accuracy. In another example, the researchers [8] investigated the detection of pneumonia by X-ray images using CNN VGG-16 and VGG-19. For comparison, they used a modified CNN-35 layer for two networks. The experiment was carried out using the open source Kaggle chest X-ray dataset. The data consisted of 2 classes: normal and pneumonia, 624 images in total. The results obtained using the VGG-16 revealed an accuracy of 94.1 %, and the VGG-19 had 95.7 % accuracy. For CNN-35, the accuracy reached 96.3 %.

In [9], the authors presented an Ensemble diagram of CNNs, inspired by decomposition and ensemble methods, to improve the performance of a computer diagnostics (CD) system. In the public ISIC 2018 dataset, this method provides the best balanced accuracy (76.6 %) among multiclass CNNs. However, the ensemble learning method is resource-intensive and requires a lot of effort to solve simple classification problems.

The study [10] considered only the detection of leukemia generally; the authors used a pre-trained serial network including InceptionV3. The authors of [11] suggested the possibility of detecting suspicious benign and malignant bone lesions from scintigraphic images of bones of the whole body with accuracy with InceptionV3 (80.61 %).

1. Overview of the architecture of the selected convolutional network models

The following Convolutional Neural Network architectures were taken as objects of research: VGG-19, ResNet-50, and InceptionV3. This choice is due to the fact that these networks have a fundamentally new approach to improving the accuracy of pattern recognition and are very well suited for comparison.

VGG is a widely used model. It was proposed by the Visual Geometry group of the University of Oxford Simonyan and Zisserman in their work Very Deep Convolutional Networks for Large Scale Image Recognition in 2014 and allowed to obtain an accurate classification of the ImageNet dataset [12]. The VGG-19 architecture consists of 19 deep layers and a 224×224 input layer. The key materials in the VGG-19 structure are the used kernels of 3×3 size and the convolution step is set to 1 pixel to cover the whole concept of the input image. The VGG-19 architecture consists of 19 deep layers and a 224×224 input layer. The key material in the VGG-19 structure is the convolutional 3×3 kernels with stride set to 1 pixel to cover the whole concept of the input image. The network uses a 3×3 stack of convolutional layers with increasing depth and max pooling 2×2 with stride 2. In the usual form, for classification according to the imagenet base, two fully connected layers of 4096 nodes each and at the output a layer of 1000 neurons with activation by Softmax. All hidden layers are trained with RELU to introduce nonlinearity [13]. In fig. 1 the structure of the VGG-19 architecture is clearly demonstrated.

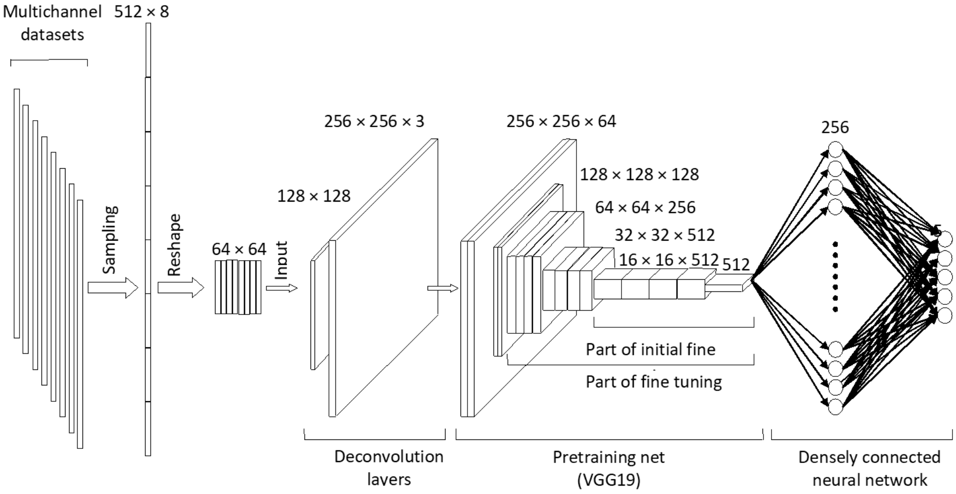


Fig. 1. VGG-19 architecture framework

The model built on the basis of VGG-19 (fig. 2) was modified as follows: the upper fully connected layers were cut off and fully connected layers with 2048, 1024 and 512 neurons were added, respectively, the activation functions were RELU. After each fully connected layer, Dropout layers were added with coefficients of 0.3. The last layer of each model was a fully connected layer with 4 neurons, which corresponded to the number of classes and were activated by the Softmax function.

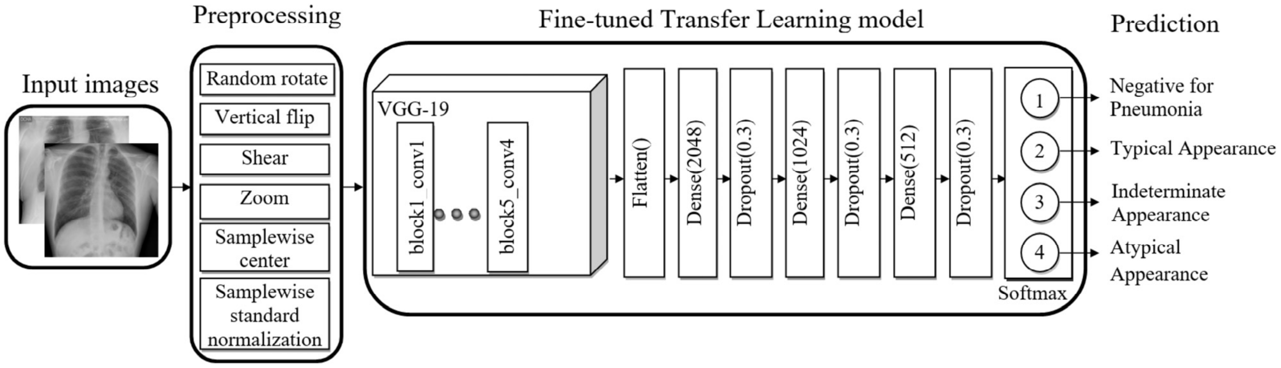


Fig. 2. Model based on VGG-19

ResNet is one of the most powerful Deep Neural Networks (DNN) that has achieved fantastic results in the 2015 ILSVRC classification. ResNet was first introduced in 2015 in the article "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. There are many options for the ResNet architecture, i.e. the same concept, but with a different number of neural network layers.

The practice of using neural networks in image recognition tasks shows that the depth of the network is crucial and all networks that have the best results on the complex ImageNet dataset use ‘very deep’ models, with a depth of sixteen layers or more. However, in neural networks with a large number of layers, a degradation problem was discovered, which means that with increasing network depth, accuracy becomes saturated and then rapidly deteriorates. This results in a higher training error. In the ResNet architecture, this problem was solved by introducing a deep residual learning system [14]. This is achieved in a simple, at first glance, way. It is known that a neural network can approximate almost any function, for example, some complex function $H(x)$. Then it is true that such a network can easily learn the residual function: $F(x)=H(x)-x$. Obviously, the original objective function will be $H(x)=F(x)+x$. If we take some convolutional network and apply (stack) 20 more layers to it, then we would like the deep network to behave at least as good as its shallow counterpart.

If for some convolutional network we apply (stack) 20 more layers, then we would like the deep network to behave at least no worse than its shallow counterpart. To reach this, add a shortcut-join, and, perhaps, it will be easier for the optimizer to make all weights close to zero than to create an identical transformation. An example of such a residual block is illustrated at fig. 3.

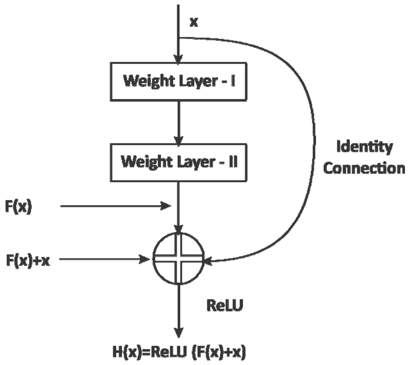


Fig. 3. Residual block of Deep Residual Network

The model built on ResNet-50 (fig. 4) was transformed by adding fully connected layers with 1024 and 512 neurons, the activation functions were also RELU. Each fully connected layer was followed by a Dropout layer with the same ratio as in the VGG-19 model.

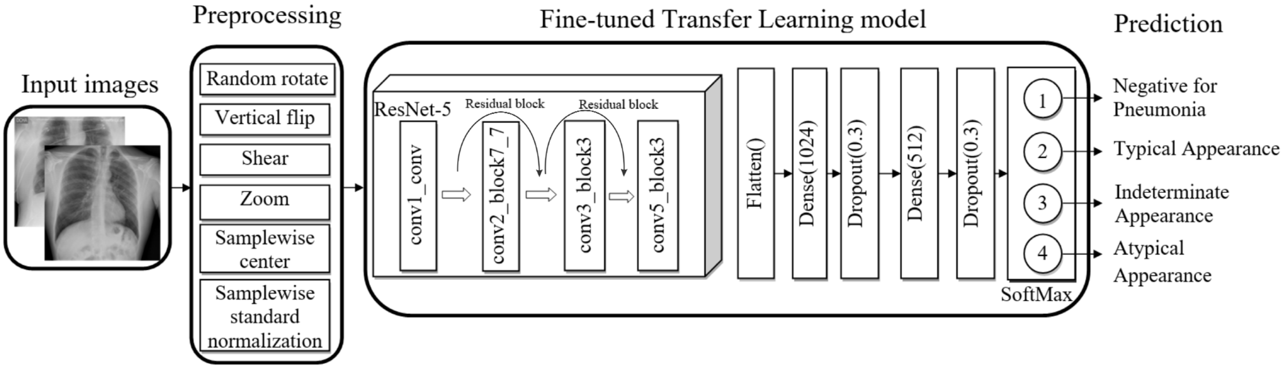


Fig. 4. Model based on ResNet-50

InceptionV3 is Google's neural network for object recognition in images. It is a pre-trained Convolutional Neural Network model with 48 levels of depth. The aim of the Inception architecture is primarily to be computationally forceful and efficient for real-world applications. Especially, it can be achieved through increasing both the width and the depth of the network. If the resources have become, for example, twice as many, it is most efficient to make the layers wider and the network deeper. On the contrary, if you only go deeper, it will be ineffective. To be precise, if the resources have become, for example, twice as many, it is most efficient to make the layers wider and the network deeper. If the depth of the network is increased without any modification, it will be ineffective.

Thus, InceptionV3 [15] talks about the ideas of factorization. The purpose of convolution factorization is to reduce the number of connections / parameters without compromising the efficiency of the network. Thanks to 48 layers, a lower error rate is achieved, and he becomes 1st prize-winner in image classification in ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2015.

The model built on InceptionV3 (fig. 5, source https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/) has been transformed by adding fully connected layers with 512 and 256 neurons, activation functions were also RELU. Each fully connected layer was followed by a Dropout layer with a factor of 0.5 as in the VGG-19 model.

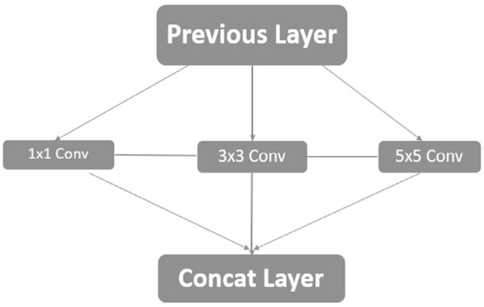


Fig. 5. The idea of the Inception module structure

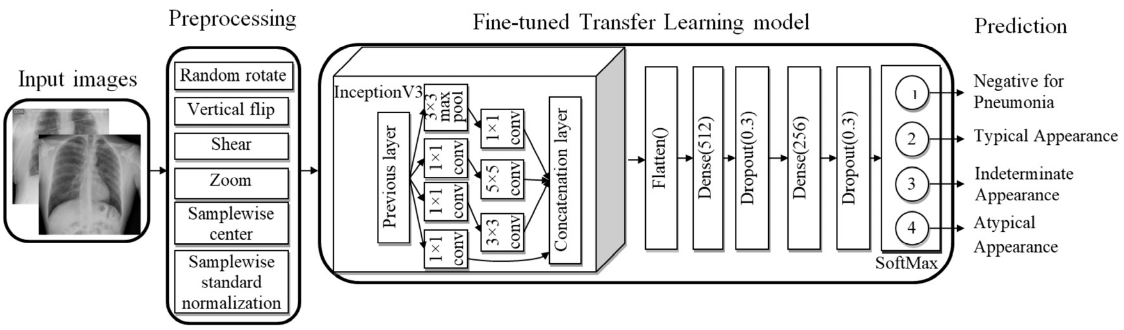


Fig. 6. Model based on InceptionV3

2. Materials and methods

2.1. Dataset description

In this research data were obtained from the database of chest X-Ray images of the Society for Imaging Informatics in Medicine (SIIM) partnered with the Foundation for the Promotion of Health and Biomedical Research of Valencia Region (FISABIO), Medical Imaging Databank of the Valencia Region (BIMCV) and the Radiological Society of North America (RSNA).

Tab. 1. Details of dataset

Classes	Dataset			
	Training	Validation	Testing	Total
Negative for Pneumonia	1067	316	158	1541
Typical Appearance	1063	317	158	1538
Indeterminate Appearance	1183	318	159	1659
Atypical Appearance	1121	316	158	1596

The X-Ray database contained 6334 DICOM images which were converted to jpeg images with a resolution of 224×224. The dataset contained 4 different varieties of COVID-19, namely 'Negative for Pneumonia', 'Typical Appearance', 'Indeterminate appearance', 'Atypical Appearance'. During the experiment, the data were divided in a ratio of 70:20:10 (training + validation + testing).

2.2. Methodology

The Transfer Learning approach involves using the accumulated knowledge of one model to solve the problem of another model. This is achieved in such ways as fine tuning, or simple use of a ready-made model, without additional training. In the simplest case, an example of Transfer Learning use is the AI Dungeon game, in which the neural network of an adventure game was not built from scratch, but using the latest GPT-2 NLP model from Google. As for the current research, Transfer Learning has been applied to solve the classification problem in Computer Vision.

The methodology that describes Transfer Learning for solving the COVID-19 classification problem is demonstrated in fig. 7.

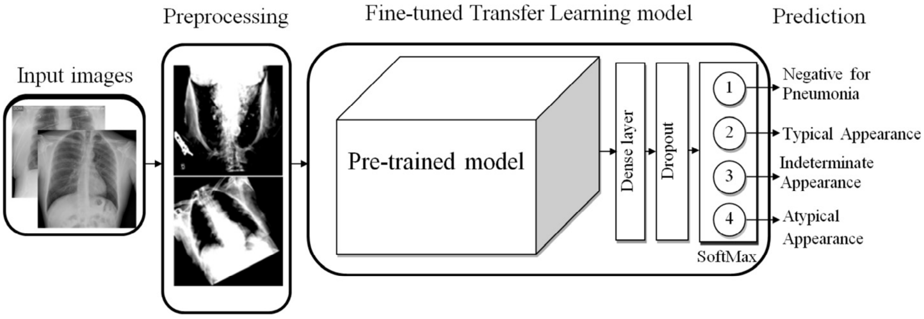


Fig. 7. Transfer learning methodology

The main goal of the experiment was to classify medical images into 4 categories. One of the important stages in the analysis of medical images is the selection of the region of interest (segmentation) [16]. Data preprocessing and cleaning are important tasks [17] that must be performed before a dataset can be used to train a model. In the previous works of the authors [18], the solution of the segmentation problem was succeeded by preprocessing using image filtering with the choice of the most optimal filter. In the current task, for preprocessing, the tools of the *Python Tensorflow 2* and *Keras* library were used. Using the *ImageDataGenerator* class, augmentations were performed on the images of the dataset. An example of using *ImageDataGenerator* for augmentations is shown in the following snippet:

```
train_datagen=ImageDataGenerator(
    samplewise_center=True,
    samplewise_std_normalization=True,
    vertical_flip = True,
    rotation_range=30,
    shear_range = 0.2,
    preprocessing_function=vgg19.preprocess_input,
    zoom_range=0.2
)
```

In this case, image augmentation techniques such as sample-wise standard normalization and centering, vertical flip, random rotation, shearing, and image magnification were applied.

It should be noted that *Keras* library provides implemented image preprocessing functions for all of regarded models. To achieve this, it is sufficient to specify the *preprocessing_function* parameter in the initialization of the *ImageDataGenerator* class by choosing the appropriate preprocessing function.

For a model using the VGG-19 architecture, the result of applied augmentation is as follows (fig. 8):

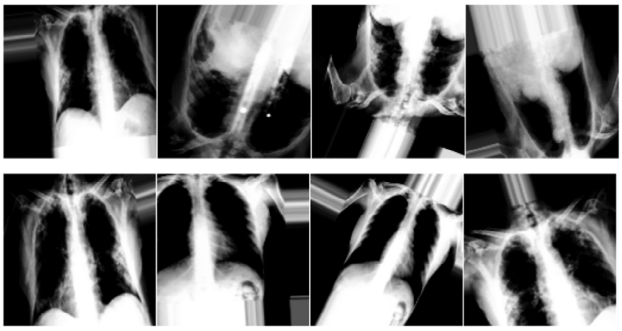


Fig. 8. Example of image preprocessing for VGG-19 network

For a model using the ResNet-50 architecture, the result of augmentation is as follows (fig. 9):

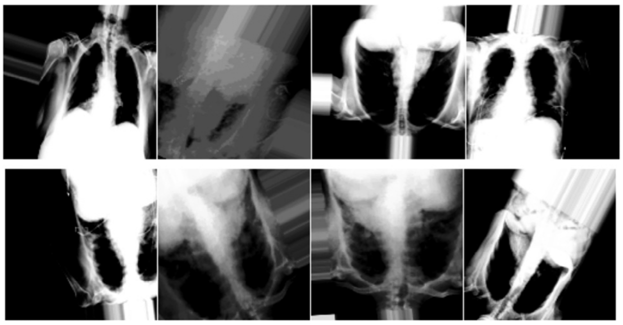


Fig. 9. Example of image preprocessing for ResNet-50 network

For a model using the InceptionV3 architecture, augmented images look like this (fig. 10):

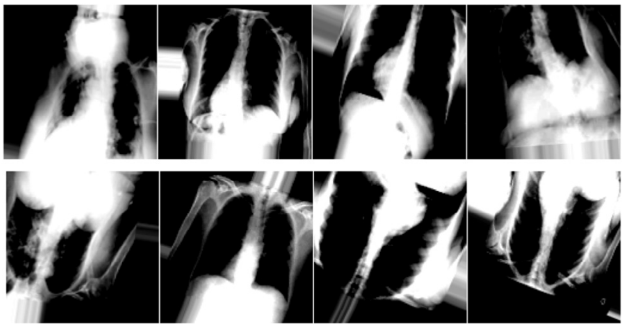


Fig. 10. Example of image preprocessing for InceptionV3 network

To illustrate the principle of operation of each of the architectures, it is necessary to visualize the work of some of their layers in the following demonstration (fig. 11).

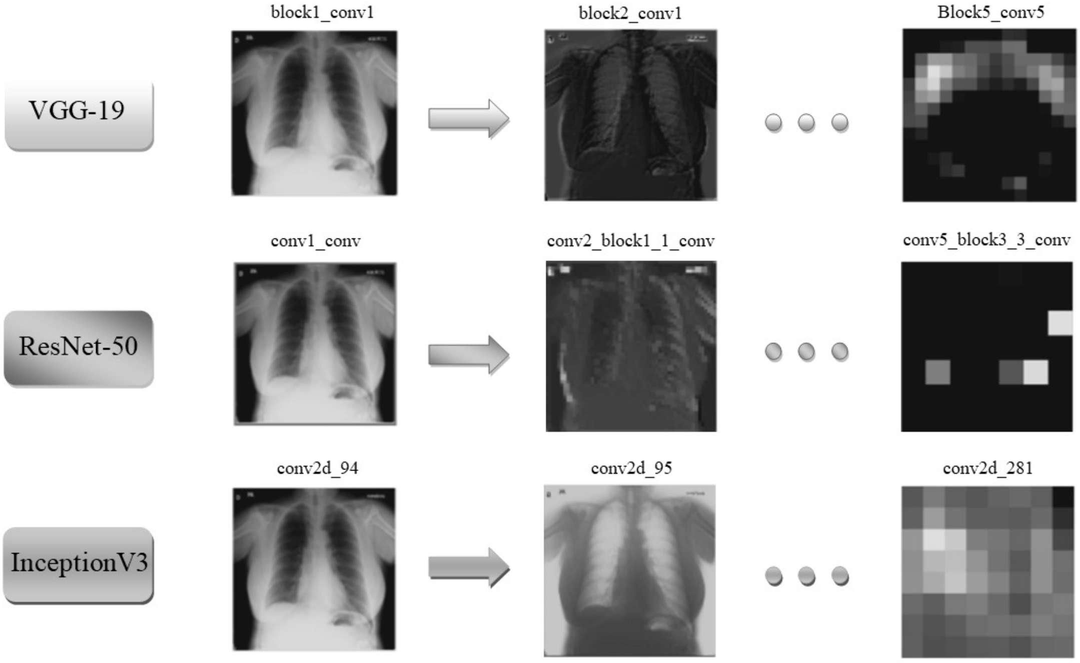


Fig. 11. Demonstration of feature maps identified by the models

Another important point in data preprocessing to solve the classification problem is performing operations on a dataset. Because machine learning models often fail to generalize well the data they were trained on, a resampling technique called cross-validation is applied to them. In our case, we are using the Stratified K-fold cross-validation method. This method allows to train the model with a lower bias, unlike other methods. Stratified K-fold has a *k* parameter that determines how many times the data in the dataset will be iterated over. Thus, the model is trained for *k* iterations on various datasets presented in the training, validation, and test samples. The numbers 5 and 10 were chosen as *k* values for the experiment to avoid the problem of overfitting and underfitting the model.

3. Implementation and results

The size of the input image for all three models was the same and equaled to 224×224. The batch size was taken 16, the number of epochs was 50. The X-ray image data were preprocessed with the corresponding preprocessing functions for each model; in order to avoid overfitting, augmentation was performed over each image.

The results of the first experiment were summed up according to the following types of evaluation metrics: training accuracy, validation accuracy, training loss and validation loss for different epochs. The results of training models are revealed in fig. 12.

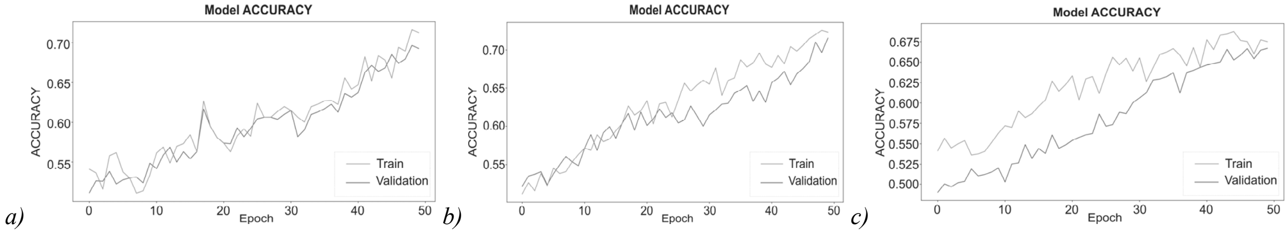


Fig. 12. Model training results a) VGG-19; b) ResNet-50; c) InceptionV3

The results of the experiments are also shown in tab. 2.

Tab. 2. Default training results for each model

Model	Epochs	Training accuracy	Validation accuracy	Training loss	Validation loss
VGG-19	1	0.51	0.54	1.41	1,81
	…	…	…	…	…
	49	0.68	0.69	0.71	0.72
	50	0.69	0.71	0.73	0.73
ResNet-50	1	0.52	0.51	1.49	1.97
	…	…	…	…	…
	49	0.69	0.71	0.83	0.81
	50	0.71	0.72	0.82	0.82
InceptionV3	1	0.49	0.54	1.13	1.76
	…	…	…	…	…
	49	0.64	0.68	0.95	0.93
	50	0.66	0.67	0.93	0.92

As the table illustrates, the model based on ResNet-50 had the best accuracy. To improve the accuracy each model was fine-tuned. In each network, one of the types of fine-tuning of hyperparameters was performed, namely, unfreezing the upper layers of the model. For the VGG-19 model, the *block5_conv4* layer was unfrozen which contained 512 filters with a size of 14×14.

In the model based on ResNet-50, the *conv5_block3_3_conv* layer was unfrozen which had a convolution kernel size of 7×7 and 2048 filters.

In the model on the InceptionV3 network, the *conv2d_93* layer was unfrozen. It had 192 5×5 filters.

Model loss has calculated by categorical crossentropy function.

The results of the training performed after unfreezing the last convolutional layer of each model are shown in the following table (tab. 3).

It is also advisable to calculate a confusion matrix based on the results of fine tuning and consider some of the most important metrics for assessing the quality of the models.

Tab. 3. Results after unfreezing the last layer of each model

Model	Epochs	Training accuracy	Validation accuracy	Training loss	Validation loss
VGG-19	1	0.54	0.57	1.38	1.83
	…	…	…	…	…
	49	0.71	0.72	0.68	0.74
	50	0.72	0.74	0.70	0.75
ResNet-50	1	0.51	0.52	1.44	1.11
	…	…	…	…	…
	49	0.70	0.72	0.78	0.76
	50	0.72	0.73	0.77	0.77
Inception V3	1	0.51	0.56	1.16	1.15
	…	…	…	…	…
	49	0.66	0.71	0.98	0.95
	50	0.68	0.69	0.96	0.94

To evaluate the performance of each model, the most important parameters were calculated: Precision, Recall, F1-Score, Accuracy. These parameters were calculated using formulas (1–4). Consider a special case for two classes (named positive and negative), since it is suitable for solving multiclass classification problems. Parameters TP (True Positive) TN (True Negative) FP (False Positive) FN (False Negative) are taken from the confusion matrix. In other words, TP is an outcome where the model correctly predicted the positive class. Additionally, TN is a result where the model correctly anticipated the negative class. FP is an outcome where the model incorrectly predicted the positive class. Accordingly, FN is a result where the model incorrectly predicted the negative class. Accuracy is a metric that describes the overall prediction accuracy of a model across all classes. Precision is a metric that displays what proportion of predicted positive predictions should really be positive. Recall is a measure that indicates what proportion of positive predictions was recognized by the algorithm as positive. F1-Score - harmonic mean of Recall and Precision.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{1}$$

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

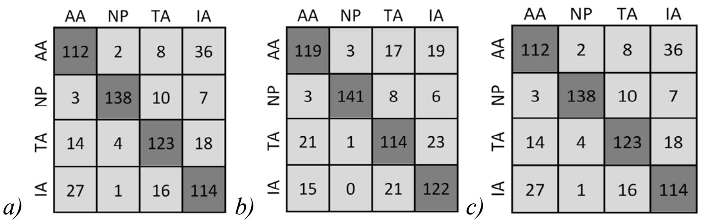
$$Specificity = \frac{TN}{TN + TP}, \tag{4}$$

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

The confusion matrix is illustrated at the following figure (fig. 13).

Tab. 4. Values of the evaluation metrics for each model

Model	Precision %	Recall %	F1-Score	Acc. %
VGG-19	78.35	78.89	78.62	78.89
ResNet-50	79.14	80.27	79.70	79.14
InceptionV3	76.93	77.61	77.27	77.61



a)

	AA	NP	TA	IA
AA	112	2	8	36
NP	3	138	10	7
TA	14	4	123	18
IA	27	1	16	114

b)

	AA	NP	TA	IA
AA	119	3	17	19
NP	3	141	8	6
TA	21	1	114	23
IA	15	0	21	122

c)

	AA	NP	TA	IA
AA	112	2	8	36
NP	3	138	10	7
TA	14	4	123	18
IA	27	1	16	114

Fig. 13. Confusion matrices for each model on the test dataset results:(a) VGG-19; (b) ResNet-50; (c) InceptionV3

Tab. 5. Comparison results of optimizers for models

Model	Optimizer		Precision %	Recall %	F1-Score	Accuracy %
VGG-19	Adam	0.001	74.31	73.92		74.76
		0.0001	78.35	78.89	78.62	78.89
		0.00001	75.19	74.27		74.53
	SGD	0.001	69.90	70.33		70.08
		0.0001	70.52	70.79		71.35
		0.00001	71.08	68.11	69,56	71.64
	RMSProp	0.001	72.54	71.85		70.97
		0.0001	71.36	72.17		71.63
		0.00001	73.13	75.45	74,27	74.02
ResNet-50	Adam	0.001	79.14	80.27	79.70	79.14
		0.0001	78.22	77.41		78.56
		0.00001	75.03	75.66		74.99
	SGD	0.001	72.06	72.18		72.83
		0.0001	73.12	72.22	72,67	72.74
		0.00001	71.59	70.84		70.94
	RMSProp	0.001	74.03	76.17	75,08	74.53
		0.0001	73.15	73.01		73.42
		0.00001	71.28	70.76		70.98
InceptionV3	Adam	0.001	69.11	70.37		69.92
		0.0001	70.54	70.61		70.12
		0.00001	72.72	71.15	71.93	72.36
	SGD	0.001	75.28	75.03		75.36
		0.0001	76.93	77.61	77.27	77.61
		0.00001	75.72	75.86		75.48
	RMSProp	0.001	74.19	74.25		74.10
		0.0001	75.03	76.31	75.66	74.92
		0.00001	74.86	74.53		74.47

4. Optimizers and learning rate

Optimization is a process that tries to reduce network error. It plays a crucial role in improving the accuracy of the model.

Optimizer options were Adaptive moment estimation (Adam), Stochastic Gradient Descent (SGD), and Root Mean Square propagation (RMSprop) [6]. Adam is simple to implement, computationally efficient, and low on memory. SGD is an incremental gradient descent algorithm that tries to find the minimum error through iteration. RMSProp can deal with stochastic objectives very nicely, making it applicable to mini batch learning. All three optimizers were applied to the two best trained models in order to compare the performance of the optimizers and select an optimizer that is applicable to all three of the studied models. The obtained parameters for VGG-19, ResNet-50 and InceptionV3 after applying the three above-mentioned optimizers are shown in the table. It can be seen that the Adam optimizer has shown promising results. among all optimizers. Consequently, the Adam optimizer was selected to be applied to three trained models.

For the experiment, learning rates were taken with the values 0.001, 0.0001, 0.00001. The tab. 5 shows the most optimal learning rate results obtained for each model highlighted in green.

5. Stratified K-fold cross-validation results

Stratified K-fold data cross-validation was applied for all three models. It allowed each model to improve its learning outcomes. The following table displays the training results with various selected k value parameters (tab. 6).

Tab. 6. Results with various selected k value parameters

Model	$k=5$				$k=10$			
	Precision %	Recall %	F1-Score	Acc. %	Precision %	Recall %	F1-Score	Acc. %
VGG-19	79.06	78.95	79.00	79.49	80.49	79.56	80.02	80.41
ResNet-50	80.23	80.87	80.55	80.17	82.69	81.34	82.01	82.74
InceptionV3	78.07	79.16	78.61	77.61	79.81	79.38	79.59	79.56

Conclusions

In this paper, fine-tuning had applied on three pre-trained Transfer Learning models (VGG-19, ResNet-50, InceptionV3) to solve a multiclass classification problem. The research was conducted on a medical chest X-Ray images dataset of the COVID-19 pneumonia variations obtained from the Society for Imaging Informatics in Medicine (SIIM) partnered with the Foundation for the Promotion of Health and Biomedical Research of Valencia Region (FISABIO), Medical Imaging Databank of the Valencia Region (BIMCV) and the Radiological Society of North America (RSNA). A total of 4433 images were used as training images, 1267 as validation images, and 633 images were taken to test the models. Such tools as the confusion matrix of the models were estimated such important metrics as accuracy, recall, F1-score, precision.

Initially, training was carried out without any modification to each model. Thereafter, various operations were sequentially carried out to fine-tune each model, including unfreezing the last convolutional layer, changing the types of optimizers, and selection the optimal learning rate. Furthermore, to make the most of the dataset the Stratified K-fold method was used which allowed each model to be trained and tested on all data, and it also tended to improve the performance of each model.

VGG-19-based model was the least accurate in classifying COVID-19 pneumonias and normal chest X-Rays; however, ResNet-50 has shown promising results in classifying COVID-19 varieties and normal images. The achieved accuracies of VGG-19, ResNet-50, InceptionV3 were 80.41 %, 82.74 %, and 79.56 % respectively. The InceptionV3 model also performed well. In various works it is confirmed, that efficiency of this model can be improved slightly by increasing the size of the input image, since the maximum input size for it is 299×299.

In addition, various optimizers were tested, and among all the tested optimizers Adam with a learning rate of 0.0001 demonstrated the best performance, and therefore, was applied to minimize errors and better optimize in the training process.

The main result of this study was the selection of an optimal predictive model for the diagnosis of pneumonia. As a result, it is recommended to use the ResNet-50 model with fine tuning according to the available dataset.

References

[1] Narin A, Kaya C, Pamuk Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Anal Appl 2021; 24: 1207-1220.

[2] Narayanan BN, Hardie RC, Krishnaraja V, Karam C, Davuluru VSP. Transfer-to-transfer learning approach for computer aided detection of COVID-19 in Chest Radiographs. AI 2020; 1(4): 539-557.

[3] Tahir H, Khan MS, Tariq MO. Performance analysis and comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the detection and counting of vehicles. Proc 2021 Int Conf on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 19-20 February 2021: 587-594.

[4] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010; 22: 1345-1359.

[5] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning – ICANN 2018. In

Book: Kůrková V, Manolopoulos Y, Hammer B, Iliadis L, Maglogiannis I, eds. Artificial neural networks and machine learning – ICANN 2018. Springer; 2018: 270-279.

[6] Rifkin R, Klautau A. In defense of one-vs-all classification. J Mach Learn Res 2004; 5: 101-141.

[7] Ikechukwu AV, Murali S, Deepu R, Shivamurthy RC. ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transitions Proceedings 2021; 2(2): 375-381. DOI: 10.1016/j.gltp.2021.08.027.

[8] Setiawan W, Damayanti F. Layers modification of convolutional neural network for pneumonia detection. J Phys Conf Ser 2020; 1477: 052055. DOI: 10.1088/1742-6596/1477/5/052055.

[9] Gouabou ACF, et al. Ensemble method of convolutional neural networks with directed acyclic graph using dermoscopic images: Melanoma detection application. Sensors 2021; 21(12): 3999.

[10] Anilkumar KK, Manoj VJ, Sagi TM. Automated detection of leukemia by pretrained deep neural networks and transfer learning: A comparison. Med Eng Phys 2021; 98: 8-19. DOI: 10.1016/j.medengphy.2021.10.006.

[11] Liu Y, Yang P, Pi Y, et al. Automatic identification of suspicious bone metastatic lesions in bone scintigraphy using convolutional neural network. BMC Med Imaging 2021; 21: 131. DOI: 10.1186/s12880-021-00662-9.

[12] Zhou J, Yang X, Zhang L, Shao S, Bian G. Multisignal VGG19 network with transposed convolution for rotating machinery fault diagnosis based on deep transfer learning. Shock Vib 2020; 2020: 8863388. DOI: 10.1155/2020/8863388.

[13] Fengzi L, Kant S, Araki S, Bangera S, Shukla SS. Neural networks for fashion image classification and visual search. SSRN Electronic Journal 2020. DOI: 10.2139/ssrn.3602664.

[14] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 770-778. DOI: 10.1109/CVPR.2016.90.

[15] Szegedy C, et al. Rethinking the inception architecture for computer vision. Proc IEEE conf on computer vision and pattern recognition 2016: 2818-2826. DOI: 10.1109/CVPR.2016.308.

[16] Pashina TA, Gaidel AV, Zelter PM, Kapishnikov AV, Nikonorov AV. Automatic highlighting of the region of interest in computed tomography images of the lungs. Computer Optics 2020; 44(1): 74-81. DOI: 10.18287/2412-6179-CO-659.

[17] Kaibassova D, La L, Smagulova A, et al. Methods and algorithms of analyzing syllabuses for educational programs forming intellectual system. J Theor Appl Inf Technol 2020; 98(5): 876-888.

[18] Nurtay M, Kaibassova D. Methods of filtering medical images for solving the segmentation problem. Youth and Modern Information Technologies. Proc XVIII Int Scientific and Practical Conf of Students, Postgraduates and Young Scientists. Tomsk: Tomsk Polytechnic University Publisher; 2021: 34-35.

Authors' information

Dinara Kaibassova (b. 1978) graduated from the Abai Almaty State University, majoring in Informatics and Computerization Manager. In 2020 obtained the degree of PhD in Information Systems. Works as Associate Professor at the Information and Computing Systems department of Abylkas Saginov Karagandy Technical University. Research interests: image processing, pattern recognition theory, data mining, natural language processing. E-mail: *dindgin@mail.ru* .

Margulan Nurtay (b. 1996), master student of Abylkas Saginov Karagandy Technical University. Research interests: computer vision & pattern recognition, deep learning E-mail: *solano.lifan2@bk.ru* .

Ardak Tau (b. 1987), graduated from the Karagandy State Technical University, master of Computer Science. Works as Senior teacher at the Information and Computing Systems department of Abylkas Saginov Karagandy Technical University. Research interests: machine learning, deep learning, image processing. E-mail: *ardak.tau@mail.ru* .

Mira Kissina (b. 1987), graduated from Karagandy State University, majoring in Math Science, master of Computer Science. Works as Senior teacher at the Information and Computing Systems department of Abylkas Saginov Karagandy Technical University. Research interests: math statistics, machine learning. E-mail: *motya.2002@mail.ru* .

Received December 10, 2021. The final version – September 1, 2022.