

# Machine Learning Models Efficiency Analysis for Image Classification Problem

Kirill Smelyakov<sup>1</sup>, Yaroslav Honchar<sup>1</sup>, Oleksandr Bohomolov<sup>1</sup> and Anastasiya Chupryna<sup>1</sup>

<sup>1</sup> Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine

## Abstract

The article is devoted to the analysis of the effectiveness of the application of modern machine learning models of convolutional neural networks, which are used for image classification. To conduct such an analysis, an actual dataset is selected and divided into training, validation, and test subsets in a standard proportion. The dataset which is selected consists of images of birds. Classification efficiency indicators are determined. ResNet and EfficientNet V2 neural networks are trained using a full training cycle and Transfer Learning technology on frozen and free weights. Pytorch framework is used to train ResNet model and Tensorflow framework is used to train EfficientNet V2 model. The effectiveness of the use of neural networks is evaluated. The evaluation is done by analyzing popular classification metrics, such as precision, recall, and f1 score. The results of experiments are given, along with conclusions and practical recommendations on the use of machine learning models.

## Keywords

Convolutional Neural Network, Image Classification, Machine Learning Model, Transfer Learning, Metrics, Efficiency Estimation

## 1. Introduction

The successful application of neural networks to solve actual computer vision problems has led to the emergence of many new models of convolutional neural networks (CNN) and their various modifications used for image detection and classification in recent years. Many CNN models are pre-trained. This means that they are focused on the detection or classification of images of a certain list of classes. This is enough for some applications. However, most often the user needs to expand the list of classes with which the CNN works. To do this, it is required to train the CNN to work with images of new classes. For such training, there are relatively large number of alternative techniques and training methods. And only on the basis of an experimental analysis of effectiveness, it is possible to answer the question of which training method will be better and by what criterion.

The aim of the work is to ensure the efficiency of machine learning of modern convolutional neural networks.

The goals of the work are to develop a plan and set up a series of experiments on the application of widely used machine learning methods in relation to modern CNNs based on actual data, evaluate the effectiveness of machine learning on free and frozen weights, formulate recommendations on the practical application of machine learning techniques and methods.

---

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland  
EMAIL: kyrylo.smelyakov@nure.ua (K. Smelyakov); yaroslav.honchar@nure.ua (Y. Honchar); oleksandr.bohomolov@nure.ua (O. Bohomolov); anastasiya.chupryna@nure.ua (A. Chupryna)  
ORCID: 0000-0001-9938-5489 (K. Smelyakov); 0000-0002-8637-8025 (Y. Honchar); 0000-0002-9539-8888 (O. Bohomolov); 0000-0003-0394-9900 (A. Chupryna)



© 2022 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

## 2. Related Works

In famous reviews [1-3] it is shown that in recent years, convolutional neural networks (CNN) have become the standard model and technology for a wide range of computer vision tasks in both image detection and classification. Mainly thanks to recent advances in deep learning [1, 3] and highly efficient post-image processing. In this regard, a detailed analysis of the effectiveness of applying the most common CNN models for image classification is provided in this work [4]. The relationship between the components of the CNN architecture and the effectiveness of their application is shown. The paper [5] proposed a solution for the classification of moving vehicles based on the application of CNN. The paper [6] describes the use of new FPGA technology to implement training and improve performance with testing on VGG-16 and ResNet-50 networks. The paper [7] describes the most important models and technologies of deep learning and the effective use of a large number of hidden layers of the CNN to improve the efficiency of training modern neural networks. At the same time the paper [8] describes the model and principles of operation of small CNN, which is important for the efficiency of relatively low-powered mobile devices, especially when processing a video stream.

An analysis of the current state of the issue allows us to conclude that the use of CNN is relevant for cars classification [9], road signs [10], lung diseases on x-rays [11], products in warehouses and in electronic stores [12], gestures [13] and in many other applications [14-16].

At the same time, the effectiveness of CNN applications is determined by the quality of network training. In such a situation, a reasonable choice of a machine learning model comes to the foreground [17, 18] for efficient tuning of neural network parameters. In this respect, a lot of recent research has been devoted to the development of combined learning technologies that are associated with freezing, training and retraining of unfrozen layers of a neural network [7, 17, 19].

In certain applications, research is being carried out related to federated learning, as well as edge computing, which is relevant for solving problems of centralized data processing. At the same time, great attention is paid to research on the effectiveness of training deep neural networks, as the most common architecture. Many aspects of the solution of these issues are provided in the work [20]; in particular, promising developments in related areas of development of the architecture of deep neural networks and deep learning methods for such neural networks are presented.

Minimizing the computational complexity of deep learning algorithms is just as important as ensuring high accuracy. The paper [21] describes the use of the Broad Learning System (BLS) as an alternative method of machine learning, which leads to a significant reduction in the amount of calculations and the duration of training.

In recent years, more and more attention has been paid to the use of Transfer Learning, which allows you to adapt pre-trained neural networks to new classes of objects by training only classification layers. Such algorithms work an order of magnitude or more faster than algorithms with a full learning cycle [7, 22] and, most often, give higher classification accuracy [23, 24].

Recently, a lot of different techniques and methods of machine learning have appeared. And theoretically it is simply impossible to determine the best method. To solve this problem, it is planned to develop a plan and conduct a series of experiments on training ResNet and EfficientNet V2 convolutional neural networks, perform a comparative analysis of efficiency, and evaluate machine learning methods, all other things being equal. Based on the results, it is planned to formulate recommendations on the practical application of machine learning methods of modern convolutional neural networks.

All these experiments are planned to be carried out on the basis of consideration “300 Bird Species” dataset [25], because it has a lot of important features: a large number of classes, some of which are similar to each other; a large number of diverse images within the same class; different shooting angles and bird backgrounds.

At the same time, for an objective assessment of the effectiveness of neural networks, metrics such as Precision, Recall and F1 Score type [26] are evaluated, as well as training time.

Successful solution of this problem through the transfer of machine learning technologies will improve the efficiency of the use of neural networks in existing video data processing services, as well as in the creation of promising multimedia traffic processing services in computer networks [27], for image analysis by robots and drones [28], in many other relevant applications [29].

### 3. Methods and Materials

Consider the data (dataset) that will be used in further methods and experiments, data analysis methods and materials, as well as metrics that will be used for effectiveness evaluation.

#### 3.1. Dataset Description

Methods and experiments are based on the use of Dataset “300 Bird Species” [25]. The dataset contains 42622 training images (approximately 130-170 images in each class), 1500 test images (5 images for each class) and 1500 test images (5 images for each class). A total of 300 bird classes were assigned for training, testing and validation. The partitioning of the original array of images differs significantly from the standard partitioning (train 70%, validation 10%, test 20%) and is inefficient. Therefore, the images of each class were first combined (train + test). All images of the dataset are presented in jpg format and are standardized - these are color images 224x224x3. Detailed information about the Dataset “300 Bird Species” can be obtained at the resource [25]. Examples of images are shown in Figure 1.

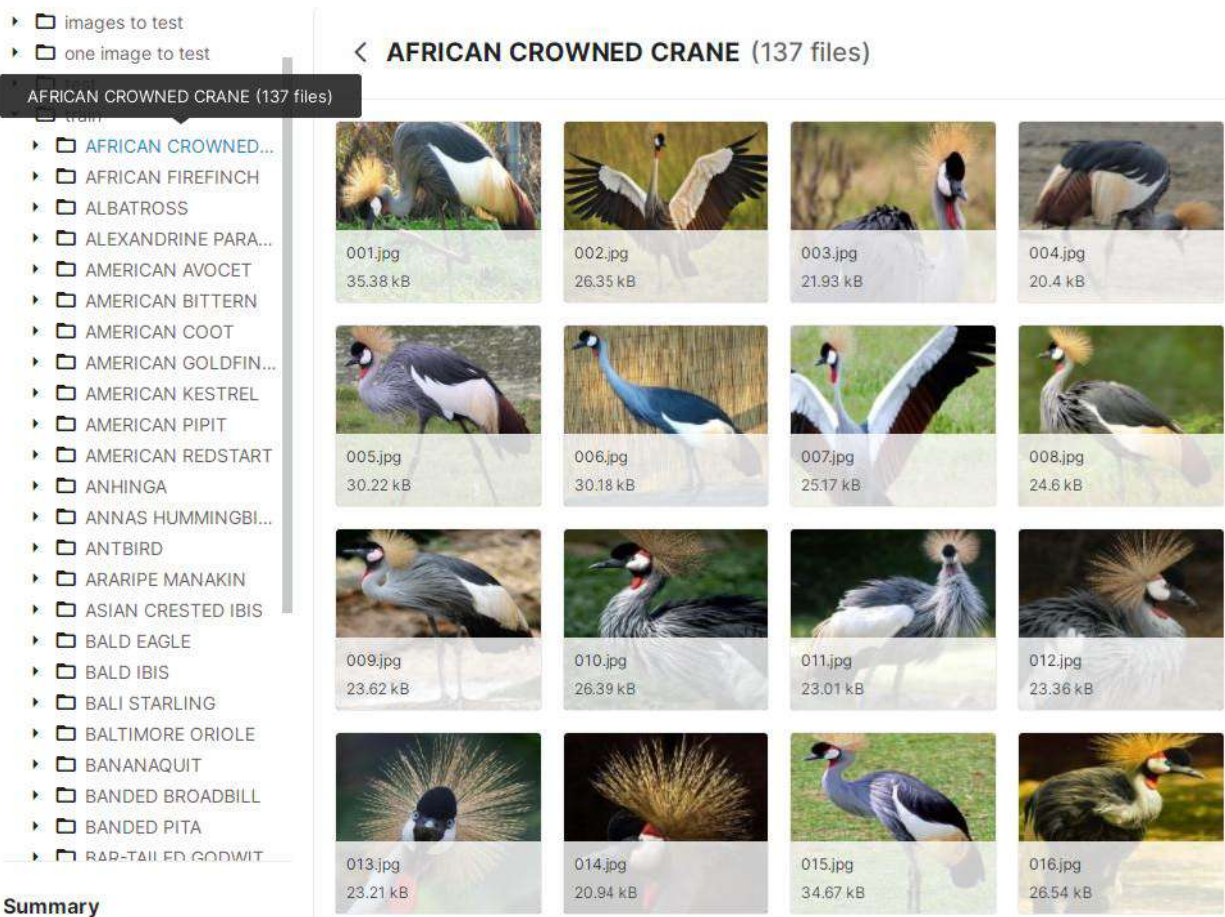


Figure 1: «African Crowned Crane» class images example of the “300 Bird Species” Dataset [25]

The “300 Bird Species” dataset was chosen for research because it has a large number of classes (300 classes). This allows you to objectively study the effectiveness of the application of machine learning models and neural networks, both in relation to errors (precision, recall types, etc.) [26], as well as training time (time efficiency). To improve the efficiency of machine learning and classification, various pre-processing technologies can be applied to source images [30, 31], including object segmentation [16, 32]. For the purity of experiments and the possibility of adequate comparative analysis with other sources, minimal preprocessing was used in the work to bring the grayscale to the required format for a given neural network.

### 3.2. Efficiency Indicators

To evaluate the quality of classification for each class, we calculate the number  $c_{i,j}$ , ( $i, j = 1, \dots, n$ ), hits of objects of class  $i$  in class  $j$ . After that for each class  $k$  evaluate three main indicators of quality [26]

$$Precision_k (= P_k) = \frac{c_{k,k}}{\sum_{i=1}^n c_{i,k}}, \quad (1)$$

$$Recall_k (= R_k) = \frac{c_{k,k}}{\sum_{j=1}^n c_{k,j}}, \quad (2)$$

$$F1\ score_k (= F_k) = \frac{2 \cdot P_k \cdot R_k}{P_k + R_k}, \quad (3)$$

where the coefficients are described in the following Figure 2.

		Predict			
A c t u a l		Class 1	...	Class n	$Recall_i$
	Class 1	$TP_1 = c_{1,1}$	...	$c_{1,n}$	$TP_1 / \sum_{j=1}^n c_{1,j}$
	...	...	...	...	...
	Class n	$c_{n,1}$	...	$TP_n = c_{n,n}$	$TP_n / \sum_{j=1}^n c_{n,j}$
	$Precision_i$	$TP_1 / \sum_{i=1}^n c_{i,1}$	...	$TP_n / \sum_{i=1}^n c_{i,n}$	

**Figure 2:** Evaluation of classification quality indicators (metrics) [26]

As a result, we get three vectors of estimates *Precision*, *Recall*, *F1 score*. When working with a large number of classes, constant monitoring of vectors *Precision*, *Recall*, *F1 score* is very computationally intensive. In such a situation of operational quality control of classification for a large number of classes, estimates of the minimum and average on these vectors are used

$$P_{min} = \min\{P_k\}_k, P_{av} = \frac{1}{n} \sum_{k=1}^n P_k; \quad (4)$$

$$R_{min} = \min\{R_k\}_k, R_{av} = \frac{1}{n} \sum_{k=1}^n R_k; \quad (5)$$

$$F_{min} = \min\{F_k\}_k, F_{av} = \frac{1}{n} \sum_{k=1}^n F_k. \quad (6)$$

Indicators (1) – (6) are estimated after training for the test sample. However, our experiments differ only in the machine learning model. Therefore, indirectly, all these estimates also characterize the effectiveness of the application of the machine learning model. Indicators (1) – (6) assess the quality of the classification. For the purposes of practical application, it is also important to know the training time. Therefore, to these estimates in experiments, we add the training time.

### 3.3. Main Methods and Techniques

To conduct research, we selected actual convolutional neural networks widely used for image classification: ResNet and EfficientNet V2 [33, 34]. For these models, relatively small types of models

have been selected CNN ResNet 50 and CNN EfficientNet V2 type B0; these networks are approximately equivalent in terms of the number of parameters and have the same input data format PT, which is a tensor ready to be submitted for training with completed preprocessing. This is done in order to:

- create approximately equivalent conditions for use;
- maximize the learning rate in order to devote more time to the comparative analysis of effectiveness, which is of the greatest interest for the purposes of the work.

Each neural network is trained using a full training cycle from scratch, as well as using Transfer Learning technology on frozen and free weights. Thus, for each network, 3 experiments are set to train it.

After that, the trained networks are tested and performance indicators are evaluated (indicators (1) – (3) and training time), summary tables of results for all classes are given, recommendations are given for the practical application of machine learning models. Since there are quite a lot of classes, the results for a certain number of classes are reflected directly in the work at the beginning and at the end of the list, and integral estimates of efficiency are also given.. Full results tables are available at the link [35].

## 4. Experiment

Consider the experiments using convolutional neural networks mentioned above.

### 4.1. Experiment Using CNN ResNet

The PyTorch Framework was used for the experiment. Therefore, the model for the experiment was taken from torchvision, which is essentially its package. In terms of size, it was agreed to take ResNet50, because it is the minimum model with available for use Transfer Learning, which best suits the smallest EfficientNet V2 model of type B0 by the number of network parameters.

Before training, standard neural network data preparation was performed: 3-channel integer array (components of the color) in the range [0; 255] converted to the corresponding tensor of floating numbers in the range [0.0; 1.0]. For better use of Transfer Learning, input tensors were also log-normalized, because the previous training on ImageNet was also performed with such a transformation.

Since the calculations on the video card are much faster than on the processor, the NVidia Tesla P100 video card on the Kaggle platform, which has 16 GB of internal memory, was used for training. It is the main indicator that has affected the size of tensor batches that are simultaneously submitted for training – 128. Using a higher value creates an overflow of memory, and a lower one potentially reduces accuracy.

Under these conditions, the neural network ResNet50 was used for 3 experiments:

- the first had a model with random weights and passed a maximum of 100 epochs, the coefficient of learning speed was 0.0001;
- the second used the principle of Transfer Learning with a pre-trained model, but the weights were frozen before (the calculation of gradients is disabled) in all layers, except the classification and training of a maximum of 100 epochs with the coefficient of learning speed equals 0.0001, and then the weights are unfrozen and a maximum of 100 epochs are performed, the coefficient of learning speed is 0.00001 (10 times less);
- the third experiment also used the principle of Transfer Learning, but without manipulating the weights of the model and also training during a maximum of 100 epochs with the coefficient of learning speed of 0.0001.

### 4.2. Experiment Using CNN EfficientNet V2

In the second part of the experiment, we used EfficientNet V2 B0 network to classify the images of birds. Unfortunately, the weights of this network had not been ported yet by the time we were performing the experiment, so we decided to use the TensorFlow implementation of this model. We also chose the B0

version of EfficientNet V2 because our input images have the 224 pixel width and height, which is the recommended input size for the EfficientNet V2 B0 model. The TensorFlow implementation of the model uses the Keras library, which already includes the training loop and this makes the experiment iteration speed much faster.

The preprocessing that we did for the images before the training is rather simple. We divided the pixel values by 255 making them fit inside the [0, 1] interval, and after that we normalized them to have zero mean and standard deviation of 1 using the ImageNet means and standard deviations.

We performed the training using the computational resources provided by Kaggle. Kaggle provides NVidia Tesla P100 GPU with 16 GB of VRAM. This was enough to use the batch size of 64 images per batch. We performed three experiments. In the first experiment, we trained the model from scratch using random weights initialization. The batch normalization layers were in training mode while training. We trained the model using the early stopping callback with the patience of 3, and the training stopped after 10 epochs. The experiment took 27 minutes and 42 seconds to complete.

In the second experiment, we firstly trained the model using the pre-trained weights with all of the layers frozen except for the last fully connected layer. We also used the early stopping callback. After the end of the first phase of the training, we unfroze all layers and performed the fine-tuning using the small learning rate. As in the first phase of the experiment, we also used the early stopping callback. The experiment took 1 hour and 35 minutes to complete.

In the third experiment, we also used pre-trained weights, but this time we trained the entire model, without freezing any layers. The experiment took 17 minutes and 38 seconds to complete.

In all three experiments we used the Adam optimizer with the learning rate of  $1e-3$ .

To evaluate the performance of the models, we used the precision, recall and f1 score metrics. We used the scikit-learn implementation of these metrics. All metrics were calculated on the test subset of the dataset.

## 5. Results

Consider the experiment results using convolutional neural networks mentioned above.

### 5.1. Experiment Results Using CNN ResNet

The results of the experiments are shown in Figure 3 – Figure 6, respectively. Each table occupies 7 pages, so in Figure 3 – Figure 6 only the initial and final parts of each table of results are shown. In order not to overload the article with data, these three tables are completely given in electronic form [35].

In three experiments, the classifier layer in models was previously replaced in order to give probabilities of 300 classes instead of 1000 (2048), which is the default.

The quality of the model was characterized by the function of cross-entropy losses, and the use of gradients relied on the Adam optimizer with a learning speed coefficient of 0.0001. It was decided not to use the scheduler due to the availability of appropriate internal mechanisms in Adam.

The training procedure was performed on only 70% of the data set to obtain real indicators during model validation and final testing. It was also planned to stop training prematurely based on changes in the indicators of the loss function over the results of the validation part of the entire data set (10%): if a lower value of loss function is not achieved within three epochs, the training ends with the restoration of the model with the lowest value of the loss function on validation. During the final test, the quality characteristics of the models were no longer determined by the loss function, but a classification report containing metric values f1 (macro and weighted) and accuracy. Also for each class you can see precision, recall and f1-score on the support pictures of the class in the test dataset. As a result of experiments, you can see the effectiveness of the Transfer Learning approach. It is necessary to follow the procedure of initial training of one classifier of the model, and then perform training after unfreezing all weights.

Duration of training in the first experiment is 30 minutes, the second before unfreezing continued for 37 minutes and after thawing another 16 minutes used (53 minutes in total for the experiment), and the third experiment lasted for 23 minutes. The model trained for 3 minutes per epoch, but when all layers except the classifier were frozen, it was reduced to 70 seconds. In order to compare speed and efficiency of different approaches we can see loss function value by epoch on Figure 7 – Figure 9.

class	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	1.0000	0.3462	0.5143	26
AFRICAN FIREFINCH	0.9167	0.3235	0.4783	34
ALBATROSS	0.2857	0.0741	0.1176	27
ALEXANDRINE PARAKEET	0.9545	0.4884	0.6462	43
AMERICAN AVOCET	0.9000	0.5455	0.6792	33
AMERICAN BITTERN	0.2366	0.8158	0.3669	38
AMERICAN COOT	0.5417	0.5000	0.5200	26
AMERICAN GOLDFINCH	0.9412	0.4444	0.6038	36
AMERICAN KESTREL	0.2558	0.3667	0.3014	30
AMERICAN PIPIT	0.5556	0.4054	0.4688	37
AMERICAN REDSTART	0.4737	0.5455	0.5070	33
ANHINGA	0.3077	0.6897	0.4255	29
ANNAS HUMMINGBIRD	0.8421	0.5926	0.6957	27
ANTBIRD	0.4615	0.1714	0.2500	35
ARARIPE MANAKIN	0.9394	0.9118	0.9254	34
ASIAN CRESTED IBIS	0.8571	0.4800	0.6154	25
BALD EAGLE	0.7037	0.5135	0.5938	37
BALD IBIS	0.1392	0.7097	0.2328	31
BALI STARLING	1.0000	0.1471	0.2564	34
BALTIMORE ORIOLE	0.3544	0.9333	0.5138	30
BANANAQUIT	0.6129	0.5938	0.6032	32
BANDED BROADBILL	0.9310	0.5625	0.7013	48
...	...	...	...	...
TURKEY VULTURE	0.3333	0.1154	0.1714	26
TURQUOISE MOTMOT	0.8571	0.6667	0.7500	36
UMBRELLA BIRD	0.6667	0.6667	0.6667	27
VARIED THRUSH	0.7317	0.6383	0.6818	47
VENEZUELIAN TROUPIAL	0.7857	0.4074	0.5366	27
VERMILION FLYCATHER	0.6486	0.6000	0.6234	40
VICTORIA CROWNED PIGEON	0.2333	0.8750	0.3684	16
VIOLET GREEN SWALLOW	0.7619	0.6957	0.7273	46
VULTURINE GUINEAFOWL	0.6250	0.5405	0.5797	37
WALL CREEPER	0.7500	0.7742	0.7619	31
WATTLED CURASSOW	0.8000	0.1667	0.2759	24
WHIMBREL	0.2571	0.3750	0.3051	24
WHITE CHEEKED TURACO	0.6364	0.2188	0.3256	32
WHITE NECKED RAVEN	0.3636	0.2857	0.3200	28
WHITE TAILED TROPIC	0.7778	0.3256	0.4590	43
WHITE THROATED BEE EATER	0.7111	0.8000	0.7529	40
WILD TURKEY	0.7273	0.2857	0.4103	28
WILSONS BIRD OF PARADISE	0.5405	0.8000	0.6452	25
WOOD DUCK	0.9231	0.5333	0.6761	45
YELLOW BELLIED FLOWERPECKER	0.2941	0.6522	0.4054	23
YELLOW CACIQUE	0.7308	0.6786	0.7037	28
YELLOW HEADED BLACKBIRD	0.7222	0.8125	0.7647	32
accuracy			0.5539	9262
macro avg	0.6592	0.5573	0.5543	9262
weighted avg	0.6719	0.5539	0.5586	9262

**Figure 3:** Results of an experiment with random weights

class	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	1.0000	1.0000	1.0000	26
AFRICAN FIREFINCH	0.7714	0.7941	0.7826	34
ALBATROSS	0.8519	0.8519	0.8519	27
ALEXANDRINE PARAKEET	0.9318	0.9535	0.9425	43
AMERICAN AVOCET	1.0000	0.9697	0.9846	33
AMERICAN BITTERN	0.9474	0.9474	0.9474	38
AMERICAN COOT	0.9259	0.9615	0.9434	26
AMERICAN GOLDFINCH	0.8929	0.6944	0.7812	36
AMERICAN KESTREL	0.8667	0.8667	0.8667	30
AMERICAN PIPIT	0.9189	0.9189	0.9189	37
AMERICAN REDSTART	0.9231	0.7273	0.8136	33
ANHINGA	0.8750	0.9655	0.9180	29
ANNAS HUMMINGBIRD	0.7333	0.8148	0.7719	27
ANTBIRD	0.7436	0.8286	0.7838	35
ARARIPE MANAKIN	0.9429	0.9706	0.9565	34
ASIAN CRESTED IBIS	0.8800	0.8800	0.8800	25
BALD EAGLE	1.0000	1.0000	1.0000	37
BALD IBIS	0.6333	0.6129	0.6230	31
BALI STARLING	0.9375	0.8824	0.9091	34
BALTIMORE ORIOLE	0.7931	0.7667	0.7797	30
BANANAQUIT	0.9062	0.9062	0.9062	32
BANDED BROADBILL	0.9091	0.8333	0.8696	48
...	...	...	...	...
TURKEY VULTURE	0.8276	0.9231	0.8727	26
TURQUOISE MOTMOT	0.9714	0.9444	0.9577	36
UMBRELLA BIRD	0.9259	0.9259	0.9259	27
VARIED THRUSH	1.0000	1.0000	1.0000	47
VENEZUELIAN TROUPIAL	0.9545	0.7778	0.8571	27
VERMILION FLYCATHER	1.0000	0.7000	0.8235	40
VICTORIA CROWNED PIGEON	0.6842	0.8125	0.7429	16
VIOLET GREEN SWALLOW	0.9512	0.8478	0.8966	46
VULTURINE GUINEAFOWL	0.8919	0.8919	0.8919	37
WALL CREAPER	0.8824	0.9677	0.9231	31
WATTLED CURASSOW	0.7600	0.7917	0.7755	24
WHIMBREL	1.0000	0.9167	0.9565	24
WHITE CHEEKED TURACO	0.8077	0.6562	0.7241	32
WHITE NECKED RAVEN	0.7931	0.8214	0.8070	28
WHITE TAILED TROPIC	0.8889	0.9302	0.9091	43
WHITE THROATED BEE EATER	0.9474	0.9000	0.9231	40
WILD TURKEY	0.9310	0.9643	0.9474	28
WILSONS BIRD OF PARADISE	0.9200	0.9200	0.9200	25
WOOD DUCK	0.8333	0.8889	0.8602	45
YELLOW BELLIED FLOWERPECKER	0.6000	0.6522	0.6250	23
YELLOW CACIQUE	0.8065	0.8929	0.8475	28
YELLOW HEADED BLACKBIRD	0.8571	0.9375	0.8955	32
accuracy			0.8753	9262
macro avg	0.8772	0.8750	0.8737	9262
weighted avg	0.8808	0.8753	0.8758	9262

**Figure 4:** The results of the experiment with frozen weights before thawing

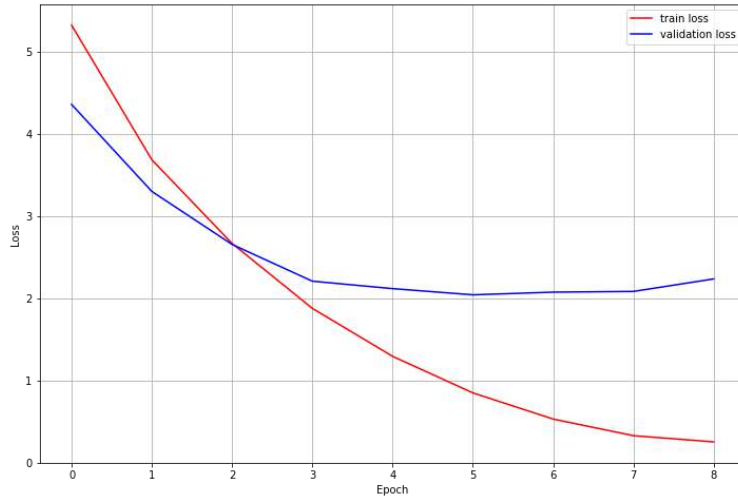


class	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	1.0000	1.0000	1.0000	26
AFRICAN FIREFINCH	0.8182	0.7941	0.8060	34
ALBATROSS	0.8214	0.8519	0.8364	27
ALEXANDRINE PARAKEET	0.9756	0.9302	0.9524	43
AMERICAN AVOCET	1.0000	0.9091	0.9524	33
AMERICAN BITTERN	0.9730	0.9474	0.9600	38
AMERICAN COOT	0.8333	0.9615	0.8929	26
AMERICAN GOLDFINCH	0.9000	0.7500	0.8182	36
AMERICAN KESTREL	0.9000	0.9000	0.9000	30
AMERICAN PIPIT	0.7955	0.9459	0.8642	37
AMERICAN REDSTART	0.8929	0.7576	0.8197	33
ANHINGA	0.9259	0.8621	0.8929	29
ANNAS HUMMINGBIRD	0.8000	0.8889	0.8421	27
ANTBIRD	0.8966	0.7429	0.8125	35
ARARIPE MANAKIN	1.0000	0.9706	0.9851	34
ASIAN CRESTED IBIS	0.8148	0.8800	0.8462	25
BALD EAGLE	0.8810	1.0000	0.9367	37
BALD IBIS	0.4828	0.4516	0.4667	31
BALI STARLING	0.8857	0.9118	0.8986	34
BALTIMORE ORIOLE	0.8889	0.8000	0.8421	30
BANANAQUIT	0.9062	0.9062	0.9062	32
BANDED BROADBILL	0.8462	0.9167	0.8800	48
...	...	...	...	...
TURKEY VULTURE	0.7576	0.9615	0.8475	26
TURQUOISE MOTMOT	0.9211	0.9722	0.9459	36
UMBRELLA BIRD	0.9259	0.9259	0.9259	27
VARIED THRUSH	1.0000	0.9574	0.9783	47
VENEZUELIAN TROUPIAL	0.9583	0.8519	0.9020	27
VERMILION FLYCATHER	0.9394	0.7750	0.8493	40
VICTORIA CROWNED PIGEON	0.9286	0.8125	0.8667	16
VIOLET GREEN SWALLOW	0.8913	0.8913	0.8913	46
VULTURINE GUINEAFOWL	0.8780	0.9730	0.9231	37
WALL CREAPER	0.8824	0.9677	0.9231	31
WATTLED CURASSOW	0.8000	0.6667	0.7273	24
WHIMBREL	0.9167	0.9167	0.9167	24
WHITE CHEEKED TURACO	0.7647	0.8125	0.7879	32
WHITE NECKED RAVEN	0.8621	0.8929	0.8772	28
WHITE TAILED TROPIC	0.9048	0.8837	0.8941	43
WHITE THROATED BEE EATER	1.0000	0.9500	0.9744	40
WILD TURKEY	0.8485	1.0000	0.9180	28
WILSONS BIRD OF PARADISE	0.9565	0.8800	0.9167	25
WOOD DUCK	0.8750	0.9333	0.9032	45
YELLOW BELLIED FLOWERPECKER	0.6667	0.7826	0.7200	23
YELLOW CACIQUE	0.9259	0.8929	0.9091	28
YELLOW HEADED BLACKBIRD	0.9091	0.9375	0.9231	32
accuracy			0.8835	9262
macro avg	0.8888	0.8838	0.8829	9262
weighted avg	0.8919	0.8835	0.8845	9262

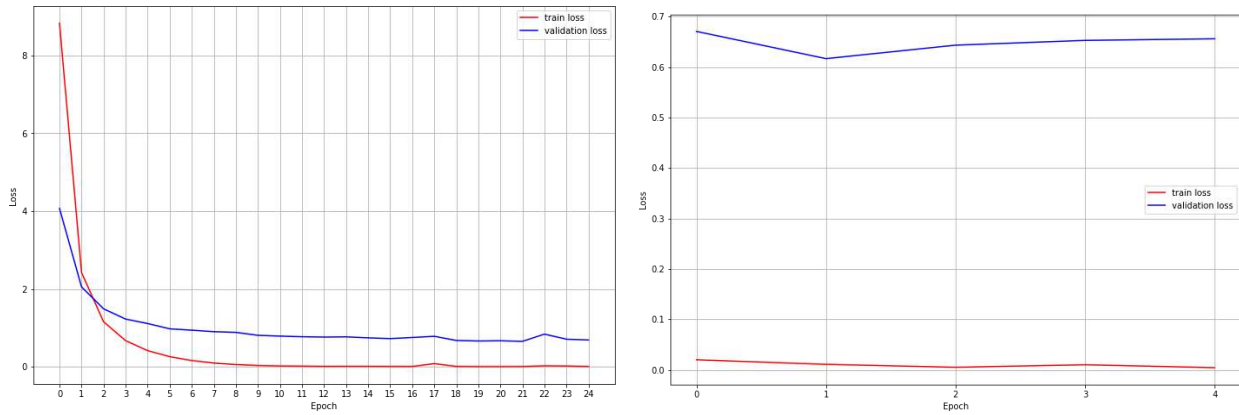
**Figure 5:** The results of the experiment with frozen weights after training after thawing

class	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	0.9259	0.9615	0.9434	26
AFRICAN FIREFINCH	0.7500	0.8824	0.8108	34
ALBATROSS	0.9231	0.4444	0.6000	27
ALEXANDRINE PARAKEET	1.0000	0.8140	0.8974	43
AMERICAN AVOCET	1.0000	1.0000	1.0000	33
AMERICAN BITTERN	0.7308	1.0000	0.8444	38
AMERICAN COOT	0.9231	0.9231	0.9231	26
AMERICAN GOLDFINCH	0.8889	0.6667	0.7619	36
AMERICAN KESTREL	0.9259	0.8333	0.8772	30
AMERICAN PIPIT	0.8462	0.8919	0.8684	37
AMERICAN REDSTART	0.9118	0.9394	0.9254	33
ANHINGA	0.7429	0.8966	0.8125	29
ANNAS HUMMINGBIRD	1.0000	0.7407	0.8511	27
ANTBIRD	0.8000	0.8000	0.8000	35
ARARIPE MANAKIN	0.9706	0.9706	0.9706	34
ASIAN CRESTED IBIS	0.9167	0.8800	0.8980	25
BALD EAGLE	0.9714	0.9189	0.9444	37
BALD IBIS	0.5122	0.6774	0.5833	31
BALI STARLING	1.0000	0.8824	0.9375	34
BALTIMORE ORIOLE	0.9000	0.9000	0.9000	30
BANANAQUIT	0.7561	0.9688	0.8493	32
BANDED BROADBILL	0.5974	0.9583	0.7360	48
...	...	...	...	...
TURKEY VULTURE	0.8462	0.8462	0.8462	26
TURQUOISE MOTMOT	0.7447	0.9722	0.8434	36
UMBRELLA BIRD	0.9286	0.9630	0.9455	27
VARIED THRUSH	0.9787	0.9787	0.9787	47
VENEZUELIAN TROUPIAL	0.9167	0.8148	0.8627	27
VERMILION FLYCATHER	0.9667	0.7250	0.8286	40
VICTORIA CROWNED PIGEON	1.0000	0.6875	0.8148	16
VIOLET GREEN SWALLOW	0.9302	0.8696	0.8989	46
VULTURINE GUINEAFOWL	0.9032	0.7568	0.8235	37
WALL CREAPER	1.0000	0.9677	0.9836	31
WATTLED CURASSOW	0.8095	0.7083	0.7556	24
WHIMBREL	1.0000	0.7917	0.8837	24
WHITE CHEEKED TURACO	0.9286	0.4062	0.5652	32
WHITE NECKED RAVEN	0.6923	0.9643	0.8060	28
WHITE TAILED TROPIC	0.7838	0.6744	0.7250	43
WHITE THROATED BEE EATER	1.0000	0.8000	0.8889	40
WILD TURKEY	0.9600	0.8571	0.9057	28
WILSONS BIRD OF PARADISE	0.9130	0.8400	0.8750	25
WOOD DUCK	0.9375	1.0000	0.9677	45
YELLOW BELLIED FLOWERPECKER	0.7619	0.6957	0.7273	23
YELLOW CACIQUE	0.7353	0.8929	0.8065	28
YELLOW HEADED BLACKBIRD	0.9091	0.9375	0.9231	32
accuracy			0.8472	9262
macro avg	0.8697	0.8466	0.8471	9262
weighted avg	0.8729	0.8472	0.8495	9262

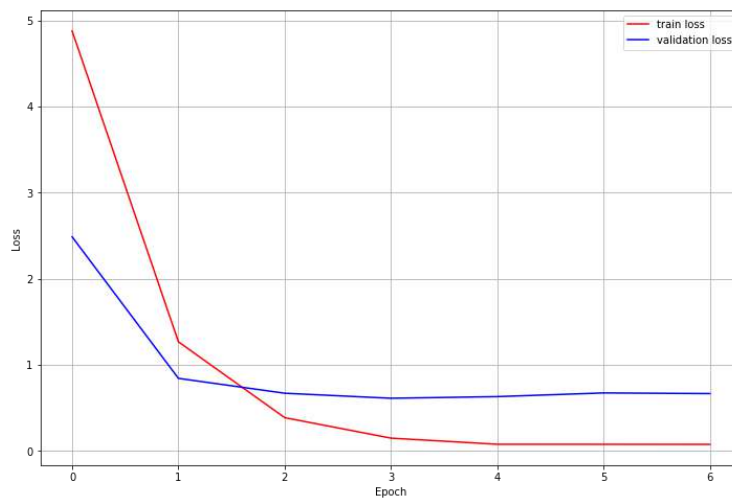
**Figure 6:** The results of the experiment with free weights



**Figure 7:** The value of the loss function in training and validation in the experiment with random weights



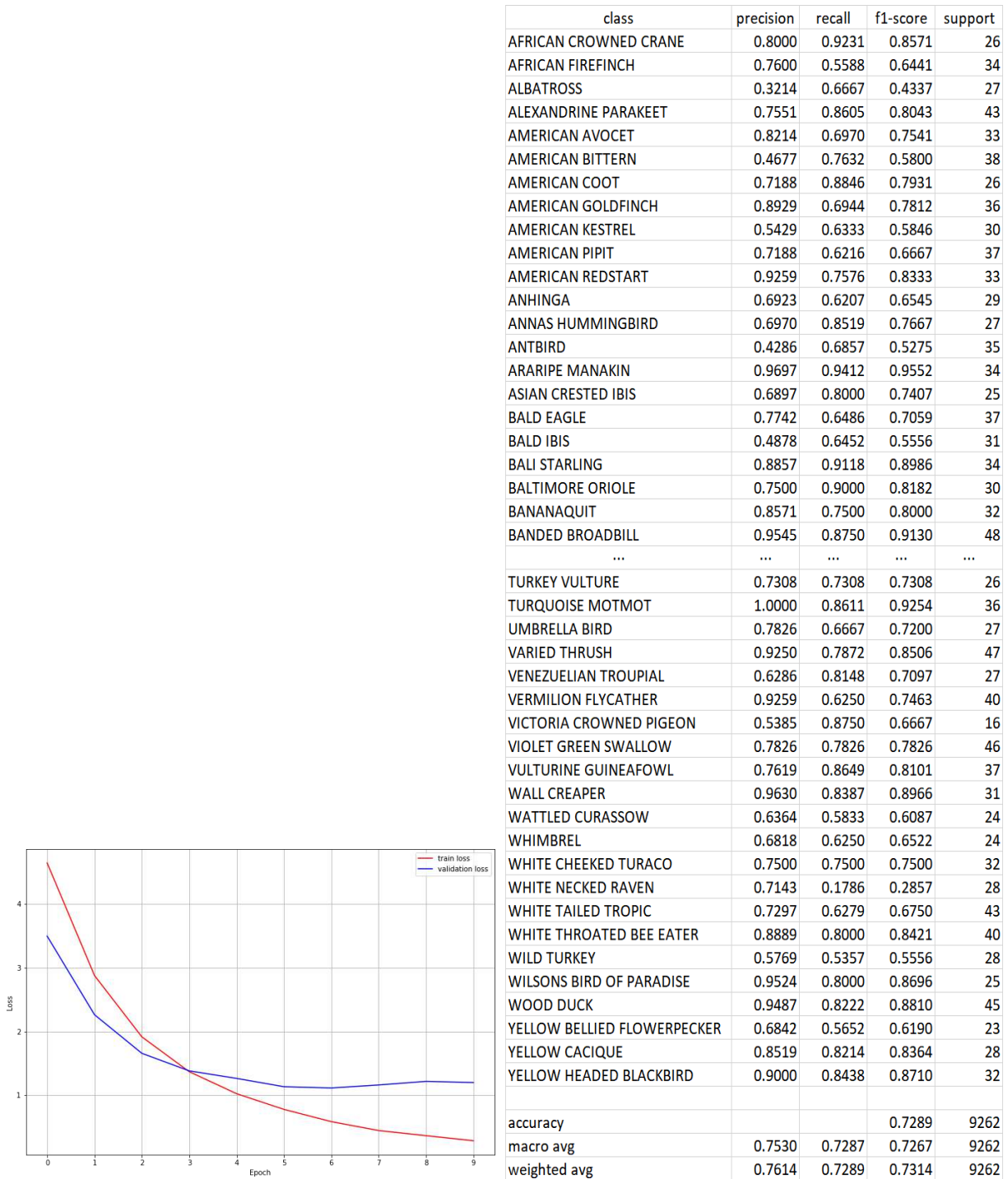
**Figure 8:** The value of the loss function during training and validation in the experiment with frozen weights (before and after unfreezing, respectively)



**Figure 9:** The value of the loss function in training and validation in the experiment with free weights

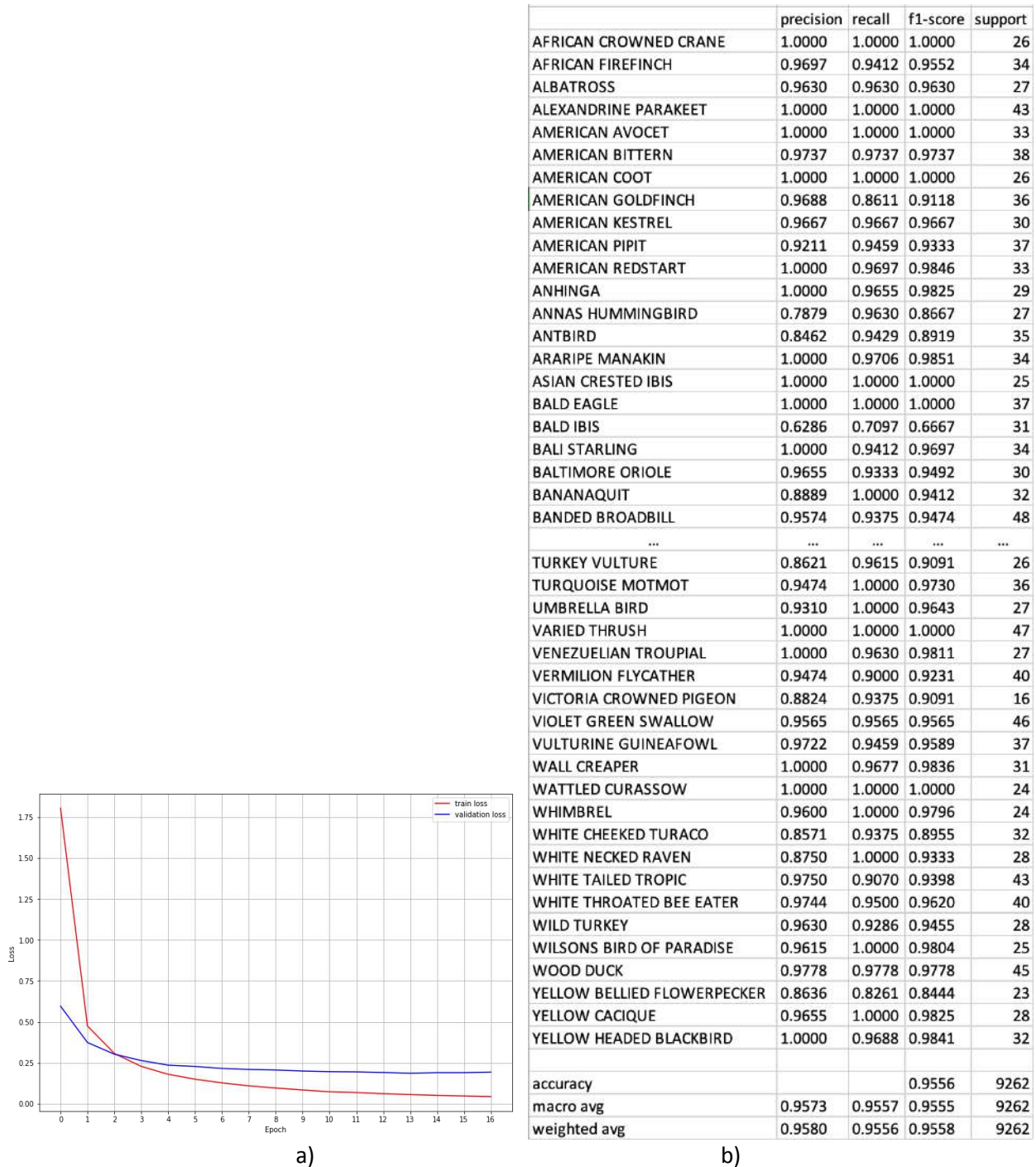
## 5.2. Experiment Results Using CNN EfficientNet V2

Similar to the previous subsection, the results of the experiments are shown in next figures. Below we present the ML curves and metrics of the first experiment based on model EfficientNet V2 (Figure 10).



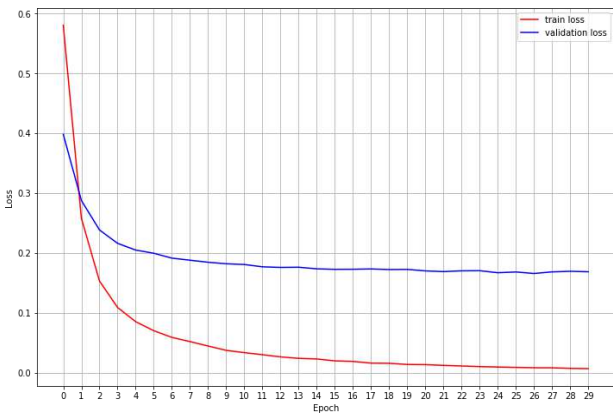
**Figure 10:** ML Result: The ML curves of the EfficientNet V2 B0 model training with random weights initialization (a), Metrics of the EfficientNet V2 B0 model with random weights initialization (b)

Below we present the learning curves and the metrics of the first stage of the second experiment based on model EfficientNet V2 (Figure 11).



**Figure 11:** ML Result: The learning curves of the EfficientNet V2 B0 first stage model training with transfer learning (a), metrics of the EfficientNet V2 B0 model with transfer learning after the first stage of training (b)

Below we present the learning curves and the metrics of the second stage of the second experiment based on model EfficientNet V2 (Figure 12).



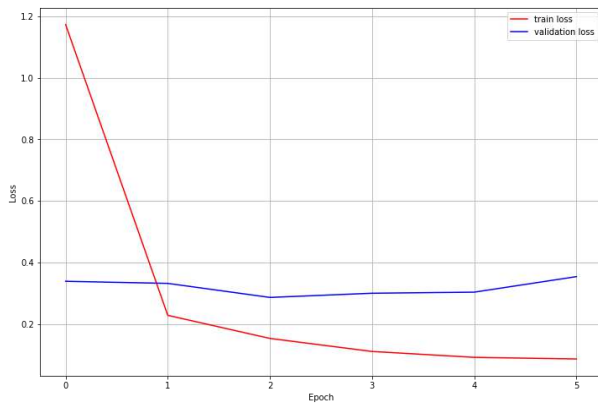
a)

	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	1.0000	1.0000	1.0000	26
AFRICAN FIREFINCH	0.9412	0.9412	0.9412	34
ALBATROSS	1.0000	1.0000	1.0000	27
ALEXANDRINE PARAKEET	1.0000	1.0000	1.0000	43
AMERICAN AVOCET	1.0000	1.0000	1.0000	33
AMERICAN BITTERN	0.9737	0.9737	0.9737	38
AMERICAN COOT	1.0000	1.0000	1.0000	26
AMERICAN GOLDFINCH	1.0000	0.9722	0.9859	36
AMERICAN KESTREL	0.9355	0.9667	0.9508	30
AMERICAN PIPIT	0.9737	1.0000	0.9867	37
AMERICAN REDSTART	1.0000	0.9394	0.9688	33
ANHINGA	1.0000	0.9655	0.9825	29
ANNAS HUMMINGBIRD	0.8125	0.9630	0.8814	27
ANTBIRD	0.8421	0.9143	0.8767	35
ARARIPE MANAKIN	1.0000	1.0000	1.0000	34
ASIAN CRESTED IBIS	1.0000	1.0000	1.0000	25
BALD EAGLE	0.9737	1.0000	0.9867	37
BALD IBIS	0.7857	0.7097	0.7458	31
BALI STARLING	1.0000	0.9706	0.9851	34
BALTIMORE ORIOLE	0.9310	0.9000	0.9153	30
BANANAQUIT	0.9412	1.0000	0.9697	32
BANDED BROADBILL	0.9375	0.9375	0.9375	48
...	...	...	...	...
TURKEY VULTURE	0.8889	0.9231	0.9057	26
TURQUOISE MOTMOT	1.0000	1.0000	1.0000	36
UMBRELLA BIRD	0.9000	1.0000	0.9474	27
VARIED THRUSH	1.0000	1.0000	1.0000	47
VENEZUELIAN TROUPIAL	0.9600	0.8889	0.9231	27
VERMILION FLYCATHER	0.9487	0.9250	0.9367	40
VICTORIA CROWNED PIGEON	0.8421	1.0000	0.9143	16
VIOLET GREEN SWALLOW	1.0000	0.9348	0.9663	46
VULTURINE GUINEAFOWL	0.9211	0.9459	0.9333	37
WALL CREEPER	1.0000	1.0000	1.0000	31
WATTLED CURASSOW	1.0000	1.0000	1.0000	24
WHIMBREL	1.0000	1.0000	1.0000	24
WHITE CHEEKED TURACO	0.8788	0.9062	0.8923	32
WHITE NECKED RAVEN	1.0000	1.0000	1.0000	28
WHITE TAILED TROPIC	0.9744	0.8837	0.9268	43
WHITE THROATED BEE EATER	0.9756	1.0000	0.9877	40
WILD TURKEY	1.0000	0.9643	0.9818	28
WILSONS BIRD OF PARADISE	0.9615	1.0000	0.9804	25
WOOD DUCK	0.9778	0.9778	0.9778	45
YELLOW BELLIED FLOWERPECKER	0.9130	0.9130	0.9130	23
YELLOW CACIQUE	0.9655	1.0000	0.9825	28
YELLOW HEADED BLACKBIRD	1.0000	1.0000	1.0000	32
accuracy			0.9639	9262
macro avg	0.9643	0.9644	0.9636	9262
weighted avg	0.9653	0.9639	0.9639	9262

b)

**Figure 12:** ML Result: The learning curves of the EfficientNet V2 B0 second stage model training with transfer learning (a), metrics of the EfficientNet V2 B0 model with transfer learning after the second stage of training (b)

Below we present the learning curves and the metrics of the third experiment based on model EfficientNet V2 (Figure 13).



a)

class	precision	recall	f1-score	support
AFRICAN CROWNED CRANE	1.0000	0.8846	0.9388	26
AFRICAN FIREFINCH	0.9118	0.9118	0.9118	34
ALBATROSS	1.0000	0.9259	0.9615	27
ALEXANDRINE PARAKEET	0.9556	1.0000	0.9773	43
AMERICAN AVOCET	0.9429	1.0000	0.9706	33
AMERICAN BITTERN	1.0000	0.9737	0.9867	38
AMERICAN COOT	0.9286	1.0000	0.9630	26
AMERICAN GOLDFINCH	0.8462	0.9167	0.8800	36
AMERICAN KESTREL	1.0000	0.8667	0.9286	30
AMERICAN PIPIT	0.9231	0.9730	0.9474	37
AMERICAN REDSTART	0.9394	0.9394	0.9394	33
ANHINGA	0.9333	0.9655	0.9492	29
ANNAS HUMMINGBIRD	0.9600	0.8889	0.9231	27
ANTBIRD	0.9677	0.8571	0.9091	35
ARARIPE MANAKIN	0.9714	1.0000	0.9855	34
ASIAN CRESTED IBIS	0.9600	0.9600	0.9600	25
BALD EAGLE	0.8537	0.9459	0.8974	37
BALD IBIS	1.0000	0.3226	0.4878	31
BALI STARLING	1.0000	1.0000	1.0000	34
BALTIMORE ORIOLE	0.9615	0.8333	0.8929	30
BANANAQUIT	0.9118	0.9688	0.9394	32
BANDED BROADBILL	0.9574	0.9375	0.9474	48
...	...	...	...	...
TURKEY VULTURE	0.8148	0.8462	0.8302	26
TURQUOISE MOTMOT	1.0000	0.9444	0.9714	36
UMBRELLA BIRD	1.0000	1.0000	1.0000	27
VARIED THRUSH	1.0000	0.9574	0.9783	47
VENEZUELIAN TROUPIAL	0.8621	0.9259	0.8929	27
VERMILION FLYCATCHER	0.9750	0.9750	0.9750	40
VICTORIA CROWNED PIGEON	0.8889	1.0000	0.9412	16
VIOLET GREEN SWALLOW	0.9773	0.9348	0.9556	46
VULTURINE GUINEAFOWL	0.9459	0.9459	0.9459	37
WALL CREEPER	0.9688	1.0000	0.9841	31
WATTLED CURASSOW	1.0000	0.8750	0.9333	24
WHIMBREL	1.0000	1.0000	1.0000	24
WHITE CHEEKED TURACO	0.7838	0.9062	0.8406	32
WHITE NECKED RAVEN	0.9286	0.9286	0.9286	28
WHITE TAILED TROPIC	0.8200	0.9535	0.8817	43
WHITE THROATED BEE EATER	1.0000	0.8500	0.9189	40
WILD TURKEY	0.9231	0.8571	0.8889	28
WILSONS BIRD OF PARADISE	0.9600	0.9600	0.9600	25
WOOD DUCK	0.9574	1.0000	0.9783	45
YELLOW BELLIED FLOWERPECKER	1.0000	0.7391	0.8500	23
YELLOW CACIQUE	0.9643	0.9643	0.9643	28
YELLOW HEADED BLACKBIRD	1.0000	0.9375	0.9677	32
accuracy			0.9263	9262
macro avg	0.9316	0.9256	0.9249	9262
weighted avg	0.9341	0.9263	0.9265	9262

b)

**Figure 13:** ML Result: The learning curves of the EfficientNet V2 B0 third experiment model training (a), metrics of the EfficientNet V2 B0 third experiment model performance (b)

## 6. Discussions

After analyzing the results (metrics) listed above, we can make the conclusion that the second experiment provides the best result. The average precision and recall metrics are considerably better than the metrics of the first experiment’s model. The second experiment also demonstrates better results than the third experiment, although the third experiment took much less time to complete (Figure 14). We can make a conclusion that freezing all layers but the last one helps with preventing overfitting. However, unfreezing all layers and performing fine-tuning produces little improvement. The overall metrics are improved by 1%. This may help in competitions, but in real world applications this improvement could be considered negligible and not worth the extra computing power spent on the model fine tuning. This effect (conclusion) is valid for the convolutional neural networks ResNet and EfficientNet V2 considered in the work. An analysis of the current state of the issue in relation to the results of machine learning of a number of other modern convolutional neural networks allows us to generalize the above conclusion.

	class	precision	recall	f1-score	support
accuracy				0.7289	9262
macro avg		0.7530	0.7287	0.7267	9262
weighted avg		0.7614	0.7289	0.7314	9262

a)

	precision	recall	f1-score	support
accuracy			0.9556	9262
macro avg	0.9573	0.9557	0.9555	9262
weighted avg	0.9580	0.9556	0.9558	9262

b)

	precision	recall	f1-score	support
accuracy			0.9639	9262
macro avg	0.9643	0.9644	0.9636	9262
weighted avg	0.9653	0.9639	0.9639	9262

c)

	class	precision	recall	f1-score	support
accuracy				0.9263	9262
macro avg		0.9316	0.9256	0.9249	9262
weighted avg		0.9341	0.9263	0.9265	9262

d)

**Figure 14:** Effectiveness comparison of ML methods, where:

- (a) Metrics of the EfficientNet V2 B0 model with random weights initialization;
- (b) The metrics of the EfficientNet V2 B0 model with transfer learning after the first stage of training;
- (c) The metrics of the EfficientNet V2 B0 model with transfer learning after the second stage of training;
- (d) The metrics of the EfficientNet V2 B0 third experiment model performance.

## 7. Conclusions

The results of the experiments showed the high efficiency of learning convolutional neural networks based on the Transfer Learning approach. At the same time, two approaches proved to be the best: Approach 1: convolutional layers are frozen, only the classification layers of the neural network are trained; Approach 2: after the implementation of the first approach, the unfrozen convolutional layers are retrained. Comparing these two approaches, it can be noted that the first approach gives high accuracy and requires approximately 3 times less time. The second approach gives a slight increase in accuracy (within 1%) and requires time up to 3 times more to train the model. In this regard, for application development purposes, including training ensembles on conventional equipment, the first approach is sufficient.

The conducted experiments and analysis of the available open experimental data allow us to conclude that for the purposes of image classification, the efficiency of EffNet networks is significantly higher than the most well-known analogues.

Experiments with TensorFlow and PyTorch have shown that training the same models in different frameworks can give significantly different results. This may be due to the implementation of different versions of the network. In the experiments performed, this was largely observed for the ResNet 50 network. The accuracy differed by 0.1.

In general, based on the results of the experiments, we can conclude that for the maximum efficiency of training a given network, from a practical point of view, it is necessary to:



- split the dataset according to the 70-20-10 standards (because the splitting of many datasets does not meet these standards, which leads to a decrease in learning efficiency);
- perform standard data preparation for the network in relation to debasing, normalization and normalization of data;
- to train a neural network based on approach 1, or approach 2 (by default, it is enough to train the classification layers by freezing the convolutional layers) on TensorFlow and PyTorch to understand where the accuracy is higher; choose the best option.

Experiments have shown that for maximum learning efficiency, it is necessary to supply images with the recommended linear dimensions for a network of this type to the input of the neural network..

## 8. References

- [1] G. R. Djavanshir, X. Chen and W. Yang, "A Review of Artificial Intelligence's Neural Networks (Deep Learning) Applications in Medical Diagnosis and Prediction," in *IT Professional*, vol. 23, no. 3, pp. 58-62, 1 May-June 2021, doi: 10.1109/MITP.2021.3073665.
- [2] M. Arredondo-Velázquez, J. Diaz-Carmona, A. Barranco-Gutiérrez and C. Torres-Huitzil, "Review of prominent strategies for mapping CNNs onto embedded systems," in *IEEE Latin America Transactions*, vol. 18, no. 05, pp. 971-982, May 2020, doi: 10.1109/TLA.2020.9082927.
- [3] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [4] C. Lai, T. Liu, R. Mei, H. Wang and S. Hu, "The Cloud Images Classification Based on Convolutional Neural Network," 2019 International Conference on Meteorology Observations (ICMO), 2019, pp. 1-4, doi: 10.1109/ICMO49322.2019.9026121.
- [5] M. Frniak, P. Kamencay, M. Markovic, J. Dubovan and M. Dado, "Comparison of Vehicle Categorisation by Convolutional Neural Networks using MATLAB," 2020 ELEKTRO, 2020, pp. 1-4, doi: 10.1109/ELEKTRO49696.2020.9130238.
- [6] R. Zhao, H. Ng, W. Luk and X. Niu, "Towards Efficient Convolutional Neural Network for Domain-Specific Applications on FPGA," 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 2018, pp. 147-1477, doi: 10.1109/FPL.2018.00033.
- [7] Ian Goodfellow, Yoshua Bengio, Aaron Courville *Deep Learning*, 3rd. ed., MIT Press, 2016, 787p. DOI/ISBN: 0262035618.
- [8] S. Tripathi and R. Kumar, "Image Classification using small Convolutional Neural Network," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 483-487, doi: 10.1109/CONFLUENCE.2019.8776982.
- [9] Stanford Cars Dataset. URL: <https://www.kaggle.com/jessicali9530/stanford-cars-dataset>.
- [10] Traffic Signs Preproc. URL: <https://www.kaggle.com/valentynsichkar/traffic-signs-preprocessed>.
- [11] COVID-19 Detection and Classification. URL: <https://www.kaggle.com/c/siim-covid19-detection>.
- [12] Fashion Product. URL: <https://www.kaggle.com/paramaggarwal/fashion-product-images-small>.
- [13] ASL Alphabet. URL: <https://www.kaggle.com/grassknotted/asl-alphabet>.
- [14] Q. Fu, S. Wang, J. Wang, S. Liu and Y. Sun, "A Lightweight Eagle-Eye-Based Vision System for Target Detection and Recognition," in *IEEE Sensors Journal*, vol. 21, no. 22, pp. 26140-26148, 15 Nov.15, 2021, doi: 10.1109/JSEN.2021.3120922.
- [15] C. Ditzel and K. Dietmayer, "GenRadar: Self-Supervised Probabilistic Camera Synthesis Based on Radar Frequencies," in *IEEE Access*, vol. 9, pp. 148994-149042, 2021, doi: 10.1109/ACCESS.2021.3120202.
- [16] K. Smelyakov, D. Tovchyrchko, I. Ruban, A. Chupryna and O. Ponomarenko, "Local Feature Detectors Performance Analysis on Digital Image," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 644-648, doi: 10.1109/PICST47496.2019.9061331.

- [17] J. B. Cloete, T. Stander and D. N. Wilke, "Parametric Circuit Fault Diagnosis Through Oscillation-Based Testing in Analogue Circuits: Statistical and Deep Learning Approaches," in *IEEE Access*, vol. 10, pp. 15671-15680, 2022, doi: 10.1109/ACCESS.2022.3149324.
- [18] J. Sresakoolchai and S. Kaewunruen, "Integration of Building Information Modeling and Machine Learning for Railway Defect Localization," in *IEEE Access*, vol. 9, pp. 166039-166047, 2021, doi: 10.1109/ACCESS.2021.3135451.
- [19] A. L. C. Ottoni and M. S. Novo, "A Deep Learning Approach to Vegetation Images Recognition in Buildings: a Hyperparameter Tuning Case Study," in *IEEE Latin America Transactions*, vol. 19, no. 12, pp. 2062-2070, Dec. 2021, doi: 10.1109/TLA.2021.9480148.
- [20] R. Chellappa, S. Theodoridis and A. van Schaik, "Advances in Machine Learning and Deep Neural Networks," in *Proceedings of the IEEE*, vol. 109, no. 5, pp. 607-611, May 2021, doi: 10.1109/JPROC.2021.3072172.
- [21] F. Yang, "A CNN-Based Broad Learning System," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), 2018, pp. 2105-2109, doi: 10.1109/CompComm.2018.8780984.
- [22] W. Wang et al., "Anomaly detection of industrial control systems based on transfer learning," in *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 821-832, Dec. 2021, doi: 10.26599/TST.2020.9010041.
- [23] G. Schwartz and K. Nishino, "Recognizing Material Properties from Images," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 1981-1995, 1 Aug. 2020, doi: 10.1109/TPAMI.2019.2907850.
- [24] Bodyanskiy Ye., Peleshko D., Setlak G., Mulesa P. Adaptive multivariate hybrid neuro-fuzzy system and its on-board fast learning, in *Neurocomputing*, 2017, vol. 230, pp. 409-416 - <http://dx.doi.org/10.1016/j.neucom.2016.12.042>.
- [25] Dataset: 300 Bird Species. URL: <https://www.kaggle.com/gpiousenka/100-bird-species>.
- [26] Comprehensive Guide to MCM. URL: <https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd>.
- [27] O. Lemeshko, O. Yeremenko and A. M. Hailan, "Two-level method of fast ReRouting in software-defined networks," 2017 4th International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2017, pp. 376-379, doi: 10.1109/INFOCOMMST.2017.8246420.
- [28] G. Churyumov, V. Tokarev, V. Tkachov and S. Partyka, "Scenario of Interaction of the Mobile Technical Objects in the Process of Transmission of Data Streams in Conditions of Impacting the Powerful Electromagnetic Field," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 2018, pp. 183-186, doi: 10.1109/DSMP.2018.8478539.
- [29] Shubin, I., Kyrychenko, I., Goncharov, P., Snisar, S., "Formal representation of knowledge for infocommunication computerized training systems," 2017 IEEE 4th International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2017, pp. 287-291, doi: 10.1109/INFOCOMMST.2017.8246399.
- [30] K. Smelyakov, A. Chupryna, M. Hvozdiev and D. Sandrkin, "Gradational Correction Models Efficiency Analysis of Low-Light Digital Image," 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), 2019, pp. 1-6, doi: 10.1109/eStream.2019.8732174.
- [31] K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrchko and O. Ponomarenko, "Efficiency of image convolution," 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019, pp. 578-583, doi: 10.1109/CAOL46282.2019.9019450.
- [32] Rafael C. Gonzalez, Richard E. Woods *Digital Image Processing*, 4th. ed., Pearson/Prentice Hall, 2018, 1168p. DOI/ISBN: 9780133356724.
- [33] ResNet and ResNetV2. URL: <https://keras.io/api/applications/resnet>.
- [34] EffNet. URL: [https://tfhub.dev/google/imagenet/efficientnet\\_v2\\_imagenet21k\\_b2/feature\\_vector/2](https://tfhub.dev/google/imagenet/efficientnet_v2_imagenet21k_b2/feature_vector/2).
- [35] Result. URL: [https://drive.google.com/drive/folders/1vW77McxK56OmLeH4BRaT5t65CxJ8p\\_8](https://drive.google.com/drive/folders/1vW77McxK56OmLeH4BRaT5t65CxJ8p_8).