

Identification of Modern Facial Emotion Recognition Models

Kirill Smelyakov¹, Oleksandr Bohomolov¹, Maksym Kizitskyi¹, Anastasiya Chupryna¹

¹*Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*

Abstract

The paper is devoted to the problem of developing a generalized algorithm for the effective identification of computational intelligence models used to recognize emotions by a person's facial expression. To solve this problem, an actual dataset was selected, alternative recognition models, algorithms and machine learning technologies were identified, as well as performance indicators and metrics that are used in the course of a comparative analysis of the obtained results. A series of numerous experiments has been carried out in relation to the identification of the parameters of alternative models of neural networks that are used to recognize emotions and evaluate the effectiveness of their application. Based on a comparative analysis of the effectiveness of the results of experiments, a generalized algorithm for identifying emotions was formulated, as well as recommendations for the use of certain architectures of neural networks in the framework of the tasks of facial emotion recognition.

Keywords

Computer vision, facial emotion recognition, face recognition, convolutional neural network, transfer learning

1. Introduction

Facial emotion recognition (FER) is a quite new and fast growing area of computer vision. Its main task is to identify what kind of emotion a person feels, using his/her facial expression. As an area of computer vision, the use of neural networks looks quite promising for this task. Because convolutional networks show, good results in other tasks. Many of the public networks are pretrained. Therefore, the question of using transfer learning for FER task arises. This will reduce the uncertainty of researchers when choosing a machine learning model and significantly speed up and increase the effectiveness of experiments in the field of FER. Therefore, the issue of transferring skills from other problem areas is rather little studied and promising.

The aim of the work is to research the possibility of neural networks and transfer learning technology applying to FER problems.

The goals of the work are to choose a dataset, based on it, plan and perform experiments, the results of which will allow:

- to formulate an effective algorithm for neural network identification and usage within the framework of the FER task;
- to determine which architecture of neural networks is better to use as a backbone for FER tasks in different situations;
- to compare the effectiveness of face recognition based backbones with standard solutions for transfer learning.

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland
EMAIL: kyrylo.smelyakov@nure.ua (K. Smelyakov); oleksandr.bohomolov@nure.ua (O. Bohomolov); maksym.kizitskyi@nure.ua (M. Kizitskyi); anastasiya.chupryna@nure.ua (A. Chupryna)
ORCID: 0000-0001-9938-5489 (K. Smelyakov); 0000-0002-9539-8888 (O. Bohomolov); 0000-0001-9771-5771 (M. Kizitskyi); 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

Researches in recent years focus on facial emotion recognition (FER) task [1-3]. Such systems often supplement to face recognition systems (Azure Face API, Face, FaceReader, etc.) [4-6] and can be used in many situations, from customer satisfaction analysis, service at the checkout, to tracking emotions at a psychologist's appointment [7, 8], in perspective drone vision services [9], etc.

The most efficient approaches that use such networks as ResNet, AffectNet, MobileNet, etc. on facial emotion recognition (FER) task are described by researchers. To simplify the access to this information they organized special list [10].

On the other hand, it includes various forms of ensembling and stacking of neural networks. It gives a win in the quality of the classification of emotions, but this approach also has disadvantages. Firstly, the model itself becomes quite huge and heavy, and a lot of time is spent on predictions. Because of this, the application of models of this kind is very complicated on mobile devices or in real-time systems. Secondly, due to the presence of several neural networks, the process of maintaining them within the production system becomes more complicated, and the task of updating models while maintaining the logic of work becomes more difficult compared to a solution in the form of an end-to-end model. Therefore, the issue of developing a model, perhaps not as effective, but much more compact and easy to maintain for use in face recognition systems, remains relevant and open.

At the same time, a wide variety of machine learning models and algorithms, as well as a high degree of uncertainty in the application conditions, often create great difficulties in choosing an appropriate network architecture and tuning its parameters effectively [11-13].

Why are neural networks and transfer learning considered to solve FER problems?

In recent years, neural networks have become the standard tool in the area of computer vision [14, 15]. A large number of diverse architectural solutions (EfficientNet, ResNet, Yolov5, etc.) and machine learning methods have been proposed to solve the problems of image classification object detection, and recognition. Their performance is affected by the quality of images [16, 17], result of image segmentation [18, 19], the architecture and hyperparameter settings of neural networks [20]. Moreover, researches on the application of convolutions are carried out to improve the effectiveness of CNN application in the case of the optimization of convolution mask parameters, the number of layers and a number of other parameters [21, 22].

For the purposes of identifying the parameters of a neural network, a wide range of machine learning algorithms is currently used. One of the most effective is transfer learning [23]. Transfer learning (fine-tuning a neural network with pre-trained weights on a huge data set (for example ImageNet) to solve a specific problem) is widely used in all areas of computer vision and increases the quality of solving different kinds of problems [24, 25].

The main advantage of this approach is that, thanks to the pre-trained weights, the model transforms the input image into a smaller set of meaningful features. Because of this, the relief of the loss function is smoothed out and the models converge faster to its minimum. Also recently, in such a field as face recognition, SOTA technique is often used, where the model is trained to compress the image into a feature vector by which a person's face can be identified [26]. Which in turn is very similar to what transfer learning is used for. That's why we decided to compare classical transfer learning models with face recognition models in more detail. Besides, this domain was selected because this is quite a popular area and many pre-trained models are in the public access [27].

For models to benefit from pre-trained weights, the task must be related to the domain on which the models were trained.

The research results are important not only for FER services, but also for solving a great number of related tasks, including the development of effective integrated E-learning services, AI solutions [28], the development of ICT solutions, network solutions and security services [12, 29, 30]. In addition, if face recognition based models show advantages over standard approaches, it means that the use of face recognition learning approaches can improve the quality of transfer learning models in other areas, increase learning speed and allow using less data for training. And it will allow specialists to conduct more experiments and reduce the outlay of cloud learning services.

3. Methods and Materials

First of all, consider the data that will be used in further experiments, some other materials and methods proposed to solve the problem under consideration.

3.1. Dataset Description

In order to test our approach, we chose a quite well known data set FER2013 [31]. The 2013 Facial Expression Recognition dataset (FER2013) is a Kaggle dataset, introduced by Pierre-Luc Carrier and Aaron Courvill at the International Conference on Machine Learning (ICML) in 2013.

This dataset was chosen because it is in the public access. It also contains photographs of people of different age, gender, race, nationality, with different background and accessories (such as glasses, masks). It allows a better evaluation of the generalization abilities in emotions recognizing.

This dataset contains grayscale images of faces. Their size is 48x48 pixels. These images have been created using an automatic face registration so that the faces on them are centered and occupy nearly the same amount of space in each image. So when making a comparison, we assume that the images have already been preprocessed in advance, therefore we will not consider this issue within the framework of our paper. Each image is labeled with one of seven emotions from the following list: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral.

The Disgust expression has the minimal number of images – 547, while other labels have nearly 5,000 samples each. More detailed information is presented in Table 1.

Table 1

Number of pictures for each class

Labels	Angry	Surprise	Disgust	Neutral	Happy	Fear	Sadness	Total
Number of examples	4953	4002	547	6198	8989	5121	6077	35887

Figure 1 shows examples of randomly selected pictures from the data set. As we can see, both men and women of different ages (from babies to old people), of different nationalities and races are represented in the data set.

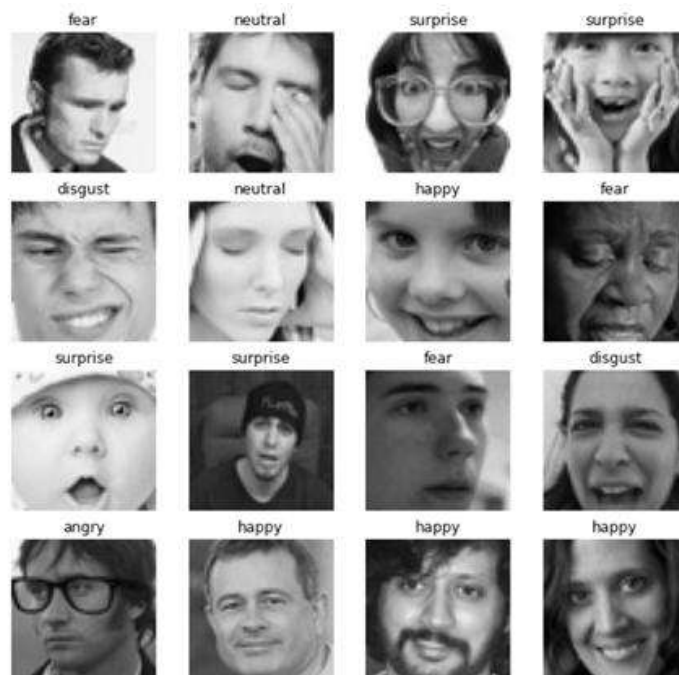


Figure 1: Examples of images [31]

In general, this data set provides a wide variety of face images, which will favorably affect the generalization ability of the model. However, it also has an imbalance of classes that is why the accuracy of determining the emotion of disgust will probably be lower in comparison with others.

To split the data set, a standard function from the sklearn package, `train_test_split`, was used. Training dataset - 70% (25,121 images). Validation dataset - 10% (3,589 image). Test dataset - 20% (7,177 images). The partition was stratified by emotion in the image with `random_state = 42`.

3.2. Methods

We have chosen as key metrics:

- accuracy on the training set;
- loss on the training set;
- accuracy on the validation set;
- accuracy on the validation set;
- mean convergence rate (MCR)

$$MCR = \frac{1}{n} \sum_{i=1}^n Metric_i - Metric_{i-1}, \quad (1)$$

where n – number of epochs; $Metric_i$ – performance metric on train data set during i 's epoch;

- mean overfitting rate (MOFR)

$$MOFR = \frac{1}{n} \sum_{i=1}^n (Metric_{train_i} - Metric_{val_i}) - (Metric_{train_{i-1}} - Metric_{val_{i-1}}), \quad (2)$$

where n – number of epochs; $Metric_{train_i}$ – performance metric on train data set during i 's epoch; $Metric_{val_i}$ – performance metric on validating data set during i 's epoch;

- initial accuracy – accuracy after training for 1 epoch. We chose this metric because it shows how well the pre-trained weights of the model fit the domain;
- initial loss – loss after training for 1 epoch.

In our experiment Metric will be accuracy and loss (categorical cross entropy).

4. Experiment

This section presents the plan of the experiment.

In order to evaluate the effectiveness of transfer learning, we will compare several popular architectures such as VGG-Face (Figure 2), OpenFace (Figure 3) which are neural networks trained for face recognition. Our hypothesis is that since the task of face recognition is in some way similar to FER, therefore, the weights of the networks will already contain the necessary features that will increase the learning performance. We also chose ResNet-50, MobileNet (Figure 4) pretrained on ImageNet dataset because they are the standard choice as a backbone in transfer learning. In these networks, the last layer was excluded, and all layers except the last 4 were frozen.

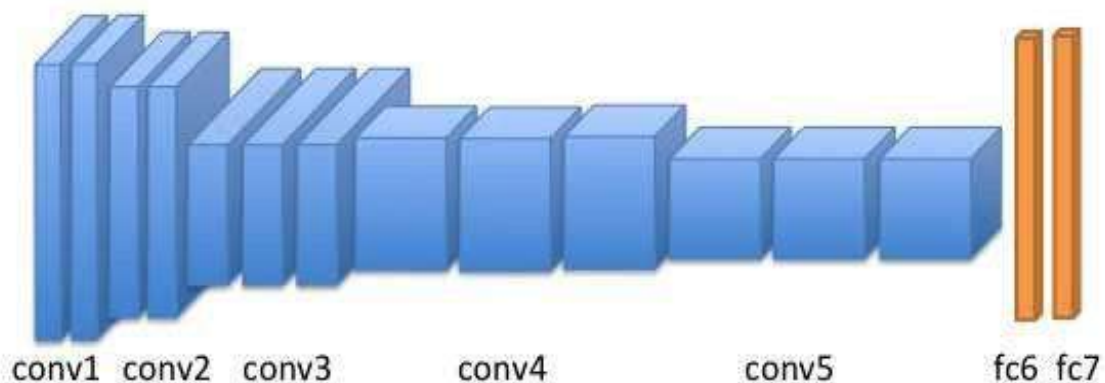


Figure 2: VGG-Face architecture [32]

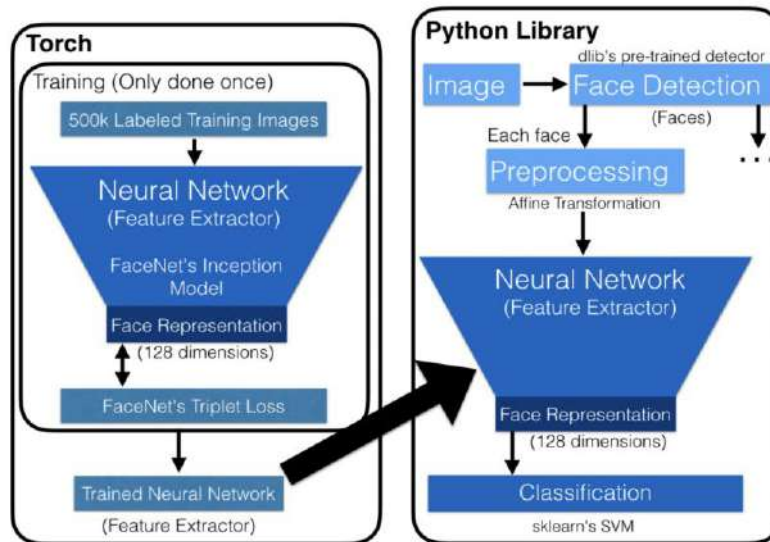


Figure 3: OpenFace architecture [33]

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 4: MobileNet-50 architecture [34]

The model structure of VGG-Face and OpenFace were loaded using deepface library [35]. The pretrained weights are available on [36-38]. ResNet-50, MobileNet were loaded using keras framfork [39]. Each model will be trained with a fixed set of hyperparameters such as the learning rate (10-4), the number of epochs is 20. Also, key metrics will be measured every 5 epochs. As a loss function we chose categorical cross entropy.

To compare the efficiency of transfer learning, we will train neural networks in 2 versions: with pre-trained weights and with randomly initialized weights. This approach will allow us to determine how and at what stages the pre-trained weights affect the efficiency of the model.

After the experiment we will find out in which model the pre-trained weights give the greatest value compared to random initialization, determine which model converges faster than others, is more resistant to overfitting and shows the highest accuracy.

Training will be carried out in the Google Colaboratory environment.

5. Results

The results of the experiments are presented in Figures 5 – Figure 8 and in Tables 2 – Table 5. The high resolution versions of all images are presented here [40].

5.1. ML Results

Figures 5 – Figure 8 show the accuracy and loss during training for MobileNet, OpenFace, ResNet-50, VGG-Face models. Each graph shows the results of the pretrained model (straight line) and the randomly initialized model (dashed line).

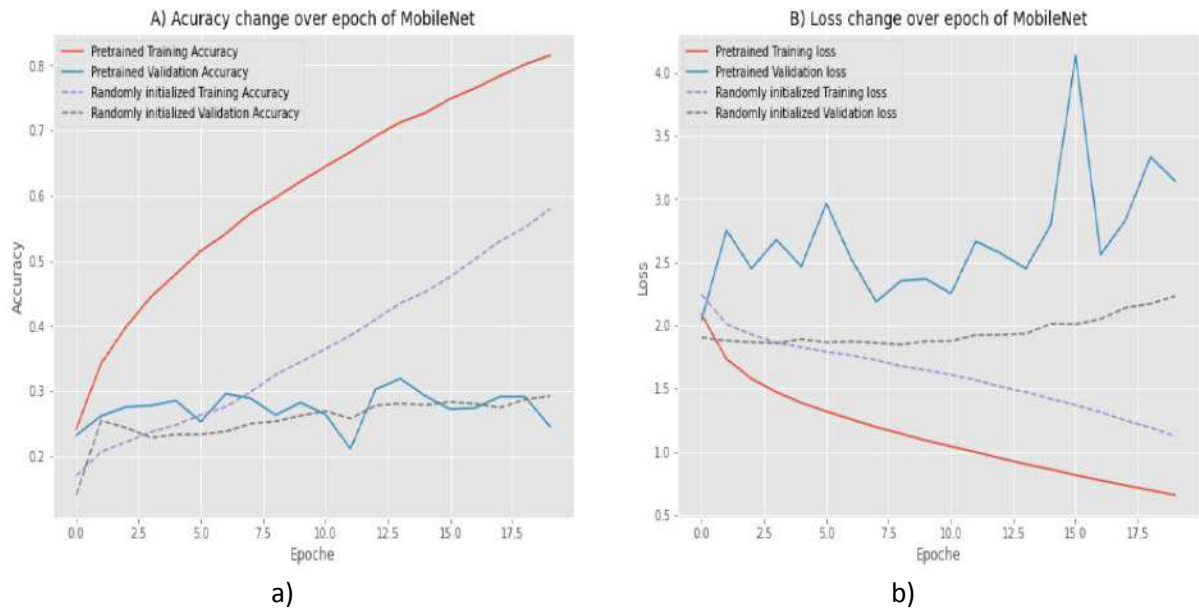


Figure 5: MobileNet training process: a) Accuracy change over epoch of MobileNet; b) Loss change over epoch of MobileNet

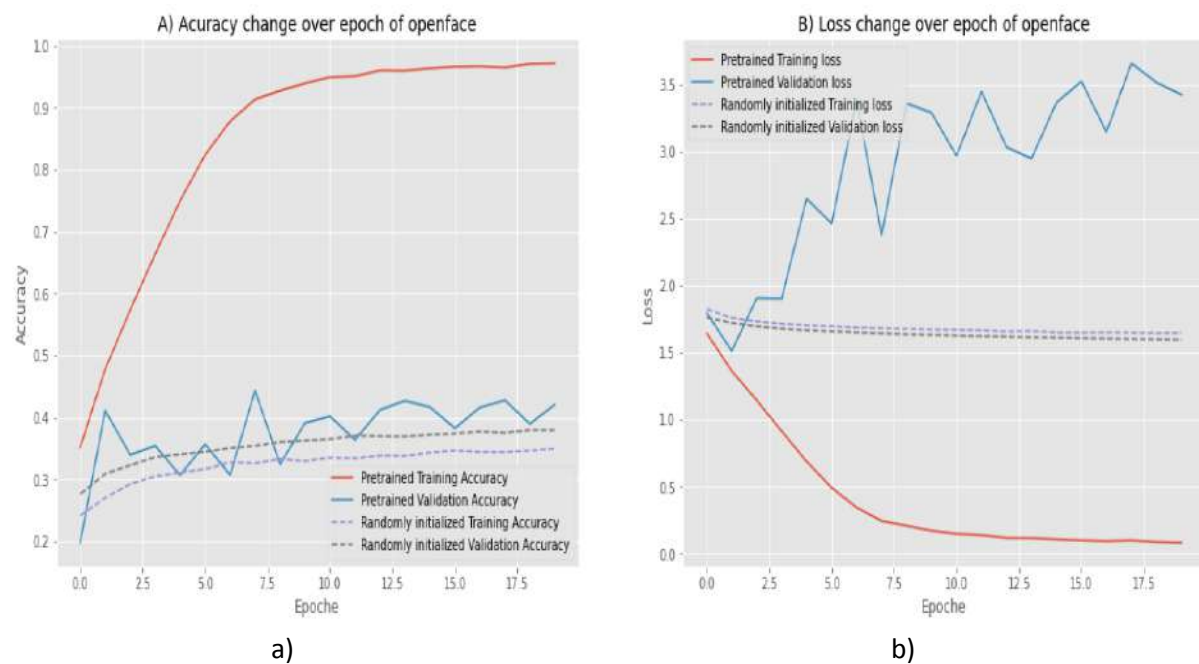
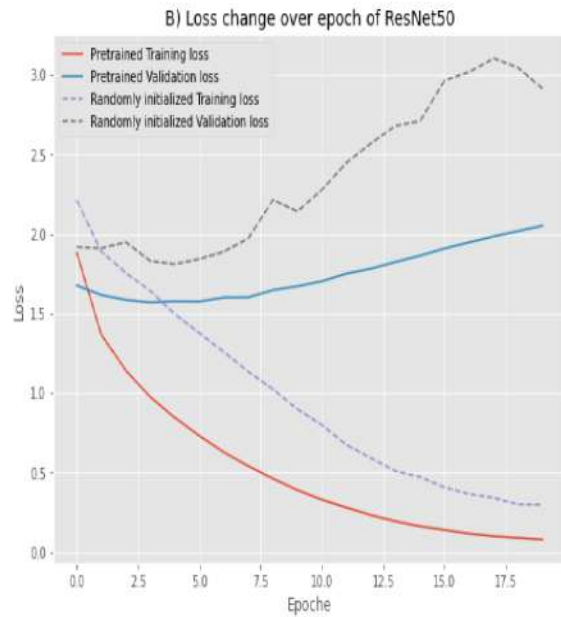
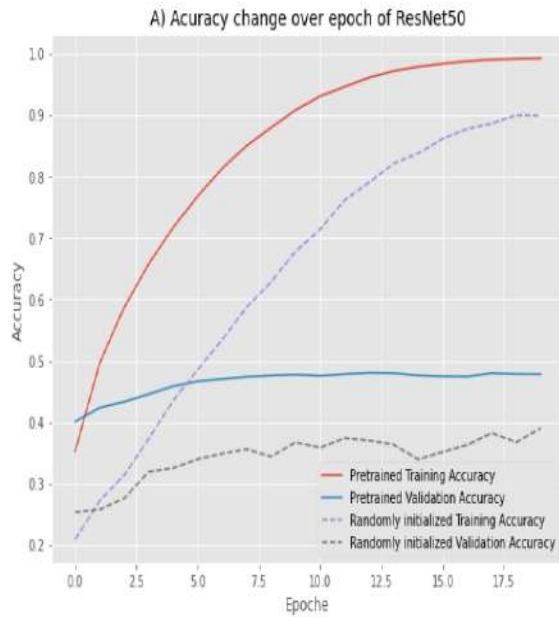


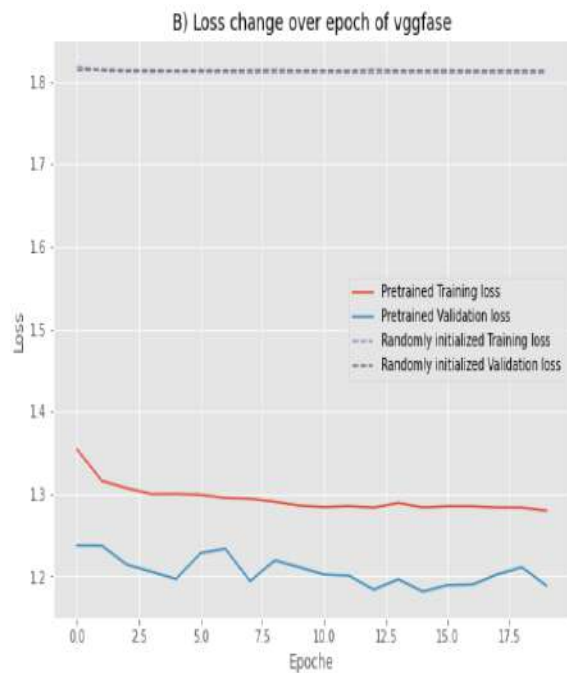
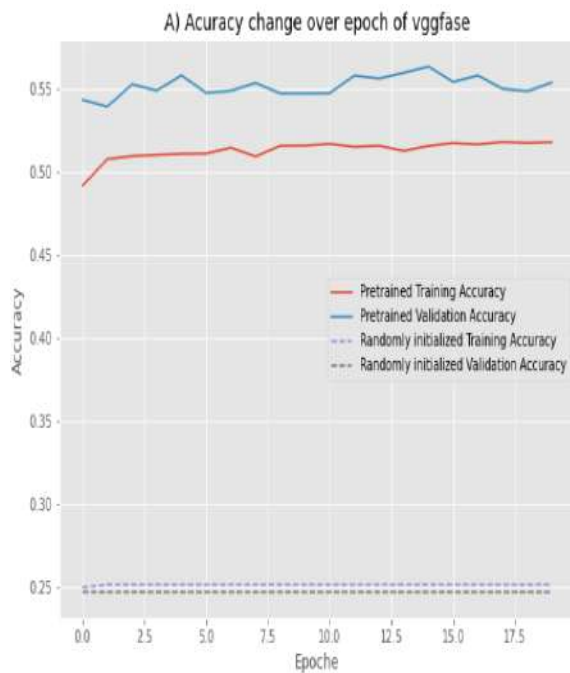
Figure 6: OpenFace training process: a) Accuracy change over epoch of OpenFace; b) Loss change over epoch of OpenFace



a)

b)

Figure 7: ResNet-50 training process: a) Accuracy change over epoch of ResNet-50; b) Loss change over epoch of ResNet-50



a)

b)

Figure 8: VGG-Face training process: a) Accuracy change over epoch of VGG-Face; b) Loss change over epoch of VGG-Face

Table 2 shows MCR (1) for each model based on accuracy. Table 3 shows MCR for each model based on loss.

Table 4 shows MOFR (2) for each model based on accuracy.

Table 5 shows MOFR for each model based on loss. It shows how much the difference between the metrics on the validation and training sets increases on average over the epoch, that is, how quickly the model over fits.

Table 2
Accuracy based MCR

Model	Epoch 1-5	Epoch 6-10	Epoch 11-15	Epoch 16-20
MobileNet_pretrained	0,039937	0,02512	0,019603	0,016789
MobileNet_random	0,013814	0,021909	0,023413	0,025798
openface_pretrained	0,080035	0,014538	0,003204	0,001637
openface_random	0,011508	0,001289	0,002027	0,0018
ResNet50_pretrained	0,063678	0,026667	0,008366	0,001567
ResNet50_random	0,052546	0,045559	0,023246	0,007233
vggfase_pretrained	0,001372	0,000111	0,000293	0,000453
vggfase_random	0	0	0	0

Table 3
Loss based MCR

Model	Epoch 1-5	Epoch 6-10	Epoch 11-15	Epoch 16-20
MobileNet_pretrained	-0,09537	-0,05194	-0,04445	-0,03934
MobileNet_random	-0,0492	-0,03906	-0,05088	-0,06358
openface_pretrained	-0,20339	-0,041	-0,00892	-0,00399
openface_random	-0,0145	-0,00418	-0,0031	-0,0008
ResNet50_pretrained	-0,14797	-0,06965	-0,0328	-0,01256
ResNet50_random	-0,12636	-0,11673	-0,06173	-0,02249
vggfase_pretrained	-0,00415	-0,00199	-3,9E-05	-0,00179
vggfase_random	-0,00034	-9,1E-05	-4E-05	-6,1E-05

Table 4
Accuracy based MOFR

Model	Epoch 1-5	Epoch 6-10	Epoch 11-15	Epoch 16-20
MobileNet_pretrained	0,033082	0,042063	0,007035	0,026356
MobileNet_random	0,017102	0,017952	0,018843	0,021943

openface_pretrained	0,100823	0,003225	-0,00725	5,79E-05
openface_random	0,003176	-0,00272	0,000745	0,001056
ResNet50_pretrained	0,054287	0,02505	0,009175	0,000267
ResNet50_random	0,034293	0,040487	0,025503	-0,00187
vggfase_pretrained	-0,00052	-0,00173	0,000293	0,001846
vggfase_random	0	0	0	0

Table 5
Loss based MOFR

Model	Epoch 1-5	Epoch 6-10	Epoch 11-15	Epoch 16-20
MobileNet_pretrained	-0,0504	-0,07935	-0,02334	-0,23275
MobileNet_random	-0,04781	-0,04951	-0,07605	-0,12429
openface_pretrained	-0,57781	-0,0532	0,051506	-0,09709
openface_random	-0,00039	0,001388	0,00041	0,002145
ResNet50_pretrained	-0,14471	-0,09968	-0,07212	-0,04759
ResNet50_random	-0,12195	-0,22848	-0,17572	0,012097
vggfase_pretrained	-0,00339	0,004537	0,002139	-0,00136
vggfase_random	-0,00013	-1,4E-05	-4,9E-05	-5E-05

Figures 9 – Figure 10 show accuracy and loss of the models after the first epoch of training.

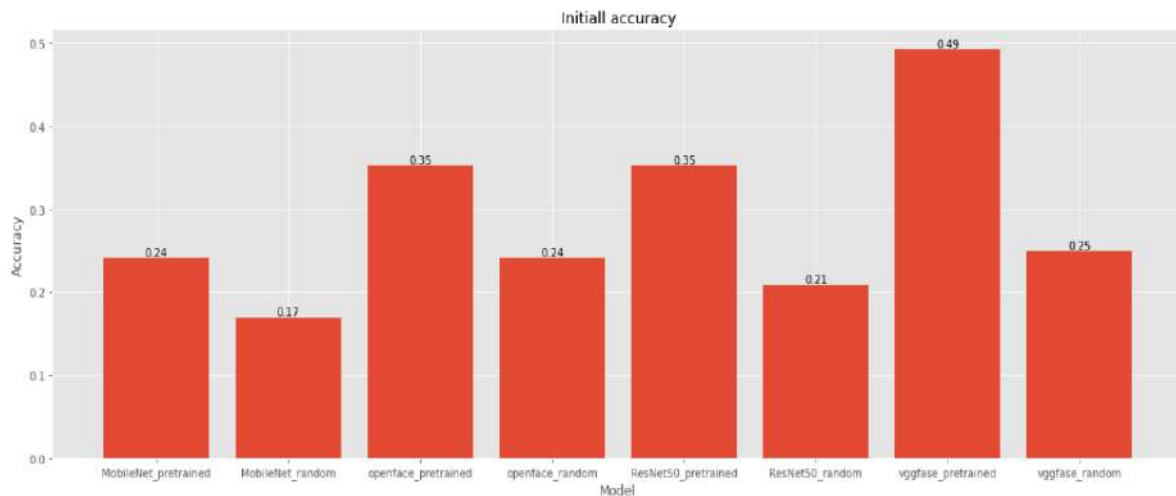


Figure 9: Initial accuracy

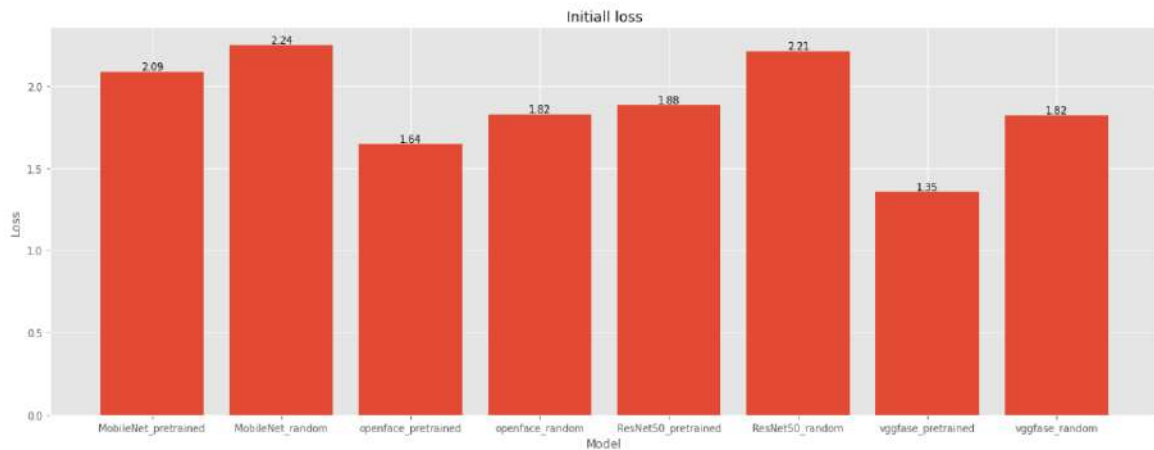


Figure 10: Initial loss

5.2. Testing Results

Figures 11 – Figure 17 show the results of image classification from a test set by various networks. An image with an emotion label is on the left, and a bar plot with neural network prediction is on the right.

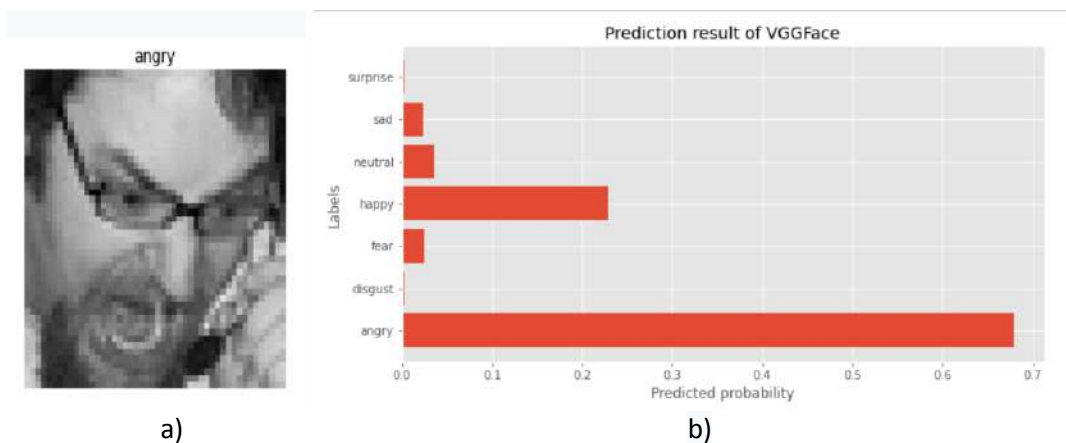


Figure 11: Classification result of emotion “angry”: a) An image example [26]; b) Predicted emotion probabilities

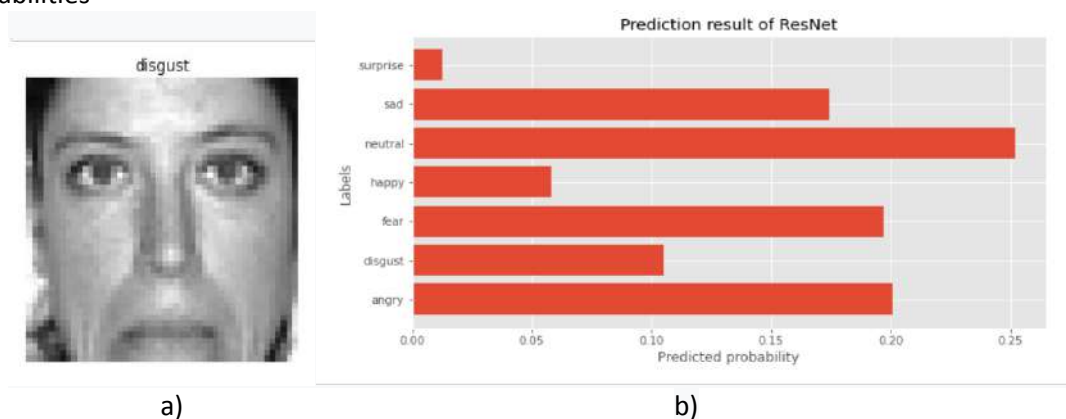
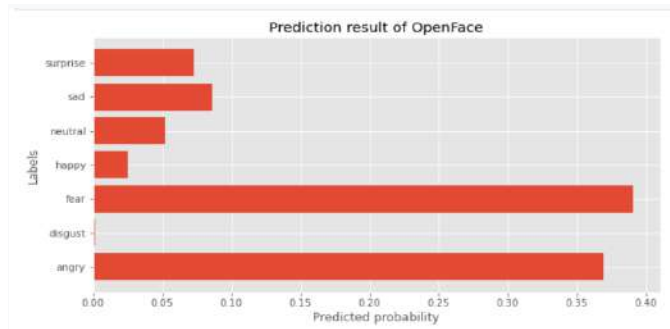


Figure 12: Classification result of emotion “disgust”: a) An image example [26]; b) Predicted emotion probabilities



a)

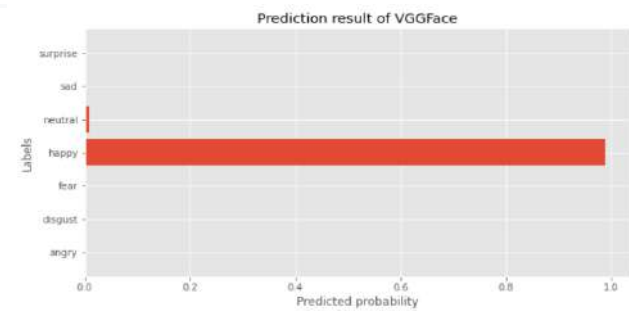


b)

Figure 13: Classification result of emotion “fear”: a) An image example [26]; b) Predicted emotion probabilities



a)

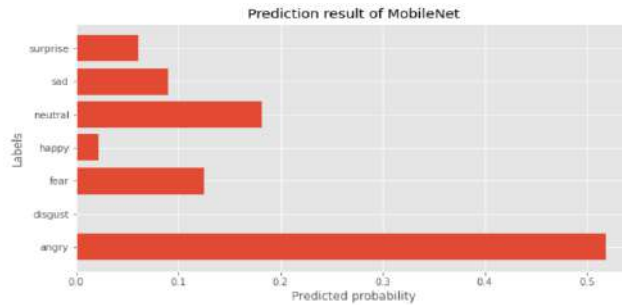


b)

Figure 14: Classification result of emotion “happy”: a) An image example [26]; b) Predicted emotion probabilities



a)

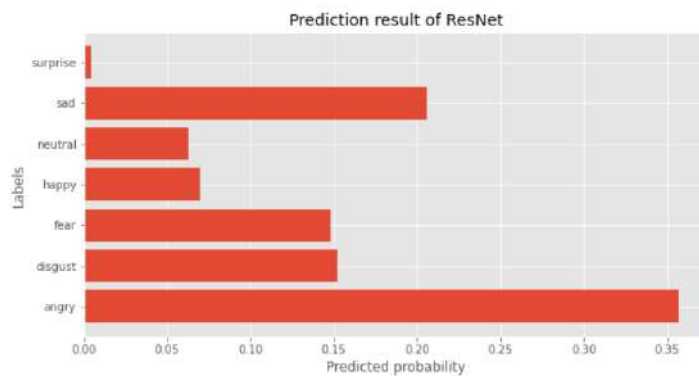


b)

Figure 15: Classification result of emotion “neutral”: a) An image example [26]; b) Predicted emotion probabilities



a)



b)

Figure 16: Classification result of emotion “sad”: a) An image example [26]; b) Predicted emotion probabilities

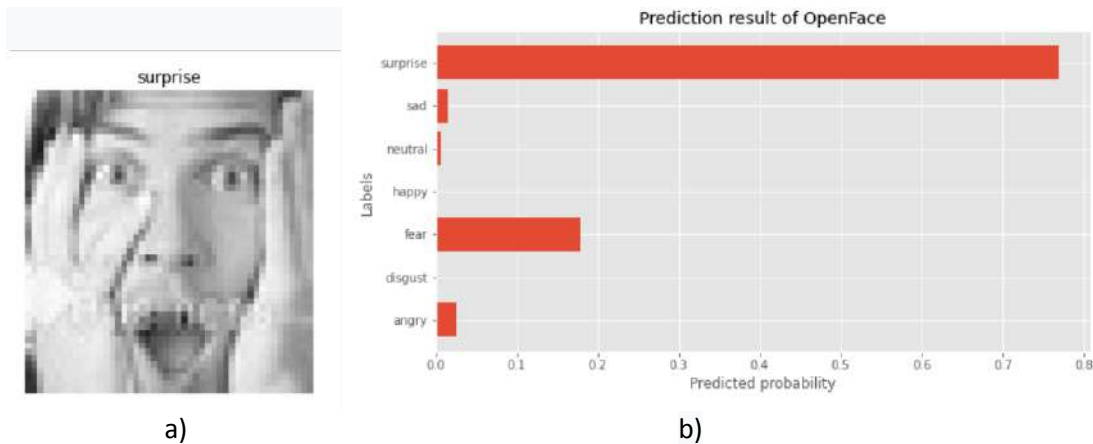


Figure 17: Classification result of emotion “surprise”: a) An image example [26]; b) Predicted emotion probabilities

6. Discussions

As a result of the experiment, it was revealed that pre-trained models showed better performance than randomly initialized ones in the FER task. Also, the pre-trained models had a higher average convergence rate at the first epochs (1-10), but then values became the same, in some cases, at epochs 15-20, the randomly initialized model converged faster. This is mainly due to the fact that the pre-trained model at that moment reached an accuracy of more than 0.8 and, accordingly, the quality increase slowed down. On the other hand, pre-trained models are more prone to overfitting, therefore, when using them, it is desirable to apply various regularization methods or data augmentation.

The best model in terms of initial and final accuracy on the validation set is VGGFace_pretrained. Therefore, its weights are initially best suited for the FER task. But in our experiment, this model had the worst performance in terms of convergence rate. That is why, for its training, other hyperparameters should be used, for example, to increase the learning rate or add more dense classification layers.

The second model for face detection - OpenFace - shows a level of accuracy comparable to the standard solutions in transfer learning - ResNet-50. But at the same time it has fewer parameters, therefore it fits and predicts faster. OpenFace has 3,743,280 parameters and ResNet-50 has 23,587,712 parameters. MobileNet has the fewest parameters (3,228,864), but its performance is lower than in OpenFace. Also, OpenFace has the highest convergence rate and overfitting rate in comparison with other models.

Thus, the face recognition based models proved to be at a fairly high level, in some cases even surpassing standard models like ResNet-50 and MobileNet.

As can be seen from Figures 8-14, such emotions as happiness, anger, fear, surprise are best recognized, and disgust is worst of all recognized. This is because this class is the least represented in the dataset. In addition, some pictures are rather controversially labeled (for example, pictures 12-13). In these examples neural networks show low confidence in the image class.

Based on the results of the experiment, the final learning algorithm was developed, which can be suggested to use in FER systems:

Preprocessing:

1) apply the face detection model to the image. You can use one of the pre-trained models, or train your own;

2) apply various augmentations to images. This will balance the classes (if the original dataset is unbalanced) and also increase the stability of the model on new data.

Training:

1) select a backbone model. If speed is more important within the task and there is enough data for training, we recommend choosing OpenFace. If the quality of recognition is more important and there are not enough resources for full model training, choose VGGFace;

2) freeze all layers of the neural network for training and add fully connected layers on top of them;

3) select hyper parameters and start the learning process with them.

7. Conclusions

As a result of the research the aim and goals of the work were reached. We formulated an effective algorithm for neural network identification and usage within the framework of the FER task; determined which architecture of neural networks was better to use as a backbone for FER tasks in different situations; compared the effectiveness of face recognition based backbones with standard solutions for transfer learning.

We found one of the most popular datasets on FER task – FER-2013. While analyzing its structure we found out that it was quite unbalanced. On the one hand it's a drawback, because models will learn how to distinguish minor class worse. But on the other hand it will show how models will work with real-world datasets that are often unbalanced.

Then we defined key metrics for analysis of networks performance during learning. Proposed metrics showed the efficiency of transfer learning for each architecture and determined what pre trained weights are most suitable for FER task and lead to faster convergence and less overfitting speed.

As part of this work, we organized an experiment and conducted a comparative analysis of the quality of the most popular neural network architectures for transfer learning (ResNet-50, MobileNet) with networks for face recognition (OpenFace, VGG-Face) within the FER task using various metrics. The obtained results show only general performance of the networks because they were all trained under the same conditions, and the best set of hyperparameters was not selected.

Based on the analysis of the experimental results, we recommend using the algorithm proposed in this article with a pretrained VGGFace. Also, under the condition of limited resources and the use of regularization methods, we recommend OpenFace as an alternative. But we also recommend specifically setting up the classifier for each specific task separately, because this will give a gain in quality.

For a deeper analysis of the effectiveness of neural networks, it is necessary to perform a deeper study, which is not the purpose of this work. It includes testing a larger class of architectures on a larger number of data sets, using various types of classifiers for embedding (including those not based on neural networks).

8. References

- [1] J. Guo et al., "Dominant and Complementary Emotion Recognition from Still Images of Faces," in *IEEE Access*, vol. 6, pp. 26391-26403, 2018, doi: 10.1109/ACCESS.2018.2831927.
- [2] H. Zhang and M. Xu, "Weakly Supervised Emotion Intensity Prediction for Recognition of Emotions in Images," in *IEEE Transactions on Multimedia*, vol. 23, pp. 2033-2044, 2021, doi: 10.1109/TMM.2020.3007352.
- [3] J. Li, S. Qiu, Y. -Y. Shen, C. -L. Liu and H. He, "Multisource Transfer Learning for Cross-Subject EEG Emotion Recognition," in *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3281-3293, July 2020, doi: 10.1109/TCYB.2019.2904052.
- [4] K. Smelyakov, A. Datsenko, V. Skrypka and A. Akhundov, "The Efficiency of Images Reduction Algorithms with Small-Sized and Linear Details," 2019 *IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, 2019, pp. 745-750, doi: 10.1109/PICST47496.2019.9061250.
- [5] L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in *IEEE Access*, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.
- [6] J. Zhao, S. Yan and J. Feng, "Towards Age-Invariant Face Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 474-487, 1 Jan. 2022, doi: 10.1109/TPAMI.2020.3011426.
- [7] N. -C. Ristea, L. C. Duțu and A. Radoi, "Emotion Recognition System from Speech and Visual Information based on Convolutional Neural Networks," 2019 *International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2019, pp. 1-6, doi: 10.1109/SPED.2019.8906538.

- [8] P. Partila, J. Tovarek, M. Voznak, J. Rozhon, L. Sevcik and R. Baran, "Multi-Classifer Speech Emotion Recognition System," 2018 26th Telecommunications Forum (TELFOR), 2018, pp. 1-4, doi: 10.1109/TELFOR.2018.8612050.
- [9] Tokariev V., Tkachov V., Ilina I., Partyka S. Implementation of combined method in constructing a trajectory for structure reconfiguration of a computer system with reconstructible structure and programmable logic // Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security", (ITS 2019), CEUR Workshop Processing, 28 Nov, 2019, pp. 71-81.
- [10] Facial Expression Rec. URL: <https://paperswithcode.com/task/facial-expression-recognition>.
- [11] K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrechko 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.
- [12] D. C. Nguyen et al., "Enabling AI in Future Wireless Networks: A Data Life Cycle Perspective," in IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 553-595, Firstquarter 2021, doi: 10.1109/COMST.2020.3024783.
- [13] S. Chaterji et al., "Lattice: A Vision for Machine Learning, Data Engineering, and Policy Considerations for Digital Agriculture at Scale," in IEEE Open Journal of the Computer Society, vol. 2, pp. 227-240, 2021, doi: 10.1109/OJCS.2021.3085846.
- [14] G. Cao, Y. Ma, X. Meng, Y. Gao and M. Meng, "Emotion Recognition Based On CNN," 2019 Chinese Control Conference (CCC), 2019, pp. 8627-8630, doi: 10.23919/ChiCC.2019.8866540.
- [15] Y. Tian, "Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm," in IEEE Access, vol. 8, pp. 125731-125744, 2020, doi: 10.1109/ACCESS.2020.3006097.
- [16] 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.
- [17] A. I. Wright, C. M. Dunn, M. Hale, G. G. A. Hutchins and D. E. Treanor, "The Effect of Quality Control on Accuracy of Digital Pathology Image Analysis," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 2, pp. 307-314, Feb. 2021, doi: 10.1109/JBHI.2020.3046094.
- [18] P. Yin, R. Yuan, Y. Cheng and Q. Wu, "Deep Guidance Network for Biomedical Image Segmentation," in IEEE Access, vol. 8, pp. 116106-116116, 2020, doi: 10.1109/ACCESS.2020.3002835.
- [19] G. Wang et al., "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1559-1572, 1 July 2019, doi: 10.1109/TPAMI.2018.2840695.
- [20] C. Nunes and F. Pádua, "A Convolutional Neural Network for Learning Local Feature Descriptors on Multispectral Images," in IEEE Latin America Transactions, vol. 20, no. 2, pp. 215-222, Feb. 2022, doi: 10.1109/TLA.2022.9661460.
- [21] N. Tian, Y. Liu, W. Wang and D. Meng, "Automatic CNN Compression Based on Hyperparameter Learning," 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533329.
- [22] L. Liao, Y. Zhao, S. Wei, Y. Wei and J. Wang, "Parameter Distribution Balanced CNNs," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4600-4609, Nov. 2020, doi: 10.1109/TNNLS.2019.2956390.
- [23] R. Gonzales-Martínez, J. Machacuay, P. Rotta and C. Chinguel, "Hyperparameters Tuning of Faster R-CNN Deep Learning Transfer for Persistent Object Detection in Radar Images," in IEEE Latin America Transactions, vol. 20, no. 4, pp. 677-685, April 2022, doi: 10.1109/TLA.2022.9675474.
- [24] X. Liu, W. Yu, F. Liang, D. Griffith and N. Golmie, "Toward Deep Transfer Learning in Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 15, pp. 12163-12175, 1 Aug. 1, 2021, doi: 10.1109/JIOT.2021.3062482.
- [25] S. Hussein, P. Kandel, C. W. Bolan, M. B. Wallace and U. Bagci, "Lung and Pancreatic Tumor Characterization in the Deep Learning Era: Novel Supervised and Unsupervised Learning

- Approaches," in IEEE Transactions on Medical Imaging, vol. 38, no. 8, pp. 1777-1787, Aug. 2019, doi: 10.1109/TMI.2019.2894349.
- [26] Deep Face Recognition: A Survey. URL: https://arxiv.org/pdf/1804.06655.pdf?source=post_page.
- [27] Deepface. URL: <https://github.com/serengil/deepface>.
- [28] Y. Lu, Q. Mao and J. Liu, "A Deep Transfer Learning Model for Packaged Integrated Circuit Failure Detection by Terahertz Imaging," in IEEE Access, vol. 9, pp. 138608-138617, 2021, doi: 10.1109/ACCESS.2021.3118687.
- [29] 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.
- [30] 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.
- [31] Learn facial expressions from an image. URL: <https://www.kaggle.com/msambare/fer2013>.
- [32] VGG-Face network architecture. URL: https://www.researchgate.net/figure/VGG-Face-network-architecture_fig2_319284653.
- [33] OpenFace architecture. URL: <https://www.cs.cmu.edu/~satya/docdir/CMU-CS-16-118.pdf>.
- [34] MobileNet-50 architecture. URL: <https://arxiv.org/pdf/1704.04861.pdf>.
- [35] OpenFace. URL: A general-purpose face recognition library with mobile applications: <http://reports-archive.adm.cs.cmu.edu/anon/2016/CMU-CS-16-118.pdf>.
- [36] VGGF. URL: https://drive.google.com/file/d/1CPSaum3HpopfomUEK1gybeuIVoeJT_Eo/view.
- [37] Openface. URL: https://drive.google.com/file/d/1LSe1YCV1x-BfNnfb7DFZTNpv_Q9jITxn/view.
- [38] ResNet and ResNetv2. URL: <https://keras.io/api/applications/resnet/#resnet50-function>.
- [39] Keras. URL: <https://keras.io/api/applications/mobilenet>.
- [40] All images. URL: https://docs.google.com/document/d/1Z_S_FpRkv4Xf2cRAqHxo23BUv7aYqtMZ59aJrpvYf-M/edit?usp=sharing.