Security
of Infocommunication Systems
and Internet of Things

2025 Vol 3, No 2

https://doi.org/10.31861/sisiot2025.2.02009

Received 30 July 2025; revised 16 November 2025; accepted 09 December 2025; published 30 December 2025

Face Detection and Identification Using Convolutional Neural Network and MobileNetV3 Model

Mykola Ilashchuk*

Department of Computer Systems Software, Institute of Physical, Technical, and Computer Sciences, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

*Corresponding author (E-mail: ilashchuk.mykola.m@chnu.edu.ua)

ABSTRACT This paper presents the results of a study of the effectiveness of applying the transfer learning methodology to the task of face detection and recognition, with a focus on processing images containing only one face. MobileNetV3 was chosen as the basic neural network architecture, which provides high performance with limited computing resources. The model was trained in two consecutive stages: the first is face recognition (detection) in photos, and the second is face identification from a face image. To ensure the unambiguity and purity of the training data, only images with one face were used. The training process combined the use of an open dataset for the initial stage of face detection with the proprietary photo set of students of Yuriy Fedkovych Chernivtsi National University for the identification phase. The model was trained and tested in the Google Colab cloud environment using an Nvidia Tesla T4 GPU. The neural network was implemented using the modern deep learning framework TensorFlow and our own program code written in Python. The model parameters were optimized by minimizing the loss function, which is the sum of the binary cross-entropy and the negative logarithm of the Intersection over Union metric, which characterizes the accuracy of determining the location of an object in an image. The built model was compared with previous approaches to face detection implemented on the basis of the OpenCV library. A comparative analysis by the metrics of recognition accuracy and processing time demonstrated the superiority of the developed system. The results obtained are of interest to researchers in the field of computer vision, automated recognition systems, and technologies for intelligent visual data processing.

KEYWORDS machine learning, computer vision, object detection, convolutional neural networks, transfer learning.

I. INTRODUCTION

Nowadays, the face recognition technologies, facial recognition is playing an increasingly important role in security, personal identification, access control automation, and personalized services. Given the rapid development of computer vision and deep learning, in particular convolutional neural networks, it is possible to significantly improve the accuracy and efficiency of face recognition algorithms. However, for the practical application of such systems on devices with limited computing resources, there is a need to use models that provide high performance while maintaining accuracy.

One of the approaches to strike a balance between performance and quality is transfer learning, which is the retraining of existing, pre-trained models for specific tasks and data sets.

The aim of this work is to implement and study an approach to face detection and recognition based on a convolutional neural network using the MobileNetV3 architecture, including face detection in an image. That is, determining the coordinates of the area in which the face is located and face recognition, specifically, identifying the person depicted in the photo by classifying the face among set of known faces.

II. DESCRIPTION OF THE METHODOLOGY USED

To accomplish the task of face detection and recognition, it was decided to apply the results of previous research in the field of computer vision, in particular, using Transfer Learning. It should be noted that the use of transfer learning is quite common in neural network training and provides a pre-designed and optimized network architecture, while changing its parameters and training the neural network for a specific data set. The MobileNetV3 model [1] was chosen for use because it is quite efficient and does not require significant computing power to train the model and perform predictions. For face recognition, we used a dataset of photos of university students who agreed to using their photos for training the neural network. However, due to the limited variety of photos, it was decided to train the neural network on an open dataset, performing only face detection, and then finally train the model for face recognition using the student photo dataset. For the sake of face certainty, we focused only on the application where there is only one face in the photo.

The transfer training of the model was performed with the TensorFlow framework [2], using the Mobilenet V3 model, which was pretrained on the Imagenet [3] dataset. The Keras-cv library [4] was used to work with this model in the Python programming language. The training was performed in the Google Colab online environment using a Nvidia Tesla T4 graphics card with 15 GB of video memory. The workspace also had 12.7 GB of RAM.

III. MOBILENET V3 NEURAL NETWORK

MobileNetV3 is a convolutional neural network (CNN) architecture designed for efficient image classification on mobile and embedded devices. It combines lightweight computation with high accuracy by integrating neural

network building blocks such as convolutions at different levels of model depth, squeeze-and-excitation modules [5], and hard-swish activation functions [6]. This model can be used as a basis for many tasks of object detection and recognition in the field of computer vision. It is worth noting that the MobileNetV3 model is presented in two modifications, namely Small and Large, which differ in efficiency and use of computing resources. It should be noted that the MobileNetV3-Large modification was used for this project. A schematic representation of the model layers is shown in Fig. 1 and is divided into two main parts:

1) MobileNetV3 (left) extracts important information from the image at several levels (in decreasing resolution: 1/4, 1/8, 1/16, etc.);

2) The Segmentation Head operator (right) is responsible for the implementation of the pixel classification itself:

- One branch performs 1x1 convolution, normalization, and activation (ReLU) to create a feature map of 128 channels;
- The next branch performs the Average Pool convolution, then 1x1 convolution and scaling;
- Both branches are multiplied to take into account the importance of each facial fragment.

The MobileNetV3-Large architecture can be logically divided into several key components, namely:

1) Feature extractor. This part of the model consists of several layers that "collapse" the image and extract important information from it. In the case of face detection, these layers find characteristic features, such as eyes, eyebrows, etc. This allows subsequent layers of the model to understand whether there is a face in the image;

2) The intermediate part of the neural network. In many computer vision systems, features obtained from different levels of the extractor are combined into a special structure - the so-called "Feature Pyramid Network". This allows the model to better recognize objects in the image, regardless of their size;

3) The detection part, which is responsible for detecting objects, consists of two blocks. The first is a regression block that determines where the object is located in the image (draws a rectangle around it). The second is the classification block, which determines what kind of object it is (face, car, or cat). This part is usually built from dense layers that work on the basis of data obtained from previous layers that extracted features. In such models, there are always two outputs: one for the position of the object (regression) and the other for its type (classification).

According to the two predictions of the model (namely, the location of the rectangle and the type of object), a loss function consisting of two parts respectively was applied. The goal of the algorithm is to minimize the model's prediction loss function, which is also a characteristic of the prediction's accuracy. The smaller the loss function, the better the model is trained.

To evaluate how well the model performs classification, we used the binary cross entropy metric. This is an adequate evaluation option for tasks in which it is necessary to determine one of two options: whether the model recognized the object correctly or incorrectly. This metric shows how similar the actual results is to the model's prediction, which is characterized by the probability value that the model gives for each object. In our case, there are two options for each object: correct identification or error. In accordance with the theoretical component of the MobileNetV3 model [1], the binary cross-entropy is calculated by the Eq. 1

$$L_{cl} = \frac{-1}{N}\sum_{i=1}^{N} y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i). \quad (1)$$

Here N is the number of dataset samples; y_i is assigned value of one if i-th object is predicted correctly, and zero otherwise, p_i is the model output for i-th object.

To assess the accuracy of object location, we used the Intersection over Union (IoU) metric [7]. The intersection-over-union area refers to two rectangles: one from the correct (labeled) image and the other predicted by the model. This metric is calculated as the ratio of the intersection area S_{int} to the union area S_u of the two boundary regions. It shows how well the rectangle predicted by the model matches the real one, i.e., where exactly the object is located in the photo. The higher the IoU value, the more accurately the model found the object

$$IoU = \frac{S_{int}}{S_u}. \quad (2)$$

If the model detected the object perfectly accurately, both rectangles will completely coincide, and the IoU value will be equal to 1. In all other cases, IoU will be less than 1.

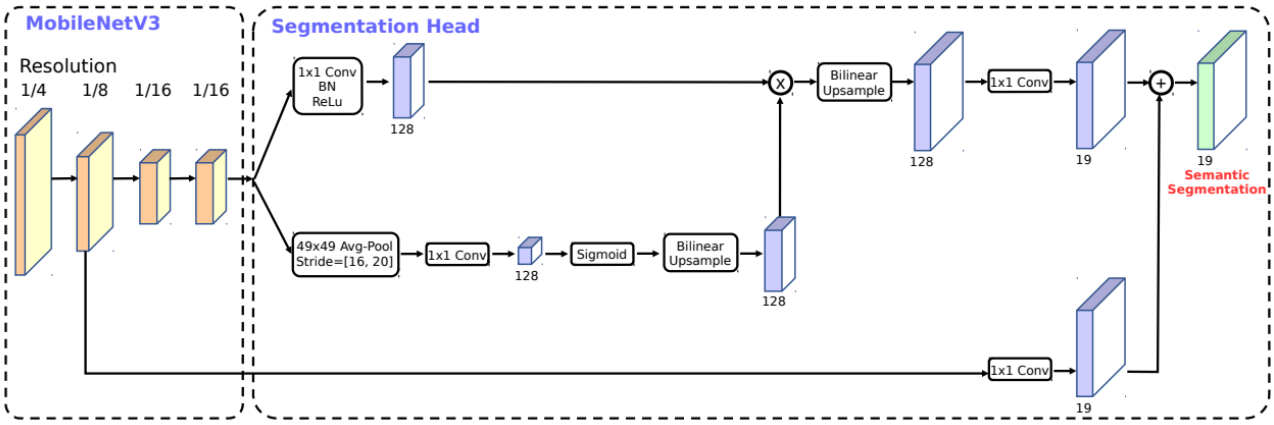

FIG. 1. Schematic representation of the structure of the MobileNetV3 model [1].

The closer the IoU value is to 1, the more accurate the model is. To use this metric in model training, it is converted into a loss function by applying the negative logarithm of IoU [7]

$$L_{rg} = -ln(IoU). \quad (3)$$

The final loss function is the sum of the regression and classification loss functions

$$L = L_{cl} + L_{rg}. \quad (4)$$

To perform this work, all input images were resized to 640 × 640 pixels in RGB color scheme.

The neural network was trained using the Adam optimization algorithm [8], which was used to find the minimum of the loss function. The "learning rate" parameter was set to 0.0036, which determines the step the model takes when updating the neural network weights during training. The input data was grouped into batches, with a batch size of 13, according to the capabilities of the graphics processing unit (GPU) used for training.

IV. TRAINING A MODEL FOR FACE DETECTION

Face recognition training was divided into two stages, the first of which is face detection serving as pretraining, and the second is face identification. For the training process of first stage, photos of different individuals, each containing a single face, were used. The dataset contained 5791 training and 1469 validation images, all of which were taken from an open dataset [9], which contains a variety of photos of general quality that can be found on social media posts. An example of a photo from the training dataset is shown in Fig. 2. As the face detection was only the pre-training stage, we opted out of scoring testing metrics on the trained model, as it will be retrained for face recognition, and then the final one will be tested.

The loss function given by Eq. 4 was monitored during the training of the neural network. The dependence of the loss functions on the number of training epoch is shown in Fig. 4. As you can see from the plot (Fig. 4), the loss function of the validation dataset reaches a plateau, indicating that further training does not make much sense. Further training can lead to poor performance on new data (e.g. overfitting in machine learning), which significantly degrades the quality of the model. To identify individuals on the second stage of training model for face recognition, the database used for the training was supplemented with a set of photographs with personal identifiers (surname, name) of students from Chernivtsi National University. This expanded dataset was used to test the accuracy of the model's predictions. Fig. 3 demonstrates the face detection by applying a pretrained face detection model on the student's dataset.

V. TRAINING A MODEL FOR FACE DETECTION

The second stage in our development is to train a model that will not only be able to detect the presence and position of a face in a photo, but also to distinguish between different faces and make an identification. To perform this step, we used the aforementioned dataset of photos with students. In this dataset, it was already known who was in the photo due to the name of the photo file (i.e., there was a match between the student and his/her photo), but there was no information about the exact location of the position of face in the image. To complete the dataset markup, a pre-trained on previous step face detection model was used to automatically determine where the face was in the photo. After that, we manually checked that the faces were found correctly before using this data to train the face recognition model.

Using the detection results described above, we perform face recognition. For this purpose, the part of the model responsible for determining the location of the face (regression) is left unchanged. The face identification model is trained on the students' photo dataset. To obtain the dataset, we asked the students to record a short video at the university auditoriums, facing the camera, and mimicking different facial expressions. The standard university room lighting was used during video recording. Then, different frames from different parts of the video were used as images for neural network training. Overall, the second stage training was used for 1680 pictures. Both validation and testing datasets contained 420 pictures. In each dataset, the number of photos was distributed equally among 30 people. Fig. 5 shows how the loss function changes during the training of this model.

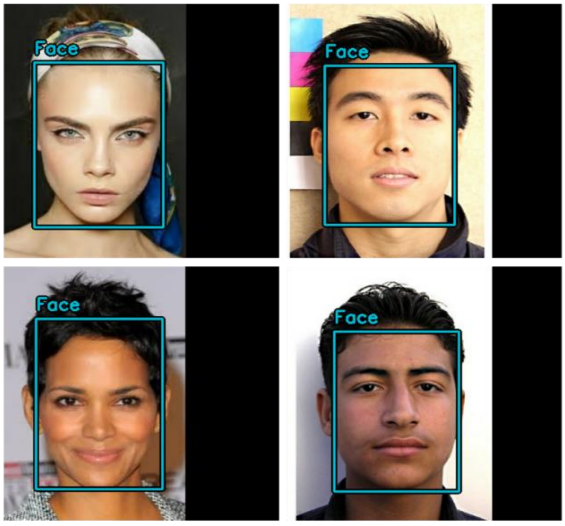

FIG. 2. An example of images from an open dataset used to train a face detection model.

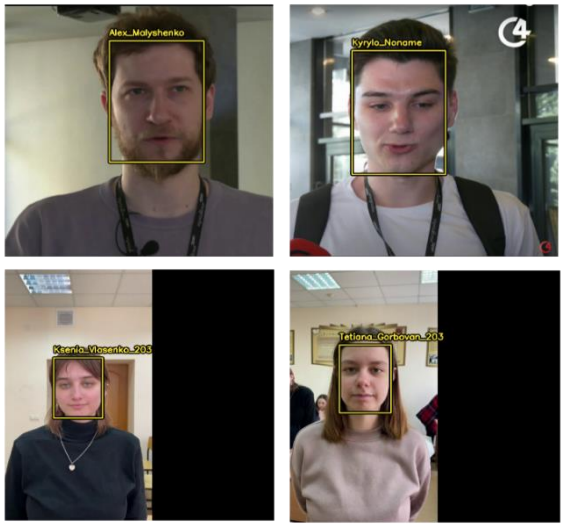

FIG. 3. Prediction of the face detection model on a dataset of student photos.

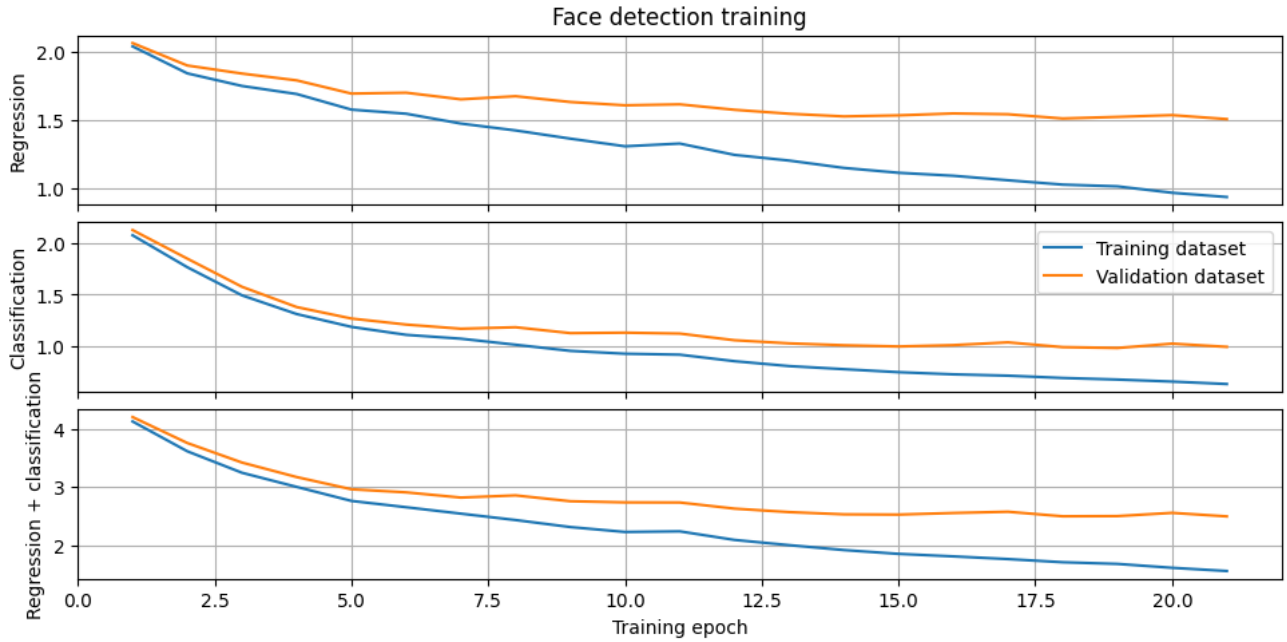

FIG. 4. Dependence of the loss function on the training epoch of a neural network for face detection.

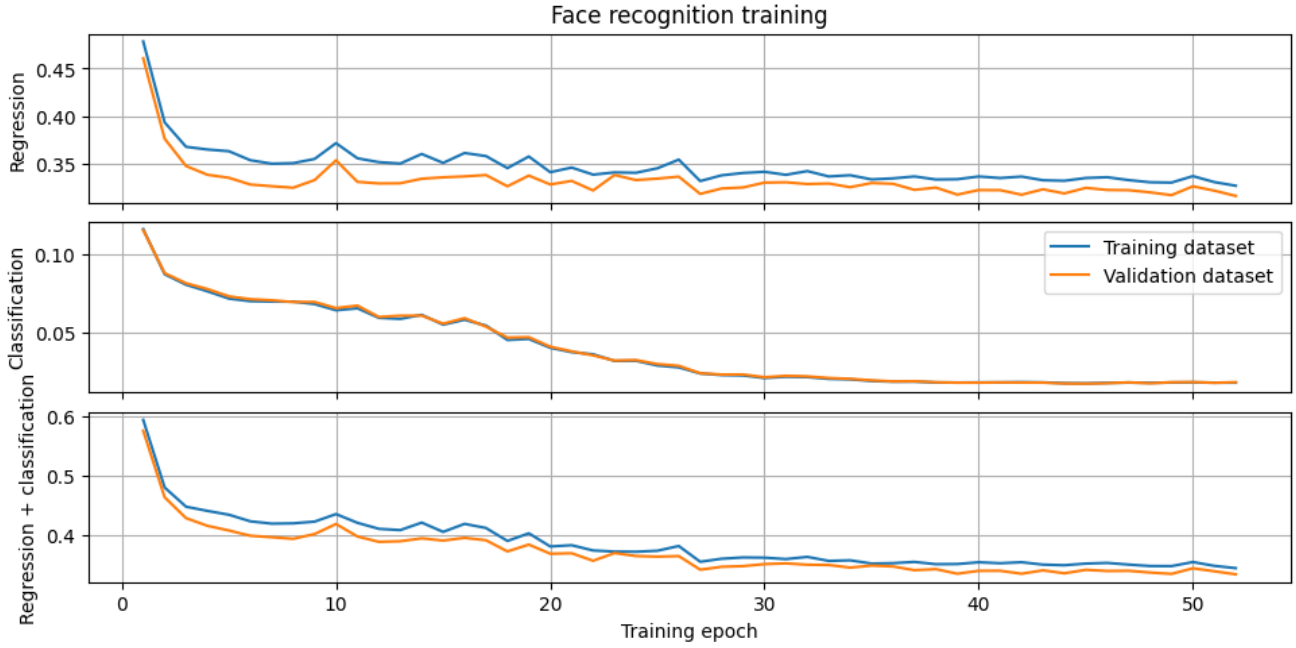

FIG. 5. Dependence of loss functions on the training epoch of a face recognition model.

As you can see from Fig. 5, the loss function associated with regression (prediction of the position and size of the rectangle indicating the location of the face) hardly changes during training. This is due to the fact that the layers of the neural network responsible for regression were fixed (i.e., not trained). The minor changes are due to changes in the classification part of the model.

The loss functions stop changing after about the 40th epoch, which indicates that further training is not make no better model. Since further training of the model with unchanged metrics of the student dataset may lead to overfitting, we limit ourselves to using 39 epochs. If we add photos of students who are not in the student database, the model will also recognize the students (see Fig. 6).

VI. EVALUATING THE EFFECTIVENESS OF THE MODEL

The model performance was tested on the testing dataset of student photos. The results of testing for 30 individuals were summarized by applying micro averaging. The resulting confusion matrix is presented in Table 1.

TABLE 1. Confusion matrix, evaluated on the testing dataset and averaged with micro averaging.

	Actually positive	**Actually negative**
Predicted positive	396	1
Predicted negative	6	0

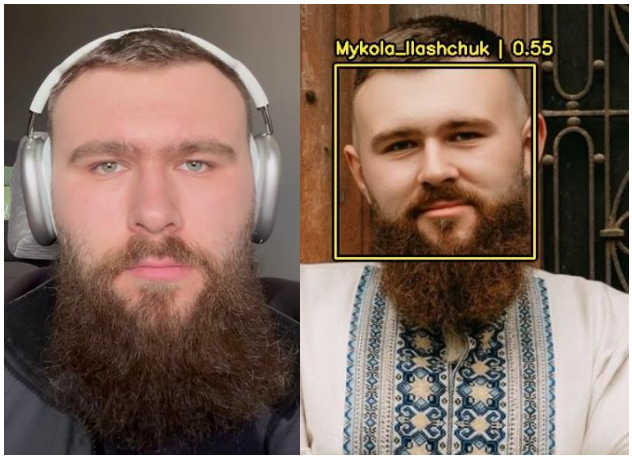

FIG. 6. On the left is a photo from the students' data, on the right is an identified face that was not used for training or model validation.

The precision of the final model, estimated on the testing dataset, is 99.75%, and the recall is equal to 98.5%. The overall F1 score is evaluated and is equal to 99.1%. Overall accuracy of the model is equal to 98.26%.

When using a GPU, the speed of the computation increases significantly compared to the calculation on the CPU alone, so it is recommended to use and train neural networks with a GPU.

Comparative characteristics of the results of the MobileNetV3 model and the previous study using the OpenCV library [10] are shown in Table 2. From the comparison table, the model has a fairly high accuracy for recognizing students' faces. The recognition quality of the neural network increases to 98.26% compared to 81% using the OpenCV library. The preparation time for recognition in the case of the neural network is half shorter.

A similar accuracy evaluation was also performed for the first stage model that only detected faces, without face recognition. The model was evaluated on a validation dataset. The accuracy decreased to 78.62%. The relatively low quality of the model is due to the large number of images and the limited quality of the dataset. Since almost the same neural network model was used its computation speed was almost the same as shown in the table.

TABLE 2. Comparative characteristics of face recognition models using OpenCV and neural network.

Amount of data, video	Accuracy, %	Preparation time for recognition, min	Video visualization quality on the screen
Results of face recognition using OpenCV [10]			
10	100	1	normal
20	90,5	4	visualization clarity
30	81	7	visualization clarity
Results of face recognition using neural network			
30	98,26	3,5	Assessment performed offline

VII. CONCLUSIONS

The face detection and recognition approach were implemented using computer vision and neural networks, in particular the MobileNetV3-Large model. The use of transfer learning made it possible to effectively adapt the pre-trained model to the specifics of the task, significantly reducing the computational cost of training and ensuring high recognition accuracy.

The model was trained in two stages: first, on an open dataset for the face detection task, and then on a local set of student photos for the identification task. This approach allowed us to achieve a balanced result even with a limited amount of our own data. The choice of metrics, such as binary cross-entropy for classification and a modified IoU function for regression, ensured efficient training and interpretation of the results.

The obtained results demonstrate that the chosen architecture and training methodology are suitable for implementing a system for automatic face recognition in images. The model allows generalization to the case when there are several faces in a photo and real-time operation.

ACKNOWLEDGMENT

Would like to thank the students of Yuriy Fedkovych Chernivtsi National University for providing the photos used for training the face recognition model.

AUTHOR CONTRIBUTIONS

M.I. – software development, neural network training, testing, optimization and validation, manuscript writing, concept development.

COMPETING INTERESTS

The author declares no competing interests.

REFERENCES

[1] A. Howard, M. Sandler, G. Chu, and L.-C. Chen, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019.

[2] M. Abadi, A. Agarwal, P. Barham, and E. Brevdo, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv preprint.* arXiv:1603.04467, 2016.

[3] O. Russakovsky, J. Deng, H. Su, and J. Krause, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[4] M. Watson, D. Shivakumar, F. Chollet, and M. Gorner, "KerasCV and KerasNLP: Vision and Language Power-Ups," *J. Mach. Learn. Res.*, vol. 25, no. 375, pp. 1–10, 2024.

[5] J. Hu, L. Shen, S. Albanie, and G. Sun, "Squeeze-and-Excitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020.

[6] S. Pydimarry, S. Khairnar, S. Palacios, and G. Sankaranarayanan, "Evaluating Model Performance with Hard-Swish Activation Function Adjustments," *arXiv preprint.* arXiv:2410.06879, 2024.

[7] J. He, S. Erfani, X. Ma, and J. Bailey, "Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression," *arXiv preprint.* arXiv:2110.13675, 2021.

[8] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv preprint*. arXiv:1412.6980, 2014.
[9] F. Elmenshawii, “Face Detection Dataset,” [Online]. Available: https://www.kaggle.com/datasets/fareselmenshawii/face-detection-dataset.
[10] M. Ilashchuk, I. Kushnir, and S. Melnychuk, “Rozpiznavannia oblych v realnomu chasi za dopomohoiu biblioteky OpenCV ta movy prohramuvannia Python,” *Herald of Khmelnytskyi Natl. Univ.*, no. 341, pp. 5–21, 2024 (in Ukrainian).

Mykola Ilashchuk

I graduated from ChNU with bachelor degree in Computer Science and with master degree in System Analysis. My research focuses on Determining the psychological state of a person using artificial intelligence. I am author of 3 scientific publications and have experience participating in conferences and research projects. Currently, I am PhD student in Department of Computer Systems Software ChNU.

ORCID ID: 0009-0002-7996-6176

Виявлення та ідентифікація облич за допомогою згорткової нейронної мережі та моделі MobileNetV3

Микола Ілащук*

Кафедра програмного забезпечення комп’ютерних систем, Навчально-науковий інститут фізико-технічних та комп’ютерних наук, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

*Автор-кореспондент (Електронна адреса: ilashchuk.mykola.m@chnu.edu.ua)

АНОТАЦІЯ У цій роботі представлено результати дослідження ефективності застосування методології трансферного навчання для задачі виявлення та розпізнавання облич із фокусом на обробці зображень, що містять лише одне обличчя. Як базову архітектуру нейронної мережі було обрано MobileNetV3, що забезпечує високу продуктивність при обмежених обчислювальних ресурсах. Навчання моделі здійснювалося у два послідовні етапи: перший – розпізнавання (детекція) облич на фотографіях, другий – ідентифікація особи за зображенням обличчя. Для забезпечення однозначності та чистоти навчальних даних використовувалися виключно зображення з одним обличчям. У процесі тренування було поєднано використання відкритого датасету для початкового етапу детекції облич із власним набором фотографій студентів Чернівецького національного університету імені Юрія Федьковича, призначеним для фази ідентифікації. Навчання та тестування моделі здійснювалися у хмарному середовищі Google Colab із використанням графічного процесора NVIDIA Tesla T4. Реалізація нейронної мережі виконувалася за допомогою сучасного фреймворку глибокого навчання TensorFlow та власного програмного коду, написаного мовою Python. Оптимізація параметрів моделі відбувалася шляхом мінімізації функції втрат, яка є сумою бінарної перехресної ентропії та від’ємного логарифма метрики Intersection over Union, що характеризує точність визначення розташування об’єкта на зображенні. Побудована модель була порівняна з попередніми підходами до детекції облич, реалізованими на основі бібліотеки OpenCV. Порівняльний аналіз за метриками точності розпізнавання та часу обробки продемонстрував перевагу розробленої системи. Отримані результати становлять інтерес для дослідників у галузі комп’ютерного зору, автоматизованих систем розпізнавання та технологій інтелектуальної обробки візуальних даних.

КЛЮЧОВІ СЛОВА машинне навчання, комп’ютерний зір, детектування об’єктів, згорткові нейронні мережі, трансферне навчання.