

Use of Convolutional Neural Networks for Identifying Additional Features on a Digital Image of Human Face

Kateryna Merkulova and Bohdan Pavliukh

Taras Shevchenko National University of Kyiv, Volodymyrs'ka str. 64/13, Kyiv, 01601, Ukraine

Abstract

This article is devoted to the study of the use of CNNs (convolutional neural networks) for the tasks of recognizing certain additional features on digital images of human faces.

Recognition of additional features in human face images has a wide range of applications, including photo and video analysis, security systems, collection of useful statistical data, and user comfort in various areas of life. This technology contributes to the improvement of safety and convenience in various life situations.

Thanks to numerous studies conducted by scientists on various image analysis methods that can be used to recognize features in an image (for example, histogram-based methods, feature allocation, contour analysis, color analysis), it was found that although the effectiveness of the method is also may depend on specific features of the task, such as lighting conditions, distance to objects and the presence of curtains or other obstacles, one of the most effective solutions is the use of machine learning methods, in particular the use of convolutional neural networks.

This paper describes the study of the effectiveness of using convolutional neural networks to recognize additional features on a digital image of a human face, namely the presence of a headdress, glasses, a medical mask, and a beard. Also, the gender of a person was determined as an additional auxiliary feature.

Keywords¹

identification, image classification, convolutional neural network

1. Introduction

Nowadays, technologies for recognition and identification are very widely used in all areas. The technology designed to identify additional features on a human face, such as the presence of glasses, a headdress, a medical mask and a beard, can be used to solve a number of important problems. The most obvious task that the technology under study can handle is the collection of statistical information. Such technology can be used in access control systems to increase security, for example by requiring users to remove a mask or glasses for identification, which can help avoid unauthorized access or circumvention [1]. Also, the technology can help identify people even when their appearance changes (that is, when using something that a system that uses recognition technology as an additional feature, for example, when wearing a glasses) [2; 3].

The tool for recognizing additional features on a digital image of a human face can also be used for narrowly focused purposes, for example, to track the observance of mask mode by visitors to a supermarket (or any other crowded place) [4; 5], and in this case it will be enough to determine only one of the additional features that the system can recognize, i.e. recognition of a medical mask on a human face (this example is especially relevant against the background of the recent pandemic). That is, there are such possibilities of application of the researched technology, in which even a limited part of the functionality of the technology can fully cope with the tasks.

In addition to the above-described possibilities of using the technology to recognize additional features in the image of a human face, it is worth paying attention to the non-obvious way of using the

Dynamical System Modeling and Stability Investigation (DSMSI-2023), December 05-07, 2023, Kyiv, Ukraine

EMAIL: kate.don11@gmail.com (K. Merkulova); bpavlyukh@gmail.com (B. Pavliukh)

ORCID: 0000-0001-6347-5191 (K. Merkulova); 0009-0001-1806-573X (B. Pavliukh)



© 2023 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)

technology, which gives commercial value. For example, facial recognition software can be used to gather statistical information about the number of bearded men passing through checkpoints at various subway stations, which can then be used to make decisions about the best location for a barbershop because the collected information will allow us to draw conclusions about the largest "places of concentration" of potential customers.

In the course of the research, a search and analysis of systems that determine additional features on people's faces was carried out, but no software systems were found that comprehensively approach the task of identifying additional features on a human face and recognize the presence or absence of all the above-mentioned features: headdress, glasses, medical mask, beard. On the Internet, in public access, you can find only information about software in which the developed functionality is partially implemented (for example, software systems designed to recognize the presence of a medical mask) or software that is able to recognize only some of the above features or others that are similar in essence additional characteristics of a human face, for example, sex, race, age, emotions, direction of gaze.

Therefore, the technology for recognizing additional features on images of a human face has a wide range of applications, contributes to improving safety and convenience in various life scenarios.

2. Task Definition, Solution Methods and Technologies

For the task of recognizing additional features in human face images, the first step is to choose the image analysis method that will be used for recognition, of which there are many these days. The most popular of them, which can be used to solve the given problem, are contour analysis, the use of image segmentation, the analysis of brightness histograms, and the use of neural networks.

Nowadays, the best "tool" for image analysis is still the visual cortex of the human brain. Computer analysis tools are already used to analyze images in the fields of medicine, security, and remote sensing, but they will not be able to replace people for a long time due to the capabilities of the human brain, because the process of information processing by the human brain is non-linear and extremely complex. That is why the most popular technologies for image analysis are those that developers try to get as close as possible to models of human visual perception. Artificial neural networks are endowed with such technologies. Convolutional neural networks are especially adapted for computer vision tasks.

Convolutional neural networks usually contain convolutional layers, subsampling (aggregation) layers, fully connected and normalizing layers in their structure [6]. Convolutional layers apply a set of filters to the input image, applying element-by-element multiplication of the filter values to the original pixel values of the image. After that, all the products are added, that is, the image is convoluted. As a result of applying a convolutional layer, we get several feature maps, the number of which is usually equal to the number of filters. Each filter determines a certain property on the image, by means of initialization with random values of the filter kernel. The following formula demonstrates the calculation of the output of a convolutional layer using filters and an activation function:

$$O(x, y) = \sigma\left(\sum_{i=1}^N \sum_{j=1}^M (I(x+i, y+j) \cdot K(i, j) + b)\right), \quad (1)$$

where $O(x, y)$ - is the output pixel at the position (x, y) , $I(x+i, y+j)$ - is the pixel of the input image at the position $(x+i, y+j)$, $K(i, j)$ - value of the filter core at position (i, j) , b - bias, σ - activating function (for example, ReLU), N and M - the size of the filter area.

The layers that will be added to CNNs being developed have the ReLU activation function:

$$f(x) = \max(0, x). \quad (2)$$

Layers of subsampling (aggregation, pooling) are used to reduce the size of the input feature map by combining the outputs of clusters of neurons of one layer to one neuron of the next layer [7]. For example, averaging aggregation uses the average value from each of the clusters of neurons in the previous layer:

$$P(x, y) = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M I(x+i, y+j), \quad (3)$$

where $P(x,y)$ - output of the pooling layer in position (x,y) .

In a fully connected layer, the input neurons are connected to all neurons (activations) of the previous layer. Usually these layers are used for the output layer. This layer calculates the class score, and outputs a vector whose dimension is equal to the number of classes. The ordinal number of the element of the vector with the largest value means the most probable belonging of the input image to the corresponding class. Therefore, a convolutional neural network is a convenient tool for image classification [8]. The identification of additional features on a human face belongs precisely to the tasks of image classification, because we assign a person detected in the image to one of the classes for each feature: for example, to identify a headdress on a human head, we have 2 classes - the presence of a headdress and its absence. To train a neural network to recognize a certain number of states of the presence of an additional feature, it is necessary to prepare an appropriate number of training data sets. The developed CNN will receive an image as an input, and the output layer will consist of the number of neurons, which is equal to the number of possible classes (states of the presence of an additional feature).

Transfer learning is an effective technology in the development of CNN. It is a task in machine learning that focuses on storing the knowledge gained during solving one problem and applying it to another, but related problem, i.e. transfer learning allows you to use the accumulated knowledge in solving one problem experience to solve another, similar problem [9]. That is, a ready-made trained neural network, which, for example, identifies the type of animal in the image, can be used to recognize additional features on the human face [10]. To do this, you can remove the last layer of the ready-made model (CNN) and add one or more own layers, the last of which, as already indicated above, must contain the number of neurons that will correspond to the number of states of the presence of an additional feature. The advantages of using transfer training are that there is no need to create a large number of layers, and that the network will already be trained on a large amount of data, after which only layers that were not part of the "original" CNN can be trained on the target set.

For research and testing of convolutional neural networks, it was decided to develop software that works with streaming video and has the following working principle: after the user launches the main function of the application, the software in real time receives an image from the camera of the device on which it is running, further on using the computer vision library OpenCV separates images of human faces [11], processes them, identifies the presence or absence of additional features with the help of five developed models and displays images with text labels of the result of their identification.

To interact with neural networks, the developed software uses Keras, an open-source library written in the Python programming language, designed to interact with neural networks, including convolutional and recurrent NNs (neural networks). Keras is one of the most popular tools used when deep learning is involved in a project. Keras offers consistent and simple APIs, minimizing the number of user actions required for typical use cases [12].

Keras includes a number of pre-trained networks that you can download and use right away. One of the most famous such networks (models) is MobileNetV2, which was trained for image classification.

The image processing after detecting human faces includes the following image processing operations:

- 1) resizing the image to 224×224 pixels. This is due to the fact that images of this size were used for training the MobileNetV2 model;

- 2) converting the image to array view. For each image, the size of the array will have a dimension of $224 \times 224 \times 3$, that is, for each pixel, 3 values corresponding to the 3 RGB color channels of the model are stored;

- 3) normalization of color channel values. The standard range of pixel values for color channels is from 0 to 255. For the correct operation of the MobileNetV2 model, this range must be changed to the range from -1 to 1. The same image processing operations are applied to the images from the training set immediately before they are used to train the CNNs. The same method of processing training images and images to which the model is applied ensures the achievement of the maximum efficiency of the model. The block diagram shown in Figure 1, demonstrates the sequence of actions performed for, as part of the training of CNNs to identify additional features on a human face.

Other imported Python programming language libraries besides Keras also play an important role in the software. The most significant for the developed software are the following used libraries: matplotlib for plotting graphs of the accuracy of CNNs, numpy for storing multidimensional arrays [13]

containing image information in a convenient form, imutils for obtaining images from a camera, OpenCV for computer vision [14], Tkinter for creating GUI.

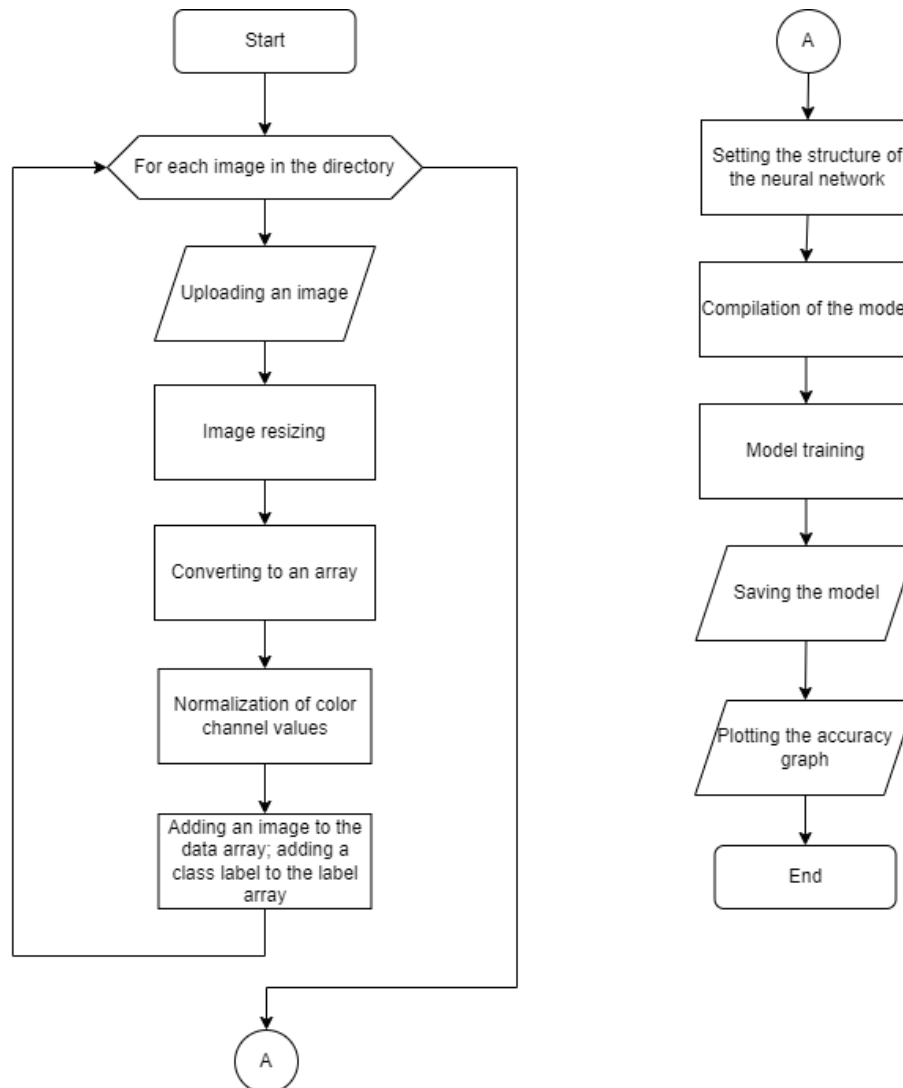


Figure 1: Block diagram of the CNN training algorithm for the identification of additional features on the human face

Therefore, using the methods and technologies described above, it was decided to conduct a study to determine the effectiveness of the application of convolutional neural networks to recognize additional features (the presence of a headdress, glasses, a medical mask, a beard and additionally determine the gender) on images of people's faces obtained from streaming video.

3. Research

The research covers the description of the process of collecting input data sets, the details of the design of convolutional neural networks for the identification of additional features on a digital image of a human face. As part of the research, software was also developed. This section also presents the results of the developed software and the achieved accuracy rates for each of the trained convolutional neural networks responsible for identifying a certain additional feature.

3.1. Data Collection

The developed software must recognize 4 additional features on the face (presence of glasses, medical mask, headdress and beard) and additionally the gender of the person, accordingly, 5

convolutional neural networks will be developed, for which 5 training data sets are required, and the images in these sets must additionally be divided into a number of groups corresponding to the number of possible resulting states recognized. Let's consider the list of signs and possible resulting states that we will determine:

- Gender – male or female;
- Glasses – sunglasses, glasses for vision or no glasses;
- Medical mask – presence, absence or incorrect wearing (for example, if the mask does not cover the person's nose);
- Headdress - presence or absence;
- Beard - presence or absence.

So, we need to form 5 sets of training data, for which a total of 12 groups of training data (images) need to be formed. To facilitate the process of forming training sets, before "presenting" the image to the neural network, we will programmatically process the image in the form of reducing the image size to the required values, this will allow the size of the found images to be neglected. It is worth paying attention to the format in which the image is saved: it should be suitable for the software tools used. The most convenient (and at the same time the most popular) formats for working with images are JPEG and PNG.

Since the identification of additional features is performed on the image of the human face itself, it is necessary to choose images for the training sets that cover only a small area around the human face.

Images of human faces can be used in different training datasets, but the same image, of course, cannot be in different groups of the same training set. Let's consider several examples of suitable images and determine to which groups of training data they can be assigned.



Figure 2: Faces of people with additional features: a) a man with glasses and a beard; b) a woman with a medical mask and headdress; c) a man wearing sunglasses and an improperly fitted medical mask (the mask was added artificially).

The image of a person's face in Figure 2.a can be assigned to all training sets: to determine gender - to the group "male", to determine the presence of glasses - to the group "with glasses for vision", to determine the presence of a medical mask - to the group "without medical masks", to determine the presence of a headdress in the "without headdress" group, to determine the presence of a beard - in the "with a beard" group.

The image in Figure 2.b can be assigned to the following training sets: to determine gender - to the group "female", to determine the presence of glasses - to the group "without glasses", to determine the presence of a medical mask - to the group "with a medical mask", to determine the presence of the main recruitment to the group "with headdress". The images cannot be assigned only to the training set for determining the presence of a beard - there should be only images of men, because the developed software, when determining the gender of a person as "female", will not use CNN to try to detect a beard in the image. For training data sets, you can use images in which additional features are artificially added to the image of a person's face using software such as Adobe Photoshop.

To the image in Figure 3, a medical mask has been artificially added with the help of software, which looks incorrectly fitted. Such an image can be used without problems for all training sets: to determine the gender - to the group "male", to determine the presence of glasses - to the group "with sunglasses",

to determine the presence of a medical mask - to the group "with an incorrectly worn medical mask", to determine the presence of a headdress - to the group "without a headdress", to determine the presence of a beard - to the group "without a beard".

In total, the prepared training data sets for all planned convolutional neural networks have about 10,750 images of people, whose detailed breakdown by set is shown in Table 1.

Table 1

The number of images of people in the training sets

| Dataset | Group | Quantity |
|--------------|--------------------------------------|----------|
| Glasses | without glasses | 800 |
| | with glasses for vision | 800 |
| | with sunglasses | 200 |
| Medical mask | without a medical mask | 750 |
| | with a medical mask | 1700 |
| | with an improperly worn medical mask | 1700 |
| Headdress | without headdress | 200 |
| | with a headdress | 200 |
| Beard | a man with a beard | 200 |
| | a man without a beard | 200 |
| Sex | male | 2000 |
| | female | 2000 |

It is noticeable that some of the used sets have significantly fewer images, this is due to the fact that for the beard and headdress it was not possible to find ready-made datasets, and it was necessary to create datasets for these features, for other features it was possible to find ready-made datasets in public access.

3.2. CNN Designing

The biggest influence on the design of convolutional neural networks was the decision to use the technology of transfer (transfer) learning, which allows us to use a ready-made neural network (with defined weights for neurons) that was created to perform a similar task as the basis for the creation of convolutional neural networks according to the essence of the problem. This approach allows you to avoid the independent creation of a large number of layers for CNN, and the neural network that will be used will already be trained on a large amount of training data (images) [15].

As already mentioned above, it was decided to use the MobileNetV2 model as such a basis for new convolutional networks. The MobileNetV2 artificial intelligence system was created for image recognition of various objects, mainly different types of animals (about 1000 classes of objects) [16]. All layers taken from the MobileNetV2 model will not be retrained. It was decided for simplification to make the same structure for all CNNs, that is, to add the same layers, namely:

- 1) A fully connected layer with 128 neurons and a ReLU (rectified linear node) activation function. Its goal is to find an additional characteristic to increase the accuracy of the model;
- 2) An exclusion layer used for regularization to reduce overtraining of the neural network by preventing complex co-adaptations of the training data (some neurons will always return 0 as a result);
- 3) A fully connected layer with a softmax activation function (returns a probability distribution). The neuron with the highest activity (highest value) indicates that the most likely outcome corresponds to the class represented by that neuron.

It was decided to train the models in 20 epochs (iterations). During training, as already mentioned, the input images are pre-processed. As an optimization algorithm that adjusts the weights of the neural network, we choose one of the most common for this type of problem, the Adam algorithm [17; 18]. A decision was made to use the sparse categorical crossentropy algorithm as a loss function.

The input data sets are divided into a training set and a validation set to test how well CNNs performs on images that were not used for training, this allows for a more adequate accuracy rate and prevents the model from being overtrained. For validation, we select 15% of the input data set.

After completing the training of the CNN, the software, using the capabilities of the matplotlib.pyplot library, plots the dependence of the accuracy and loss of the CNN on the iteration number (for both training and validation input data sets). The loss value is calculated as follows:

$$L(y, \hat{y}) = - \sum_{i=1}^N (y_i \cdot \log(\hat{y}_i)), \quad (4)$$

where $L(y, \hat{y})$ - is the value of the loss function, y_i is the true answer, \hat{y}_i is the answer of the model prediction.

The developed CNNs consists of 158 layers and 2,422,210 parameters for CNNs with two resulting classes and 2,422,339 parameters for CNNs with three resulting classes, of which 162,266 parameters for CNNs with two resulting classes and 162,355 parameters for CNNs with three are subject to training. resulting classes.

3.3. Research Results

A demonstration of the identification of additional features on people's faces by the developed software is shown in Figure 4.

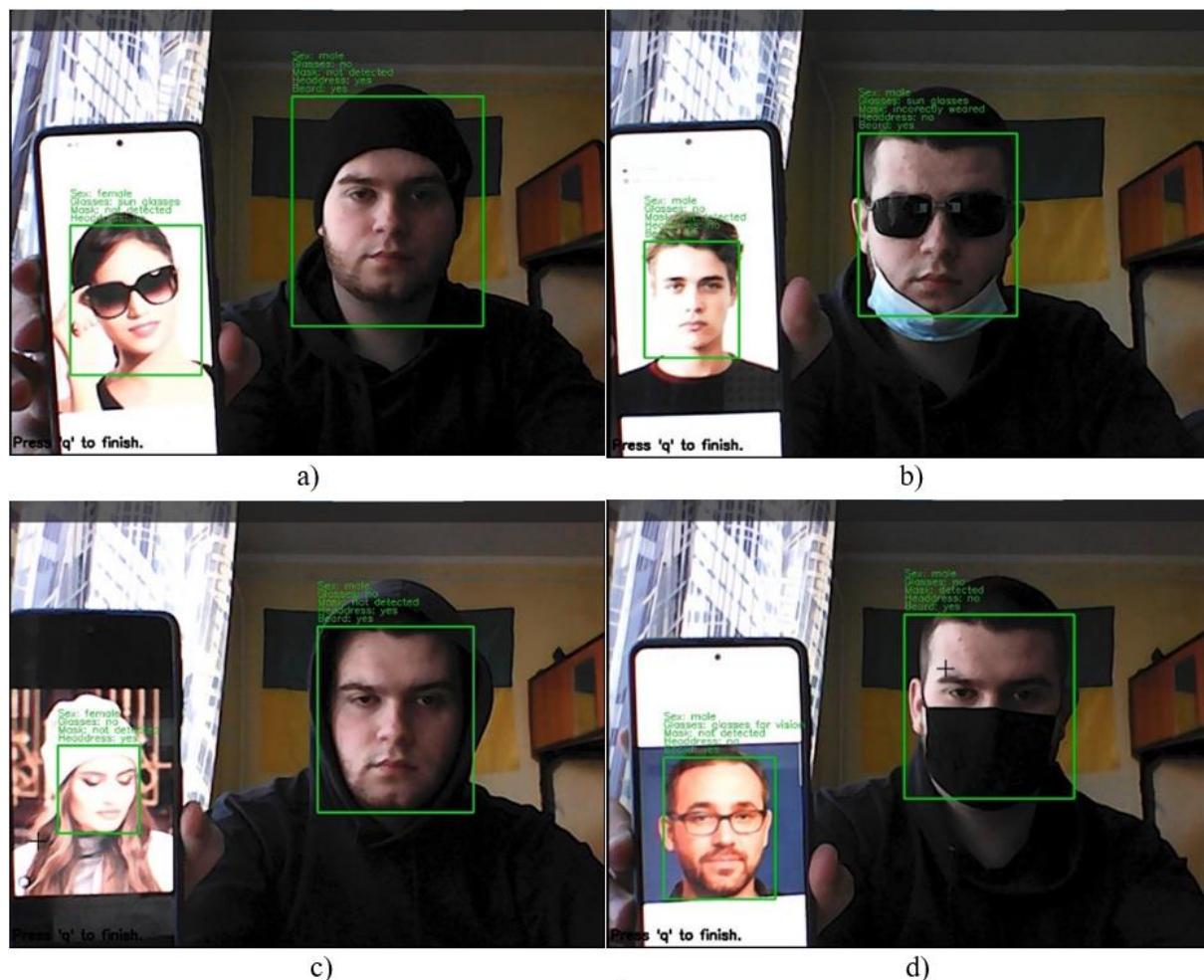


Figure 4: Determination of additional features on human faces with the help of developed CNNs: a) a man with a beard in a headdress and a woman in glasses; b) a man with an improperly worn medical mask and sunglasses and a man without additional features on his face; c) a man with a beard and a headdress and a woman with a headdress; d) a man in a medical mask and a man with a beard and glasses.

To confirm the correct operation of the program, the operation of the program is demonstrated with two people in the image (a photo of a person is also suitable for demonstrating the operation of the program), and in different images, there are different states of the presence of glasses, a medical mask, a headdress, a beard (and different glasses and masks of different colors are used) , as well as different

genders of people, which allows you to check the correctness of the work of all trained convolutional neural networks.

Tables 2-4 contain the obtained results of accuracy and loss (for training and validation data sets) for all 5 created CNNs for identifying additional features.

Table 2
Accuracy and loss results for CNNs for determining gender and presence of glasses

| Epoch | sex | | | | glasses | | | |
|-------|--------|----------|----------|---------|---------|----------|----------|---------|
| | loss | accuracy | val_loss | val_acc | loss | accuracy | val_loss | val_acc |
| 1 | 0.3132 | 0.8797 | 0.2522 | 0.8737 | 0.3775 | 0.8597 | 0.1091 | 0.9667 |
| 2 | 0.1516 | 0.9431 | 0.4833 | 0.8062 | 0.0808 | 0.9722 | 0.1081 | 0.9583 |
| 3 | 0.1197 | 0.9581 | 0.2034 | 0.9100 | 0.0620 | 0.9840 | 0.0166 | 0.9972 |
| 4 | 0.0928 | 0.9681 | 0.1754 | 0.9250 | 0.0354 | 0.9944 | 0.0312 | 0.9889 |
| 5 | 0.0693 | 0.9769 | 0.1407 | 0.9362 | 0.0359 | 0.9903 | 0.0141 | 0.9972 |
| 6 | 0.0554 | 0.9828 | 0.0839 | 0.9663 | 0.0293 | 0.9903 | 0.0232 | 0.9944 |
| 7 | 0.0334 | 0.9897 | 0.0961 | 0.9688 | 0.0255 | 0.9937 | 0.0439 | 0.9889 |
| 8 | 0.0350 | 0.9887 | 0.0662 | 0.9762 | 0.0183 | 0.9931 | 0.0697 | 0.9750 |
| 9 | 0.0286 | 0.9928 | 0.1014 | 0.9700 | 0.0166 | 0.9951 | 0.0299 | 0.9861 |
| 10 | 0.0253 | 0.9922 | 0.0288 | 0.9887 | 0.0110 | 0.9986 | 0.0586 | 0.9806 |
| 11 | 0.0191 | 0.9950 | 0.0426 | 0.9887 | 0.0156 | 0.9965 | 0.0246 | 0.9889 |
| 12 | 0.0210 | 0.9937 | 0.0645 | 0.9812 | 0.0114 | 0.9972 | 0.0684 | 0.9806 |
| 13 | 0.0193 | 0.9944 | 0.0870 | 0.9750 | 0.0081 | 0.9979 | 0.0157 | 0.9944 |
| 14 | 0.0186 | 0.9953 | 0.0485 | 0.9825 | 0.0104 | 0.9944 | 0.1213 | 0.9500 |
| 15 | 0.0195 | 0.9950 | 0.0391 | 0.9875 | 0.0091 | 0.9979 | 0.0372 | 0.9861 |
| 16 | 0.0140 | 0.9956 | 0.0395 | 0.9862 | 0.0041 | 0.9993 | 0.0211 | 0.9944 |
| 17 | 0.0146 | 0.9969 | 0.0561 | 0.9837 | 0.0036 | 1.0000 | 0.0222 | 0.9917 |
| 18 | 0.0119 | 0.9956 | 0.0559 | 0.9875 | 0.0017 | 1.0000 | 0.0207 | 0.9917 |
| 19 | 0.0134 | 0.9950 | 0.0223 | 0.9950 | 0.0039 | 1.0000 | 0.0916 | 0.9639 |
| 20 | 0.0124 | 0.9947 | 0.0377 | 0.9862 | 0.0031 | 1.0000 | 0.0139 | 0.9972 |

Table 3
Accuracy and loss results for CNNs to determine the presence of a beard and headdress

| Epoch | beard | | | | headdress | | | |
|-------|--------|----------|----------|---------|-----------|----------|----------|---------|
| | loss | accuracy | val_loss | val_acc | loss | accuracy | val_loss | val_acc |
| 1 | 0.3970 | 0.8125 | 0.3440 | 0.8500 | 0.3757 | 0.8281 | 0.0834 | 0.9875 |
| 2 | 0.2341 | 0.9000 | 0.2273 | 0.9500 | 0.1070 | 0.9625 | 0.0079 | 1.0000 |
| 3 | 0.1711 | 0.9375 | 0.4524 | 0.7875 | 0.0748 | 0.9812 | 0.0059 | 1.0000 |
| 4 | 0.1268 | 0.9563 | 0.2017 | 0.9375 | 0.0402 | 0.9844 | 0.0061 | 1.0000 |
| 5 | 0.0974 | 0.9656 | 0.2716 | 0.9000 | 0.0245 | 1.0000 | 0.0035 | 1.0000 |
| 6 | 0.0542 | 0.9875 | 0.3199 | 0.8625 | 0.0222 | 0.9906 | 0.0022 | 1.0000 |
| 7 | 0.0526 | 0.9781 | 0.2712 | 0.8750 | 0.0111 | 0.9969 | 0.0035 | 1.0000 |
| 8 | 0.0507 | 0.9844 | 0.2864 | 0.8500 | 0.0066 | 1.0000 | 0.0022 | 1.0000 |
| 9 | 0.0346 | 0.9906 | 0.2169 | 0.8875 | 0.0055 | 1.0000 | 0.0017 | 1.0000 |
| 10 | 0.0303 | 0.9906 | 0.2112 | 0.9000 | 0.0056 | 1.0000 | 0.0021 | 1.0000 |
| 11 | 0.0278 | 0.9937 | 0.3007 | 0.8875 | 0.0033 | 1.0000 | 0.0018 | 1.0000 |
| 12 | 0.0125 | 1.0000 | 0.2385 | 0.9000 | 0.0029 | 1.0000 | 0.0012 | 1.0000 |
| 13 | 0.0167 | 1.0000 | 0.1978 | 0.9000 | 0.0038 | 1.0000 | 0.0008 | 1.0000 |
| 14 | 0.0211 | 0.9937 | 0.2655 | 0.9000 | 0.0050 | 1.0000 | 0.0012 | 1.0000 |
| 15 | 0.0144 | 0.9969 | 0.1816 | 0.9375 | 0.0019 | 1.0000 | 0.0010 | 1.0000 |
| 16 | 0.0167 | 1.0000 | 0.3053 | 0.9000 | 0.0021 | 1.0000 | 0.0006 | 1.0000 |
| 17 | 0.0197 | 0.9937 | 0.2157 | 0.9125 | 0.0017 | 1.0000 | 0.0008 | 1.0000 |
| 18 | 0.0154 | 0.9969 | 0.3927 | 0.8500 | 0.0016 | 1.0000 | 0.0009 | 1.0000 |
| 19 | 0.0085 | 1.0000 | 0.1733 | 0.9375 | 0.0021 | 1.0000 | 0.0006 | 1.0000 |
| 20 | 0.0141 | 0.9969 | 0.4428 | 0.8500 | 0.0011 | 1.0000 | 0.0005 | 1.0000 |

The accuracy graph, which visualizes the achieved accuracy and loss rates for the training set and the validation set depending on the training iteration, is shown in Figure 5.

Table 5 shows the values of accuracy and loss for the validation set obtained by convolutional neural networks after 20 iterations (epochs) of training.

It is worth noting that the training process was started several times, the difference between the results was very insignificant.

Table 4

Accuracy and loss results for the CNN to determine the presence of a medical mask

| Epoch | medical mask | | | |
|-------|--------------|----------|----------|---------|
| | loss | accuracy | val_loss | val_acc |
| 1 | 0.2038 | 0.9253 | 0.1975 | 0.9205 |
| 2 | 0.0806 | 0.9708 | 0.1073 | 0.9542 |
| 3 | 0.0509 | 0.9825 | 0.1353 | 0.9470 |
| 4 | 0.0438 | 0.9852 | 0.1949 | 0.9337 |
| 5 | 0.0267 | 0.9892 | 0.1483 | 0.9482 |
| 6 | 0.0268 | 0.9919 | 0.1698 | 0.9386 |
| 7 | 0.0170 | 0.9955 | 0.1644 | 0.9446 |
| 8 | 0.0210 | 0.9925 | 0.0994 | 0.9590 |
| 9 | 0.0143 | 0.9958 | 0.1191 | 0.9590 |
| 10 | 0.0191 | 0.9931 | 0.1421 | 0.9542 |
| 11 | 0.0178 | 0.9940 | 0.1900 | 0.9398 |
| 12 | 0.0100 | 0.9964 | 0.1042 | 0.9651 |
| 13 | 0.0104 | 0.9976 | 0.0713 | 0.9759 |
| 14 | 0.0188 | 0.9925 | 0.1316 | 0.9590 |
| 15 | 0.0131 | 0.9955 | 0.1400 | 0.9554 |
| 16 | 0.0104 | 0.9967 | 0.1263 | 0.9614 |
| 17 | 0.0067 | 0.9982 | 0.1707 | 0.9494 |
| 18 | 0.0039 | 0.9988 | 0.2399 | 0.9373 |
| 19 | 0.0034 | 0.9997 | 0.1668 | 0.9554 |
| 20 | 0.0037 | 0.9985 | 0.1537 | 0.9614 |

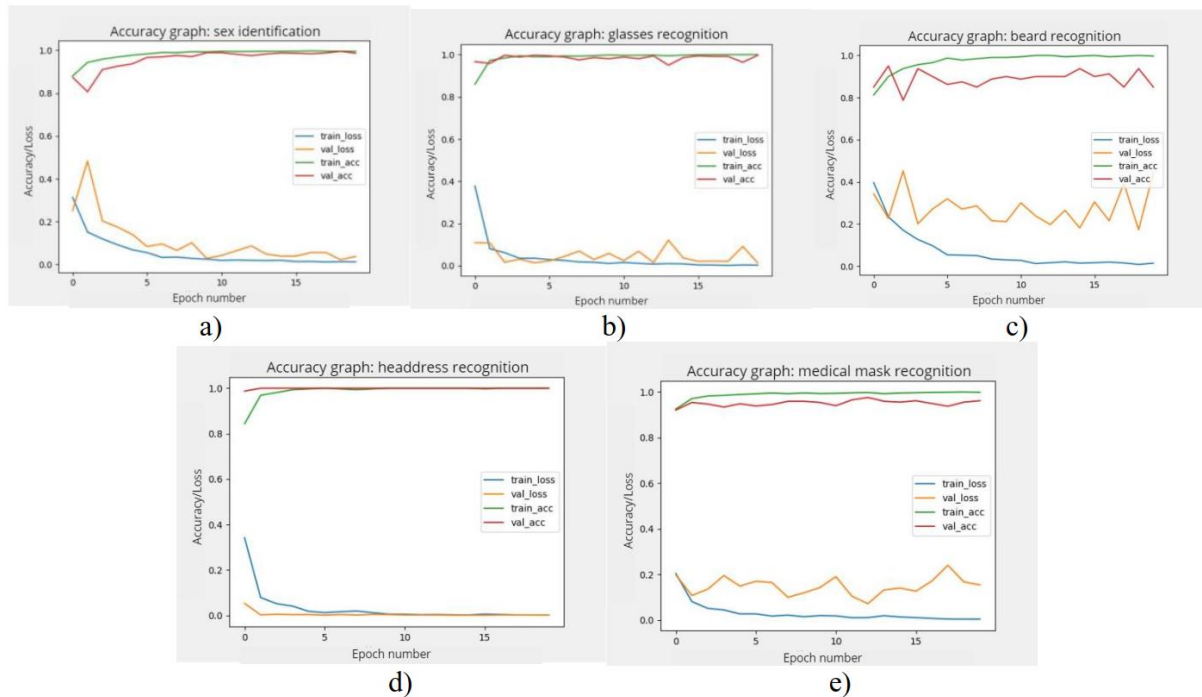


Figure 5: Accuracy graphs of the developed CNNs for: a) gender identification; b) glasses recognition; c) beard recognition; d) headdress recognition; e) medical mask recognition.

Table 5

Final results of accuracy and loss on the validation sets for the developed CNNs

| CNN purpose | Accuracy, % | Loss, % |
|--------------------------|-------------|---------|
| Glasses recognition | 99.72 | 1.39 |
| Beard recognition | 85 | 44.28 |
| Headdress recognition | 100 | 0.05 |
| Medical mask recognition | 96.14 | 15.37 |
| Sex identification | 98.62 | 3.77 |

4. Conclusion

This article is devoted to the study of the effectiveness of using convolutional neural networks for the task of recognizing additional features on a digital image of a human face. For this purpose, CNNs was developed to recognize the following additional features: the presence of glasses, a beard, a headdress, a medical mask, and a person's gender.

All the developed models showed a good result both on training and validation data sets (which is indicated by the presented accuracy tables and graphs), and during their experimental application for streaming video. The final values of accuracy and loss on the validation data sets for developed CNNs are as follows: for the detection of glasses, the accuracy is 99.72%, the loss is 1.39%; for detecting a medical mask, the accuracy is 96.14%, the loss is 15.37%; for headdress detection, the accuracy is 100%, the loss is 0.05% (such a high result is probably related to the not very good content of the input data set); for detecting a beard, the accuracy is 85%, the loss is 44.28%; for gender determination, the accuracy is 98.62%, the loss is 3.77%. Among the shortcomings of the developed models, we can highlight a decrease in the accuracy of work when the image was taken in poor lighting conditions (which was expected).

One of the interesting results obtained is the fact that the fewest images in the training sets had a beard and a headdress, and, at the same time, their models gave the opposite result - based on the obtained accuracy results, the model for recognizing the beard has the largest loss, and the model for recognizing headdress is the smallest. This result is most likely related to the small number of images in the training sets.

For further research on this topic, models can be developed to identify other additional features, for example, a mustache, as well as such features of a person as age, race, emotions. It is also possible to increase the number of resulting classes for CNNs, for example, to define different types of headdress or to define more types of glasses.

To improve the quality of the obtained results, it is possible to expand data sets, as well as apply additional methods of image preprocessing, which would help solve some difficulties, for example, reduce the influence of the level of illumination on the result of the work of CNNs.

5. References

- [1] O. Bychkov, K. Merkulova and Y. Zhabska, "Information Technology for Person Identification by Occluded Face Image," 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2022.
- [2] O. Bychkov, K. Merkulova, Y. Zhabska, A. Shatyko, "Development of information technology for person identification in video stream," Proceedings of the II International Scientific Symposium "Intelligent Solutions" (IntSol-2021), CEUR Workshop Proceedings, 3018, pp. 70-80, Kyiv - Uzhhorod, Ukraine, September 28-30, 2021. URL: http://ceur-ws.org/Vol-3018/Paper_7.pdf.
- [3] V. Martsenyuk, O. Bychkov, K. Merkulova and Y. Zhabska, "Exploring Image Unified Space for Improving Information Technology for Person Identification," in IEEE Access, vol. 11, pp. 76347-76358, 2023, doi: 10.1109/ACCESS.2023.3297488.
- [4] Face SDK, Regula. URL: <https://regulaforensics.com/products/face-recognition-sdk>.

- [5] V. Petrivskiy, V. Shevchenko, S. Yevseiev, O. Milov, O. Laptiev, O. Bychkov, V. Fedoriienko, M. Tkachenko, O. Kurchenko and I. Opirskyy, "Development of a Modification of the Method for Constructing Energy-Efficient Sensor Networks Using Static and Dynamic Sensors", *Eastern-European Journal of Enterprise Technologies*, vol. 1 (9 (115)), 2022, pp. 15–23, doi: <https://doi.org/10.15587/1729-4061.2022.252988>.
- [6] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, H. Ghayvat, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope", India, 2021. URL: <https://www.mdpi.com/2079-9292/10/20/2470>.
- [7] N. Ketkar, J. Moolayil, "Convolutional neural network", Berkeley, CA, USA, April 10, 2021, pp. 197–242. URL: https://doi.org/10.1007/978-1-4842-5364-9_6.
- [8] S. Mathot, "Introduction to deep learning", 2021. URL: <https://pythontutorials.eu/deep-learning/introduction>.
- [9] "Face Mask Detection and Correct Mask Wearing Recognition Software. How to Save Your Business from Quarantine and Closure", SYTOSS, 2021. URL: <https://www.sytoos.com/blog/face-mask-detection-and-correct-mask-wearing-recognition-software-how-to-save-your-business-from-quarantine-and-closure>.
- [10] A. Hossain, S. Sajib, "Classification of Image using Convolutional Neural Network (CNN)", Pabna University of Science & Technology, 2019.
- [11] S. Aparna, "Face Recognition using OpenCV", Dublin Business School, 2020, pp. 10-12.
- [12] "Keras. Simple. Flexible. Powerful", Keras. URL: <https://keras.io>.
- [13] S. Shell, "Introduction to Numpy and Scipy", San Francisco, 2019, pp. 7-11.
- [14] M. Khan, S. Chakraborty, R. Astya, S. Khepra, "Face Detection and Recognition Using OpenCV", International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), India, October 18-19, 2019. URL: <https://ieeexplore.ieee.org/abstract/document/8974493>.
- [15] Kutyrev A., Kiktev N., Kalivoshko O., Rakhmedov R. Recognition and Classification Apple Fruits Based on a Convolutional Neural Network Model. (2022) CEUR Workshop Proceedings, 3347, pp. 90 – 101. https://ceur-ws.org/Vol-3347/Paper_8.pdf
- [16] S. Mathot, "Classifying images with MobileNetV2", 2021. URL: <https://pythontutorials.eu/deep-learning/image-classification>.
- [17] Shatyrko A., Khusainov D. On the Interval Stability of Weak-Nonlinear Control Systems with Aftereffect // Open Source Journal. The Scientific World Journal, vol. 2016, Article ID 6490826, 8 pages, 2016. doi:10.1155/2016/6490826 <https://www.hindawi.com/journals/tswj/2016/6490826>.
- [18] S. Chaganti, I. Nanda, K. Pandi, T. Prudhvith, N. Kumar, "Image Classification using SVM and CNN", International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020. URL: <https://ieeexplore.ieee.org/abstract/document/9132851>.