ABSTRACT

The purpose of this article is to summarize the knowledge gained in the development and implementation of a neural network for facial recognition. Neural networks are used to solve complex tasks that require analytical calculations similar to what the human brain does. Machine learning algorithms are the foundation of a neural network. As input, the algorithm receives an image with people's faces, then searches for faces in this image using HOG (Histogram of oriented gradients). The result is images with explicit face structures. To determine unique facial features, the Face landmark algorithm is used, which finds 68 special points on the face. These points can be used to center the eyes and mouth for more accurate encoding. To get an accurate “face map” consisting of 128 dimensions, you need to use image encoding. Using the obtained data, the convolutional neural network can determine people's faces using the SVM linear classifier algorithm.

Keywords: face recognition; method; algorithm; neural network; machine learning; deep learning; histogram; directed gradients; affine transformations.

1. Introduction

The interest in face recognition and recognition processes has always been significant, especially in connection with increasing practical needs: security systems, credit card verification, forensic analysis, teleconferences, etc. Despite the clarity of the everyday fact that a person is good at identifying people's faces, it is not obvious how to teach a computer to do this, including how to decode and store digital images of faces. Human authentication technology based on facial recognition, unlike technologies based on fingerprint scanning, iris or retina, does not require physical contact of the person with the device. Based on the rapid development of digital technology, human authentication based on the image of the face is the most acceptable for mass use. In turn, neural networks are one of the most common methods of facial recognition.

Neural networks are trained using various training examples. The essence of training is reduced to correcting the weights of inter-neural connections in the process of solving an optimization problem using the gradient descent method. During the study, key features are automatically identified, their significance is determined, and the relationship between them is built. It is assumed that, the trained NN will be able to apply the experience gained in the learning process to unknown images by generalizing abilities.

Various machine learning algorithms are used to create a neural network. After conducting an empirical study of the most popular algorithms, the most optimal ones were selected.

2. System of classification and recognition of human face

To solve these tasks, the following main steps can be identified:

2.1 Search for all faces in the photo: this is necessary in order to select the area of the image that is passed to further processing. A HOG (Histogram of Oriented Gradients) is used for this purpose[1].

According to this algorithm, the image is converted into black and white format, because color data is not needed to search for faces. The loop looks at each pixel and its neighboring pixels in order to find out how dark the current pixel is compared to the surrounding ones. Then an arrow is added to indicate which direction the image is getting darker. After performing this procedure for each individual pixel in the image, each pixel is replaced with an arrow. These arrows are called gradients, and they show the direction from light to dark pixels throughout the image. If you use dark and light images of the same person, the pixels will have different brightness values, but when you consider the direction of brightness changes, you get the same image regardless of the brightness of the original image. To save resources and get rid of redundant information, the image is divided into blocks of 16x16 pixels. Then replace this square in the image with arrows pointing in the same direction as the majority[2].

Eventually, the original image is transformed so that the basic structure of the face is clearly visible. To find a face in a HOG image, you need to find the part of the image that most closely resembles the well-known HOG drawing obtained from a variety of other faces during training. An example of a fragment that is highlighted in an image is shown in Figure 1.
2.2 Face position: to make it easier for the computer to work with faces turned, you must convert each image so that the eyes and lips are always in a specific place. To solve this problem, we use the face landmark estimation algorithm [4]. The basic idea is that there are 68 special points (called landmarks) that exist on each face – the upper part of the chin, the outer point of each eye, the inner point of each eyebrow, and so on. Then a machine learning algorithm is trained to find these 68 special points on any face. Figure 2 and 3 show the selected face landmarks.
When the eyes and mouth are defined in an image, you can rotate, zoom, and shift it so that the eyes and mouth are centered as best as possible. For centering, only basic image transformations are used, such as rotation and scaling, which preserve parallel lines- affine transformations.

This method of image transformation is based on the following principle: for each point in the final image, a fixed set of points in the source image is taken and interpolated according to their relative positions. The affine transformation is the most General one—to mapping of a plane to a plane, which preserves straight lines and the ratio of lengths of segments lying on a single line. After this transformation, the face is centered in approximately the same position in the image, which makes the next step more accurate.

2.3 Face encoding: the face image obtained after the first steps must be "encoded" in a unique way, since comparing the resulting image with all the previous ones is an irrational approach when there is a large database of images. So we need a way to take a few basic measurements from each face, which we can compare with the closest known measurements and find the most similar face. In 1960, Woodrow Bledsoe proposed an algorithm that identifies obvious facial features. However, these measurements for the human brain (eye color, nose size, and so on) do not really make sense for a computer that looks at individual pixels in an image. Researchers have shown that the most accurate approach is to let the computer measure what it needs. Deep learning, determines which parts of the face to measure, better than people[6].

To solve this problem, a convolutional deep learning neural network is created that generates 128 dimensions (face map) for each person (Figure 4). The idea of converting images into a list of computer-generated numbers is extremely important for machine learning (especially for automated translation).
The algorithm of image analysis:
3 images are analyzed to train the network:
1. Training image of a famous person's face
2. Another photo of the same famous person
3. An image of a completely different person
The algorithm then looks at the measurements it makes for each of these three images. Then it tweaks the neural network a little to make sure that the dimensions created for images #1 and #2 are more similar, and the dimensions for #2 and #3 are less similar. A schematic representation of the algorithm is shown in Figure 5.

A single 'triplet' training step:

![A single 'triplet' training step](image)

After repeating this step m times for n images of different people, the neural network is able to reliably create 128 characteristics for each person. Any 10-15 different images of the same person may well give the same characteristics.

The algorithm allows you to search the database for an image that has characteristics that are closest to the characteristics of the image you are looking for[8].

By training a network with different images of the same person, the algorithm creates approximately the same dimensions. The resulting 128 dimensions are called a face map. The idea of converting an array of raw data, such as an image, into a list of computer-generated numbers is extremely important for machine learning (especially for automated translation).

2.4 Search name by face map: the SVM linear classifier algorithm (Support Vector Method) is used to search for a person's name on a face map. This algorithm is based on the concept of the solution plane. The plane divides the received data with different classes. One of the main advantages of SVM is the high speed of learning the algorithm, and, consequently, the ability to use a fairly large amount of source data for training. Using this method solved the problem of binary classification, in this example determined the identity photographs to the existing sample. To do this, you need to train a classifier that takes measurements from the image being checked and shows who this person is most similar to from the entire training sample of the system.

Other classifiers include the RVM - Relevance Vector Machine method. In contrast to SVM, this method gives the probabilities with which the face image belongs to a given person. i.e., if SVM says "the image belongs to person 1", then RVM will say "the image belongs to person 1 with probability p and class person 2 with probability 1-p" [9]. In addition, the reference vector method is unstable with respect to noise in the source data. If the training sample contains noise emissions, they are significantly taken into account when constructing the dividing hyperplane. The method of relevant vectors does not have this disadvantage.
Despite the existence of various algorithms, it is possible to distinguish a General scheme of the face recognition process as shown in Figure 6.

Before you start the face recognition process, you must complete the following steps:

− localize faces in an image
− to align the image of the face (geometrical and luminance)
− detect signs
− face recognition itself is a comparison of the found features with the standards previously included in the database.

3. Convolutional neural network for solving the recognition problem

The pre-trained model is a convolutional neural network (CNN), a type of neural network that builds state-of-the-art models for computer vision. The CNN which is used in this article itself was trained by “Davis King” on a dataset approximately of 1.2 million images and evaluated on the ImageNet classification dataset that consist of 1000 classes. Architecture of this pre-trained model is based on ResNet-34, with 34 layers(Figure 7).
The convolutional layers of ResNet (Figure 7 - the third column) is based on the plain network (Figure 8 - the second column). But the difference with Plain Network is shortcut connections, which turns the network into a residual version of it. Shortcut connections skip one or more layers and perform ID mapping. Their outputs are added to the outputs of the layers being laid (Figure 8).
Here is simple way of using ResNet-34 for face recognition:

- Firstly, make sure that you have installed Python, OpenFace, dlib.
- Then, execute this command: `pip3 install face_recognition`.
- After that you have to create 2 folders, “pictures_of_people_i_know” and “unknown_pictures”. First folder should contain people that you already know, second one should contain people that you want to identify.
- The next step is running the command `face_recognition` that passes through both folders, and tells you who is in each image:
  ```
  face_recognition ./pictures_of_people_i_know/
  ./unknown_pictures/ [11]
  ```
  The results will look like this:
  ```
  /unknown_pictures/obama.jpg, Barack_Obama.jpg
  ```
  “obama.jpg” and “Barack_Obama.jpg” are the same person.

To evaluate this model, we need to consider other ResNet models with different layers. The ResNet CNN model could contain 152 layers. In this research, TensorFlow ResNet Library is used for its simple deployment and for obtaining accuracy. An ultimate training accuracy of 0.8651 and validation of 0.8519 was discovered in the model. Experimentally determined ultimate measures of loss was 1.5983 whereas validation loss was 1.006. Table 1 provides a summary results of the analyzed ResNet models. Next, Figure 9 and Figure 10 summarize time accuracy and compare training and testing accuracy of the analyzed models.

```
<table>
<thead>
<tr>
<th>Model</th>
<th>Training accuracy</th>
<th>Validation accuracy</th>
<th>Loss</th>
<th>Validation Loss</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet18</td>
<td>0.83</td>
<td>0.87</td>
<td>1.0436</td>
<td>1.006</td>
<td>2701s</td>
</tr>
<tr>
<td>ResNet34</td>
<td>0.8651</td>
<td>0.8519</td>
<td>1.5983</td>
<td>1.7510</td>
<td>4800s</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.8662</td>
<td>0.8095</td>
<td>4.3967</td>
<td>4.5914</td>
<td>5580s</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.8594</td>
<td>0.7884</td>
<td>8.4274</td>
<td>8.7475</td>
<td>6112s</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.8798</td>
<td>0.8836</td>
<td>11.943</td>
<td>12.05</td>
<td>9248s</td>
</tr>
</tbody>
</table>
```

Table 1 - Comparison of ResNet models [13]
Figures 11-15 show screenshots, code that demonstrate the developed system.
After making face comparisons more strict by setting lower tolerance to 0.54, we got good accuracy, you can see it in Figure 13, there is no Bill Gates look alike person, but in Figure 12 that person shown. That is because of default value of tolerance settled as 0.6 in Figure 12. By making this value lower, you will get more better results.

Figure 13 – The results of setting lower tolerance

Figure 14 – Example of facial recognition code for group photos
5. Conclusion
In the course of the research, a full-fledged convolutional neural network was implemented that can solve the task set for it. The proposed classification algorithm is simple to implement, which makes it possible to effectively recognize people’s faces in images or in real-time video. The recognition of faces was implemented through certain algorithms mentioned above that were also observed to have effective results. The first algorithm, that is HOG, includes searching for faces in the given image. The second one, Face landmark, is used for finding 68 special points on the face. The next algorithm deals with constructing a “face map”, consisting of 128 dimensions. Lastly, SVM linear classifier algorithm can help CNN to determine people’s faces.

The current implementation of the facial recognition algorithms is a scalable solution, so the results can be used to build complex functional systems in any area where it may be necessary to solve this kind of problem.

The developed model can be used in photos or in real time applications or special solutions for embedded systems.

References