**Stanisław OSOWSKI[2], Krzysztof SIWEK[1]**

Warsaw University of Technology (1), Military University of Technology (2),

doi:10.15199/48.2020.03.31

# CNN application in face recognition

***Abstract.*** *The paper presents application of the convolutional neural network (CNN) in face recognition. The CNN is regarded nowadays as the most efficient tool in image analysis. This technique was applied to recognition of two databases of faces: the own base containing 68 classes of very different variants of face composition (grey images) and 244 classes of color face images represented as RGB images (MUCT data base). This paper will compare different solutions of classifiers applied in CNN, autoencoder and the traditional approach relying on classical feature generation methods and application of support vector machine classifier. The numerical results of experiments performed on the face image database will be presented and discussed.*

***Streszczenie*** *Praca przedstawia zastosowanie sieci CNN w rozpoznaniu obrazów twarzy. Twarze poddane eksperymentom pochodzą z dwu baz danych. Jedna z nich jest własną bazą zawierającą 68 klas reprezentowanych w postaci obrazów w skali szarości i drugą (MUCT) zawierającą 244 klasy reprezentujące obrazy kolorowe RGB. Zbadano i porównano różne metody rozpoznania obrazów. Jedna z nich polega na zastosowaniu konwolucyjnej sieci neuronowej CNN z dwoma różnymi klasyfikatorami końcowymi (softmax i SVM). Inne głębokie podejście stosuje autoenkoder do generacji cech i SVM jako klasyfikator. Wyniki porównano z klasycznym podejściem wykorzystującym transformację PCA w połączeniu z klasyfikatorem SVM.* ***Zastosowanie sieci CNN w rozpoznaniu obrazów twarzy***

**Słowa kluczowe:** CNN, transfer learning, obrazy widzialne, rozpoznawanie twarzy, transformacje danych, klasyfikacja.
**Keywords:** CNN, transfer learning**,** visible imagery, face recognition, transformation of data, classification.

## Introduction

A face recognition system is aimed to identifying or verifying a person from a set of digital images or a video sources. It finds applications in many different areas [1,2,3], such as payments (customers open the application to confirm a payment using their camera), access and security (instead of using passcodes the person will be granted access via his face image), criminal identification or even a healthcare (medical professionals could identify illnesses by looking at patient's features).

The general problem in this paper is formulated as follows: given the image of person, identify or verify him using a stored database of faces. The most often used methods nowadays apply artificial intelligence, especially artificial neural networks, cooperating with some preprocessing steps. Traditional methods split the problem into two tasks: 1) generation of proper set of numerical descriptors of the image and 2) use the developed descriptors as the input attributes to the final classifier, responsible for pattern recognition and classification [2,4]. Such approach to the problem needs some additional knowledge how to generate the efficient set of diagnostic features.

However, nowadays the most popular is application of deep learning, in which the multilayer structure performs automatically these two tasks without inference of human into the processing stages. The user simply delivers the raw image to the input of such system and get final result in the form of class membership on the output. This is a great simplification of problem from the user point of view. Moreover, the experiments have shown, that such approach not only simplifies the task, but also allows obtaining better accuracy of pattern recognition.

This paper will compare different variants of application of convolutional neural network (CNN) in recognition of facial images. CNN is nowadays the most important tool in image processing [5,6]. Our work will be based on so called transfer learning approach, in which the pre-trained CNN network is adapted to the particular task of image recognition. We compare the efficiency of such system in recognition of faces taken from two bases: one (individually prepared by us) is composed of 68 families of faces, each family member represented by 25 images, and the second – the base MUCT [7] containing families of 244 individuals, each family represented by 15 images. Two different final classifiers cooperating with the locally connected layers will be applied and compared: the so called softmax (the typical probabilistic classifier built-in the CNN structure) and classical support vector machine (SVM) of Gaussian kernel adapted to this particular task.

## CNN structure - illustration of activations in different layers of CNN

CNN is a deep multilayer neural network, which combines in one structure two functions: automatic unsupervised generation and selection of diagnostic features of the images and final classification [4]. The first layers are locally connected by applying the operations of convolution, nonlinear ReLU (Rectified Linear Unit) activation, pooling and normalization. The data are organized in the form of tensors, where the first two dimensions represent width and height of the images and the third one - the succeeding images. These layers are responsible for generation of diagnostic features. Finally, the tensors are converted to vector form, which starts fully connected structure. The elements of this vector represent the input attributes to the classifier, which may be incorporated in the final CNN structure in the form of so called softmax or any classical classifier, for example SVM. We have chosen SVM as the most successful form of classification system.

The characteristic fact of CNN is that succeeding convolutional layers represent the primitive features found out in the image. They represent the preserved spatial relationship between pixels, by learning the features corresponding to small squares of image regions. By applying different filters (kernels) it is possible to detect different types of primitive features existing in the image, for example blobs, edges, crossing points, curves at different locations and color representation.

As CNN goes deeper into succeeding layers, the kernels in these layers try to build its own abstract features based on the features found in the preceding layer. In this way deep layers can capture the global high-level features, which are used as the input attributes to fully connected layers of CNN, responsible for final recognition of patterns.

Fig. 1 illustrates the succeeding steps of image processing in different levels of the CNN structure [8].

Graphical results of convolutional layer 1, illustrated in Fig. 1a by 30 channels, represent the most primitive features in the form of some blobs, different distribution of colors, etc. Applying different filter parameters (kernels) we

get different feature maps. This diversity is reflected well in the form of 30 different sub-images in figures 1a to 1c. The succeeding layers of local connections (Fig. 1b and 1c) represent the features created as the combinations the features from the preceding layers.

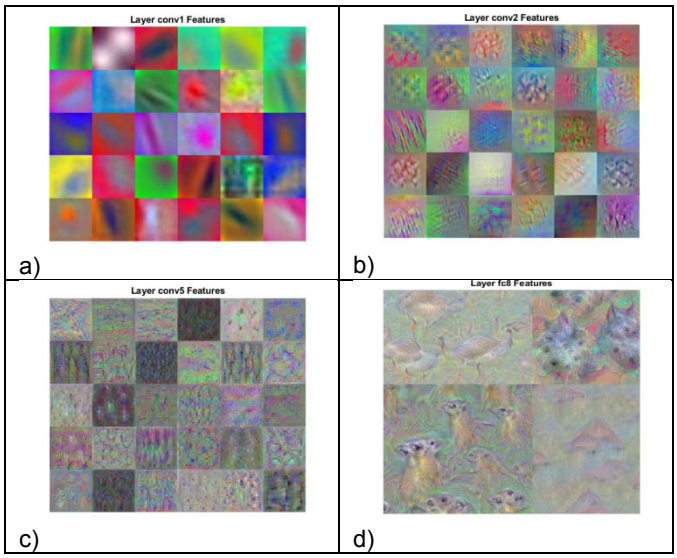


Fig. 1 The illustration of activations of different layers in CNN. The images are presented after linear convolution in: a) the first layer, b) the second layer, c) the fifth layer, d) the last fully connected layer.

The four images depicted in Fig. 1d represent responses of CNN in the form of four object images in the last fully connected layer (FC8). They are responses to 4 different input excitations (images of: goose, Scotch terrier, meerkat and boathouse). The figures have been reconstructed from 1000 elements of the output layer of the network. The images resemble the shape of 4 animal images applied to the input of CNN and correspond to the learning base (goose, Scotch terrier, meerkat and boathouse). With different distorted input images, CNN successfully captures the representative patterns. CNN is able to learn a global pattern representatives rather than local details.

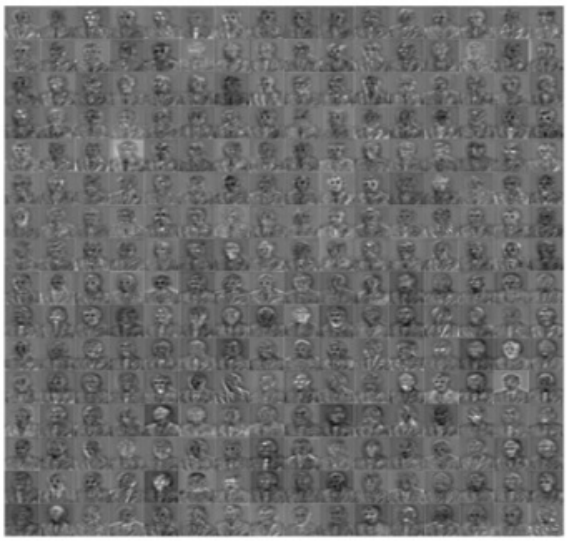

Fig.2 The illustration of effect of linear convolution of the image (256 images of the same person of fifth convolutional layer)

This is also well illustrated in Fig. 2, representing the output of 256 images of the same person in fifth convolutional layer, obtained in ALEXNET [8,9] as a results of application of 256 kernels. It presents direct results of linear convolution (in grey scale). The 256 images forming the tensor represent diagnostic features of the images. They characterize different points of view of the image, which have been provided by different kernels. Thanks to such multiple representation the total information provided by all images is enhanced. This knowledge, applied to the final, fully connected classifier results in better efficiency and robustness to some changes in input images, following from some differences in image acquisition (different lightning conditions, variety of poses, some noise, etc.). The output layer predicts the likelihood of the possible decisions of class membership of the input data.

The most efficient way of application of CNN is to use the pre-trained network structure. The user adapts only this structure to his own task by learning only last layers applying his own data sets and leaving the first locally connected layers unchanged.

There are already many pre-trained CNN networks, available for free [10]. In actual Matlab packets (Matlab 2019a) we can find such pre-trained CNN structures as: ALEXNET (the first pre-trained implementation of convolutional neural network), googlenet, inception3, resnet18, resnet50, resnet101, inceptionresnetv2, etc. They differ by the depth, size, number of parameters and the size of input image. For example ALEXNET operates with 61 million parameters, while googlenet needs only 7 million. The numerical investigations in this paper will use the ALEXNET structure.

**Numerical experiments**

The numerical experiments have been performed using ALEXNET CNN structure available in Matlab [8] by applying the strategy of transfer learning, in which the locally connected layers were fixed and only fully connected layers were subject to adaption according to the actually applied learning data. The built-in ALEXNET structure used in experiments is as follows (in Matlab notation [8]).

1 'data' Image Input 227x227x3 images with 'zerocenter' normalization
2 'conv1' Convolution 96 11x11x3 convolutions - stride [4 4] and padding [0 0 0 0]
3 'relu1' ReLU
4 'norm1' Cross Channel Normalization cross channel normalization with 5 channels per element
5 'pool1' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6 'conv2' Convolution 256 5x5x48 convolutions - stride [1 1] and padding [2 2 2 2]
7 'relu2' ReLU ReLU
8 'norm2' Cross Channel Normalization cross channel normalization with 5 channels per element
9 'pool2' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10 'conv3' Convolution 384 3x3x256 convolutions - stride [1 1] and padding [1 1 1 1]
11 'relu3' ReLU ReLU
12 'conv4' Convolution 384 3x3x192 convolutions - stride [1 1] and padding [1 1 1 1]
13 'relu4' ReLU ReLU
14 'conv5' Convolution 256 3x3x192 convolutions - stride [1 1] and padding [1 1 1 1]
15 'relu5' ReLU ReLU
16 'pool5' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17 'fc6' Fully Connected 4096 fully connected layer
18 'relu6' ReLU ReLU
19 'drop6' Dropout 50% dropout
20 'fc7' Fully Connected 1500 fully connected layer (*subject to change*)
21 'relu7' ReLU ReLU
22 'drop7' Dropout 50% dropout
23 'fc8' Fully Connected M fully connected layer (*subject to change*)
24 'prob' Softmax softmax
25 'output' Classification Output crossentropyex with 'tench' and 999 other classes

The number of output neurons in the layer fc8 (denoted as M) is subject to changes according to the number of actually recognized classes. In transfer learning we have changed only fully connected substructure by reducing the layer neurons from 4096 to 2500 and limiting the output neurons to the already assumed number M of classes. In the first experiment the built-in Softmax was used as the classifier and in second experiment we have changed the classifier from Softmax to SVM of Gaussian kernel [11]. In

the second case the activations were taken from fc7 (layer #20 in Matlab). These 4096 signals served at the input attributes to the SVM. The assumed hyperparameters of SVM have been set to C=1000, σ=1 and one against one strategy (for example, at 68 classes it means construction of 2278 two-class recognition units, at 244 classes it is 29646 two-class units).

**Results of class recognition for the first data base**

The numerical experiments have been performed using the face images of maximum 68 persons in grey scale, treated as 68 classes subject to recognition. They have represented both males and females. Each class contained 20 photographs of the same person made in different poses and lighting conditions. The size of the original images for all persons was the same and equal 100×100. The examples of original images for 2 persons are presented in Fig. 3. The same person has been represented in different poses and at varying lighting. Some faces are with glasses and some without glasses. The faces are shown in different scale, representing either full face or only some part of it. These facts are evidence of differences among images representing the same class of persons.

The experiments of learning have been performed on 70% of the available data, leaving the other 30% to the testing purposes. 10 runs of classification process were performed, each time at different (random) selection of learning and testing part of the data base. The statistical results for different number of classes are presented in Table 1. They correspond to testing samples only. For comparative reasons we have also shown the results at application of traditional PCA [4,12] cooperating with SVM of Gaussian kernel and the application of other form of deep learning by using autoencoder [1].

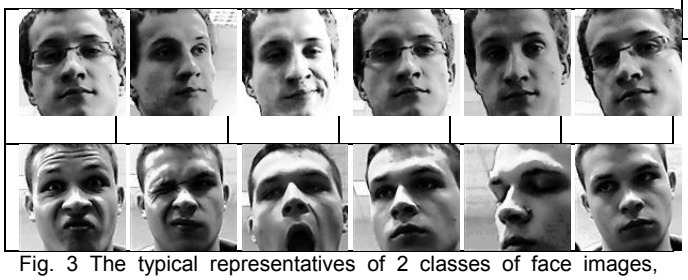

Fig. 3 The typical representatives of 2 classes of face images, which are subject to recognition.

Table 1 The comparative average accuracy results in face recognition (testing mode) for different number of classes.

| Classes | PCA+SVM | Autoencoder | CNN+Softmax | CNN+SVM |
|---|---|---|---|---|
| 20 | 96.6% | 96.7% | 97.5% | 100% |
| 51 | 86.5% | 90.4% | 95.7% | 99.5% |
| 68 | 81.3% | 88.6% | 94.1% | 99.1% |

Application of CNN with built in softmax classifier has resulted in 94.12% average class accuracy on the testing data for 68 classes. Reduction of the number of classes has increased the efficiency of recognition. The observed results were better at reduced number of classes (97.5% for 20 classes and 95.7% for 51 classes). Standard deviation obtained in 10 runs of classification process at random choice of learning and testing data was only 1.33% in the case of 68 classes.

Changing final classifier to SVM has allowed to obtain accuracy close to 100% on the same testing sets irrespective of the number of classes. It is an evident advantage of SVM over simple Softmax solution. However, this was paid by much longer learning of the system (instead of around 3 minutes for Softmax the system needed around 10 minutes for SVM, all on PC computer with GPU. Both results obtained by CNN represent the great advantage in comparison to autoencoder or application of classical approach based on PCA and SVM [1].

**Results of class recognition for the second data base**

To find the dependence of CNN classifier on the number of classes the next experiments have been performed for enlarged number of classes. This time we have performed experiments on images taken from the freely available MUCT FACE DATABASE [7]. It consists of 3660 faces divided into 244 different classes. Each class was represented by 15 images of the same person, in different acquisition conditions. This time the images were represented in RGB channels. The members of the same class differ by the pose, colors of the dress and also of the background. The database has provided diversity of lighting, age of person, orientation of face, ethnicity and colors. Some chosen examples of images taken from this data base are shown in Fig. 4.

.

Fig. 4 The chosen representatives of images in MUCT database

The learning experiments for this database have been performed also on 70% of the available data, leaving the other 30% to the testing purposes. Similarly to the previous experiments 10 runs of classification process have been performed, each time at different (random) selection of learning and testing part of the data base.

The experiments have confirmed high independence of the CNN classifier on the number of recognized classes. In spite of the fact that this base contained 244 classes, the accuracy obtained was even better than in the previous experiments on the images belonging to 68 classes. This time the solution of CNN with Softmax classifier in 10 runs of program has achieved average accuracy equal 97.16% with std=2.87%. Application of SVM classifier in the final stage has resulted always in 100% of accuracy. The source of this very high efficiency may be higher similarity of images representing different classes (similar positions of face, the same dress of persons representing class, etc.)

**Conclusions**

The paper has studied the performance of CNN network with different arrangement of the final classification system. Two different final classifiers in fully connected subnetwork of CNN have been checked and compared. One has applied the probabilistic solution in the form of Softmax and the second SVM of Gaussian kernel. On the basis of experiments the advantage of SVM has been observed.

The interesting conclusion is that the number of classes was not crucial in CNN application, contrary to non-deep learning, where the number of classes was very important and decided on the efficiency of the system. The main

factor influencing the quality of CNN solution is the difference in similarity among the representatives forming the same class and differences among classes. The second base MUCT, containing persons of different ethnicity, has happened to be much easier, in spite of much larger number of recognized classes. Moreover, the color information of the images seems to be also quite important factor. Its inclusion helps the system to differentiate the representatives of different classes.

***Authors**: prof. dr hab. inż. Stanisław Osowski, Warsaw University of Technology, Institute of the Theory of Electrical Engineering, Measurement and Information Systems, Military University of Technology, Institute of Electronic Systems, Email: sto@iem.pw.edu.pl.*
*dr hab. inż. Krzysztof Siwek, Warsaw University of Technology, Institute of the Theory of Electrical Engineering, Measurement and Information Systems, Email: ksiwek@iem.pw.edu.pl.*

REFERENCES

[1] Siwek K., Osowski S., Autoencoder versus PCA in face recognition, Conference *"Computational Problems of Electrical Engineering"* (CPEE), Kutna Hora, 2017.
[2] Kasar M. M., Bhattacharyya D., Kim T. H., Face recognition using neural network: a review, International Journal of Security and Its Applications, vol. 10, no 3, pp. 81-100, 2016.
[3] Kim K. I., Jung K., Kim H. J., Face recognition using kernel principal component analysis, IEEE Signal Process. Lett., vol. 9, no 2, 2002.
[4] Tan P. N., Steinbach M., Kumar V., Introduction to data mining, Pearson Education Inc., Boston, 2006.
[5] Goodfellow I., Bengio Y., Courville A., Deep learning, MIT Press, 2016.
[6] Zeiler M. D., Fergus R., Visualizing and understanding convolutional networks, European Conference on Computer Vision, pp. 818-833, 2014.
[7] Milborrow S., Morkel J., Nicolls F., The MUCT landmarked face database, Pattern Recognition Association of South Africa, 2010, http://www.milbo.org/muct.
[8] Matlab user manual, MathWorks, Natick, USA, 2019a.
[9] Krizhevsky A., Sutskever I., Hinton G., Imagenet classification with deep convolutional neural networks, NIPS, 2012.
[10] https://github.com/BVLC/caffe/tree/master/models/bvlc_google net.
[11] Schölkopf B., Smola A., Learning with kernels, Cambridge, MIT Press, MA, 2002.
[12] Siwek K., Osowski S., Deep neural networks and classical approach to face recognition – comparative analysis, Przegląd Elektrotechniczny, vol. 94, no 4, pp. 1-4, 2018.