sciendo

VOLUME 19, N° 1 2025
Journal of Automation, Mobile Robotics and Intelligent Systems

Convolution Neural Network for Face Similarity Estimation

Submitted: 4th September 2024; accepted: 5th October 2024

Wojciech Domski, Adam Jankowiak

DOI: 10.14313/jamris-2025-007

Abstract:
We present a convolution neural network used to determine face similarity given two images as input, i.e. a face identification task. The main focus is on the shape of the input data. We propose schemes where two pictures are connected in four different ways. The input sample is concatenated horizontally and vertically, giving the first two schemes. The other two input shapes include the intertwining by column and by row. Analysis of precision versus recall has been provided for each input schema. Some of the traditional approaches focus on deriving the feature vectors of an individual and then comparing the obtained vectors with each other. Our paper offers a new approach to face identification problems where two images of an individual are directly fed to the neural network. Then, it is the task of the neural network to determine the similarity score.

Keywords: *CNN, face identification, input shaping*

1. Introduction

Solutions created with artificial intelligence are gaining more and more attention. For example, ChatGPT is an optimised dialogue model while offering a human-like conversation experience. AI finds applications in many different domains such as engineering, medicine and finance. The use of artificial intelligence is much broader and is not limited to the fields mentioned. A particular field where such methods have proven to be useful is computer vision, especially face detection and identification [6]. This field is not solely reserved for human face recognition but can be applied to animals as well [7]. One of the methods in computer vision that can be used for face recognition is the Haar Cascade Classifier [15]. It is a classic object detection algorithm based on Haar-like features and the Adaboost algorithm. It can be adopted to the face detection task by training it on face and non-face images. It is efficient, but it requires a pre-trained database, so it is not able to identify faces that the algorithm has not seen previously. Another classical approach to face detection and identification is the Eigenface detector [14]. In the course of this algorithm, a face is projected onto "face space" where deviations from a normalised patterns are measured. This approach uses principal component analysis (PCA) to extract features from images and then uses those features to perform face recognition.

Additionally, this algorithm can be extended to recognize previously unseen faces through an unsupervised learning process. A similar approach to Eigenfaces is known as Fisherfaces [2]. Its projection method is based on Fisher's linear discriminant and gives results in a low-dimensional subspace that are well separable. Research on this method showed that it is invariant or presents high robustness to illumination conditions.

A separate group of algorithms is based on neural networks such as ImageNet [5], MTCNN [16], VGGFace [8] and ResNet [4]. These solutions are based on deep convolutional neural networks. ImageNet and ResNet are not directly intended to be used as a model to identify faces but rather as various object classifiers that could be adopted in order to perform this task. In turn, MTCNN and VGGFace are dedicated solutions to perform face recognition tasks. It is important to know the underlying difference between two terms: face detection and face identification. The former limits its scope to the task of finding a face in an image, while the latter is capable of comparing two faces and drawing a conclusion if the face belongs to the same person or not. While it is clear that the face identification task is harder to perform, finding a person's face on an image cannot be neglected either. Moreover, as the term face recognition gains more friction, it seems that is is being used in a simultaneous task which combines previous two. In this article we focus our effort on the face identification task. Thus we assume following that a picture containing two faces is provided. Through the inference process, the model determines probability describing if the provided image represents the same person. The main focus of the letter is put on how structure of input data impacts overall performance of a deep convolution neural network. We present four similar structures of the input data which differ in image representation and compare the final performance of the presented deep CNN.

In this paper, we are presenting results obtained during training of a deep neural network for the task of face identification. The deep neural network is provided with two photo samples. The goal is to determine whether the two pictures represent the same person. Moreover, we focus on input data shape and how it influences accuracy and performance. We have proposed four different input data schemes. The results of the trained models are presented in the form of precision vs. recall.

Open Access. © 2025 Wojciech Domski and Adam Jankowiak, published by Sciendo. BY-NC-ND This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License

Figure 1. Example faces from the ORL database [10]

In Section 2 we discuss four different image representations. Section 3 contains description of used neural network architecture. Next, Section 4 presents detailed information gathered during training of presented deep CNN model. Discussion of achieved results is presented in Section 5. Additionally, it offers conclusion and further research plans.

2. Input Data

The input data set is the driving force of deep neural network solutions. To facilitate this, we have used the ORL (Olivetti Research Labs) database [1]. This database contains photographs of 40 distinct subjects, 10 samples per person. The data set was divided into three subsets: training dataset, directly used during training process; validation dataset, to validate network performance during training; and finally testing dataset that was used during evaluation of a trained model. The original data set (before data augmentation) was divided into three parts with following shares:

- training – 32 subjects (80%),
- validation – 4 subjects (10%),
- testing – 4 subjects (10%).

Therefore, the training dataset contains exactly 320 raw samples. The samples were not directly fed to the neural network.

The size of each image is 92x112 pixels in 8-bit grey scale. The databases contain multiple images of the same subject but taken under different conditions. These conditions include different lighting conditions, facial expression, and facial details. The presented ORL database is used in various research areas [3,10]. In Figure 1, 9 randomly chosen faces were presented.

2.1. Data Preparation

Before the training dataset could be constructed, each image had to be cropped to resemble a square shape. This was achieved by cropping the original images from 92x112 to 92x92 pixels.

Afterwards, each image was scaled down to 64x64 pixels. This process was required to limit the number of inputs to the neural network. In order to satisfy deep neural network with sufficient data, the post-processed dataset was augmented. Each image was transformed with a three-stage pipeline:

- noise addition,
- rotation, and
- zooming.

This allowed us to significantly enlarge the number of samples. To each image, noise was added based on normal distribution. Additionally, each image was rotated within the range of [-5°, +5°]. The rotation angle was chosen at random. In addition, each image was slightly zoomed in or out. These operations allowed creation of an automated augmentation pipeline that produced a number of different samples.

2.2. Training Set Curation

The goal was to train a network capable of discovering if two images belong to the same person, thus performing a face identification task. This requires the neural network to be fed with not a single image but a sample consisting of two images. Each such sample is then labeled with a zero value representing the situation when two pictures do not belong to the same person and value of one otherwise. Through the process of data augmentation followed by image concatenation, we have extended our dataset considerably allowing the deep neural network to be trained properly. After this process the datasets were of the following magnitude:

- 20160 training samples (including 10080 positive and 10080 negative samples),
- 2520 validation samples (including 1260 positive and 1260 negative samples),
- 2520 training samples (including 1260 positive and 1260 negative samples).

In the scope of the classification task (the same person or two different persons), it is important to ensure that the magnitude of datasets representing positive and negative samples is comparable. Therefore, the data augmentation process allowed us to satisfy these two requirements. The first one concerning the total number of samples; a large number of training samples leads to better performance. The second condition requires that the magnitude of datasets representing each class for a classification problem is similar, ideally identical. Based on this fact, the number of positive samples (concatenation of images representing the same person) is much smaller than the number of negative samples. For original dataset the number of positive samples can be calculated as combination given as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \quad (1)$$

Figure 2. Two images concatenated horizontally

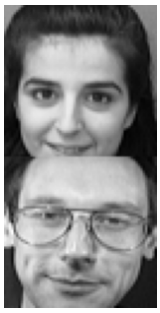


Figure 3. Two images concatenated vertically

Figure 4. Two images intertwined by row

For a single person with 10 photographs it yields 45 positive samples. Given 32 individuals the total number of positive samples is 1440 which is insufficient. In turn, the number of negative samples is far greater for the original dataset.

As already mentioned, in order to achieve face identification, two images had to be represented as a single input sample. Therefore, we propose four different sample representations. The first schema is based on concatenating images together, as it was shown in Figure 2.

The second representation is similar to the first one, but now the pictures of individuals are concatenated vertically. The result is presented in the Figure 3.

The two images were concatenated horizontally, so the resulting input shape was 128x64. In the case of vertical concatenation, the resulting resolution is 64x128. The third schema was based on intertwining pattern. See the example in Figure 4. As can be seen, the resulting image is a concatenation of two images where rows are intertwined. The resulting size of the image is 128x64 pixels (similar to vertical concatenation).

Similar to the third input data schema, we propose a representation where an input sample was created through concatenation, but columns were intertwined instead of rows. The result was show in Figure 5.

Figure 5. Two images intertwined by column

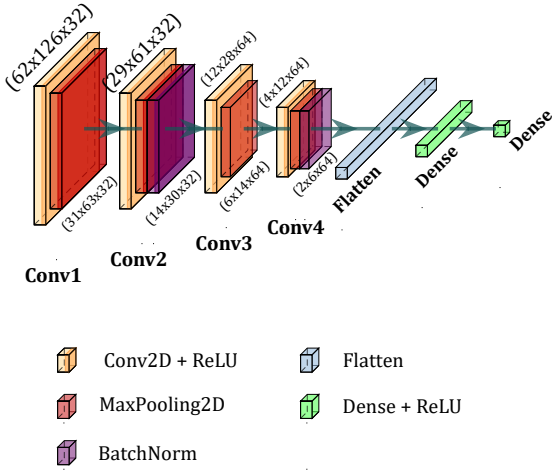


Figure 6. Face similarity convolution network architecture

The resulting resolution of the image is equal to the first representation; thus it is now 128x64 pixels.

3. Network Architecture

The convolution network architecture is presented in Figure 6. Almost the same network architecture was utilized for all four input schemes. Therefore, the input layer was 128x64 or 64x128. Similar changes had to be adopted across all convolution and pooling layers. In Figure 6 we have presented network architecture for the input layer with dimensions 64x128, thus for images which were concatenated vertically (one under the other) or intertwined by row.

It is worth noting that the ReLU activation function was used for the last output layer. Since we focus on values from 0 to 1, a different activation function could be used for the last layer, e.g. the sigmoid function could be used. However, we have observed that using ReLU as an activation layer for the network output gave good results.

4. Model Training

We have trained four models where each model differs by shaping of input samples. We present training statistics such as accuracy and loss during the training and during validation phase.

To better understand how the trained model performs, it was tested against the testing dataset since it was not used during training stage. We provide two metrics which give insight how well it performs – precision and recall. Precision is expressed as

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2}$$

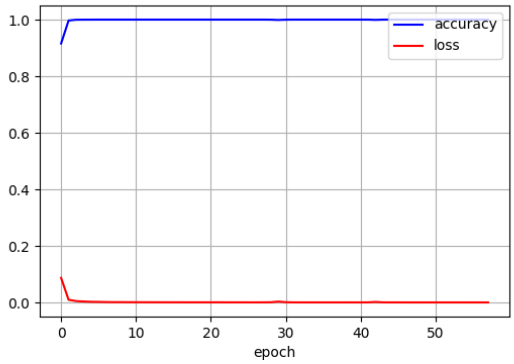


Figure 7. Accuracy and loss during training for horizontally concatenated pictures

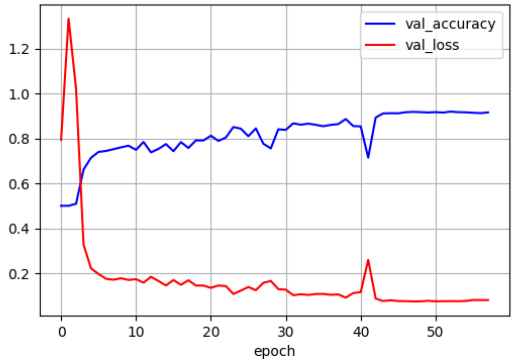


Figure 8. Validation accuracy and loss for horizontally concatenated pictures

where TP are true positives meaning positive samples that were correctly classified and FP are false positives reflecting negative samples that were misclassified as negative samples. Recall is defined as

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{3}$$

where TP is as previously, while FN are false negatives reflecting positive samples that were misclassified as negative samples. Observing how precision and recall behave in a function of assumed threshold allows us to decide which behaviour of the classifier is more desirable. Depending on the selected threshold we can decide if the model should tend to prefer strong cases while some positive samples might not be detected over more robust positive classification with some false positives.

Our first model was trained on horizontally concatenated data where pictures were stacked side-by-side. In Figure 7 a plot shows accuracy and loss function during training for the model prepared for horizontally concatenated images. In turn, in Figure 8 we show accuracy and loss values for the validation dataset.

In Figure 9 recall and precision indicators for the first model are presented. Based on the plots the balanced threshold is equal to 0.72.

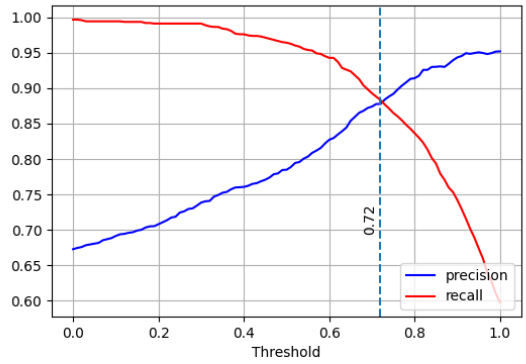
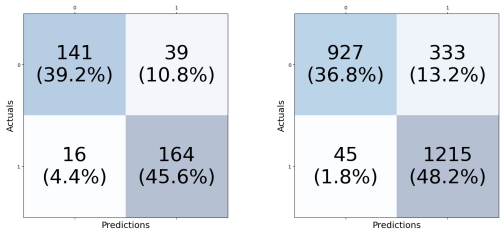


Figure 9. Precision and recall for horizontally concatenated pictures



(a) Original test dataset (b) Augmented test dataset

Figure 10. Confusion matrices for model trained on horizontally concatenated images

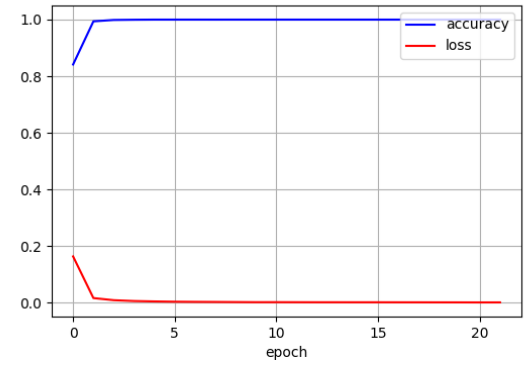


Figure 11. Accuracy and loss during training for horizontally concatenated pictures

In Figures 10(a) and 10(b) confusion matrices have been presented for our first model. As can be seen, results obtained for the original test and the augmented test datasets are similar. Overall accuracy for this model is around 85%.

The second model was trained for the same dataset but the input data was post processes by concatenating two pictures vertically, thus one picture is above the other as already been presented in Figure 3. The results of training process were depicted in Figure 11 were accuracy vs. loss function was presented. To get more insightful data we have presented accuracy vs. loss function in Figure 12.

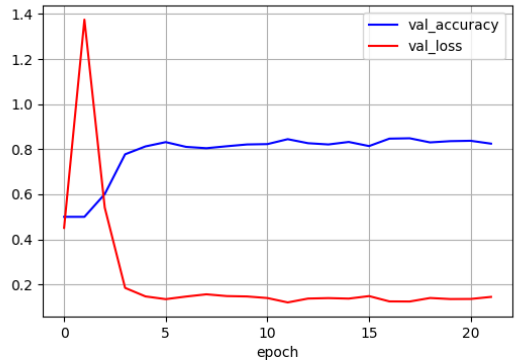


Figure 12. Validation accuracy and loss for horizontally concatenated pictures

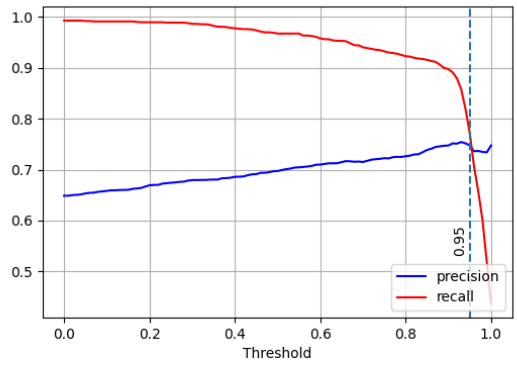
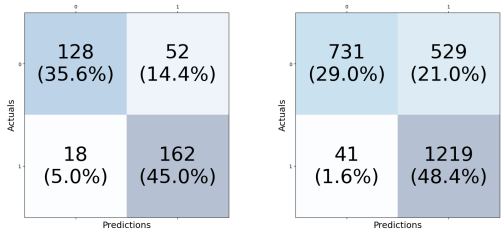


Figure 13. Precision and recall for horizontally concatenated pictures



(a) Original test dataset (b) Augmented test dataset

Figure 14. Confusion matrices for model trained on vertically concatenated images

Finally, the precision and recall were presented in Figure 13 where these two curves were calculated in a function of fixed threshold from 0 to 1. The best threshold – providing balance between recall and precision is 0.95.

In Figures 14(a) and 14(b) confusion matrices have been presented for our second model trained on vertically concatenated images. It can be noticed that the overall performance of this model is around 80%. It is slightly less compared to previous model using horizontal image concatenation.

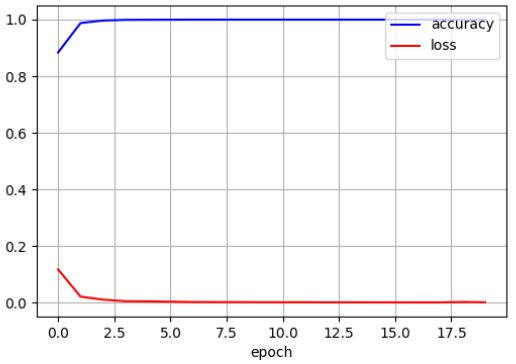


Figure 15. Accuracy and loss during training for pictures intertwined by row

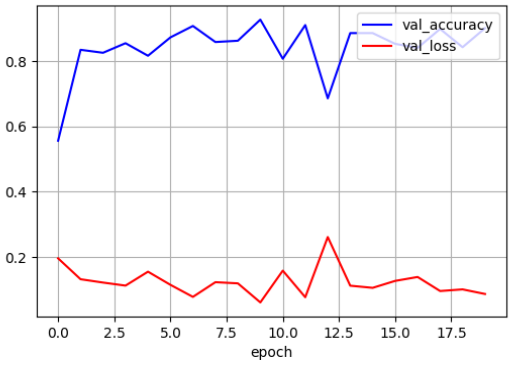


Figure 16. Validation accuracy and loss for pictures intertwined by row

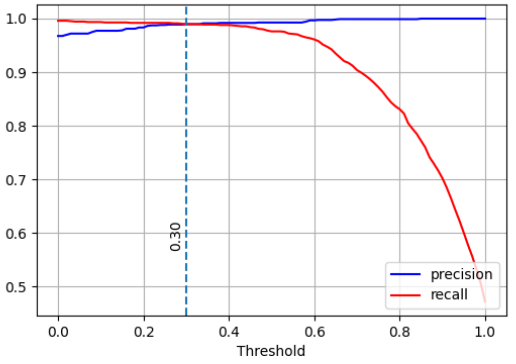
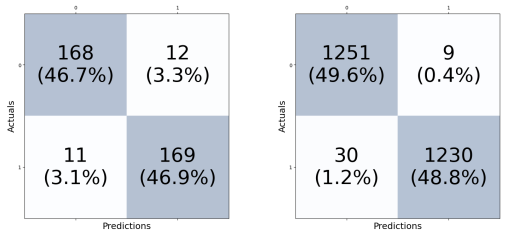


Figure 17. Precision and recall for pictures intertwined by row

The second group of models was tested on intertwined input data, the third model (pictures intertwined by row) and the last model (pictures intertwined by column). In Figure 15 accuracy vs loss was shown during training process. Validation results were presented in Figure 16 showing accuracy and loss for validation dataset.

In turn, in Figure 17, precision and recall for the model with intertwined rows are presented. The balanced threshold between recall and precision for the model was computed as 0.30.



(a) Original test dataset (b) Augmented test dataset

Figure 18. Confusion matrices for model trained on row intertwined images

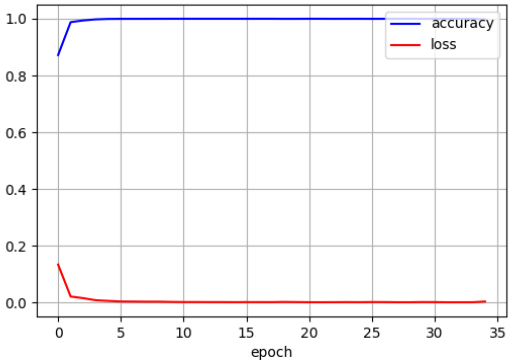


Figure 19. Accuracy and loss during training for pictures intertwined by column

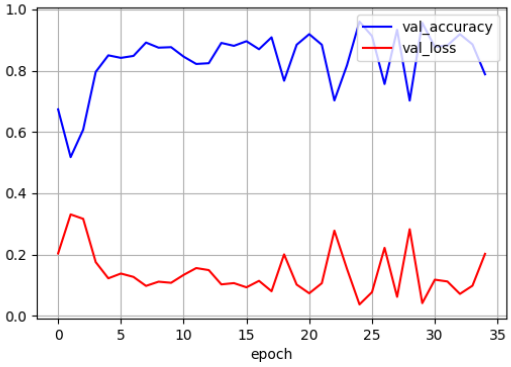


Figure 20. Validation accuracy and loss for pictures intertwined by column

In Figures 18(a) and 18(b) we show results in the form of confusion matrices obtained while testing model trained on row intertwined images. It was tested on original and augmented datasets. There is visible, however small, ca. 5%, discrepancy between results obtained for these two datasets. This model is capable of achieving accuracy on the level of 93%.

Similar, to the previous model, we present results obtained for input data that was originally intertwined by column. In this case, the training results were presented in Figure 19. Validation process was conducted also on a previously unseen validation dataset. The results were presented in Figure 20.

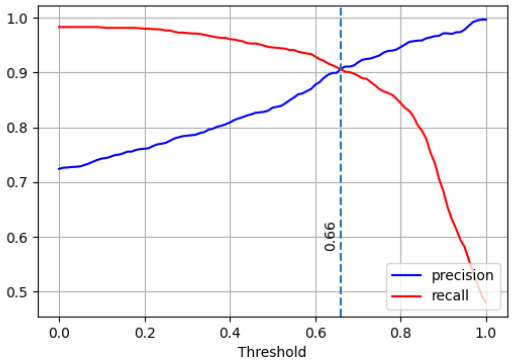
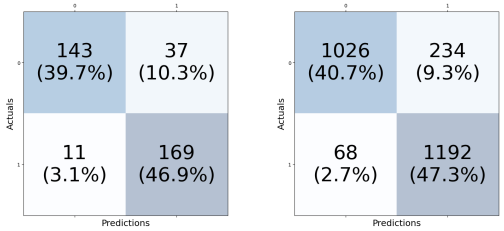


Figure 21. Precision and recall for pictures intertwined by column



(a) Original test dataset (b) Augmented test dataset

Figure 22. Confusion matrices for model trained on column intertwined images

To complement the analysis, we also present recall and precision metrics achieved on a testing dataset (Figure 21). For this model the balanced threshold is equal 0.66.

We present confusion matrices (Figures 22(a) and 22(b)) obtained during the testing routine for a model trained on column intertwined images. As in previous cases the difference between a model tested on original and augmented datasets is small, but larger than in the case of models trained on concatenated images. The overall performance of this model is around 87%, which is higher compared to models trained on concatenated images.

5. Conclusion

In the presented work we have focused on a well-know problem of face recognition. There are many state-of-the-art methods and algorithms that focus on this problem [9, 11–13]. Usually, the method is based on obtaining a numerical representation of a face that is then compared to other representations using a predefined metric. In our approach we investigate if a CNN can learn to determine if input data, consisting of two pictures, represents the same person. To further examine this we have proposed four different schemes that differentiate in input data shape. Two images are concatenated (vertically or horizontally) or intertwined (by row or by column). No similarity metric is being computed other than the output of the convolution neural network itself.

The proposed architecture allowed us to build a model that is capable of differentiating between two photographs of human faces and determining if the images represent the same person. Research results provided in Section 4 show that the input preprocessing has significant influence on the model performance. It was shown that when the input image is intertwined, either by row or column, it achieves better classification results. The best model achieving 93% accuracy was trained on intertwined images by row. What is more, all models were tested on two datasets – an original and an augmented testing dataset. It is clear that models trained on intertwined images have a higher accuracy dispersion. It is around 3-5% while for models trained with concatenated images the accuracy dispersion is 0.5%-3%. During the course of the research it was determined that the intertwined input data performs better than concatenated input data. It was shown how input data is shaped has significant impact on model performance. It was determined by providing exactly the same training sets for each model and utilizing the same architecture layout.

Since the system is not based on metric calculation it could be deployed immediately on premises without additional adjustments. The model provided with two images (intertwined or concatenated) is able to provide a prediction if the two images belong to the same person. What is more, it does not require a preexisting database of subjects since it compares two images, so its maintenance is low. However, the drawback of such systems is the necessity of running inference between all subjects, while a metric based system could be more efficient in terms of calculating only the distance between two numerical representations of faces.

In future research, we plan to significantly enlarge the original data set. This might allow us to discover potential discrepancies. Additionally, we plan to introduce a different variant of the proposed neural network. It would be trained not to identify if two pictures represent the same person but if there is any level of kinship and later on what kind of kinship level it is. It is a challenging task which would require a large data set and a specific one. Furthermore, since neural networks require a significant amount of data it would be worth to investigate if artificially augmented dataset, by the means of generative adversarial networks (GANs), influences accuracy of such architecture.

AUTHORS

Wojciech Domski* – Department of Cybernetics and Robotics, Wrocław University of Science and Technology, ul.Janiszewskiego 11/17, 50-372 Wrocław, Poland, ORCID: 0000-0001-5768-8051, e-mail: wojciech.domski@pwr.edu.pl, www: edu.domski.pl.

Adam Jankowiak – ORCID: 0009-0009-1587-7394, e-mail: adamjankowiak4@gmail.com.

*Corresponding author

References

[1] AT&T Laboratories Cambridge. "The database of faces", 2023. https://cam-orl.co.uk/facedatabase.html.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997, 711–720, 10.1109/34.598228.

[3] A. Eleyan and H. Demirel, "Pca and lda based face recognition using feedforward neural network classifier". In: B. Gunsel, A. K. Jain, A. M. Tekalp, and B. Sankur, eds., *Multimedia Content Representation, Classification and Security*, Berlin, Heidelberg, 2006, 199–206.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, 770–778, 10.1109/CVPR.2016.90.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Commun. ACM*, vol. 60, no. 6, 2017, 84–90, 10.1145/3065386.

[6] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology", *IEEE Access*, vol. 8, 2020, 139110–139120, 10.1109/ACCESS.2020.3011028.

[7] X. Li, Y. Xiang, and S. Li, "Combining convolutional and vision transformer structures for sheep face recognition", *Computers and Electronics in Agriculture*, vol. 205, 2023, 107651, https://doi.org/10.1016/j.compag.2023.107651.

[8] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition". In: X. Xie, M. W. Jones, and G. K. L. Tam, eds., *Proceedings of the British Machine Vision Conference (BMVC)*, 2015, 41.1–41.12, 10.5244/C.29.41.

[9] B. S. Peng Lu and L. Xu, "Human face recognition based on convolutional neural network and augmented dataset", *Systems Science & Control Engineering*, vol. 9, no. sup2, 2021, 29–37, 10.1080/21642583.2020.1836526.

[10] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification", *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 1994, 138–142, 10.1109/ACV.1994.341300.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 815–823, 10.1109/CVPR.2015.7298682.

[12] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification". In: Z. Ghahramani, M. Welling,

C. Cortes, N. Lawrence, and K. Weinberger, eds., *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 1701–1708, 10.1109/CVPR.2014.220.

[14] M. Turk and A. Pentland, "Face recognition using eigenfaces". In: *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, 586–591, 10.1109/CVPR.1991.139758.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, I–I, 10.1109/CVPR.2001.990517.

[16] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks", *IEEE Signal Processing Letters*, vol. 23, no. 10, 2016, 1499–1503, 10.1109/LSP.2016.2603342.