Comparisons of Facial Recognition Algorithms Through a Case Study Application

https://doi.org/10.3991/ijim.v14i14.14997

Amir Dirin (✉), Nicolas Delbiaggio, Janne Kauttonen
Haaga-Helia University of Applied Science, Helsinki, Finland
amir.dirin@haaga-helia.fi

Abstract—Computer visions and its applications have become important in the contemporary life. Hence, researches on facial and object recognitions have become increasingly important both from academicians and practitioners. Smart gadgets such as smart phones are nowadays capable of high processing power, memory capacity, along with high resolutions camera. Furthermore, the connectivity bandwidth and the speed of the interaction have significantly impacted the popularity of mobile object recognition's applications. These developments in addition to computer vision's algorithms advancement has transferred object's recognitions from desktop environments to mobile world. The aim of this paper to reveal the efficiency and accuracy of the existing open source facial recognition algorithms in real-life setting. We use the following popular open source algorithms for efficiency evaluations: Eigenfaces, Fisherfaces, Local Binary Pattern Histogram, the deep convolutional neural network algorithm and OpenFace. The evaluations of the test cases indicate that among the compared facial recognition algorithms the OpenFace algorithm has the highest accuracy to identify faces. The findings of this study help the practitioner on their decision of the algorithm selections and the academician on how to improve the accuracy of the current algorithms even further.

Keywords—Facial Recognition Algorithms; OpenFace; Mobile Facial recognitions.

1 Introduction

We deal with photos and images regularly through our smart phones. These devices and images are seen and taken everywhere, e.g., in streets, in supermarkets, and in many other public locations. We use cameras for various purposes such as security [1], health [2] . In addition, athletics use cameras such as GoPro's [3] for recording extreme sports and share the excitement with others either as live stream or offline. Furthermore, the use of camera has extended to the overcome some of the human constrain such as helping the color blind people [4]. In addition to the hand-held cameras for offline and online detection ,e.g. [5], the usage of drone based cameras has increased [6]. Drones are used to take and compare images especially for areas that are not easy to explore.

The usage of cameras has also been increased through mobile applications for entertainment such as Snapchat application [7]. Snapchat uses the computer vision algorithm for entertainment in mobile phones. In Snapchat, user is able to detect face through camera and use effects to manipulate the face (e.g., extend and re-shape). Furthermore, social media applications such as Facebook uses object recognition widely in their platform [8] and in Instagram [9]. Facebook is able to detect faces on pictures and recognize individual persons which can be then automatically tagged [10]. Another well-known example is Google search engine; the engine is able to automatically detect the content of a picture and show related pictures without need for manual tagging. These applications are based on machine learning [11] and computer vision algorithms (Rohrbach et al. 2016).

Training an accurate image recognition models, such as those by Facebook and Google, requires large amounts of training data, which is one of the main reasons behind their success. These companies collect enormous amounts of data from their users [13]. However, the algorithms behind the models are typically publicly available and open-source. For example, Google has published their facial recognition model named FaceNet (Schroff et al., 2015) and Facebook published a model called DeepFace (Taigman et al., 2014). The traditional facial recognition algorithms use statistical approaches with manually engineered features and pre-defined patterns, while the recent state-of-art model apply deep neural networks [16]. What are the differences between these two and how accurate they are in detecting a face when compared to each other? In the first part of this study, we provide an overview of the logic behind different facial recognition algorithms. In the second part, the performance of the algorithms is tested through a case study.

The aim of this paper is to test accuracies of different facial recognition algorithms in real-life setting. To ensure the reliability of the test, each algorithm is trained and tested with the identical data sets. The test results are compared and the algorithms are ranked accordingly allowing us to pinpoint the best ones.

2 Related Research

2.1 Computer vision and facial recognition

The term Computer Vision (CV) refers as a field of research that aim to develop proper techniques to enable computer to see and process the content of images and videos. The computer vision is a part of Artificial Intelligent (AI). The objects detection in an image is the main task of the CV algorithm as what and where the objects are seen. Additionally, the CV algorithms must identify the properties of identified objects, for example whether it is a face, building, or a door. In most cases these identified images are stored and compared against new objects. Furthermore, the CV enables us to have multiple metrics on the selected objects. Therefore, the CV has been utilized in various sectors, such as in safety, health, security, entertainment, cars, robotics, and in sports.

Before trying to recognize a face from an image, it is essential to detect and extract the face region from the original image. Other elements in the image that are not part of a face can deteriorate the recognition process. Facial recognition includes multiple steps that are illustrated in Figure 1.

Fig. 1. Facial Recognition process.

Research on computer vision has a long history on object and facial recognition. The aim of the research has been to mimic how humans perceive and process the faces or objects [17]. There are many algorithms for facial detection with the most common being Haar-Cascade [18] and Histogram of Oriented Gradients (HOG) [19] classification methods. The Haar-Cascade classifier needs to be trained with a large number of pictures with and without faces. The classifier is based on detecting Haar-like features which are rectangles split into two, three or four smaller rectangles with black or white pixels. Using these as feature extractors with a multi-stage weak classification process (cascading), one can build a high-accuracy face classifier. HOG is another approach for detecting faces and the one applied in this work. The HOG algorithm begins with calculation of the gradient values in horizontal and vertical directions to detect edges. Next, the image is divided into small connected regions (cells), and for the pixels within each cell, a histogram of gradient directions is compiled. Finally, cells are grouped together into larger, spatially connected and overlapping blocks, e.g., 16x16 pixels, and the most frequent gradient direction in a block is kept. The resulting HOG descriptors may be then used to train a classifier, such as a Support Vector Machine, to detect faces.

After detecting a face, image is then further preprocessed. This includes different adjustment and modifications of images, such as cropping and resizing to fit requirements of the facial recognition method that follows. Most of the algorithms for facial recognition require the same size for the entire training set.

2.2 Defining facial recognition

Facial Recognitions (FR) are often done in two ways: Verification and identification. In verification, the system compares a given object with the existing stored objects. In identification, the system identifies the object and gives a rank of the matches. In both cases, the biggest challenge is teaching the machine to recognize faces. The FR technology implementation consist of several stages such as image acquisition, image processing, characteristic identifications, eye sockets, nose shape, template creation, and template matching. Facial recognition algorithm often measures the distance between the eyes, width of the nose, depth of the eye socket, cheekbones, and chin. Traditional FR algorithms use statistical approach or search for patterns, while the more recent ones use deep neural networks. In the following these two approaches are discussed in more detail.

2.3 Facial recognition algorithms

Popular methods using traditional machine learning include Eigenface [20], Fisherfaces [21] and local binary patterns histogram algorithm [29]. Eigenface[20] is a method which perform the facial recognitions statistically by measuring variations of an extracted picture. The prediction of faces is based on the training set. The modified version of the Eigenface is called Fisherfaces [21]. While in the Eigenface we do not make a difference between two pictures from different classes during the training part, the Fisherfaces uses Linear Discriminant Analysis [22] method in order to make a difference between two pictures from a different class. In contrast to previous algorithms, the Local Binary Histograms (LBPH) is not a holistic approach; LBPH is based on 3x3 block of pixels, where the center pixel is compared to its neighbors. The pixel which is smaller than middle, the value 0 will be added to the threshold square and otherwise 1.

A Neural Network (NN) model [23] contains at minimum two layers of nodes; an input and output layer. The deep neural network [24], [25] is a neural network which has more than two hidden layers. The nodes ("neurons") that combine response from earlier layers, apply nonlinear transformation (activation) and feed the result into the next layer. Convolutional Neural Network (CNN) is a special type of neural network that applies convolution operations to data [26]. In convolutional layer each pixel in the picture represent by a number between -1 (dark pixel) and 1 (bright pixel). After each convolutional layer, it is recommended to use activation function. An activation function is a nonlinear function with the goal to add nonlinear input-output mapping properties in the model. CNN architecture enables using 2D pictures an input data, which makes it well-suited for a facial recognition task. When the input of first layer is a picture of a face, the output of the last layer is the predicted class, i.e., a specific person. Here we applied OpenFace [27] face recognition library based on Google's FaceNet (Schroff et al., 2015) CNN systems. The version of OpenFace facial recognition model used here was trained with 500K images.

2.4 Application of facial recognitions

Facial recognitions have become popular among all ages and sectors in society. We have seen facial recognition in entertainment and games (e.g., snapchat) [28], security (e.g., unlocking phone) [29] and privacy applications [30]. The facial recognition algorithms has been applied in various high-importance sectors, such as assisting the forensic investigation through the pattern recognitions [31]. Or in health sector facial recognitions in patient check-in process such as in schizophrenia and in bipolar [32]. The applications of facial recognition in the health sector have been increasing in a rapid pace for example assisting physician for diagnosis, managing pain, safety, and patient check-in process specially with the high-risk patient.

There are however many ethical, legal, and security concerns to apply facial recognitions in public sector. These include where to collect images, and how to save the images and manipulate images in secure ways. Therefore, encryption of the collected images has also become part of the some of the facial recognition's algorithm.

3 Research Question and Methods

3.1 Scope

The facial recognition algorithms come in various architectures and flavors. We selected algorithms to test using the following criteria: Most used algorithms in commercial sector, the popularity of the algorithms and with open-source availability. We did a case study involving these algorithms, involving developing a test application and collecting a novel dataset.

3.2 Research objective and question

The main objective of this study to answer the following question:

Which facial recognition algorithms or models are most accurate in recognition of the faces?

To answer this, we performed a case study by developing a facial recognition application and collected picture dataset. A script that uses OpenCV (https://opencv.org) and Dlib (http://dlib.net) libraries to perform facial recognition for our custom dataset.

3.3 Test process

The test process contained the following steps:

- Capture and image acquisition: A sample image is captured by the test application
- Extraction: The face is extracted from the sample image
- Comparison and image processing: The sample is compared to stored samples
- Collecting results, template creation and template matching: The test application returns the name of the person in image which is compared against the ground truth (match or no match)

The pictures in the training set are faces with various emotional gestures. The test set consists of pictures from each subject that were not included in the training set. The same training and test sets are used for all algorithms.

4 Case Study Design and Development

4.1 Testing the algorithms

All the pictures from the training data were taken with a Nikon D3100 digital camera. After obtaining images, we ran HOG algorithm to locate faces in them. The landmark detector of the Dlib library was used to center the face into the picture based on the nose. For example, to the detects the landmarks on the faces we applied following figure 2 open source code [33].

```python
# Load the image, resize and convert it to grayscale
image = cv2.imread(image_path)
image = imutils.resize(image, width=500)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Initialize face detector, create landmarks predictor
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(landmark_predictor)

# Detect faces in the image
rectangles = detector(gray, 1)

# Loop over faces and draw facial landmarks
for (i, rect) in enumerate(rectangles):
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)
    (x, y, w, h) = face_utils.rect_to_bb(rect)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    for (x, y) in shape:
        cv2.circle(image, (x, y), 1, (0, 0, 255), -1)

cv2.imshow("Face with Landmarks", image)
```

Fig. 2. Sample of code for detecting landmarks on the face [33]

The landmarks are then drawn on the picture and displays.

The picture was also rotated to have the eyes on the same level. After that, the images were cropped based on the external landmarks of the faces and resized into 96x96 pixels. Sample of the face detection and the landmark is presented in figure 3.

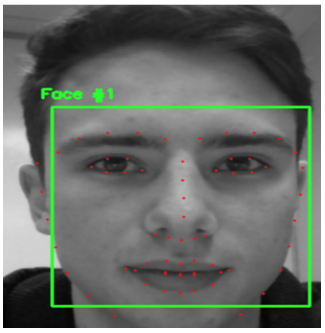

Fig. 3. Sample of detected face.

The algorithms used for the tests were Eigenfaces, Fisherfaces and local binary patterns histograms, which were all available in OpenCV library (https://opencv.org).

Eigenfaces and Fisherfaces were used with Euclidean distance metric. For deep convolutional neural network we used OpenFace [34] model.

Each algorithm had three categories of tests. The first category had 5 subjects with all pictures taken in the same environment and light conditions. The second category had 10 subjects and the third 15 subjects. Therefore, the corresponding chance levels of picking the correct person were 20%, 10% and 6.67% accordingly. The pictures in categories two and three were taken in different environments and with varying levels of luminosity. Each category was divided into three sets with either 10, 20 or 40 pictures each. The pictures in the test set were taken at the same time as the ones used in training, thus the environment of the training and test sets were the same. Figure 3 shows the summary of all tests. In addition, we took two additional pictures per subject. These pictures were taken in a completely different environment than the training data. This allowed us to test robustness of the face recognition algorithms in varying conditions.

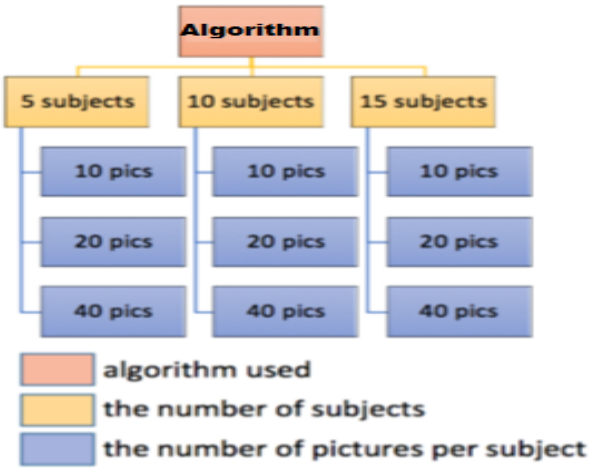

Fig. 4. All dataset sizes used in the testing.

Figure 4 presents examples of the face detection and the landmark using the OpenCV and Dlib's 68-point facial landmark.

Fig. 5. Sample of detected faces by applying openCV and dlib algorithms with 68-point landmarks.

5 Results

The comparison results of the algorithms' test in same training and test environments are presented in Table 1. Here each row corresponds to an algorithm, with last being the random chance level. Each algorithm is tested with 5, 10 and 15 subjects (#categories) with varying number of training samples (10, 20 or 40). Testing set contained 5 samples from each person. Accuracy percentages were calculated by comparing the number of pictures that were correctly recognized compared to the total number of pictures tested (either 25, 50 or 75).

Table 1. Face identification results as percentages for training and testing data in same environment. 5 pictures were tested per person.

	5 subjects			**10 subjects**			**15 subjects**		
	n=10	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***
Eigenfaces	92	100	100	96	100	100	96	99	100
Fisherfaces	92	100	100	96	98	100	93	95	100
LBPH	96	100	100	98	100	100	99	99	100
OpenFace	100	100	100	100	100	100	100	100	100
Random	20	20	20	10	10	10	6.7	6.7	6.7

As expected, the accuracies increased as the number of training samples became higher. All algorithms were able to recognize pictures correctly with 40 training samples. With less samples, only OpenFace retained full 100% accuracy in all situations. With 5 subjects and 10 images per subject, Eigenfaces and Fisherfaces reached accuracy of 92%, which was the worst accuracy among all tested settings. In general, all algorithms performed well (over 90% accuracy) and far above random change level. With a perfect score, OpenFace was the best algorithm. Differences between the remaining three were small (within 4% margin).

Next, we relaxed the requirement for testing data having the same environment as the training data. This requires a classifier to have higher tolerance against data variability (noise), resulting is notably harder task. Results for this second test type are in Table 2, again with 5 to 15 subjects and 10 to 40 training samples.

Table 2. Face identification results for training and testing data in different environment. 2 pictures were tested per person.

	5 subjects			**10 subjects**			**15 subjects**		
	n=10	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***
Eigenfaces	60	60	60	30	30	30	13	17	20
Fisherfaces	70	50	50	10	25	20	17	17	13
LBPH	30	40	40	15	30	30	13	23	23
OpenFace	100	100	100	100	100	100	93	93	97
Random	20	20	20	10	10	10	6.7	6.7	6.7

The differences between algorithms became more evident. The mean accuracy over all algorithms dropped on average by 50.8% compared to the Table 1. While the drop

for classical (non-neural) methods was 67.1%, it was only 1.9% for the OpenFace, with only a minor reduction in performance. With 10 samples, LBPH had the worst performance (average accuracy 19.3%), Fisherfaces was third (32.3% accuracy) and Eigenfaces was second (34.3%). With 20 samples or more, differences between the non-neural methods decreased notably.

Finally, in order to better evaluate whether algorithms were sensitive to the particular subjects, we recalculated the accuracies for the 5 subject that were present in all test sets. In other words, the despite the number of subjects in training data, test was only performed for those 5 subjects with total 10 samples (2 per subject). Results for this test are shown in Table 3.

Table 3. Face identification results (percentages) for training and testing data in different environment. 2 pictures were tested per person. Here only the data from 5 subjects, that were fixed, were used in testing phase (total 10 test images).

	5 subjects			**10 subjects (5 tested)**			**15 subjects (5 tested)**		
	n=10	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***	***n=10***	***n=20***	***n=40***
Eigenfaces	60	60	60	40	40	40	20	20	20
Fisherfaces	70	50	50	20	50	30	20	10	10
LBPH	30	40	40	0	10	20	10	30	30
OpenFace	100	100	100	100	100	100	80	80	90
Random	20	20	20	10	10	10	6.7	6.7	6.7

Now the differences between non-neural algorithms became somewhat more apparent. Performance for LBPH dropped overall and even below the change level for 10 subjects. Also, the performance of OpenFace dropped to 80% with 15 subjects. These results demonstrate that accuracy not only depends on number of training and testing samples, but also on the specific samples. Despite the slight drop in performance, OpenFace was found highly robust to variance caused by samples size, environment and selection of subjects.

6 Discussion

Facial recognition's algorithms are being constantly developed with increasing accuracy and reliability. The accuracy and reliability are extremely important in when the application deals with security and privacy [35]. The main objective of this study was to assess and evaluate the performance of the popular facial recognition's algorithms. The outcome of our study helps specially the practitioner for facial recognition's algorithm selection for different contexts.

In the first phase, we tested the performance of the algorithms such that training and test images came from the same environment. This is the traditional setting in machine learning. In this context, our results demonstrated that all algorithms performed well and there were no notable differences between algorithms. In fact, all four algorithms resulted in perfect accuracy with n=40. Some differences arouse only with limited number of training samples (n=10), with OpenFace taking the lead

(100%) and LBPH coming in second (96-99%). Eigenfaces and Fisherfaces were tied at 92-96%. Our results were similar to those reported in literature: 94.4% for Eigenfaces with dataset of 3040 images [36], 93% for Fisherfaces with 73 images [37], and 93.1% for LBPH with 1280 images [38].

In the second phase, the environment between the train and test sets was different. In this context, its assumed that images are coming from environments and conditions not necessarily available during training. This is closer to the real-life usage of the system and requires robustness in noise tolerance from the algorithm. In this context the accuracy of all but OpenFace dropped dramatically, e.g., from 100% to 13-23% with 15 subjects and n=40. There was also lots of variability in accuracies, e.g., performance decrease for FisherFaces with increased number of samples (n=40 vs. n=10), which further indicates poor generalization and noise tolerance of the algorithm against environment mismatch. Only LBPH consistently improved when adding more samples. Apart from the clear winner (OpenFace), there was no clear runner-up. For limited samples (n=10), EigenFaces was better than FisherFaces and LBPH, while LBPH become better with more samples (n=40). High variability of FisherFace performance could be problematic particularly in real-life applications.

OpenFace based on deep neural networks shows to be by far the best algorithm in our experiments. This algorithm had perfect accuracy in the first phase and also in the second phase with 5 and 10 subjects. Some errors emerged only in the second phase with 15 subjects. Error was reduced with more training data. OpenFace was only lightly impacted by a change in the environment compare to the others. This is align also with the findings by Santoso and Kusuma [39].

Based on these results, we can conclude that using a convolutional neural network is more efficient for performing a facial recognition than statistical approaches or searching for patterns. CNN does not have a unique structure, but can be customized based on application. OpenFace can be optimized by changing hyperparameters and different combinations of hidden layers. Increasing the amount of data in the data set for extracting the features is also a way to potentially improve the results.

7 Conclusion

We tested three classical facial recognition algorithms, Fisherfaces, Eigenfaces, and Local Binary Pattern Histogram against deep neural network model OpenFace. The tests involved varying number of training samples and subjects with matched or mismatched environments between training and test data. The main objective of our study was to find differences of the algorithms in real-life environment with a practical setting.

Our study reveals that OpenFace is the only algorithm robust against mismatched training and testing data, while there was a major drop in performance of classical algorithms. At the same time, OpenFace was able to work with only 10 training samples, while 20 or more was often required by the classical algorithms. These results demonstrate that models based on convolutional neural networks are superior in facial recognition task.

As a future work we plan to conduct a research on the emotion recognition through facial recognitions. Instead of differentiating persons, the aim is to recognize the emotion of a person. The concept of class is also used, though instead of having a person representing a class it is a facial expression such as happy, sad, surprised, fear, anger or neutral. Fisherface or a convolutional neural network could be used for recognizing emotions due to their approach to cluster pictures. However, what accuracy can be expected for emotion recognition? Could Fisherface or OpenFace provide satisfying results?

8 Reference

[1] T. J. L. van Rompay, D. J. Vonk, and M. L. Fransen, "The Eye of the Camera. Effects of security Cameras on Prosocial Behavior," Environ. Behav., vol. 41, no. 1, pp. 60–74, 2009. https://doi.org/10.1177/0013916507309996

[2] B. C. Jones, A. C. Little, I. S. Penton-Voak, B. P. Tiddeman, D. M. Burt, and D. I. Perrett, "Facial symmetry and judgements of apparent health: Support for a 'good genes' explanation of the attractiveness-symmetry relationship," Evol. Hum. Behav., 2001. https://doi.org/10.1016/s1090-5138(01)00083-6

[3] P. Vannini and L. M. Stewart, "The GoPro gaze," Cult. Geogr., vol. 24, no. 1, pp. 149–155, 2017.

[4] L. A. Elrefaei, "Smartphone based image color correction for color blindness," Int. J. Interact. Mob. Technol., 2018. https://doi.org/10.3991/ijim.v12i3.8160

[5] L. Rai, Z. Wang, A. Rodrigo, Z. Deng, and H. Liu, "Software Development Framework for Real-Time Face Detection and Recognition in Mobile Devices," Int. J. Interact. Mob. Technol., vol. 14, no. 04, 2020. https://doi.org/10.3991/ijim.v14i04.12077

[6] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, "Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments," 2016. https://doi.org/10.1007/978-94-6265-132-6_2

[7] M. Weinberger, "Facebook vs. Snapchat in computer vision - Business Insider," 2017. .

[8] K. Hazelwood et al., "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective," in Proceedings - International Symposium on High-Performance Computer Architecture, 2018.

[9] J. Y. Jang, K. Han, P. C. Shih, and D. Lee, "Generation Like: Comparative Characteristics in Instagram," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15), 2015. https://doi.org/10.1145/2702123.2702555

[10] J. P. Mello Jr., "Facial Recognition Beyond Facebook.," PCWorld, vol. 29, no. 12, pp. 13–14, 2011.

[11] E. Alpaydın, "Introduction to machine learning," Methods Mol. Biol., vol. 1107, pp. 105–128, 2014.

[12] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Computer Vision – ECCV 2016," 1511.03745V1, 2016. https://doi.org/10.1007/978-3-319-46448-0_49

[13] Y. Ganin et al., "Domain-adversarial training of neural networks," in Advances in Computer Vision and Pattern Recognition, 2017.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering." https://doi.org/10.1109/cvpr.2015.7298682

[15] Y. Taigman, M. Yang, and M. A. Ranzato, "Deepface: Closing the gap to humal-level performance in face verification," CVPR IEEE Conf., pp. 1701–1708, 2014. https://doi.org/10.1109/cvpr.2014.220
[16] M. Majumder, "Artificial Neural Network," 2015.
[17] "Computer vision: algorithms and applications," Choice Rev. Online, 2013.
[18] S. Soo, "Object detection using Haar-cascade Classifier," Inst. Comput. Sci. Univ. Tartu, 2014.
[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005, vol. I, pp. 886–893, 2005. https://doi.org/10.1109/cvpr.2005.177
[20] J. Zhang, Y. Yan, and M. Lades, "Face recognition: Eigenface, elastic matching, and neural nets," Proc. IEEE, 1997. https://doi.org/10.1109/5.628712
[21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs . Fisherfaces : Recognition Using Class Specific Linear Projection," vol. 19, no. 7, pp. 711–720, 1997. https://doi.org/10.1109/34.598228
[22] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," Inst. Signal Inf. Process., 1998.
[23] B. Dasgupta, D. Liu, and H. T. Siegelmann, "Neural networks," in Handbook of Approximation Algorithms and Metaheuristics, 2007.
[24] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," Neurocomputing, 2017. https://doi.org/10.1016/j.neucom.2016.12.038
[25] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[26] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," IEEE Trans. Neural Networks, 1997. https://doi.org/10.1109/72.554195
[27] T. Baltrusaitis, P. Robinson, and L. P. Morency, "OpenFace: An open source facial behavior analysis toolkit," in 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016. https://doi.org/10.1109/wacv.2016.7477553
[28] M. Quinn, "snapchat," Quartz, 2016. .
[29] H. Alshamsi, H. Meng, and M. Li, "Real time facial expression recognition app development on mobile phones," in 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016, 2016. https://doi.org/10.1109/fskd.2016.7603442
[30] K. A. Gates, Our biometric future: Facial recognition technology and the culture of surveillance. 2011. https://doi.org/10.18574/nyu/9780814732090.001.0001
[31] S. Moore and R. Bowden, "Local binary patterns for multi-view facial expression recognition," Comput. Vis. Image Underst., 2011.
[32] J. Addington and D. Addington, "Facial affect recognition and information processing in schizophrenia and bipolar disorder," Schizophr. Res., 1998. https://doi.org/10.1016/s0920-9964(98)00042-5
[33] A. Rosebrock, "Facial landmarks with dlib, OpenCV, and Python." [Online]. Available: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/. [Accessed: 10-Apr-2020].
[34] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications," no. June, 2016.
[35] Y. Nakashima, T. Koyama, N. Yokoya, and N. Babaguchi, "Facial expression preserving privacy protection using image melding," in Proceedings - IEEE International Conference on Multimedia and Expo, 2015. https://doi.org/10.1109/icme.2015.7177394

[36] M. üg. Çarıkçı and F. Özen, "A Face Recognition System Based on Eigenfaces Method," Procedia Technol., 2012. https://doi.org/10.1016/j.protcy.2012.02.023
[37] M. Anggo and La Arapu, "Face Recognition Using Fisherface Method," in Journal of Physics: Conference Series, 2018. https://doi.org/10.1088/1742-6596/1028/1/012119
[38] C. Shan and T. Gritti, "Learning discriminative LBP-histogram bins for facial expression recognition," in BMVC 2008 - Proceedings of the British Machine Vision Conference 2008, 2008. https://doi.org/10.5244/c.22.27
[39] K. Santoso and G. P. Kusuma, "Face Recognition Using Modified OpenFace," in Procedia Computer Science, 2018. https://doi.org/10.1016/j.procs.2018.08.203

9 Authors

Art Amir Dirin, **Nicolas Delbiaggio** and **Janne Kauttonen** works for Haaga-Helia University of Applied Science in Finland.

Article submitted 2020-04-19. Resubmitted 2020-05-26. Final acceptance 2020-05-28. Final version published as submitted by the authors.