A performance comparison of two emotion-recognition implementations using OpenCV and Cognitive Services API

Luis Antonio Beltrán Prieto[1,a] and Zuzana Komínkova-Oplatková[1]

[1] *Faculty of Applied Informatics, Department of Informatics and Artificial Intelligence, Tomas Bata University in Zlín, Nad Stráněmi 4511,76005 Zlín, Czech Republic*

Abstract. Emotions represent feelings about people in several situations. Various machine learning algorithms have been developed for emotion detection in a multimedia element, such as an image or a video. These techniques can be measured by comparing their accuracy with a given dataset in order to determine which algorithm can be selected among others. This paper deals with the comparison of two implementations of emotion recognition in faces, each implemented with specific technology. OpenCV is an open-source library of functions and packages mostly used for computer-vision analysis and applications. Cognitive services is a set of APIs containing artificial intelligence algorithms for computer-vision, speech, knowledge, and language processing. Two Android mobile applications were developed in order to test the performance between an OpenCV algorithm for emotion recognition and an implementation of Emotion cognitive service. For this research, one thousand tests were carried out per experiment. Our findings show that the OpenCV implementation got a better performance than the Cognitive services application. In both cases, performance can be improved by increasing the sample size per emotion during the training step.

1 Introduction

The aim of this paper is to compare the performance of two emotion-recognition implementations. The first application consists of Python code which uses OpenCV face-recognizer classes combined with Fisher Face technique. The second one is a C# application which makes requests to a Cognitive Services API. 500 facial expressions from the Cohn-Kanade (CK+) dataset were examined by each program for evaluation purposes.

Facial emotion recognition seeks to predict the real feeling that a person expresses based on facial images, with a wide range of possible applications, such as improving student engagement [1], building smart health environments [2], analyzing customers' feedback [3] and evaluating the quality of children's games [4], just to name a few.

Deep learning is a recent, revolutionary technique in machine learning which pursues the objective of bringing artificial intelligence to solve practical applications across different, diverse fields, such as recommender systems [5], plasma tomography reconstruction [6], facial age estimation [7], and neuroimaging [8], among others. Recognizing faces within images and videos has been one of the challenges that deep learning has tested thoroughly, with significant performance and improvement. This progress has been achieved thanks to development of Convolutional Neural Networks and the availability of huge training datasets [9].

As a consequence, detecting the emotion expressed by a person is the next step into facial analysis. Recent research [10] has proven that emotion detection can be achieved by the usage of machine learning and artificial intelligence algorithms. While it is not an easy task, several open-source libraries and packages, such as OpenCV, TensorFlow, Theano, Caffe and CNTK (Microsoft Cognitive Toolkit) simplify the process of building deep-learning-based algorithms and applications. Emotions such as anger, disgust, happiness, surprise, and neutrality can be detected.

This paper is organized as follows. In the first section, a theoretical background on emotion recognition, OpenCV, Fisherfaces algorithm, Cognitive Services, and the extended Cohn-Kanade (CK+) database is presented. Afterwards, the methods and methodology that was used for this comparison are described. Then, evaluation results, comparison and discussion are presented together. Finally, conclusions are shown at the end of the paper.

2 Background Information

Emotions are strong feelings about people's situations and relationships with others. Most of the time, humans show how they feel by using facial expressions. Speech, gestures, and actions are also used to describe a person's current state.

Emotion recognition can be defined as the process of detecting the feeling expressed by humans from their

[a] Corresponding author: beltran_prieto@fai.utb.cz

facial expressions, such as anger, happiness, sadness, deceitfulness, and others. Even though a person can automatically identify facial emotions, machine learning algorithms have been developed for this purpose. Emotions play a key role in decision-making and human-behaviour, as many actions are determined by how a person feels at some point.

Typically, these algorithms use either a picture or a video (which can be considered as a set of images) as input, then they proceed to detect and focus their attention on a face and finally, specific points and regions of the face are analysed in order to detect the affective state.

Machine Learning (ML) algorithms, methods and techniques can be applied to detect emotions from a picture or video. For instance, a deep learning neural network can perform effective human activity recognition with the aid of smartphone sensors [11]. Moreover, a classification of facial expressions based on Support Vector Machines was developed for spontaneous behavior analysis.

2.1 OpenCV and Fisherfaces

OpenCV [12] is a free, yet powerful, open-source library developed by Intel Corporation which has been widely used in computer vision and machine learning tasks, such as image processing, real-time image recognition, and face detection. With more than 2500 optimized algorithms included, this library has been extensively used for research and commercial applications from both global and small entrepreneurs. OpenCV contains an optimized set of libraries written in C language, with bindings to other languages and technologies, including Python, Android, iOS, and CUDA (for GPU fast processing), and wrappers in other languages, such as C#, Perl, Haskell, and others. Moreover, it works under Windows and Linux.

Among its capabilities, OpenCV contains a FaceRecognizer class which, as the name suggests, is helpful for face recognition tasks. There are three algorithms available for this purpose: Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms. While the first technique considers a linear combination of facial features in order to maximize total variance in data, thus representing data in a powerful, but classless, way, Fisherfaces takes a Linear Discriminant Analysis approach in which class-specific dimensionality reduction is performed so the combination of features that separate the best classes is taken into account. If there exists any external source, such as light, which affects the representation of the image, the Eigenfaces technique is not able to accurately classify the faces. Fisherfaces, on its side, is not affected by this factor.

The algorithm goes as follows: First, let $V = \{V_1, V_2, \ldots, V_c\}$, $V_i = \{v_1, v_2, \ldots, v_N\}$ be a random vector with samples obtained from c classes. Then, the total mean, μ, and the mean of class i, μ_i, where $i \in \{1, 2, \ldots, c\}$ are computed as described in equations (1) and (2). These values are used in equations (3) and (4) in order to calculate the Scatter matrices, S_B and S_W.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{1}$$

$$\mu_i = \frac{1}{|V_i|}\sum_{v_j \epsilon V_i} v_j \tag{2}$$

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{3}$$

$$S_W = \sum_{i=1}^{c}\sum_{x_j \epsilon x_i}(x_j - \mu_i)(x_j - \mu_i)^T \tag{4}$$

A projection W maximizes the class separability criterion by following equation (5):

$$W_{opt} = \arg max_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{5}$$

The General Eigenvalue Problem solves this optimization task (6):

$$S_B v_i = \lambda_i S_W v_i$$
$$S_W^{-1} S_B v_i = \lambda_i v_i \tag{6}$$

The rank of the scatter matrix, S_w, is at most (N samples – c classes). In problems such as pattern recognition tasks, the number of samples is smaller than the dimension of the data input. Thus, S_W becomes singular and can be solved using a linear discriminant analysis. At the end, the optimization problem is rewritten to equations (7) and (8)

$$W_{pca} = arg\ max_W |W^T S_T W| \tag{7}$$

$$W_{fld} = \arg max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \tag{8}$$

And the transformation matrix W, projecting a sample into the $(c - 1)$ dimensional space is given by equation (9):

$$W = W_{fld}^T W_{pca}^T \tag{9}$$

2.2 Cognitive Services

Cognitive Services [13] are a set of machine learning algorithms developed by Microsoft which are able to solve artificial intelligence problems in several fields, such as computer vision, speech recognition, natural language processing, machine learning search, and recommendation systems, among others. These algorithms can be consumed through Representational State Transfer (REST) calls over an Internet connection, allowing developers to use artificial intelligence research to solve problems. These services are open-source and can be consumed by many languages, including C#, PHP, Java, Python, and implemented in desktop, mobile, console, and web applications.

The Computer Vision API of Cognitive Services provides access to machine learning algorithms capable of image processing tasks. Either an image stream is uploaded or an image URL is specified to the service so the content can be analyzed for label tagging, image categorization, face detection, color extraction, text detection, and emotion recognition. Video is also supported as input. The Emotion API analyses the sentiment of a person in an image or video and returns the confidence level for eight emotions mostly understood by a facial expression, including anger, contempt, disgust, fear, happiness, neutrality, sadness and surprise.

2.3 The Extended Cohn-Kanade database

The Cohn-Kanade AU-Coded Facial Expression Database [14, 15] is a well-known repository of face images used for research purposes into the facial recognition field, with an increased interest in emotion detection research. An Extended version 2 of the database, also known as CK+, was developed in order to address some limitations of version 1, such as non-validation of emotion labels, the absence of a common performance metric against which to evaluate the latest algorithms and standard protocols for typical databases and quantitative meta-analysis. It contains the facial expressions of 210 adults between 18 and 50 years of age, from which 31% were male, 81% Euro-Americans, 13% Afro-Americans, and 6% from other groups. For each participant, 23 facial displays were performed. 593 sequences were labelled with a basic emotion from a pool of seven categories: anger, contempt, disgust, fear, happiness, sadness, and surprise.

3 Methods and Methodology

The objective of this experiment is to compare the performance of 2 emotion-recognition implementations. The first one (Experiment A) is a Python-based application which makes use of the OpenCV machine learning algorithms for facial and emotion detection. The second one (Experiment B) is a C# software application which makes requests to a Cognitive Services API for emotion detection. In both cases, the Extended Cohn-Kanade dataset of images was used as input for the analysis. Both applications were developed by the authors for this experiment.

For Experiment A, we considered 327 sequences which actually show a relevant sequence of emotion, from a neutral feeling to the emotion itself. First step is then to obtain both the neutral and emotional images. From this subset, OpenCV library is used to detect the face on each picture by using a custom Haar-filter. Effective object detection is possible thanks to the Haar feature-based cascade classifiers proposed in [16], a machine learning based approach which takes advantage of cascade functions training from both positive and negative images. OpenCV already provides several Haar-filter functions; however, a cascade function of boosted classifiers was used for this experiment in order to detect the faces in the pictures. Thereafter, all images were standardized by converting them to grayscale and resized to the same dimensions. Table 1 presents how half of the subset, i.e., 327 images, were distributed among the different labelled emotions. The other half corresponds to 327 emotionless faces, i.e., showing neutrality. Then, we proceeded to randomly split the subset in two new sets: training set and prediction set. For training, we considered 523 images, which corresponds to 80% of the pictures. The remaining 20% was considered for the prediction set. For better evaluation purposes, 10 random training and classification sets were generated. The training process consists of getting the characteristics of each face along with the labelled emotion, i.e. the expression shown by the person. This data is used to create and train a Fisherface classifier. Then, evaluation of the classifier proceeds by comparing the outcome of its predict function of each face with the actual labelled emotion.

Table 1. Sample size of each labelled emotion

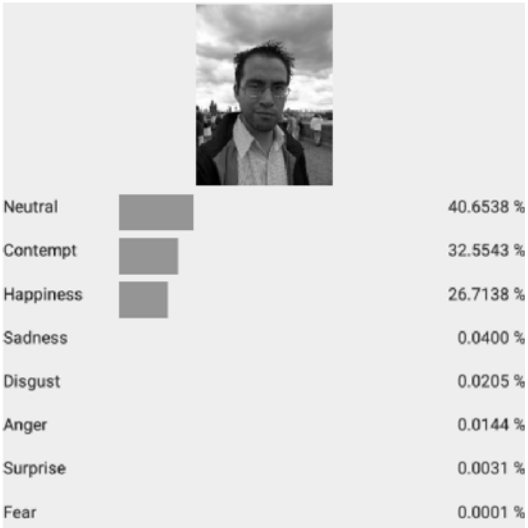
Labelled Emotion	Number of faces
Anger	45
Contempt	18
Disgust	58
Fear	25
Happiness	69
Sadness	28
Surprise	84

Experiment B starts with the 654 images extracted with the Python code from Experiment A. 10 random groups consisting each of 20% of the collection were generated in order to evaluate samples of the same size as in Experiment A. For each experiment, every face was submitted to the Cognitive Services API for its evaluation, as its training has already been developed by Microsoft. A C# application was developed for running this experiment. The service returns a JSON content which contains the score for each emotion. Only the highest score, considered as the predicted facial expression detected by the service, was compared with the actual labelled emotion for evaluation purposes. Fig. 1 shows the application developed in C# for this experiment. It takes a previously provided picture, then finds a face on it and finally submits a request in order to detect the emotions expressed by the person in the photo. The analysis is performed by the Emotion Cognitive Service.

5 Results and Discussion

After running each of the 10 tests from Experiment A, the results which are presented in Table 2 were obtained. An average of 75.87% correct predictions was calculated as an outcome. On the other side, Table 3 illustrates the results of each test in Experiment B. As a result, a 68.93% average of efficiency was obtained after running 10 tests of this implementation.

The findings of the experiments show that the Python-based implementation with OpenCV using Fisherfaces proved to be more accurate than the Cognitive Services implementation in C# by approximately a 7% difference. For experiment A, several of the mistakes occurred when trying to predict emotions with low occurrences in the dataset, such as fear and contempt mistakenly classified as neutral expressions. Moreover, there were a few errors when trying to predict a neutral face, most of the time identified as a sad face. Regarding experiment B, neutral images were wrongly identified as either contempt or sadness emotions; however, by looking closely to the scores obtained by the Cognitive Services, a minimum difference between the wrong prediction and the actual emotion was detected. Thus, in most cases, the second best prediction was correct. However, for evaluation purposes, this was considered as an error.



Neutral	40.6538 %
Contempt	32.5543 %
Happiness	26.7138 %
Sadness	0.0400 %
Disgust	0.0205 %
Anger	0.0144 %
Surprise	0.0031 %
Fear	0.0001 %

Fig. 1. Analysis of emotions detected on a picture by the Emotion Cognitive Service.

Table 2. Evaluation results of experiment A

Test Number	Correct (%)	Incorrect (%)
1	103 (78.62%)	28 (21.37%)
2	100 (76.33%)	31 (23.66%)
3	98 (74.80%)	33 (25.19%)
4	104 (79.38%)	27 (20.61%)
5	96 (73.28%)	35 (26.71%)
6	95 (75.25%)	36 (27.48%)
7	99 (75.57%)	32 (24.42%)
8	100 (76.33%)	31 (23.66%)
9	97 (74.04%)	34 (25.95%)
10	102 (77.86%)	29 (22.13%)

Table 3. Evaluation results of experiment B

Test Number	Correct (%)	Incorrect (%)
1	89 (67.93%)	42 (32.06%)
2	95 (72.51%)	36 (27.48%)
3	86 (65.64%)	45 (34.35%)
4	91 (69.46%)	40 (30.53%)
5	88 (67.17%)	43 (32.82%)
6	90 (68.70%)	41 (31.29%)
7	95 (72.51%)	36 (27.48%)
8	93 (70.99%)	38 (29.00%)
9	84 (64.12%)	47 (35.87%)
10	92 (70.22%)	39 (29.77%)

5 Conclusion

The objective of this experiment was to compare the performance of two different implementations of emotion-recognition applications by using OpenCV and Python with a Fisherface technique in the first case, while considering a C#-based solution which makes requests to a Cognitive Services API for Emotion detection for the second solution. While the first implementation got the best results, the performance could be improved either by increasing the sample size of those emotions with few faces, so the training phase gets benefited, or by removing them from the subset, as not enough cases were collected.

Acknowledgement

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014) and also by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, further it was supported by Grant Agency of the Czech Republic—GACR 588 P103/15/06700S and by Internal Grant Agency of Tomas Bata University in Zlin under the project No. IGA/CebiaTech/2017/004. L.A.B.P author also thanks the doctoral scholarship provided by the National Council for Science and Technology (CONACYT) and the Council for Science and Technology of the State of Guanajuato (CONCYTEG) in Mexico.

References

1. A. C. Garn, K. Simonton., T. Dasingert, A. Simonton, Psychol. Sport Exerc. **29** pp. 93-102 (2017)

2. A. Fernandez-Caballero, A. Martinez-Rodrigo, J. M. Pastor, J. C. Castillo, E. Lozano-Monasor, M. T. Lopez, R. Zangroniz, J. M. Latorre, A. Fernandez-Sotos, J. Biomed. Inform. **64** pp. 55-73 (2016)
3. A. Felbermayr, A. Nanopoulos, J. Interact. Mark. **36** pp. 60-76 (2016).
4. R. Gennari, A. Melonio, D. Raccanello, M. Brondino, G. Dodero, M. Pasini, S. Torello, Int. J. Hum-Comput. St. **101** pp. 45-61 (2017)
5. J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, Expert Syst. Appl. **69** pp. 29-39 (2017)
6. F. A. Matos, D. R. Ferreira, P. J. Carvalho, Fusion Eng. Des. **114** pp. 18-25 (2017)
7. H. Liu, J. Lu, J. Feng, J. Zhou, Pattern Recogn. **66** pp. 82-94 (2017)
8. S. Vieira, W. H. L. Pinaya, A. Mechelli, Neurosci. Biobehav. R. **74** pp. 58-75 (2017)
9. O. M. Parkhi, A. Vedaldi, A. Zisserman: *Proceedings of British Machine Vision Conf.* (BMVA Press, Durham 2015)
10. C. K. Yogesh, M. Hariharan, R. Ngadiran, A.H. Adom, S. Yaacob, K. Polat, Appl. Soft. Comput. **3** (2017)
11. C. A. Ronao, S. Cho, Expert Syst. Appl. **59** (2016)
12. OpenCV library. http://opencv.org [Online: accessed 01-Jun-2017]
13. Cognitive Services. http://microsoft.com/cognitive [Online: accessed 03-Jun-2017]
14. T. Kanade, J. F. Cohn, Y. Tian. *Proceedings of the Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition* (Grenoble, 2000).
15. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews. *Proceedings of the Third Int. Workshop on CVPR for Human Communicative Behavior Analysis* (San Francisco, 2010).
16. M. Turk, A. Pentland. J. Cognitive Neurosci **3** pp. 71–86 (1991)