Feeding a DNN for Face Verification in Video Data acquired by a Visually Impaired User

Jhilik Bhattacharya[a,b], Stefano Marsi[b], Sergio Carrato[b], Herbert Frey[c], and Giovanni Ramponi[b]
[a] *Thapar University, India*
[b] *University of Trieste, Italy*
[c] *Ulm University of Applied Sciences, Germany*

***Abstract*—Some experiments on a face verification tool based on FaceNet are presented in this paper. The task of the system is to perform face verification in a real-time assistive system aiming at facilitating the approach between a blind person and a preselected acquaintance of his/her who enters the field of view. Face detection is made easier by the fact that an almost frontal view of the face is highly probable; verification on the contrary is difficult due to the poor quality of the acquired images and to the necessity of achieving a very low error rate.**
A custom database consisting of subjects required for verification is populated with face images provided by a suitable detection tool. The cascade of FaceNet and a Bayesian Classifier proves to be an effective tool for this unconstrained face verification task.

***Index Terms*—face detection, face verification, convolution neural network, face recognition**

I. Introduction

We are developing a system for facilitating a blind person to interact with other people in a way similar to the one of a person with normal vision [1], [2]. The scenario we have agreed upon with the users is the one of a blind person who needs to meet one of his/her acquaintances in a public place, and is not willing to wait for the acquaintance to engage interaction e.g. by speaking: the users prefer to autonomously recognize the person they are meeting, in order to be able to behave consequently. In computer vision, this is a problem of face verification. To enable such a scenario, the system has to access the visual information of the surrounding environment and process it to extract information which gives an understanding of different non-verbal communication cues. Some of them may include the number of people in the scene, distance and position of identified people, physical appearance, gesture and expression of known people. This research devises the various steps needed to verify the presence or absence of particular people in the scene captured by the blind user, i.e. video acquisition, face detection, preprocessing, feature extraction and finally classification for end use.

The scene is simultaneously recorded by two commercial devices: one camera is mounted on the bridge of a pair of sunglasses, another is held by a short necklace on a light support. The glasses-mounted camera has a resolution of 1280×720 pixel and an angle of view of 135 deg.; the resolution and angle of view of the necklace-mounted camera are 1920×1080 pixel and 124 deg. respectively. In order to keep the prototype system close to its final goal, several test videos used for experimentations and validations are actually recorded by users who are fully blind from birth. Consequently, the acquired video data suffer from geometrical distortion due to wide angle camera optics, back-lighting, and disturbances due to fast and wide movement of the blind person. As the user of course lacks any feedback about the subjects in the field of view of the cameras, faces can be partially occluded or partially outside the frame. Moreover, the field of view of both cameras may be partially occluded, typically by a tuft of hair or by a lapel of the dress.

The video sequences are acquired in different indoor and outdoor environments where it may be required to identify subjects. These include a university library, a coffee shop, the hall of a public building, the neighborhood of a bus stop. These scenes reflect some typical scenarios in terms of natural and artificial lighting and crowd where the user may have to find and approach his/her acquaintance [3]. Our research focuses on preprocessing detected faces from these video sequences and feeding a feature extractor which provides face representation embeddings for classification.

The performances of face recognition tools have gradually increased even in unconstrained situations with the application of biologically-inspired Deep Neural Networks (DNN), which have been shown to largely outperform shallow nets. The literature reports both the existence of standard DNNs trained with millions of face images and the continuous evolution in layer architecture and patch selection [4], [5], [6], [7], [8], [9], [10], [11].

The performance of recognition or verification is greatly influenced by the preceding face detection and preprocessing steps. In our system, faces are detected using PICO [16]; they are validated based on a quality parameter, preprocessed and then passed on to pretrained networks which are variants of FaceNet, developed by Google for feature extraction. The features extracted from the second last layer of the network are fed into a Bayesian classifier for the face verification tasks. The method is hence able to exploit the deep layered feature extraction of FaceNet and adapt it for recognition or verification with a classifier-training phase which uses a customized dataset. Moreover, we also analyse verification results of Euclidean distance classifiers on the two different FaceNet versions [11] and [4].

Novel contributions of this paper are related to the usage of truly-in-the-wild video data acquired by a blind user, a refined method for region of interest (RoI) preprocessing, and a study of intensity preprocessing methods and their effects. The rest of the paper is organized as follows. Section II discusses face detection and RoI processing; the feature extraction models are discussed in III, while classification results are given in Section IV; section V provides the conclusions and future directions of the current research.

II. Face Detection and RoI processing

A lot of datasets have been reported in the literature for face detection. These differ in their level of annotation detail, which may vary from a simple bounding box to few or more facial landmarks such as eyes, nose, lips etc. The use cases considered in our project require to work on video sequences and hence the need to cope with larger amounts of data. Moreover, the final deliverable in this respect includes eye-related detections for gaze estimation and pose modifications between successive frames as a suggestion of an intention to communicate; although this counts as a future direction and is not in the scope of the work discussed here, the currently used face detector is chosen to accommodate this scope as an add-on without major modifications.

Popular face detectors like Viola Jones [12], Visage [13], NPD [14], FaceID [15], PICO [16], and GMS Vision [17] were tested to get an idea of which works best for the current dataset. Indeed, as elaborated in [18], all the face detectors performed poorly in the considered sequences. PICO and NPD, however, provide a confidence value which can be utilized to successfully refine the outcome by discarding the detected regions for which a low confidence is obtained. PICO is chosen in our experiments as it gives a higher average precision compared to NPD for the data under consideration.

The PICO software reports a rectangular RoI and a score for each object found. When a face is actually present, PICO reports several RoIs in slightly different positions for the same face in each frame; these RoIs are reported in subsequent frames as long as the face is inside the scene. In turn, for false positive results PICO often reports only one or two RoIs in isolated frames. To get rid of false positive results we implement a filter that uses the RoIs and their scores as well as their occurrence in subsequent frames. For each RoI reported by the detector we test if this RoI belongs to an already existing face object of our filter by calculating the distance of the center of the region from the center of all the face objects. It the distance is below a given threshold, the score of the RoI is added to the score of this face object, but only up to a given maximum. If a RoI does not fit any of the already existing face objects, a new face object with the data of this RoI is generated. All face objects that were not hit by a new RoI in the current frame are penalized by subtracting a given value from their score; if the score is below zero the face object is deleted. Finally, all face objects with a score above a suitable threshold are reported as a positive result of the face detector for the current frame, as depicted in Figure 1.

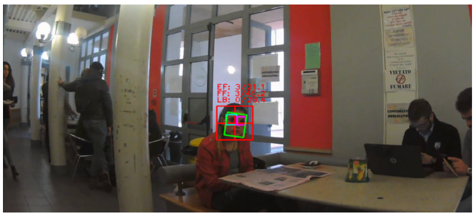


Fig. 1. Example of Face Detection.

III. Face Feature Representation

Deep convolutional networks have recently become the core of face recognition and verification tools even in unconstrained situations. The various networks reported in the literature differ in patch selection and network architecture. The models consist of multiple interleaved layers of convolutions, non-linear activations, local response normalizations, and max pooling layers. 1×1×d convolution layers and inception models [4], [7] are two variants of using a large number of wide kernel sized filters for convolution in some deep layers; the latter consists in parallel mixed layers of convolutional and pooling layers concatenated together, and have been reported to provide almost twenty times reduction in time complexity and an improved feature representation.

In general, to use these networks for feature representation the CNN bottleneck layer output is further processed using PCA for dimensionality reduction and an SVM or Bayesian classification tool [19], [20], [21].

The performances of these different networks are improved in various ways:

- The input to the network is an aligned or frontalized face. DeepFace uses 3D frontalization to align the face, whereas OpenFace utilizes a 2D affine transform to align and get a tight crop of the face
- Different networks on different face patches or alignments are computed and their responses are combined. [21] combines the responses of 25 networks and predicts the distance using PCA and Joint Bayesian Model; [20] uses SVM to combine the predictions of three networks using different face alignments
- The network is trained with a combination of classification and verification loss [4], [22]. This also avoids the extra dimension reduction and the nonlinear classification tasks.

According to an analysis of popular networks in [23] the following facts are highlighted:

- Large fully connected layers are inefficient for small batches of images as the operations are better optimized over large matrices rather than small ones, hence utilizing resources more efficiently. For example AlexNet takes 84% of its inference time for batch size 1 and 33% for batch size 16

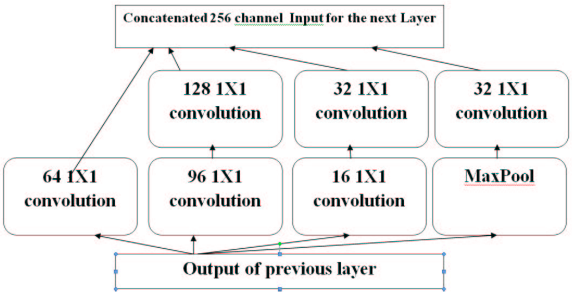


Fig. 2. Inception Layer 1:$Inception_1$.

- Accuracy and inference time are in a hyperbolic relationship: in general, model averaging is carried out for a better accuracy thus increasing the inference time. Consequently, the accuracy vs. inference time graph shows a steep slope which eventually flattens when cost complexity outgains accuracy
- Energy constraints are an upper bound on the maximum achievable accuracy and model complexity as it is obvious that in order to achieve a greater accuracy resource usage, power-consumption, and latency increase to a large extent
- The number of operations is a reliable estimate of the inference time.

In real-time situations, it may be meaningless to consider combined networks (for example the concatenation of 25 network outputs which extracts features from 25 different patches of a face) for performance elevation. Although alignment plays a crucial role, also the face detection phase has to be optimized in order to get good performances. Consequently, we need to balance the cost of the overhead (in terms of both time and computational effort) with improvements in accuracy.
The current work utilizes two models as shown in Table I and II. Network 1 is a modified version of FaceNet, which was kindly provided by the e-lab laboratory at Purdue University; Network 2 is the OpenFace network nn4.small2.v1 [11]. In Table I, we provide number of filters and filter sizes for the Spatial Convolution layers, window size and stride for the Maxpool layers, and output feature size for the View and Linear layers. The Inception Layer in Table II is a concatenation (indicated with the symbol ":") of two or four operations: $(f1, n1|f2, n2)$ denotes a layer with $f1$ and $f2$ filters of size $n1$ and $n2$. This is further shown in Figure 2. Both networks provide a 128-dimensional feature representation.

We feed the networks with the images of the detected faces; they are not aligned and may not contain a tight face crop (e.g. in case of a miss of the PICO preprocessing), as shown in Figure 3.

IV. Classification Results

Even if the final goal of our study is face verification, the results we show are for top-1 face recognition in a set of 1700

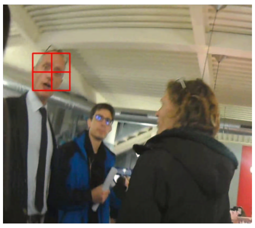


Fig. 3. Face Detection without a tight face crop.

Layers	no.of filters,filter size
Spatial Convolution	48,11
Maxpool	3,2
ReLU	
Spatial Convolution	128,5
Maxpool	3,2
ReLU	
Spatial Convolution	192,3
ReLU	
Spatial Convolution	192,3
ReLU	
Spatial Convolution	256,3
Maxpool	3,2
ReLU	
View	9216
Linear	1024
ReLU	
Linear	1024
ReLU	
Linear	128
Normalize	128

TABLE I
Network 1.

Layers	no.of filters,filter size
SpatialConvolution	64,7
Batch Norm	
ReLU	
MaxPooling	3,2
CrossMapLRN	
SpatialConvolution	64,1
Batch Norm	
ReLU	
SpatialConvolution	192,3
Batch Norm	
ReLU	
CrossMapLRN	
MaxPooling	3,2
$Inception_1$	64,1:(96,1|128,3):(16,1|32,5):32p
Inception	64,1:(96,1|128,3):(32,1|64,5):64p
Inception	(128,1|256,3):(32,1|64,5)
Inception	256,1:(96,1|192,3):(32,1|64,5):128p
Inception	224,1:(112,1|224,3):(32,1|64,5):128p
Inception	(160,1|256,3):(64,1|128,5)
Inception	384,1:(192,1|384,3):(48,1|128,5):128p
Inception	384,1:(192,1|384,3):(48,1|128,5):128p
AveragePool	1024
View	896
Linear	
Normalize	128

TABLE II
Network 2.

faces; in this phase we found this approach more informative, since it does not require to set a threshold to determine the reliability of the system. The results are analysed with three different classifiers; the effect of some basic preprocessing on the images fed to the network is also evaluated. It should be noted that an initial preprocessing is first carried out to obtain the best possible crop of the face RoI; this is followed by histogram and other normalizations of the cropped faces.

The ground truth for the various classifiers consists in 11 classes having 32 images each. 11 different sets of scene sequences are used for testing. After face detection, the presence or absence of a particular person is searched for in each scene; then, the detected face images are preprocessed, and finally they feed the network. The network requires a 3-channel input and is tested with (a) a gray image on all the three channels and (b) an RGB color image; it may be observed (Table III) that RGB color images provide a better average performance with a much lower standard deviation between different test sets. Even though grayscale images show some improvement for a few sets, their performances on others (2, 7, 10) are poor, so that the use of RGB images seems to be a better and stable solution.

As already mentioned, three kinds of classifiers are used, namely Bayesian (B), Euclidean Distance (E), Euclidean Distance with the mean face of each class (EM). E and EM are computed as follows:

$$E = \arg\min_{i=1}^{n}(\overline{f_{test_x} - f_{train_i}}) \tag{1}$$

$$EM = \arg\min_{i=1}^{c}(\overline{f_{test_x} - f_{train_i}}) \tag{2}$$

where f_{train_i} is the feature vector for each sample in Eqn. 1 and the mean feature vector of each class in Eqn. 2, respectively for n samples and c classes.

The results are shown in Table IV. As shown in Equation 2, Euclidean mean provides better results with respect to standard Euclidean distance 1) for most cases. The performances of Bayesian and Euclidean mean are almost the same with the exception of Sets 2 and 7. In Figure 4) it may be seen that Set 2 has little or no variations in terms of pose and illumination but faces are not tightly cropped in most cases, whereas Set 7 has huge variations in terms of scale and illumination. Consequently, we suppose that the Bayesian classifier provides the best verification results on using tight image crops, while the Euclidean mean distance classifier performs better with aligned images.

Set	Gray	RGB
1	100	100
2	74.68	87.34
3	98.27	86.20
4	94.54	92.59
5	96.77	100
6	98.12	95.78
7	47.82	94.88
8	88.88	87.30
9	96.55	94.31
10	84.37	94.89
11	97.82	95.58
mean	88.89	93.53
std	15.61	4.78

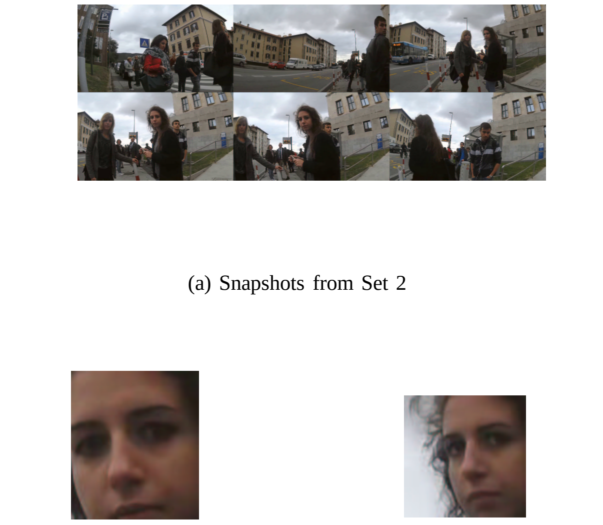
TABLE III
PERFORMANCE (ACCURACY IN %) OF BAYESIAN CLASSIFIER FOR FACE RECOGNITION ON NORMALIZED GRAYSCALE AND COLOR IMAGE.

Set	Bayes	Euclidean	Euclidean mean
1	100	100	96.87
2	79.74	92.40	92.40
3	91.37	100	98.27
4	94.44	96.29	96.29
5	100	100	100
6	97.19	97.54	97.89
7	96.69	72.77	89.43
8	87.30	88.88	88.88
9	96.59	97.72	96.59
10	97.95	96.93	96.93
11	91.17	100	92.64
mean	93.85	94.77	95.11
std	6.12	8.10	3.68

TABLE IV
PERFORMANCE (ACCURACY IN %) OF BAYESIAN, EUCLIDEAN AND EUCLIDEAN MEAN CLASSIFIER FOR FACE RECOGNITION ON NORMALIZED COLOR IMAGE.

The different preprocessing variants used in this work include normalized histogram equalization (HN) and normalized average histogram equalization (HAN) (mean of original and equalized image). Two normalizations have been tested, namely global normalization of the face according to neural network training data (NT), and classifier training set mean and standard deviation (CT). As shown in Figure 5-a,-b, the global histogram equalization and global average histogram normalization are computed over three different kinds of data referred to as RGB-RGB, RGB-HSV, RGB-YUV, where the first part of the name denotes the color channel data that feeds the network whereas the second part denotes the color channel used to perform the histogram equalization. For example, in case of RGB-HSV the RGB image is converted to the HSV color space and the V channel is normalized before converting the image back to RGB and feeding the network. Using the Bayesian classifier, we verified that HN,NT on RGB-RGB and HAN,NT on RGB-YUV provide better performances with lower deviation between sets. The best performing categories are also evaluated using the two other classifiers as shown in Table V. The results show that histogram normalized images with NT data give the best results with a Bayesian classifier.

Normalization (N) results on RGB images using the different classifiers are depicted in Figure 6. It may be seen that all the classifiers give the same average performance; however, the Euclidean mean has a lower standard deviation across the different datasets and can be considered the best one. It may be also observed that the deviation in verification accuracy results among the different test set is larger when using CT normalization instead of NT normalization for E and EM; it is



(a) Snapshots from Set 2

(b) Candidate 1 for identification

(c) Cropped face by face detector

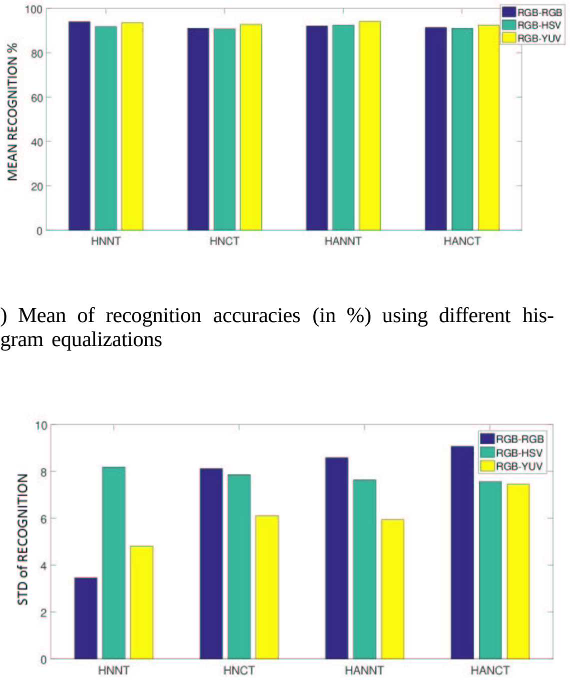
Fig. 4. Snapshots and their RoI detections.

Set	HN NT-E	HAN NT-B	HN NT-B	HN NT-EM	HAN NT-EM	HAN NT-E
1	100	100	96.87	93.75	96.87	100
2	96.20	93.67	92.40	97.46	97.46	96.20
3	96.55	81.03	89.65	96.55	86.2	94.82
4	98.14	98.14	90.74	90.74	100	100
5	100	100	100	100	100	100
6	97.54	98.94	97.54	97.19	98.94	98.24
7	66	94.71	93.23	86.30	89.43	79.53
8	93.65	85.71	96.82	92.06	85.71	87.30
9	92.04	94.31	89.77	90.90	90.90	93.18
10	97.95	94.89	95.91	96.93	91.83	96.93
11	95.58	95.88	92.64	89.70	92.64	98.52
mean	93.97	94.27	94.14	93.78	93.64	94.97
std	9.58	5.94	3.47	4.17	5.3	6.36

TABLE V
PERFORMANCE (ACCURACY IN %) OF BAYESIAN CLASSIFIER FOR FACE RECOGNITION ON COLOR IMAGE INPUTS AFTER APPLYING DIFFERENT KINDS OF PREPROCESSING TECHNIQUES.

just the opposite for the Bayesian classifier. Consequently, we decided to use the Bayesian classifier with CT normalization for the comparison of the two networks, shown in Table III.

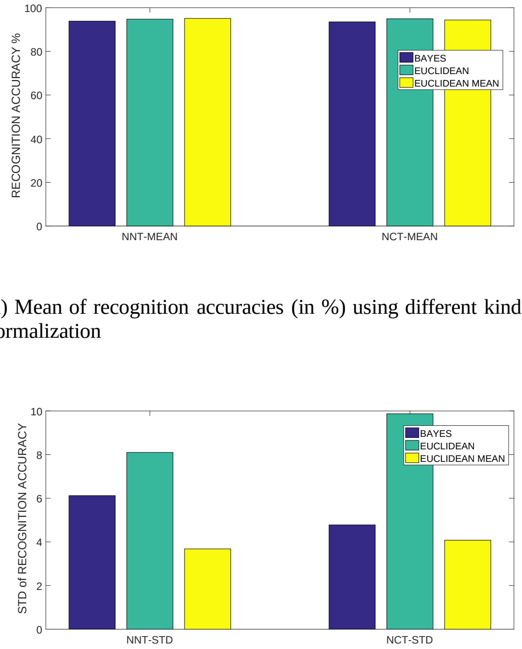
In Figure 7 it may be seen that the OpenFace network performs poorly when compared to Network 1. This may be due to underlying network differences, the fact that the images are not aligned as expected by the network, and the use of 96 X 96 patches of already very poor quality images, whereas the other network utilizes 240 X 240 patches. We are aware of the fact that the small amount of experiments we have performed does not permit to draw final conclusions about the performances of the system we are building. However, we



(a) Mean of recognition accuracies (in %) using different histogram equalizations

(b) STD of recognition accuracies using different histogram equalizations

Fig. 5. Preprocessing results.



(a) Mean of recognition accuracies (in %) using different kinds of normalization

(b) STD of recognition accuracies across sets using different kinds of normalization

Fig. 6. Effect of different kinds of normalization on CNN Input.

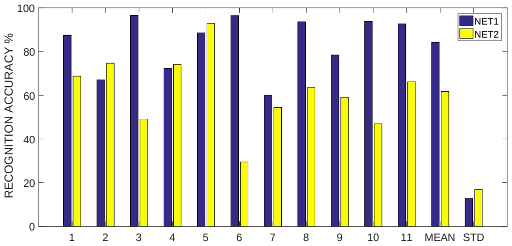


Fig. 7. Face recognition performance with two different models using the Euclidean classifier.

think that useful indications are already present (e.g. the larger or smaller standard deviations of the performances) that can guide a reader interested in the design of a system in this field of application.

V. Conclusion

The identification of faces in video sequences captured by devices worn by a blind person has been studied in this paper. This includes face detection using PICO filter, preprocessing, and face representation using deep convolution filters. The already challenging task of face recognition/verification from an unconstrained environment is further aggravated here by the fact that the faces may be partially or completely occluded and have to be detected from videos subject to backlighting, low resolution, distortion due to the use of wideangle cameras, and fast movements. Although the overall performance of the face detectors in such scenarios is quite poor, the current task is motivated by the fact that the user mainly has to recognize or verify faces who are approaching, interacting or at least looking directly at him/her; this favors the condition to a certain extent. The current work analyses the performance of two convolution models without face alignment (2D or 3D). The obtained results are satisfactory considering the single unaligned patch approach and promise considerable improvement subject to alignment operations, even if realtime implementation issues have to be taken into account. Indeed, for the same reason the authors do not plan to consider a multi-patch feature extraction for performance enhancement. Work is in progress towards testing the effect of 3D frontalization and 2D affine transformation on the detected faces before feature extraction. As the OpenFace network is supposed to be trained with aligned faces, the former step may significantly enhance the performance on our dataset. Moreover, the effect of finetuning on the networks instead of training a classifier will also be analyzed.

VI. Acknowledgement

This work has been supported by the University of Trieste - Finanziamento di Ateneo per progetti di ricerca scientifica - FRA 2014, and by a private donation in memory of Angelo Soranzo (1939-2012).
The authors also thank Eugenio Culurciello and Alfredo Canziani for kindly providing Network 1.

References

[1] S. Carrato, G. Fenu, E. Medvet, E. Mumolo, F.A. Pellegrino, G. Ramponi, ”Towards More Natural Social Interactions of Visually Impaired Persons”, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2015, Catania, Italy, Oct. 26-29, 2015

[2] S. Carrato and G. Ramponi, ‘’Assistive technologies based on image processing for people with visual impairments: a tool to help social interaction of the blind,” in Proc. UNIversal Inclusion Rights and Opportunities for Persons with Disabilities in the Academic Context, (Torino (Italy)), May 2016

[3] S. Carrato, S. Marsi, E. Medvet, F. A. Pellegrino, G. Ramponi, and M. Vittori, ‘’Computer vision for the blind: a dataset for experiments on face detection and recognition,” in Proc. MIPRO 2016 - 39th International Convention on ICT, Electronics and Microelectronics, (Opatija (HR)), 30 May - 3 June 2016

[4] Florian Schroff, Dmitry Kalenichenko, James Philbin, ”FaceNet: A Unified Embedding for Face Recognition and Clustering”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015

[5] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, ‘’Learning convolutional feature hierarchies for visual recognition,” in Proc. NIPS, 2010, pp. 1090-1098

[6] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi MaPCANet: ‘’A Simple Deep Learning Baseline for Image Classification”, IEEE Trans. on Image Processing, vol. 24, no. 12, Dec. 2015.

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, ”Rethinking the Inception Architecture for Computer Vision”, arXiv:1512.00567v3 [cs.CV] 11 Dec 2015

[8] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf ”Deepface: Closing the gap to human-level performance in face verification”. In CVPR, pages 1701-1708, 2014.

[9] M. D. Zeiler and R. Fergus. ”Visualizing and understanding convolutional networks”. CoRR, abs/1311.2901, 2013. 2, 4, 6

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. ”Going deeper with convolutions”. CoRR, abs/1409.4842, 2014. 2, 4, 5, 6, 9

[11] B. Amos, B. Ludwiczuk, M. Satyanarayanan, ”Openface: A general-purpose face recognition library with mobile applications,” CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[12] Viola, P., Jones, M.J., ”Robust real-time face detection”. International journal of computer vision 57(2) (2004) 137154

[13] Visage Technologies- Face Tracking and Analysis, https://visagetechnologies.com/products-and-services/visagesd

[14] Liao, S., Jain, A.K., Li, S.Z., ”A fast and accurate unconstrained face detector”. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(2) (2016)

[15] Dundar, A., Jin, J., Martini, B., Culurciello, E., ”Embedded streaming deep neural networks accelerator with applications”. IEEE Transactions on Neural Networks and Learning Systems (2016)

[16] Markus, N., Frljak, M., Pandzic, I.S., Ahlberg, J., Forchheimer, R., ”Object detec- tion with pixel intensity comparisons organized in decision trees”. arXiv preprint arXiv:1305.4537 (2013)

[17] Google Developers, https://developers.google.com

[18] M. De Marco, G. Fenu, E. Medvet, and F.A. Pellegrino, ‘’Computer Vision for the Blind: a Comparison of Face Detectors in a Relevant Scenario,” in Proc. GoodTechs 2016 - 2nd EAI International Conference on Smart Objects and Technologies for Social Good, Venice, Italy, November 30 - December 1, 2016.

[19] Z. Zhu, P. Luo, X. Wang, and X. Tang. ”Recover canonicalview faces in the wild with deep neural networks”. CoRR, abs/1404.3543, 2014. 2

[20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. ”Deepface: Closing the gap to human-level performance in face verification”. In IEEE Conf. on CVPR, 2014. 1, 2, 5, 8

[21] Y. Sun, X. Wang, and X. Tang. ”Deeply learned face representations are sparse, selective, and robust”. CoRR, abs/1412.1265, 2014. 1, 2, 5, 8

[22] K. Q.Weinberger, J. Blitzer, and L. K. Saul. ”Distance metric learning for large margin nearest neighbor classification”. In NIPS. MIT Press, 2006. 2, 3

[23] Alfredo Canziani & Eugenio Culurciello, ”An Analysis of Deep Neural Network Models for Practical Applications”, arXiv:1605.07678v2 [cs.CV] 30 May 2016.