Face Verification and Recognition for Digital Forensics and Information Security

Giuseppe Amato[1], Fabrizio Falchi[1], Claudio Gennaro[1], Fabio Valerio Massoli[1], Nikolaos Passalis[2], Anastasios Tefas[3], Alessandro Trivilini[4] and Claudio Vairo[1]

***Abstract*—In this paper, we present an extensive evaluation of face recognition and verification approaches performed by the European COST Action MULTI-modal Imaging of FOREnsic SciEnce Evidence (MULTI-FORESEE). The aim of the study is to evaluate various face recognition and verification methods, ranging from methods based on facial landmarks to state-of-the-art off-the-shelf pre-trained Convolutional Neural Networks (CNN), as well as CNN models directly trained for the task at hand. To fulfill this objective, we carefully designed and implemented a realistic data acquisition process, that corresponds to a typical face verification setup, and collected a challenging dataset to evaluate the real world performance of the aforementioned methods. Apart from verifying the effectiveness of deep learning approaches in a specific scenario, several important limitations are identified and discussed through the paper, providing valuable insight for future research directions in the field.**

***Index Terms*—Forensics; Face Verification; Deep Learning; Surveillance; Security.**

I. Introduction

This paper bases its results in the European COST Action entitled MULTI-modal Imaging of FOREnsic SciEnce Evidence (MULTI-FORESEE) - tools for Forensic Science. In particular, it focuses its attention on activities performed by the working group dedicated to digital forensics strand, by exploiting the possibility to explore and applying face recognition approaches through specific Round Robin Studies in the security environment. The aim of the Action is to promote innovative, multi-informative, operationally deployable and commercially exploitable imaging solutions/technology to analyze forensic evidence. Forensic evidence includes, but it's not limited to, images, digital evidence, fingermarks, biofluids, fibers, documents, and living individuals.

The motivation of the group takes into account the main goals of the European Action, such as taking advantages of the unique networking and capacity-building capabilities provided by the COST framework to bring together their knowledge and expertise. By acting in this way, the main contribution can be measured in terms of engaging in a synergistic approach to boost face recognition developments, allowing highly reliable and multi-informative intelligence to be provided to investigators, prosecutors, and defence.

To fulfill this objective, we have created a challenging application scenario that allows us to analyze different aspects of facial recognition in a real world setting. In particular, the scenario we want to address in this paper is to determine if a person entered in an unconstrained monitored environment on a certain date (for example when something happened in that environment) is the same person that entered in the monitored environment some other day. This is done by comparing a pair of faces and determine if the faces belong to the same person or not.

Several approaches use facial landmarks as representative information of the face to be recognized [1]–[4]. These approaches do not achieve very good results in recognizing people and in performing the face verification task. However, they can provide an analytical measure of the similarity between two faces, and this can be used as a valuable forensic evidence in trials. On the other hand, less interpretable, yet more powerful deep learning approaches have been recently used in security and surveillance [5]–[7] and in face verification and recognition [8], [9], with very good results.

In this paper, we measure the accuracy in verifying if two faces belong to the same person or not by using both approaches based on facial landmarks and on Convolutional Neural Networks (CNN). We also provide a face dataset collected to perform the experiments. The dataset is composed by 39,037 faces images belonging to 42 different identities and it is publicly available for download, serving as a challenging benchmark which allows other researchers to readily evaluate their methods on the employed realistic face verification setup.

The rest of paper is organized as follows: Section II describe the proposed approaches and the techniques used. In Section III, we briefly present the face dataset we provide and that we used to perform the experiments, which are reported in Section IV. Finally, Section V concludes the paper.

II. Facial Matching Approaches

In this section, we present the approaches that we used to perform our experiments of face recognition and face verification for the COST Action. We exploit and compare facial landmarks and off-the-shelf CNN models pre-trained on large face datasets to build facial features used to perform face verification. We also investigate the training from scratch of a CNN model with the proposed dataset to perform face verification.

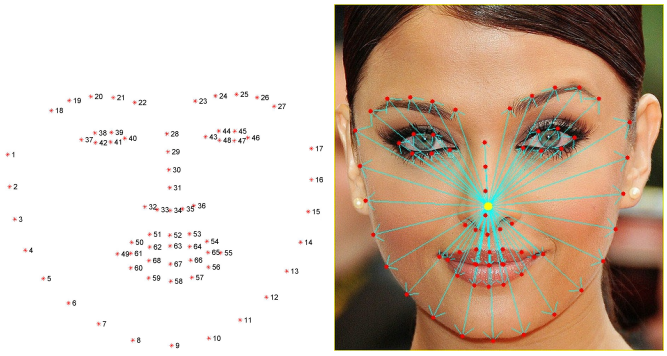
[1]Institute of Information Science and Technologies of the National Research Council of Italy (ISTI-CNR), via G. Moruzzi 1, 56124, Pisa, Italy name.surname at isti.cnr.it

[2]Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland, e-mail: nikolaos.passalis@tuni.fi

[3]Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, e-mail: tefas@csd.auth.gr

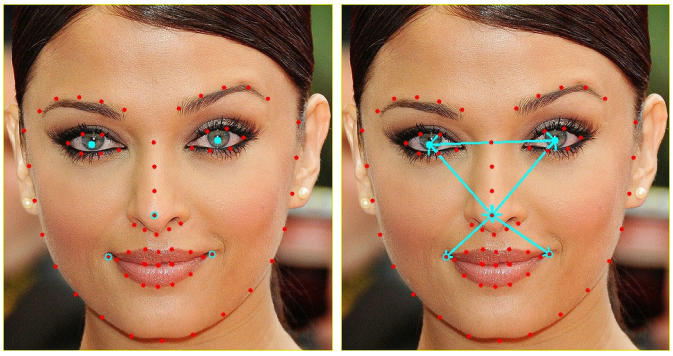
[4]University of Applied Sciences of Southern Switzerland, Department of innovative technologies, Digital forensics lab, Lugano, Switzerland, e-mail: alessandro.trivilini@supsi.ch

978-1-7281-2827-6/19/$31.00 2019 European Union



(a) 68 facial landmarks. (b) 68-points feature.

Fig. 1: Facial landmarks used and distances from the centroid of the face to all 68 facial landmarks, used to build the 68-points features.



(a) Selected 5 nodal points. (b) 5 nodal points distances.

Fig. 2: Nodal points and distances used to build the 5-points features.

A. Facial Landmarks

Facial landmarks are key points along the shape of the face that can be used as face features to perform several tasks like improve face recognition, align facial images, distinguish males and females, estimate the head pose, and so on.

Some of these points and other points computed from the facial landmarks (for example the center of the eye computed from the points delimiting the eye) may be more representative than others. For example, the eyes, the nose, and the mouth are very representative parts of a person's face, so points relative to these parts of the face can be more relevant to represent that face. We refer to these points as *nodal points*.

In order to extract the facial landmarks from an image, we used the dlib library [10]. In particular, the facial landmark detector is an implementation of the approach presented by Kazemi et al. in [11]. It returns an array of 68 points in form of (x,y) coordinates that map to facial structures of the face, as shown in Figure 1(a).

The distances between both nodal points and facial landmarks can be used to build a feature of the face that can be compared with other faces features. In particular, we computed three features based on the distances between nodal points and facial landmarks: a) the *5-points* feature, b) the *68-points* feature and c) the *pairs* feature. All the distances used to compute these features are normalized to the size of the bounding box of the face. In particular, each distance is divided by the diagonal of the bounding box.

1) 5-points feature: In order to build the 5-points feature, we used five specific nodal points: the centroids of the two eyes, the center of the nose, and the sides of the mouth. The centroids of the two eyes are computed from the six facial landmarks of each eye returned by the dlib library. For the nodal points of the nose and of the mouth, instead, we used directly some of the facial landmarks, respectively the landmark #31 for the nose and the landmarks #49 and #55 for the sides of the mouth (see Figure 2(a)). We used these nodal points to compute the following 5 distances (see Figure 2(b)):

- left eye centroid - right eye centroid
- left eye centroid - nose
- right eye centroid - nose
- nose - left mouth
- nose - right mouth

This produces a 5-dimensional float vector that we used as 5-point feature of the face.

2) 68-points feature: For the 68-points feature, we computed the centroid of all the 68 facial landmarks returned by the dlib library and we computed the distance between this point and all the 68 facial landmarks (see Figure 1(b)). This produces a 68-dimensional float vector that we used as 68-feature of the face.

3) Pairs feature: The pairs feature is obtained by computing the distance of all unique pairs of points taken from the 68 facial landmarks computed on the input face, as suggested in [4]. This produces a vector of 2,278 float distances that we used as pairs feature of the face.

B. Off-the-shelf Pre-trained Models

Convolutional Neural Networks are widely used to perform classification tasks with very good results, while their recent application in face verification and recognition tasks also demonstrated their great potential in security and surveillance [5]–[9]. The immense power of CNNs arises from the ability of convolutional layers to detect various features (relevant to the task at hand) and extract representations that capture increasingly complex concepts, as the depth of a network increases. In particular, the output of the last layer before the output of the network is, in fact, a high-level representation of the input image, that can be used as a global descriptive feature for that image. In the rest of this paper, we call this representation of a face *deep feature*, to distinguish it from the traditional facial landmark-based features. This feature can be compared to other deep features computed on other faces. Close deep features vectors mean that the input faces from which the features are extracted are semantically similar. Therefore, if their distance is below a given threshold, we can conclude that the two faces belong to the same person.

In this paper, we use the ResNet-50 CNN (shortly ResNet-50_ft) [12], which is a 50-layers CNN pre-trained to recognize

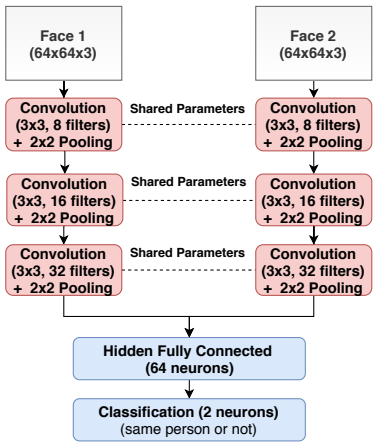


Fig. 3: Convolutional siamese architecture used for face verification. The network was trained to identify image pairs that belong to the same person.

faces. In particular, the model has been trained on the MS-Celeb-1M [13] dataset (10 million images of 100 thousand different identities) and fine-tuned on the VGGFace2 [14] dataset (3,31 million images of 9,131 different identities). We take the output of the pool5|7x7_s1 layer as deep feature, which is a 2,048 size float vector.

C. Training from scratch

For training a face verification neural network from scratch we employed a convolutional siamese architecture [15]. The used network architecture is depicted in Figure 3. Two image pairs of size 64×64 each were fed to the two streams of the network, followed by a convolutional layer composed of 8 filters of size 3×3 and a 2×2 max pooling layer. Then, another two convolutional and pooling layers with 16 and 32 filters respectively follow. These layers were applied separately on the input images and the representations extracted from each of them were merged and fed to the final stream. Then, two fully connected layers with 64 and two neurons follow, allowing for performing face image verification. The output of the network corresponds to the two possible face verification outcomes: same person or not. The *relu* activation function was used for all the layers, except for the final classification layer where the *softmax* activation was used. This architecture was used for most of the conducted experiments, unless otherwise stated. The network was trained to identify image pairs that belong to the same person using the cross-entropy loss. To this end, the Adam optimizer with the default hyper-parameters [16] was used (the optimization ran for 50,000 training iterations with batches of 64 randomly sampled pairs).

III. Face Dataset

In order to perform the experiments, we built a face dataset. We collected several images, over a period of time of more than one and a half year, of people entering two monitored environments. We manually labeled the acquired images of the dataset with the corresponding identity. The dataset is

(a) Pairs of same person.

(b) Pairs of different persons.

Fig. 4: Example of pairs of same person and different persons.

organized in folders, one for each identity, containing the faces corresponding to that person. Each filename has the following structure: $PERSON\_SEQNUM@DATETIME.jpg$, where $PERSON$ is the identifier of the person whom the face belongs to, $SEQNUM$ is an increasing counter of four digits ranging from 1 to the total number of faces for that person, and $DATETIME$ is the acquisition timestamp in ISO 8601 format (for example: p0042_0001@2017-10-17T13-45-34Z.jpg). The dataset is composed of 39,037 faces images belonging to 42 different identities. The name of each person has been replaced with a numerical ID in order to preserve the people's privacy. The minimum number of images per identity is 21, the maximum is 8,304, the average is 950.

The dataset is available for the download at the link: *http://deepfeatures.org/AIMIRFace.zip*.

A. Pairs View

We provide a view of the dataset organized in pairs of same and different persons, in order to provide a specific test set for the face verification scenario. In particular, we prepared a list of pairs of faces belonging to the same person, and a list of pairs of faces of different persons (see Figure 4 for some examples). With these pairs, it is possible to validate the analyzed approaches by comparing the faces images of each pair defined in both lists and report the results in determining the correct pairs of same persons and the correct pairs of different persons.

The pairs are presented as text files. We provide 3,444 positive pairs and 3,444 negative pairs listed, respectively, in

TABLE I: Verification accuracy on pairs with different face features.

Feature	Our dataset	LFW
5-points feature	55.8%	55.0%
68-points feature	54.4%	52.5%
Pairs feature	54.2%	53.7%
Deep feature	98.15%	98.9%

the files posPairs.txt and negPairs.txt located in the zip file of the dataset. Each row in both positive and negative pairs files is composed of two elements separated by a space containing the name of the file as presented before, without the file extension (an example of negative pair is: p0007_0175@2018-04-20T11-50-42Z p0038_0010@2018-07-27T14-35-28Z). We also provide a division of the pairs in train set and test set for both positive and negative pairs. The training pairs are 2,444, the test pairs are 1,000.

In order to create the negative pairs, we selected, for each person, four pairs with all the other persons in the dataset. Both faces of the pairs are selected randomly. Regarding the positive pairs, we selected 82 pairs for each person. If a person has less than 82×2 images, we randomly selected the pairs among the available images. Otherwise, we selected pairs with the greatest time difference between the acquisition times of the two images of the pair. This in order to have the most different conditions in the pairs of the same person.

IV. Experiments and Results

In this Section, we present the experiments we performed on the proposed dataset for the face verification task. We tested the accuracy in analyzing the pairs using the different approaches described in Section II and at different resolutions of the images. We also measured the performance in classifying the persons of the dataset with the deep features.

A. Verification Accuracy on Pairs

In this experiment, we show the accuracy in verifying that two faces belong to the same person or not. To this purpose, we use the pairs, both the training and the test sets, presented in Section III-A. In particular, we extracted the different features from the faces of each pair of the training set. We then computed the euclidean distance between the extracted features of the pair. We compared the obtained distance with a threshold in order to determine if the faces belong to the same person or not, i.e. if the distance is below the threshold. We performed the experiment for increasing values of the threshold and for each threshold we computed the corresponding accuracy measured as: $(TP + TN)/N$, where TP is the number of true positives (i.e. the pairs of same people correctly recognized as same), TN is the number of true negatives (i.e. the pairs of different people correctly recognized as different), and N is the total number of pairs in the training set. We used the threshold value that results with the biggest accuracy on the training set, and we computed the corresponding accuracy value on the test set.

Fig. 5: Different image resolutions used for the experiment. From original resolution on the right to 8×8 pixels on the left.

We performed the same experiment on our dataset and on the dataset Labeled Faces in the Wild (LFW) [17]. LFW is a very famous face dataset, which contains more than 13,000 faces of 5,750 different identities. All images in LFW are 250×250 pixels and the face is aligned to be in the center of the image. We performed the face detection and we computed the facial landmark points by using the dlib library. We then extracted the four different features (5-points, 68-points, Pairs, and deep features), from all the faces in the proposed dataset. LFW provides a pairs view, split into train pairs and test pairs, as well as the proposed dataset. Also in this case, we computed the best threshold on the training set and we computed the accuracy value on the test set using the threshold determined.

We carried out this experiment by using the three features based on landmarks presented in Section II-A and the deep features from ResNet_50 presented in Section II-B.

Table I reports the accuracy of the different features used on both the datasets. We can see that the deep feature heavily outperforms all the landmarks-based features in both the analyzed datasets. This is probably due to the fact the faces are not frontalized, which is among the future activities of this work. We also notice that the deep feature works slightly better with LFW dataset than with the proposed dataset (98.9% against 98.15%). On the other hand, the accuracy achieved by the landmarks-based features is higher on the proposed dataset than on the LFW dataset.

B. Verification Accuracy at Different Resolutions

In this experiment, we investigate how the resolution of the input faces can affect the accuracy while performing the face verification task. Sometimes it is needed to perform this task on very low-resolution input images, such as frames of surveillance cameras, or faces acquired in the very background of a picture, which results very small. Therefore, we took into consideration the following additional resolutions for the faces of our dataset: 64×64, 32×32, 16×16, 8×8 pixels (see Figure 5 for some examples). We used the free image editing library ImageMagick [1] to resize the images. In particular we used the *convert* command with the desired *geometry* parameter.

We replicated the experiment presented in Section IV-A but we only used the deep feature in this case. We studied both

[1] https://www.imagemagick.org/

TABLE II: Verification accuracy on pairs at different face resolutions.

	res8	res16	res32	res64	orig
res8	50.45				
res16	59.05	85.50			
res32	50.35	81.48	97.60		
res64	50.13	74.33	97.60	97.95	
orig	50.10	72.28	97.40	98.08	98.15

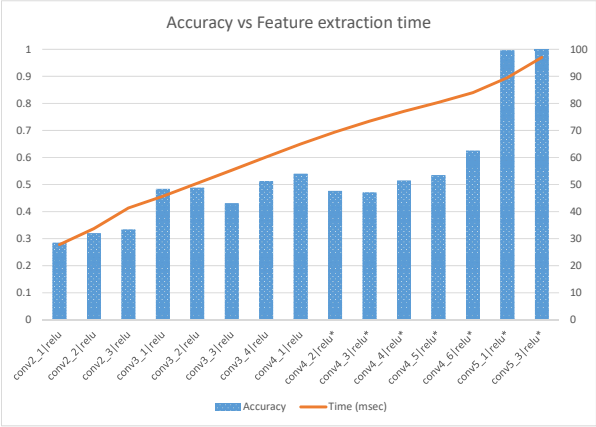


Fig. 6: Comparison between the classification accuracy and the (CPU) extraction time of the features for the different levels of the neural network ResNet-50_ft. The features of the levels with the name followed by * were averaged across the two dimensions of the filter.

the scenario in which the two faces of each pair are at the same resolution and the scenario in which the resolutions of the faces of the pairs are different (cross resolution). Results are shown in Table II. We can notice that with resolutions down to 32×32 pixels the results are very good, even in the cross resolution scenario. When the resolution goes down to 16×16, the scenario with the same resolution still maintains a good accuracy of 85.5%, but it quickly drops in the cross resolution scenario as the resolution of the other face of the pair increases. Finally, with a 8×8 resolution we see that the accuracy goes down to 50% in almost every scenario, with the exception of cross resolution scenario with the 16×16 resolution that achieves an accuracy of almost 60%. Anyway, we can conclude that, with the analyzed approaches, the face verification task can be executed only with faces down to 16×16 pixels.

C. Classification

In this experiment we take into consideration the whole face dataset, ignoring the pairs, and we computed the accuracy of the proposed approach with deep feature in classifying correctly the people.

TABLE III: Training from scratch: Evaluating different regularization methods.

Method	Accuracy	Precision	Recall	F1
No regularization	75.04	79.29	73.09	76.05
BN [18]	77.96	81.25	76.48	78.49
BN [18] + Dropout [19]	**81.37**	**86.45**	**78.47**	**82.26**

TABLE IV: Training from scratch: Evaluating different pooling approaches.

Method	Accuracy	Precision	Recall	F1
Max Pool.	81.37	86.45	78.47	82.26
Avg Pool.	**81.93**	**87.48**	78.75	**82.87**
Global Max Pool.	70.43	78.07	67.70	72.51
Global Avg Pool.	73.12	84.00	69.01	75.76
Max Pooling + SPP [5]	74.51	83.31	70.86	76.57
Avg. Pooling + Remove last	81.52	80.90	**81.95**	81.38

In particular, we evaluated the performance of the different levels of the neural network ResNet-50_ft in terms of classification accuracy and the computational effort required to extract the related features. Figure 6 shows the results of this analysis, where the computational effort is expressed as the time required (in msec) for one core of a x86 based processor @ 3GHz to extract the one feature. The names of the layers are the same as the ones used in [14]. Note the significant loss of accuracy when moving from the penultimate level to the third last of the network.

D. Training from scratch

The main purpose of these experiments is to evaluate the accuracy of deep neural networks on face verification when a small, yet specialized for the task at hand, dataset was used for training. There are several challenges associated with the dataset and/or the network architecture that was used for these experiments. First, the relatively small size of the dataset poses several challenges for training a network from scratch. Therefore, first, we evaluated different regularization approaches using the network architecture described in Section II-C. We report the experimental results for training with different regularization methods in Table III. The best results are obtained when the batch normalization (BN) approach [18], is combined with the dropout method (the dropout rate was set to 0.2) [19].

Then, we examined the effect of using different pooling techniques on the verification metrics. The results are reported in Table IV. Several conclusions can be drawn from the reported results since the type of the pooling layers used in the network seems to have a significant effect on the verification metrics. First, using average pooling seems to leads to slightly better results in most of the cases. However, the most important finding is that using global pooling methods (when applied in the layer before the fully connected layers) can significantly reduce the verification accuracy, since it discards a large amount of spatial information. This is confirmed by the last two experiments, since a) using spatial pyramid pooling [20] performs better than using global pooling and b) completely

TABLE V: Training from scratch: Varying the number of images per person.

# images per person	Accuracy	Precision	Recall	F1
2	56.88	16.51	85.98	27.63
5	68.68	44.69	85.96	58.78
10	77.19	64.92	86.09	73.94
20	82.08	77.49	85.40	81.22
50	83.43	82.57	84.12	83.27

removing the last pooling layer significantly boost the results achieving almost the best verification results (close to the best performing method).

Finally, we also evaluated the effect of the number of training samples per person on the verification accuracy, after fine-tuning the network architecture (doubling the number of filters in the convolutional layers and increasing dropout rate to $p = 0.3$, as determined by additional experiments). The results are reported in the Table V. As it was expected, using more training samples per person significantly improves the accuracy of the networks. Also, when only two training samples were used per person, the network was not able to perform any useful predictions regarding the identity of the persons in the pairs (the performance was close to the performance of a random classifier).

V. Conclusions

In this article, we presented a review of state-of-the-art tools for facial verification and recognition along with a dataset of faces for evaluating their performance. The objective was to verify their effectiveness and possibly identify their limitations. The pre-trained convolutional networks exhibit astonishing performance on facial verification but have the disadvantage of not providing any information on which attributes the possible matching of two faces was based. Indeed, facial landmarks are of significant importance in forensics as they are generally accepted in court as evidence.

Several interesting conclusions were also drawn from the results obtained when training siamese architectures from scratch for the task of face verification. First, selecting the most appropriate architecture is especially important, e.g., we should avoid using global pooling, since it discards useful spatial information, which is especially important for the task of face verification. Furthermore, it was confirmed again that the amount of the data is among the most crucial factors when training the network from scratch, while using dropout and other regularization methods can improve the performance, especially when relatively small datasets are used. Using more advanced network architectures, e.g., residual networks [6], combining this dataset with other face datasets and/or fine-tuning existing networks, that were already trained for face verification, can potentially further improve the verification results.

Acknowledgments

This publication is based upon work from COST Action 16101 "MULTI-modal Imaging of FOREnsic SciEnce Evidence" (MULTI-FORESEE), supported by COST (European Cooperation in Science and Technology).

We also thank the support of NVIDIA Corporation with the donation of the Jetson TX2 board used for this research.

References

[1] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Advances in neural information processing systems, 2014, pp. 1988–1996.

[2] C. Sanderson, M. T. Harandi, Y. Wong, and B. C. Lovell, "Combined learning of salient local descriptors and distance metrics for image set face verification," in Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on. IEEE, 2012, pp. 294–299.

[3] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 365–372.

[4] A. G. Rassadin, A. S. Gruzdev, and A. V. Savchenko, "Group-level emotion recognition using transfer learning from face identification," arXiv preprint arXiv:1709.01688, 2017.

[5] E. Granger, M. Kiran, L.-A. Blais-Morin et al., "A comparison of cnn-based face and head detectors for real-time video surveillance applications," in 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, 2017, pp. 1–7.

[6] H. Kavalionak, C. Gennaro, G. Amato, C. Vairo, C. Perciante, C. Meghini, and F. Falchi, "Distributed video surveillance using smart cameras," Journal of Grid Computing, 2018, pp. 1–19.

[7] P. Barsocchi, A. Calabrò, E. Ferro, C. Gennaro, E. Marchetti, and C. Vairo, "Boosting a low-cost smart home environment with usage and access control rules," Sensors, vol. 18, no. 6, 2018, p. 1886.

[8] X. Wu, L. Song, R. He, and T. Tan, "Coupled deep learning for heterogeneous face recognition," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[9] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Facial-based intrusion detection system with deep learning in embedded devices," in Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing, ser. SSIP 2018. New York, NY, USA: ACM, 2018, pp. 64–68. [Online]. Available: http://doi.acm.org/10.1145/3290589.3290598

[10] "Dlib library," http://dlib.net/, accessed: 2018-04-13.

[11] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867–1874.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[13] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in European Conference on Computer Vision. Springer, 2016, pp. 87–102.

[14] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," arXiv:1710.08092, 2017.

[15] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in Proceddings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 539–546.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[17] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, 2014, pp. 1929–1958.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in Proceedings of the European Conference on Computer Vision, 2014, pp. 346–361.