4th International Conference on Advanced Technologies
For Signal and Image Processing – ATSIP' 2018
March 21-24, 2018 – Sousse, Tunisia

State-of-the-art Face Recognition Performance Using Publicly Available Software and Datasets

Mohamed Amine Hmani
Samovar CNRS UMR 5157, Télécom SudParis
Université Paris-Saclay
Evry, France
mohamed.hmani@telecom-sudparis.eu

Dijana Petrovska-Delacrétaz
Samovar CNRS UMR 5157, Télécom SudParis
Université Paris-Saclay
Evry, France
dijana.petrovska@telecom-sudparis.eu

***Abstract*—We are interested in the reproducibility of face recognition systems. By reproducibility we mean: is the scientific community, and are the researchers from different sides, capable of reproducing the last published results by a big company, that has at its disposal huge computational power and huge proprietary databases?**

With the constant advancements in GPU computation power and availability of open-source software, the reproducibility of published results should not be a problem. But, if architectures of the systems are private and databases are proprietary, the reproducibility of published results can not be easily attained. To tackle this problem, we focus on training and evaluation of face recognition systems on publicly available data and software. We are also interested in comparing the best Deep Neural Net (DNN) based results with a baseline "classical" system. This paper exploits the OpenFace open-source system to generate a deep convolutional neural network model using publicly available datasets. We study the impact of the size of the datasets, their quality and compare the performance to a classical face recognition approach. Our focus is to have a fully reproducible model. To this end, we used publicly available datasets (FRGC, MS-celeb-1M, MOBIO, LFW), as well publicly available software (OpenFace) to train our model in order to do face recognition. Our best trained model achieves 97.52% accuracy on the Labelled in the Wild dataset (LFW) dataset which is lower than Google's best reported results of 99.96% but slightly better than FaceBook's reported result of 97.35%. We also evaluated our best model on the challenging video dataset MOBIO and report competitive results with the best reported results on this database.

***Index Terms*—Deep Learning, Triplets, MOBIO, LFW, MS-Celeb-1M**

I. Introduction

In the last years, mainly due to the advances of deep learning, more concretely convolutional networks, the quality of image recognition and object detection has been progressing at a dramatic pace. With the advent of GPU computation and big datasets, neural networks saw a huge resurgence. This results in huge improvements in image recognition and consequently face recognition. Many works [1]–[5] report near perfect biometric performance. But in most cases, all systems are either proprietary or trained on private datasets. This raises the problem of the difficulty of reproducing published results [6].

In this paper we try to reach the best reported results on the Labeled Faces in the Wild (LFW) [7] database, by using the open-source OpenFace [8] software. This software is based on the Google's FaceNet architecture [5] that achieves the best results on LFW, but is fully proprietary. CMU has already worked in this direction, but their published results of 92.92% are far from the 99.96% that Google got on LFW. We have chosen to exploit the publicly available MS-celeb-1M [9] dataset. We evaluate the performance of our newly trained system on the (LFW), as well as the MOBIO [10] dataset (a very challenging audio-visual dataset). The rest of the paper is organized as follows: Section II summarizes the latest achievements of Convolutional Neural Nets and DNN based face recognition. Section III explains our approach to try to reach best published and reproducible results. Our experimental results are given in Section IV, followed by conclusions and perspectives.

II. Related Works

Most of the state-of-the-art face recognition systems are based on artificial neural networks. The best result on the LFW is reported by Google's team [5], exploiting convolutional networks. With the OpenFace software the researchers implemented the architecture of [5]. Therefore in the following paragraph we will first provide a short history related to convolutional networks, followed by a summary of best performing DNN-based face recognition systems.

A. Convolutional Neural Networks

One of the very first convolutional neural networks is LeNET5 [11]. It can be summarized as convolution, pooling and non-linearity. The extracted features are fed into a Multi Layer Perceptron (MLP) to do the final classification. However, due to the low computational power at that time and the fact that datasets were small, the architecture did not find huge success. With the advent of GPUs, the training time saw a drastic reduction. This resulted in many architectures based on LeNET5, such as AlexNET [12], VGG network [13], GoogleNET and Inception [14]. Inception was developed at Google to provide state-of-the-art performance on the ImageNet Large-Scale Visual Recognition Challenge and to be more computationally efficient than other architectures. The inception module acts as a multi-level feature extractor by computing 1x1, 3x3, and 5x5 convolutions within the same module of the network. The output of these filters are then stacked along the channel dimension before being fed into the next layer in the network.

978-1-5386-5239-8/18/$31.00 ©2018 IEEE

B. Deep Neural Network Based Face Recognition Systems

Table I summarizes the most prominent Deep Neural Network (DNN) based face recognition systems. Most of them are either proprietary, where only a description of the system is provided and/or trained on private datasets.

FaceNet [5] was developed by Google. It is a unified system for face verification, identification and clustering. It extracts Euclidean representations from images with the advantage of being general purpose. The features are also compact (with a dimension of 128) in comparison with traditional representations (bottleneck features for example). The system was trained on a huge private dataset of 260 M images from 8 M subjects. It was trained for 1000 hours. FaceNet has the best reported accuracy of 99.96% on the LFW database.

DeepID2 [3] was developed by the Department of Information Engineering of the Chinese University of Hong Kong. The features are learned using deep convolutional networks. The face identification task increases the inter-personal variations by drawing apart DeepID2 features extracted from different identities, while the face verification task reduces the intra-personal variations by pulling DeepID2 together extracted from the same identity, both of which are essential to face recognition. It was trained on a private dataset consisting of 200k images from 10k subjects. Compared to other datasets such as Google's or Facebook's systems, the size of the database can be considered relatively small. It gives 99.15% verification accuracy on the LFW dataset.

VGG-DeepFace [4] was developed by the Visual Geometry Group (VGG) from the university of Oxford. The system was trained on 2.6 M images containing 2.6 k identities. The published performance on LFW is 98.95%. The VGG system is essentially a very deep convolutional neural network. It leverages two distinct methods for the training, N-way classification and triplet embedding. In the case of this system, the N-way has the advantage of faster training, while on the other hand triplet embedding gives better overall performance.

DeepFace [1] is developed by Facebook. It processes images in two steps. First it corrects the angle of a face so that the person in the picture becomes forward facing, using a 3-D model of an 'average' forward-looking face. The second step is to propagate the face to the DNN in order to extract it's representation. The system was trained on a private dataset consisting of 4.4 M images from 4 thousand subjects (average of 1k per subject). It has 97.35% accuracy on LFW.

CASIANet [2] was developed by the Institute of Automation, Chinese Academy of Sciences (CASIA). The system is inspired from many recent successful networks including very deep architecture, low dimensional representation and multiple loss functions. It was trained on the publicly available CASIA dataset which consists of 500 thousand images representing 10 thousand identities. The reported performance of the system on LFW is 96.13%.

OpenFace [8] is an implementation of the FaceNet system based on [5]. The source code is publicly available as well as the trained model. It was trained on the CASIA-webfaces and FaceScrub, publicly available databases. The system has 92.92% accuracy on LFW.

In the rest of this paper we limit the study to using OpenFace as the DNN architecture for technical limitation and limited time, and because it implements the best performing architecture reported in [5]. In fact, the average DNN training session takes one week thus preventing us from reproducing multiple systems. Our goal was to obtain better results by using a bigger publicly available database. In the next section we will explain our approach to have a fully reproducible training model and results using OpenFace.

III. Approach to fully reproducible training Model and results

In order to detail our approach, first, we will explain how the images were preprocessed before feeding them to the neural network. Afterwards, we will expose the neural network architecture, the training as well as the evaluation datasets, the protocols that were followed in the evaluation process, before finally giving the training conditions.

A. Preprocessing

The preprocessing is done using the Open Source Computer Vision Library (OpenCV) [15] and the DLIB library [16]. First, the face is detected using DLIB face detector. Then, landmarks of the face are detected also using DLIB. The landmarks that are used for normalization are the eyes and the nose. Using these landmarks, the face is rotated, scaled and cropped. The resulting image has 96x96 pixels. Fig1 Shows the effects of preprocessing on one image.

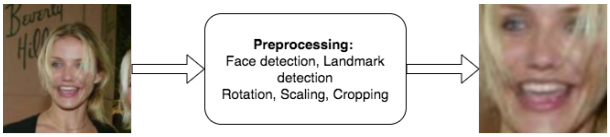

Fig. 1. Example of the preprocessing of an image from LFW using eyes and nose positions

B. OpenFace Neural Network Architecture

The DNN architecture used in OpenFace is an implementation of the FaceNet model based on [5]. It is a deep convolutional neural network. It was inspired from the inception network [17]. For our work we have chosen the nn4.small2 architecture because it is less complex than the default architecture nn2 and because our tests show better overall performance given the same data and the same training time. It consists of an input layer, an output layer and 24 hidden layers among which there are 7 inception layers. The whole network counts 3733968 parameters. It aims to extract feature vectors that give the best possible separation between subjects. It uses triplet embedding to optimize the representations. [5] details the process of the triplet selection and optimization. The loss function defined in Eq.1 is based on the triplet loss optimization scheme which consists of choosing two samples from the same class (the anchor and the positive) and a sample for a different class (the negative). The goal of the training is to separate the Anchor-Positive pairs from the Anchor-Negative pairs by at least the margin α. The triplet mining is done online by selecting the triplets that are not arranged correctly. We select the hard negative

TABLE I
STATE-OF-THE-ART DEEP NEURAL NETWORK BASED FACE RECOGNITION SYSTEMS

System	Availability of the source code	Training Dataset	Validation Set	Results	Reproducibility
FaceNet [5]	Private	260M images, 8000k subjects, private	LFW	99.96%	No
DeepID2 [3]	Private	0.2M images 10k subjects , private	LFW	99.15%	No
VGG-DeepFace [4]	Public	2.6M images 2.6k subjects, public	LFW	98.95%	Yes
DeepFace [1]	Private	4.4M images 4k subjects, private	LFW	97.35%	No
CASIANet [2]	Public	0.5M images 10k subjects, public	LFW	96.13%	Yes
OpenFace [8]	Public	0.6M images, 11k subjects, public	LFW	92.92%	Yes

triplets where the anchor negative-distance is less than the anchor-positive distance.

$$L(\theta) = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \quad (1)$$

In Eq.1, θ represents the network parameters, x_i^a is the anchor sample, x_i^p the positive sample, and x_i^n the negative sample for subject i. $f(x)$ is the DNN representation of the image x. In order for the training to be efficient (to save computing time), only the triplets that verify Eq2 rule are selected, as other triplets will not improve the network performance.

$$\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha > 0 \quad (2)$$

This selection process allows for the training to run faster and be more efficient because we won't need to backpropagate triplets that have little effect. If a triplet does not verify the inequality from Eq.2 then the considered samples contain too little intra-class variance and a high inter-class variance. As a result of such training, the network outputs a low-dimensional representation of an input image which consists in a feature vector of size 128 and norm 1. This representation can be leveraged to do either verification, identification or clustering. In this article, we will focus on the verification performance.

C. Training Datasets

The training datasets that were used in this work are the FRGC dataset (because it is a relatively big dataset at the time that it was introduced), and the MS-celeb-1M [9] (because this is to our knowledge among the biggest publicly available datasets). More details about these databases are given below.

1) FRGC: The Face Recognition Grand Challenge (FRGC) [18] dataset contains 568 subjects with a total of 39328 images. The dataset was captured under controlled and uncontrolled conditions. In the controlled conditions, the face orientation and the illumination as well as the pose were fixed, which is not the case for the uncontrolled conditions as the pictures were taken in different settings. The dataset was labeled manually, thus there is almost no mislabeling. We decided to exploit the whole dataset for the training of the OpenFace model. This databases offers a reasonable number of subjects (568), with variable acquisition conditions, as well as enough images per subjects to allow the triplet training that we intend to use.

2) MS-celeb-1M: The MS-celeb-1M is one of the largest publicly available datasets. It has 100K subjects and almost 10M images. Popular search engines are used to provide about 100 images for each subject. The images are collected based on metadata. This results in the dataset having a considerable amount of mislabeled images. The dataset is constructed by Microsoft and is available for noncommercial use. [9] further describes the process of assembling the images and the metric used for the choice of the 100K celebrity provided in the dataset. We used the whole dataset for the training of our neural network.

D. Evaluation Datasets

We report our results on two datasets. Labeled Faces in the Wild (LFW) [7] and MOBIO [10]. Both datasets are publicly available for noncommercial use. LFW is one of the most used benchmarking datasets. As for MOBIO, we decided to use this dataset because it is a hard bimodal dataset where faces are hard to detect in opposition to LFW. Compared to LFW where only still images are given, the MOBIO dataset is providing videos, as faces were captured while subjects were speaking. We can also exploit those multiple images extracted from videos in order to study their influence on the triplet loss optimization.

1) Labeled Faces in the Wild: The LFW dataset contains 13233 target face images with a very large degree of variability in facial expressions, age, race, occlusion and illumination conditions. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector [19]. The protocol specifies two views of the data set. View 1 is for model selection and algorithm development. It contains two sets: 1100 pairs per each class (matched/mismatched) for training and 500 pairs per each class for testing. View 2 is designed for performance reporting. It is divided into 10 sets (folders), each with 300 matched pairs and 300 mismatched pairs. The cross-validation evaluation can be adopted among these 10 folders. The final verification performance is reported as the mean recognition rate and standard error over the 10 fold cross validation. It has to be noted that the task is to do pair matching: given a pair of images the goal is to decide weather they belong to the same subject. This task is similar to face verification, except that the evaluation metrics proposed by the database collectors is to report the accuracy of the pair matching.

2) MOBIO: The MOBIO database [10] is a bi-modal (face/speaker) database recorded from 152 people. The database has a female-male ratio of nearly 1:2 (52 females 100 males). In total 12 sessions were captured for each individual. It consists of 3 sets; training, development and evaluation. In our experiments we used only the development and evaluation sets. We report the result on the protocol

described in [20]. The results are reported separately for males and females because for speaker recognition separating males from females gives better results. Therefore face recognition experiments follow the same principle.

E. Training Conditions

The main target of this study is to understand the impact of the training dataset on the performance. In order to be able to study the effect of the database we first made a baseline system based on recommended parameters from [5]. We set the parameters as follows. The embedding size, meaning the length of the representation, was set to 128. We decided to stop the training based on two criteria, either we reach 1000 epochs or after 170 hours with the condition that results are stagnant. Each epoch consisted of 250 batches. 20 subjects were uniformly sampled in each batch from the dataset and 18 images per subject were also uniformly sampled form the available images for each subject. If less than 18 images are available, we take all available images. Because we are using the triplet loss function we need at least 2 images per subject. Before the training we removed all subjects from the dataset who have less than 2 images where DLIB successfully detected a face. α is a margin used in the process of triplet selection and serves also in separating the anchor form the negatives. α's impact is further explained in [5]. It is set to 0.2 which constitutes a compromise between the complexity of the triplet mining and the separation between the triplets. The hardware configuration is as follows: an Intel core i7 7700k, 64 Go of DDR4 RAM, 1 TB SSD for storage and a NVIDIA Geforce GTX 1080Ti with 12 GB of VRAM. Each epoch of the training consists in optimizing the loss function 250 times (once every batch). The batch training is done as follows:

1) Generate a batch by random sampling from the database.
2) Represent every image in the batch (forward propagation).
3) Select triplet verifying Eq.2. If no triplets are found, return to step 1. Else compute the loss function.
4) Optimize the network parameters (backward propagation).

For the specified training parameters, the batch generation takes 0.02 seconds. The forward propagation takes 0.4 seconds. The triplet selection, if enough triplets are found, takes 0.001 seconds and the backward propagation takes 0.3 seconds. Thus, the batch lasts for almost 0.7 seconds on average. However, if no triplets are found (for example due to not enough variability in the training dataset) the processing time for the batch increases considerably.

Fig 2 illustrates the evolution of the epoch time (250 x average batch time) where there are not enough triplets. This training was done to study the limits of the triplet selection process. We used a small dataset with 50 subjects with 4000 images taken from the MOBIO database. In the beginning, the model can not separate the dataset correctly, thus we find enough triplets to optimize the network. As the network performance improves, it becomes able to discern the identities. This results in less triplets verifying Eq.2. The training process is stacked at step 1, trying different samples in order to find the triplets it needs to compute the loss function. The process may try thousands of configurations before finding hard-negative triplets. This results in exponential increase of the training time. This made us decide to add another condition to stop the training: if the training period exceeds one week (170h) and the results are stagnant.

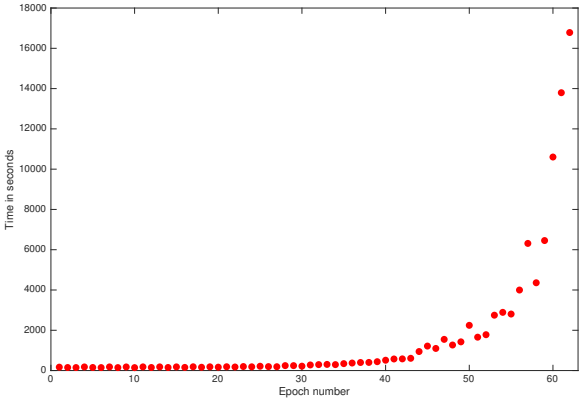

Fig. 2. Illustration of the evolution of the epoch training time using a low variability dataset originating from the MOBIO dataset

IV. Experimental Results

In this section we detail the performance of the different DNN models we obtained by training the architecture using the datasets mentioned in III.3 on the LFW dataset and the MOBIO dataset. We also compare the result of the DNN with a "classical" (not DNN based) approach. For this purpose we have chosen the Direct Linear Discriminant (DLDA) [21] based system, because it has a similar strategy of building a compact image (face) representation model (at the training phase) that we can use to project the new incoming faces in order to be able to compare two face images.

A. Performance on the LFW dataset

Our goal is to obtain the best performance with the available datasets. We achieved a pair matching accuracy of 97.6% on LFW using all of the available images from the MS-Celeb-1M for training the DNN.
We used the preprocessing of OpenFace. As the preprocessing is based on DLIB face detector, it was not able to detected faces in 58 images from LFW. As a fallback, we used images from the deep funneled set of LFW in order to do the verification tests.
We followed the 10-folds cross validation protocol provided by LFW on the view two. In which, 6000 pair matching tests are split in 10 partitions. The accuracy is defined as the mean value of correctly matched pairs divided by the number of pairs in each of the 10 folds.

Table II summarizes the most interesting experiments we did using OpenFace. In **Exp.1**, we used the FRGC dataset, we stopped the training process at 700 epochs because the training time became too long due to not finding enough triplets satisfying the constraint defined in Eq. 2. In **Exp.2**, we tried to get better results using the same dataset by pushing the training further. The loss on the training partition and the accuracy on the LFW were both improved by 3 percentile.

TABLE II
OUR RESULTS ON THE LFW DATASET REPORTING THE INFLUENCE OF THE TRAINING IMAGES COMPARED WITH GOOGLE AND CMU RESULTS

Exps	Preprocessing	Training Dataset	subjects	images/subject	Total number of images	Epochs	Loss on training dataset	Accuaracy
Google	FaceNet	private	8 M	30	260 M	-	-	99.96%
CMU	OpenFace	FaceScrub CASIA-WebFace	11k	50	600k	-	-	92.92
Exp.1	OpenFace	FRGC	568	70	39328	700	6%	77%
Exp.2	OpenFace	FRGC	568	70	39328	1000	3%	80%
Exp.3	Microsoft	MS-celeb-1M	100k	80	8 M	1000	19%	86%
Exp.4	OpenFace	MS-celeb-1M	100k	40	4 M	1000	19%	96.82%
Exp.5	OpenFace	MS-celeb-1M	100k	80	8 M	1000	18%	97.52%

Nevertheless, the results were not convincing. This made evident the need for bigger datasets. The biggest public dataset that we found was MS-celeb-1M. This dataset was the core of the remaining experiments. Microsoft provides a pre-aligned version as well as a raw version of the dataset. In **Exp.3** we used the pre-aligned version by Microsoft. However, the preprocessing was not adequate to the input of the DNN. The images were of varying sizes. After 1000 epochs we obtained 86% accuracy on LFW. The results are better than when using only FRGC as trainig data, but still not at the level of the reported results in the literature. Thus we decided to apply OpenFace alignment on the raw data. This resulted in better overall performance as shown in experiments 4 and 5. In **Exp.4**, only half of the images were used, and at 1000 epochs we obtained 96.82% accuracy on LFW. When we used the whole dataset in **Exp.5** we got 97.52% accuracy on LFW after 1000 epochs. The performance only improved by less than 1 percentile even when doubling the number of images used. We deduced from both this experiments that the most important aspect is the variability in the dataset. It is more beneficial to have more identities than to have more samples per person as the limit for the intra-class variability is achieved fast. We retained the model created in Exp 5 for the remaining tests. Further on we will refer to it as **OpenFace_best**.

B. Performance on the MOBIO dataset

The MOBIO dataset is divided into 3 partitions: training, development and evaluation. For the purpose of this work we did not use the training partition as we wanted to validate the model obtained from training on the MS-celeb-1M. TableIII details the results on MOBIO of our model with the best performance on LFW (**OpenFace_best**). In the table we report the verification performance on both still and automatic protocols. Both these protocols are described further in [20]. For the still protocol we used the still images provided in the framework of the ICB2013 challenge. For the automatic protocol we used 3 and 10 frames from the videos. The frames were selected uniformly from the videos, ie: for 3 frames we took the first, the middle, and the last frame. The results that we obtained on MOBIO are equivalent if not better than the commercial system studied in [20]. To measure the performance on MOBIO we used the HTER metric which is defined as follows:

$$HTER = \frac{FAR(\theta) + FRR(\theta)}{2} \quad (3)$$

θ is the threshold at the Equal Error Rate (EER) defined on the development partition. The False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are then computed on the evaluation dataset using the threshold θ.

TABLE III
RESULTS OF OUR **OPENFACE_BEST** MODEL ON MOBIO

Openface_best	Eval Female (HTER)	Eval Male (HTER)
Still	14.57%	6.43%
3 frames	10.04%	4.79%
10 frames	8.84%	3.99%

The MOBIO dataset is biased towards males with females representing about 30%. We trained OpenFace on a gender independent database. However we find relatively different results when comparing the performance between males and females. The same tendency appear in the systems studied in [20]. The best reported results are 9% on the eval female partition and 5.5% on the eval male partition when using 10 frames, whereas we got 8.8% on the eval female partition and 4% HTER on the eval male. We can attribute the difference in the performance to the poor performance of the face detector on the female images. OpenCV fails to detect the face in 80 female images and only 19 in male images. This may be explained either by a bias in the pretrained face detector module or by bad illumination in the female images.

C. Comparaison with a Traditional Approach (DLDA)

We studied the impact of the size of the training data on the performance in both cases of traditional DLDA approach using the SudFrog software and the deep neural network architecture provided by OpenFace. We decided to compare OpenFace to the DLDA approach because of fundamental similarities. Both, triplet embeddings and DLDA try to reduce the intraclass distance and enlarge the interclass distance. SudFrog is a face recognition system that was developed in Institut Mines Telecom, Telecom SudParis[1]. It is based on space reduction techniques SudFrog does not do neither face detection nor landmark detection. Moreover, SudFrog aims to construct an Euclidean projection space, similar to OpenFace. It must be provided with the eyes, nose and mouth positions for it to do face recognition. For face detection and landmark detection, we use a combination of OpenCV and DLIB. OpenCV was used for face detection.

[1]https://github.com/sudfrog/sudfrog

TABLE IV
COMPARISON OF OUR RESULTS OF THE DNN AND THE DLDA ON MOBIO STILL IMAGES AND LFW

System	Training Dataset	Subjects	Images	MOBIO		LFW
				Eval Female (HTER)	Eval Male (HTER)	
SudFrog_1	FRGC	568	39328	17.43%	10.9%	79.94%
OpenFace_1	FRGC	568	39328	21.87%	18.97%	80%
SudFrog_best	Mobio train set + FRGC	100	4000	12.64%	7.68%	86%
OpenFace_best	MS-celeb-1M	100K	10M	14.57%	6.43%	97.52%

DLIB was used for landmark detection. We used the default detectors provided by the software (front_face.xml for face detection and shape_predictor_68_face_landmarks.dat for landmark detection). In comparison, OpenFace uses DLIB both for face as well as landmark detection. OpenCV is slower but detects more faces than DLIB on the somehow difficult MOBIO dataset. This results in fewer errors for SudFrog as shown in table IV on the MOBIO dataset. Using the same amount of data, SudFrog outshines the DNN. However, once we use the huge MS-celeb-1M dataset, the positions are reversed. We can not train SudFrog with MS-celeb-1M dataset as it is technically infeasible. The feature space becomes too huge for the memory.

V. CONCLUSIONS AND PERSPECTIVES

This paper details how to obtain a state-of-the-art face recognition system based on publicly available software and using public datasets. We try to give the most possible details to allow for the reproducibility of the results. When CMU implemented OpenFace, reproduciblity was one of their main goals. Thus we were able to reproduce their results and improve upon them. However, we couldn't get the same results as Google who used huge proprietary datasets and a proprietary face alignment system. Our **OpenFace_best** DNN model gives good verification results on both evaluation datasets, MOBIO and LFW. From the results that we obtained we can infer that the performance bottleneck lays in the preprocessing, notably the face detection phase. Given enough data, the DNN is unmatched. Nevertheless, in situations where the data are not available classical approaches give better performance. In order to improve our results, we plan to study further the preprocessing and use neural networks for the face detection. We also plan to reduce the mislabelling of the MS-celeb-1M dataset in order to study the effect of the mislabelling on the generalization of the system.

ACKNOWLEDGMENT

This work is partially supported by the SpeechXRays project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 653586. We are also grateful for the constructive feedback from the reviewers.

REFERENCES

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[2] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[3] 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.

[4] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[6] D. Petrovska-Delacrétaz, G. Chollet, and B. Dorizzi, *Guide to biometric reference systems and performance evaluation*, 2009, ch. The BioSecure Benchmarking Methodology for Biometric Performance Evaluation, pp. 11–23.

[7] 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.

[8] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, 2016.

[9] 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.

[10] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernockỳ, N. Poh, J. Kittler, A. Larcher, C. Levy *et al.*, "Bi-modal person recognition on a mobile phone: using mobile phone data," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 635–640.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv preprint arXiv:1409.1556*, 2014.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[16] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[18] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1. IEEE, 2005, pp. 947–954.

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

[20] T. Bourlai, *Face Recognition Across the Imaging Spectrum*. Springer, 2016, ch. Face Recognition in Challenging Environments: An Experimental and Reproducible Research Survey, pp. 269–270.

[21] H. Yu and J. Yang, "A direct lda algorithm for high-dimensional datawith application to face recognition," *Pattern recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.