TAKING ROLL: A PIPELINE FOR FACE RECOGNITION

I. Gallo, S. Nawaz, A. Calefati

Dip. di Scienze Teoriche e Applicate
University of Insubria
21100, Varese, Italy

G. Piccoli

Louisiana State University
2222 Business Education Complex South,
LA, 70803, USA

ABSTRACT

We propose a generic pipeline for a face recognition system capable of creating or cleaning datasets when videos or images come from a finite set of identities. Face recognition has wide practical applicability for organizations and can be solved using an approach based on Convolutional Neural Networks, such as FaceNet. Differently from FaceNet, we proposed a solution based on a Convolutional Neural Network model with center loss, that speeds-up the labeling of faces in a video. With this pipeline, we show that cleaning a dataset of faces is a fully automatable process and improves the performance of the face recognition system. Together these two elements of the pipeline significantly improve face recognition results.

1. INTRODUCTION

Convolutional Neural Networks (CNNs) have significantly improved state-of-the-art results in many applications including face recognition [1, 2]. A face recognition system starts with the creation of a large scale dataset from videos or still images [2, 3]. This is an extremely important step because the performance of deep CNN depends on the availability of large scale datasets. Typically, large scale datasets are created from search engines and are prone to noise [4]. Although, deep CNN withstands certain amount of noise, significant noise presence can deteriorate face recognition performance. To address these challenges, we present a generic pipeline for face a recognition system based on learning embedding using a deep CNN, similar to [1]. The pipeline is capable of creating a dataset either from video or still images. In addition, it can remove noise from existing datasets [4]. Our proposed pipeline offers solutions to a class of problems occurring when organizations seek to measure the recurrent presence of a specific set of individuals (e.g., detecting students in attendance at a lecture, identifying members at a fitness club).

2. PROPOSED PIPELINE

We propose a strategy similar to the one described in [1] to recognize and label faces belonging to a set of identities. We split our strategy into two phases: a) collect a set of training data; b) use a classifier to recognize a finite set of identities. Both phases use a CNN model previously trained on large scale face datasets, as shown in Fig. 1. This CNN model is used as image embedding technique in both phases of the proposed pipeline. Interestingly, the CNN model is trained on a large scale dataset, typically still images, that does not contain the target identities of our system. Thus, we can use a model already trained in a different context - as a black box. We used a CNN model based on the *center loss* [5] function instead of the *Triplet Loss* [1]. Triplet loss suffers from dramatic data expansion when constituting sample triplets from the training set. Conversely, center loss has the same requirement as the softmax loss function and needs no complex recombination of training samples [5]. Consequently, the supervised learning of the CNN is easier to train.

2.1. CNN model

The CNN model used in our work is the Inception-ResNet-v1 [6] trained with center loss function. The training process starts with the selection of a large scale faces dataset, then aligning faces in images or frames and finally training the deep network, as described in [5].

Embeddings extracted from this trained CNN model are used for two purposes: (a) to create the face recognition dataset; (b) to transform faces into embeddings for recognition, as shown in Fig. 1. This CNN can also be used to clean an existing unlabeled faces dataset with the help of an expert or in case of an already labeled dataset, to automatically remove noise.

We trained the CNN model on the VGGFace2 [7] dataset. We downloaded loosely cropped faces dataset from the VGGFace2 website[1]. The dataset is aligned with 160×160 image size and 32 pixels margin based on Multi-task CNN [8]. We trained the model on aligned dataset for 100 epochs with an

[1] http://www.robots.ox.ac.uk/ vgg/data/vgg_face2

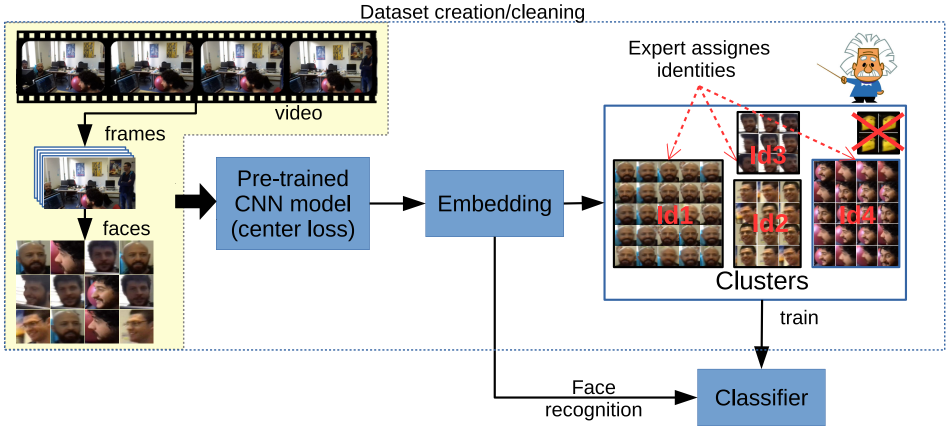

Fig. 1. Overview of the proposed pipeline. The model accepts a set of aligned faces and a pre-trained model is used to transform faces into embeddings, which then are fed to a clustering algorithm that groups all faces belonging to the same person. An expert selects best clusters and assigns them a label (identifying the person the cluster represents). Labeled clusters are used to train a classifier to recognize people in a particular context.

RMSProp optimizer. In addition, we used two off-the-shelf models[2] trained on CASIA-WebFace [9] and subset of MS-Celeb-1M [4] datasets.

2.2. Dataset creation and cleaning

The alignment process may add noise along with faces due to false face detections as shown in the left portion of Fig. 1. Feeding aligned faces to a clustering algorithm, we separate identities and noise using embeddings from the pre-trained CNN model.

In this work we used the popular density-based clustering algorithm called DBSCAN [10], which does not require a priori specification of the number of clusters in the data. Intuitively, the clustering algorithm is able to group together points (faces of an identity) that are closely packed together. Our next task is to label clusters into identities. To assign the correct identity to each cluster, an expert must annotate clusters. A similar pipeline is used in [11] for the creation of the MF2 large scale dataset with 672K identities and 4.7M images. In this scenario, the authors automatically assign an artificial identifier to each cluster because such dataset is used to obtain a pre-trained model and not for recognition.

Our proposed pipeline can also be used to clean a dataset. Existing labeled datasets with noise can use this pipeline to select the largest cluster as identity and interpreting other smaller clusters as noise, eliminating the need of an expert. This hypothesis holds if the number of noise images is less than the number of face images of an identity.

2.3. Face recognition

Given an image x we obtained an embedding $f(x) \in \mathbb{R}^d$ using the pre-trained CNN model. In all our experiments we used $d = 128$ as embedding dimension, similar to [1]. Once the embedding is produced, face recognition becomes a classification problem as described in [1]. In this work we use a standard SVM [12] for classification, to perform face recognition. The SVM receives an embedding $f(x)$ and classifies it to one of the known identities or as an unknown.

3. DATASETS

Datasets made up from videos, such as the YouTube Faces dataset [13], have low pose variability for each identity, this means that the majority of frames are similar in terms of pose and expression variation. We created a small scale dataset named 7Pixel-Faces with 25 identities in an office setting, using the approach explained in Sec. 2.2. We recorded $8 - 10$ seconds videos with: (a) single identity and pose variability; (b) multiple identities. The first strategy is considered ideal, however, it needs considerable effort to record videos of all identities. The second strategy is more realistic, but it presents a challenge with multiple identities in a single video. Typically, a recorded video consists of $300 - 400$ frames. We extracted each frame from a video and feed it to a face detection and alignment algorithm [8].

In addition, we used 4 publicly available datasets in our experiments: VGGFace2 [7], MS-Celeb-1M, YouTube Faces and UMDFaces [3].

[2] https://github.com/davidsandberg/facenet

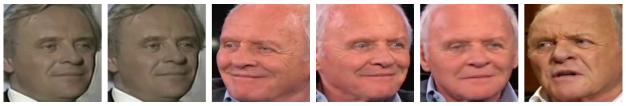

Fig. 2. Some aligned faces of a single cluster obtained merging 3 different videos from YouTube. In each video of Anthony Hopkins we have different environment settings and input sources.

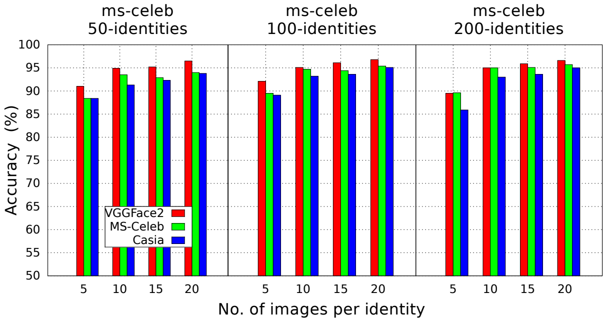

Fig. 3. Face recognition accuracy obtained from a trained SVM on the MS-Celeb-1M dataset varying the number of identities and the number of images per identity. The SVM receives embeddings obtained from a CNN trained on a center loss function with VGGFace2, MS-Celeb and Casia datasets.

4. EXPERIMENTS

In this section we present two groups of experiments to evaluate elements of our proposed pipeline: a) dataset creation-cleaning b) the face recognition processes. In Sec. 4.1 we present experiments of the semi-supervised approach that speeds-up the creation of a dataset. Moreover, we evaluate a generic automated cleaning process for existing datasets. Finally, in Sec. 4.2 we evaluated the face recognition phase.

4.1. Evaluation of dataset creation and cleaning processes

To apply the face recognition pipeline in a real scenario, it is necessary to construct a dataset containing multiple images for each identity to be recognized. During the dataset creation process, we can experience the following challenges that can deteriorate face recognition performance: different input source, different environment settings, various identities in a video. We conducted the following experiments to analyze these challenges.

In the first experiment we answer the question *Q1*: *are we able to obtain automatically a single cluster of an identity merging various videos with different environment settings and sources?* To answer this question we selected 5 celebrities and downloaded 3 videos for each one from YouTube. We obtained an average accuracy of 91.59% selecting only the biggest cluster from each video representing one identity.

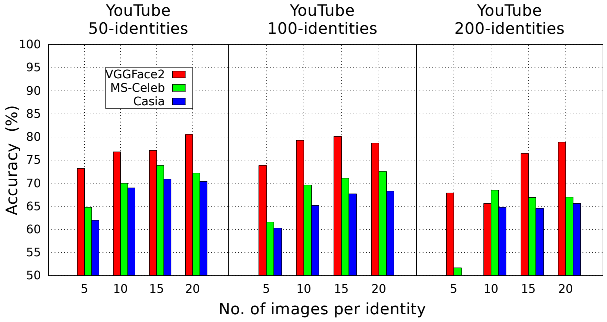

Fig. 4. Face recognition accuracy obtained from a trained SVM on the YouTube Faces dataset varying the number of identities and the number of images per identity. The SVM receives embeddings obtained from a CNN trained on a center loss function with VGGFace2, MS-Celeb and Casia datasets.

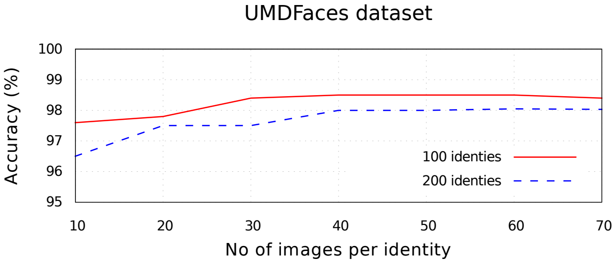

Fig. 5. Face recognition accuracy obtained from a trained SVM on the UMDFaces dataset varying the number of identities and the number of images per identity. The CNN model is trained with center loss function with the VGGFace2 dataset.

The total number of face images belonging to the selected identities is 2104 while 1921 is the number of images in selected clusters. This experiment shows that we are able to merge videos or images from different sources and environment settings because we have no wrong identities in selected clusters. Fig. 2 shows 6 images taken from 3 different videos and merged into the selected cluster.

Table 1. Accuracy (Acc), precision (P) and recall (R) of the cleaning process when applied 50 randomly selected identities of the MS-Celeb dataset. Positive images belong to a selected identity, while negative are all the other images.

	Positive	Negative	Acc=97.35%
Biggest cluster	2617	86	P=99.32%
Other clusters	18	1206	R=96.82%

In the second experiment we answer the following question *Q2*: *are we able to extract clusters of different identities present in a video?* This scenario is also illustrated in Fig. 1. In this experiment we selected 5 different videos having 4, 4, 6, 3, 4 identities respectively. We expected to obtain a single cluster for each identity in each video. For 3 videos, selecting just the biggest cluster of each identity, we get no wrong images in each selected cluster. However, for other 2 videos, we are not able to separate 2 similar identities. These results indicate that we are able to separate multiple identities in a video.

In the final experiment we answer this question *Q3*: *are we able to merge frames extracted from video(s) and still images to obtain a cluster for a single identity?* In this experiment we selected 5 celebrities from the MS-Celeb-1M dataset and downloaded 5 videos from YouTube. We expected to obtain a single biggest cluster for each identity containing faces from still images and video frames. We got a total of 2474 correct faces selecting the biggest cluster from each of the 5 combinations over a total of 2686 aligned faces. The overall accuracy computed on merged images is 99.33% with a Recall of 100% and a Precision of 99.28%. This result shows that we can merge faces extracted from different sources.

In addition to the dataset creation process, the proposed pipeline can be used to remove noise from an available dataset as described in Sec. 2.2. To illustrate how this noise removal process works, we randomly selected a sub-set ($10,000$ identities) of MS-Celeb-1M dataset to remove noise from each identity (celebrity). We automatically selected the biggest cluster obtained using DBSCAN, thus removing noise from each identity. State-of-the-art deep neural network learning algorithms can tolerate a certain level of noise in the training data but in case of dataset like MS-Celeb-1M the amount can be considerable and we obtained better result after noise removal, as shown in Fig. 6. In addition to this experiment, we manually labeled images of 50 randomly selected identities to present numerical result of the cleaning process. Table 1 shows performance values, in terms of accuracy, precision and recall. 32.90% of the images have noisy labels and our cleaning process is able to eliminate them almost completely, leaving only 2.19% of the noise into selected clusters.

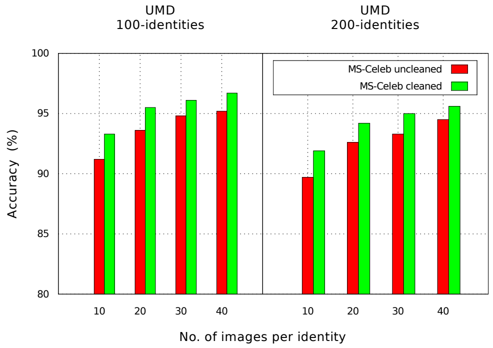

Fig. 6. Comparison results obtained using the *Inception resnet v1* with center loss, as pre-trained model, when trained on 10,000 identities of the original (uncleaned) MS-Celeb dataset and from the cleaned version (using our pipeline) of the same dataset. On the left, the accuracy results of an SVM trained on 100 randomly selected identities from UMDFaces dataset. On the right, we trained the SVM adding 100 new identities to the previously 100 selected from the UMDFaces dataset.

4.2. Evaluation of the face recognition phase

We are interested to find out the capability of the face recognition phase on both still images and video frames. We conducted two experiments comparing different pre-trained CNN models on different sets of identities. In these experiments, we used the Youtube Faces and MS-Celeb-1M datasets randomly selecting 50, 100 and 200 identities. Gradually increasing the number of images per identity, we want to find out which one is the best model to use for extracting embedding from each face image. The extracted embeddings were fed to an SVM, trained to classify all identities. Analyzing plots in Figs. 3 and 4, we can see that the best pre-trained model is the VGGFace2, so we decided to use it. We can also conclude that a CNN trained and tested on still images perform better than a CNN trained on still images and tested with videos frames. In a deeper analysis on the YouTube Faces dataset we found out that the dataset has low pose variability, which contributes to low accuracy values as shown in Fig. 4. These results indicate that we can still increase the recognition performance by increasing the number of images per identity. We conducted another experiment by increasing number of images per identity to find out the asymptote as reported in Fig. 5. We selected the UMDFaces dataset because it is considered a deeper dataset (higher number of images per identity). This result indicates that the highest accuracy is achieved with 40 images per identity.

Finally, we conducted an experiment on the 7Pixel-Faces dataset obtaining an accuracy of 93.6%. Comparing with the result obtained from the YouTube Faces dataset we conclude that pose variability plays a significant role to increase the performance in face recognition.

5. CONCLUSION

In this paper we proposed a generic pipeline for face recognition systems capable of creating, cleaning and recognizing faces. We proposed a semi-supervised solution based on a CNN model with center loss, that speeds-up the faces labeling process in a video composed of a finite set of identities. With this pipeline, we showed that cleaning a dataset is a fully automatable process and improves the performance of the system. Attention must be paid to the characteristics of the videos used for training the recognition model: videos with low pose variability can lead to poor recognition performance. In the future we are interested in creating a large scale faces dataset from videos, exploiting the proposed dataset creation pipeline.

6. REFERENCES

[1] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering.," in *CVPR*. 2015, pp. 815–823, IEEE Computer Society.

[2] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al., "Deep face recognition.," in *BMVC*, 2015, vol. 1, p. 6.

[3] Ankan Bansal, Anirudh Nanduri, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa, "Umdfaces: An annotated face dataset for training deep networks," *arXiv preprint arXiv:1611.01484*, 2016.

[4] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds. 2016, vol. 9907 of *Lecture Notes in Computer Science*, pp. 87–102, Springer.

[5] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[6] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017, pp. 4278–4284.

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *arXiv preprint arXiv:1710.08092*, 2017.

[8] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[9] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, KDD'96, pp. 226–231.

[11] Aaron Nech and Ira Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," *arXiv preprint arXiv:1705.00393*, 2017.

[12] Vladimir N. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, 1998.

[13] Lior Wolf, Tal Hassner, and Itay Maoz, "Face recognition in unconstrained videos with matched background similarity.," in *CVPR*. 2011, pp. 529–534, IEEE Computer Society.