IVCNZ Publication License for PID # 5652977

Paper Title: A Pipeline to Improve
Face Recognition Datasets and Applications
Author: I. Gallo, S. Nawaz, A. Calefati, G. Piccoli

I. L ICENSE

A Pipeline to Improve Face Recognition Datasets and Applications

I. Gallo, S. Nawaz, A. Calefati
University of Insubria, Varese, Italy
{ignazio.gallo,snawaz,a.calefati}@uninsubria.it

G. Piccoli
Louisiana State University, LA, USA
gpiccoli@cct.lsu.edu

***Abstract*—Face recognition has a wide practical applicability in various contexts, for example, detecting students attending a lecture at university, identifying members in a gym or monitoring people in an airport. Recent methods based on Convolutional Neural Network (CNN), such as FaceNet, achieved state-of-the-art performance in face recognition. Inspired from this work, we propose a pipeline to improve face recognition systems based on Center loss. The main advantage is that our approach does not suffer from data expansion as in Triplet loss. Our pipeline is capable of cleaning an existing face dataset to improve the recognition performance or creating one from scratch. We present detailed experiments to show characteristics and performance of the pipeline. In addition, a small-scale application for face recognition that makes use of the proposed cleaning process is presented.**

***Index Terms*—face recognition, convolutional neural network, center loss, cleaning dataset**

I. Introduction

Convolutional Neural Networks (CNNs) obtained state-of-the-art results in many applications including face recognition [1]–[3]. A face recognition system based on CNN models, usually, begins with the creation of a large scale dataset from videos or still images [2], [4]. This is an extremely important step because the performance of the whole system depends on the availability of large quantity of data and on its quality [1]. Typically, large scale datasets created in a semi-supervised way from search engines are prone to noise [5]. Even though deep CNN withstand certain amount of noise, a significant presence can deteriorate performance of recognition systems. To tackle these challenges, we present a generic pipeline for face recognition systems based on learning embeddings using a deep CNN, similar to Facenet [1] where the use of the Triplet loss enhanced the discriminative power of the deeply learned face features. Our pipeline is capable of creating a dataset either from video or still images from scratch. In addition, it can be used to remove noise from existing datasets such as MS-Celeb-1M [5]. Our proposed pipeline is general and can be employed to various problems occurring, for example, when organizations want 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 or monitoring people in a airport.

978-1-7281-0125-5/18/$31.00 ©2018 IEEE

II. Related work

Face dataset cleaning is a very common problem in computer vision [6], [7], especially with the advent of deep models that need a huge quantity of data to be trained. The aim of our paper is similar to [6] where authors created a pipeline to clean datasets. This approach applies traditional methods, such as Local Binary Patterns (LBP) and Three-patch LBP, to extract image descriptors from 48-pixels horizontal striped patches cropped from the center of the image containing a face. This leads to embeddings that are computed only on a specific portion of images rather than whole images, thus being inaccurate. In [1] authors propose Triplet loss function to obtain similar embeddings for images of a class, while different ones for images of other classes. However, this approach suffers from dramatic data expansion when constituting sample triplets from the training set. It turns out that to use Triplet loss, the availability of high-end hardware is required. Conversely, Center loss [8] does not need complex recombination of training samples. In our work we employed Multi-task Cascaded Convolutional Network (MTCNN) [9] and Center loss [8] to obtain embeddings from face images extracted from videos or still images. Moreover, this is the first work which leverages on a clustering algorithm to remove noise from large scale dataset.

III. Proposed Pipeline

The proposed pipeline for face recognition has two uses: (a) to create a brand new dataset from scratch and (b) to clean an existing dataset. In the former case we can create a dataset using either still images or videos of an identity, while in the latter we start our pipeline from existing datasets.

To create datasets from videos, first of all, we extract frames and then apply MTCNN to obtain aligned faces with image size of 160×160 and margin of 32 pixels. Next, using a CNN model previously trained on a large scale face dataset, we extract embeddings, represented by 128-dimensional vectors, which then are fed to a density-based clustering algorithm called DBSCAN [10]. The advantage of DBSCAN is that the number of output clusters is automatically computed from the algorithm itself and thus must not be selected a priori. The CNN model is used as image embedding technique in both threads of the proposed pipeline. We use a model based on the Center loss [8] instead of the Triplet loss [1]. The model is trained on a large scale dataset that does not contain the target

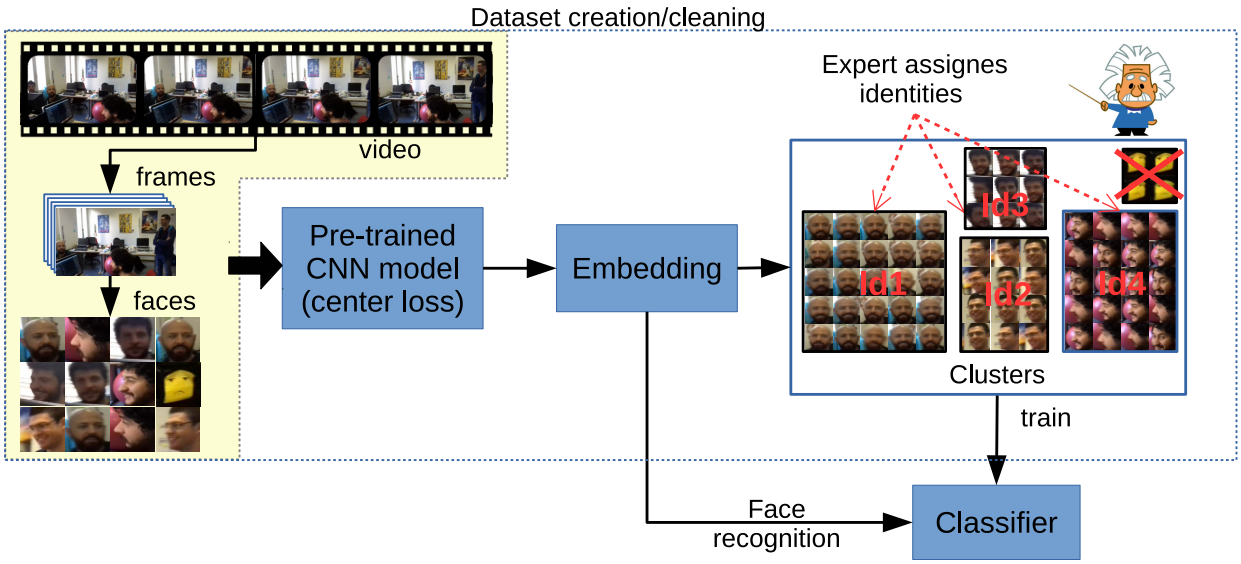

Fig. 1. Overview of the proposed pipeline. The pipeline accepts a set of aligned faces and using a pre-trained model it transforms faces into embeddings, which then are fed to a clustering algorithm to group all faces belonging to the same person. An expert selects best clusters and assigns them a label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

identities of our system. After the feature extraction phase, an expert must select relevant clusters containing images of the considered identity and removes the ones which contain false positive images obtained from the detection phase. Finally, with the resulting set of identities, we train our face recognition classifier. The dataset creation from still images is very similar to the pipeline described for videos, except for the frame extraction phase which is skipped.

Our pipeline can be also used to clean existing datasets, removing noise and thus improving classifiers performance. In this scenario, we extract embeddings from the pre-trained model and use them as input for the clustering algorithm. At this stage, the intervention of experts can be avoided selecting automatically the biggest cluster obtained and assigning it the label of the considered class.

The entire pipeline is summarized in Fig.1.

A. CNN model

The CNN model used in our work is the Inception-ResNet-v1 [11] trained with Center loss function. The training process begins with the selection of a large scale faces dataset, followed by face alignment and ends with the training of a deep network, as described in [8].

Embeddings extracted from the pre-trained CNN model are used for three purposes: (a) to create the face recognition dataset, (b) to clean existing datasets and (c) to use a face recognition system.

We downloaded loosely cropped faces dataset from the VGGFace2 [12] website[1] and then trained the CNN model. This dataset is aligned with 160×160 image size and 32 pixels margin based on Multi-task CNN [9]. We trained the model on aligned dataset for 100 epochs with an RMSProp optimizer. In addition, we used two off-the-shelf models[2] trained on CASIA-WebFace [13] and subset of MS-Celeb-1M [5] datasets. Further details about experiments on these datasets are provided in Sec. V-B.

B. 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 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 them. A similar pipeline is used in [14] for the creation of the MF2 large scale dataset with 672K identities and 4.7M images. In this scenario, 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 already existing dataset. They are often created using semi-supervised approaches which lead to the introduction of noise which lowers the performance of classifiers. Our pipeline can automatically clean a dataset, selecting the largest cluster as identity and interpreting other smaller clusters as noise, eliminating the intervention of experts.

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

[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 environmental settings, qualities and lightings.

IV. Datasets

We used four publicly available datasets in experiments: VGGFace2 [12], MS-Celeb-1M [5], YouTube Faces [15] and UMDFaces [4]. VGGFace2, MS-Celeb-1M and UMDFaces dataset contain still images while YouTube Face dataset contains images from videos. Datasets made from videos have low pose variability for each identity, this means that the majority of frames are similar in pose and expression. In contrast, datasets built from still images contain high pose variability. We selected these datasets to evaluate accuracy trends varying the number of images per identity.

In addition, we developed a small application which is used to perform taking roll in a controlled environment. For this purpose, we created a small scale dataset named 7Pixel-Face containing 25 identities in an office setting, using the approach explained in Sec. III-B, thus the entire pipeline is employed.

We recorded $8-10$ seconds videos featured with: (a) single identity and pose variability; (b) multiple identities. The first strategy is considered ideal, however, it takes considerable effort to record videos of all identities individually. 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 fed it to a face detection and alignment algorithm [9]. Finally we applied a clustering mechanism to create the face dataset.

Some examples of images/frames representative of the various datasets used in this paper are shown in Figure 3.

A. Face recognition application

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. Once the embedding is produced, face recognition becomes a common classification problem as described in [1]. In this work we used a standard SVM [16] for classification, to perform face recognition, but many other models can be employed. The SVM receives an embedding $f(x)$ and classifies it to one of the known identities or as unknown.

V. Experiments

In this section we present two groups of experiments to evaluate elements of our proposed pipeline: (a) dataset creation-cleaning step and (b) the face recognition process.

In Sec. V-A 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 already available datasets. Finally, in Sec. V-B we evaluate the face recognition phase.

Fig. 3. In each row some examples of representative images/frames of datasets used in this paper: VGGFace2 (a), UMDFaces (b), MS-Celeb-1M (c), YouTube Faces (d) and 7Pixel-Face (e).

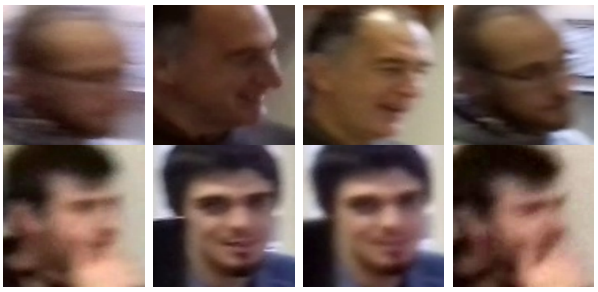

Fig. 4. In each line some problematic examples of faces extracted from videos that contain more than one identity of the 7Pixel-Face dataset. Blur and motion problems, in some cases, lead to clusters of faces with errors: the top row and the bottom row represent some errors found in two different clusters.

A. Evaluation of dataset creation and cleaning processes

To apply the face recognition pipeline in a real scenario, it is necessary to build a dataset containing multiple images for each identity to be recognized. During the dataset creation process, we experience the following challenges that can deteriorate face recognition results: differences in input sources, different environmental settings and various identities in a video. We conducted many experiments to analyze how these features affect performance.

In the first experiment we want to check if we can obtain automatically a single cluster of an identity merging various videos taken from heterogeneous sources with different environment settings. This experiment is useful to understand

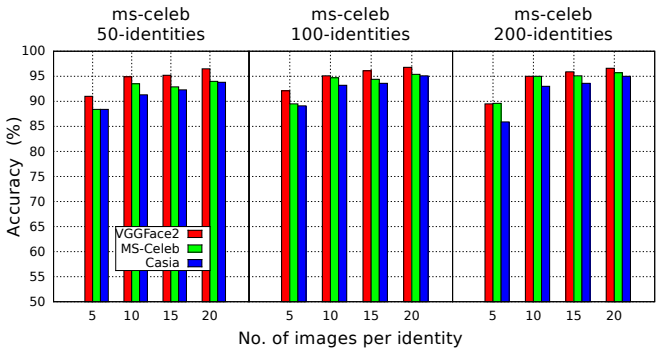

Fig. 5. Face recognition accuracy obtained from a trained SVM on the MS-Celeb-1M dataset varying the number of identities and the number of images per identity. The SVM receives embeddings obtained from a CNN trained on a Center loss function with VGGFace2, MS-Celeb and Casia datasets.

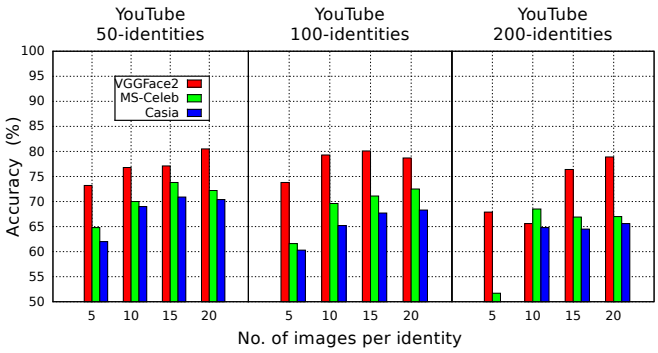

Fig. 6. Face recognition accuracy obtained from a trained SVM on the YouTube Faces dataset varying the number of identities and the number of images per identity. The SVM receives embeddings obtained from a CNN trained on a Center loss function with VGGFace2, MS-Celeb and Casia datasets.

if we are able to automatically obtain a single cluster of an identity from videos with different settings and environments. In fact, creating a new dataset, we want to extract frames from all available videos and not only from a single video. Our expectation is to get only one main cluster from all the videos used. For this purpose, as a proof of concept, we selected 5 celebrities and downloaded 3 short videos for each one from YouTube. The total number of face images belonging to the target identities is 2104 while 1921 is the number of images contained in the selected clusters. We got an average accuracy of 91.59% from each video representing a single identity and selecting only the biggest cluster. On average, this means that we discarded 8.41% of faces for each identity but the selected clusters contain no errors or noise. This experiment shows that we are able to merge videos or images coming from various sources with different quality and environmental settings with zero false positives in the selected clusters. Fig. 2 shows 6 images taken from 3 different videos and merged into the selected cluster.

The second experiment tests whether our approach can extract clusters of different identities from 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 from the 7Pixel-Face dataset. We expected to obtain a single cluster for each identity in each video. For 3 videos, selecting the biggest cluster of each identity, we got no wrong images in each selected cluster. However, for the remaining 2 videos, we were not able to separate 2 similar identities. These results indicate that we are able to separate multiple identities contained in a video except for cases where identities are considered similar by the clustering algorithm due to motion/blur problems. Figure 4 shows some examples of the two clusters containing two identities instead of the single one we expected.

In the last experiment we wanted to test if we could merge frames coming from videos and still images to obtain a cluster for a single identity. To tackle this problem, we selected 5 celebrities from the MS-Celeb-1M dataset and downloaded 5 videos from YouTube. We obtained a total of 2474 correct faces selecting only 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 clearly shows that we can merge faces extracted from different sources.

TABLE I
ACCURACY (ACC), PRECISION (P) AND RECALL (R) OF THE CLEANING PROCESS APPLIED TO 50 RANDOMLY SELECTED IDENTITIES OF THE MS-CELEB DATASET. POSITIVE IMAGES BELONG TO A SELECTED IDENTITY, WHILE NEGATIVE ARE ALL REMAINING IMAGES.

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

In addition to the dataset creation process, the proposed pipeline can be used to remove noise from a publicly available dataset, as described in Sec. III-B. To illustrate how the noise removal process works, we randomly selected a sub-set of $10,000$ identities from the MS-Celeb-1M dataset to remove noise from each identity. We automatically selected the biggest cluster obtained using DBSCAN to remove noise from each identity. The MS-Celeb-1M dataset is strongly affected by noise. Even though state-of-the-art deep neural network learning algorithms are noise tolerant, to obtain better results it is recommended to remove noise. Thus we applied the cleaning process. This experiment has been conducted using Inception ResNet-v1, trained using Center loss, as a pre-trained model on a subset of $10,000$ identities from MS-Celeb dataset. To prove the effectiveness of our cleaning process, we trained this feature extraction model with 2 versions of MS-Celeb dataset: the first with the original (containing noise) training set, while the second with the cleaned one. We computed the accuracy as the ratio between correct predicted identities over total using an SVM trained on 100 and 200 randomly selected identities from UMDFaces dataset. Results are shown in Fig. 8.

In addition to this experiment, we manually labeled images of 50 randomly selected identities from MS-Celeb dataset to present qualitatively results of the cleaning process. We

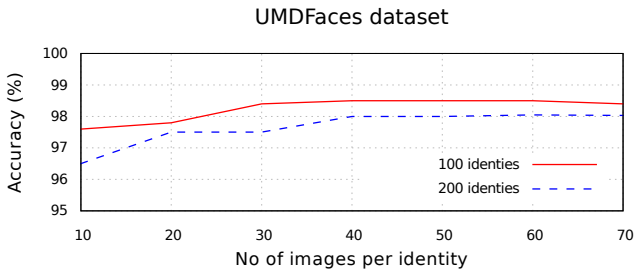

Fig. 7. Face recognition accuracy obtained from a trained SVM on the UMDFaces dataset varying the number of identities and the number of images per identity. The CNN model is trained with Center loss function with the VGGFace2 dataset.

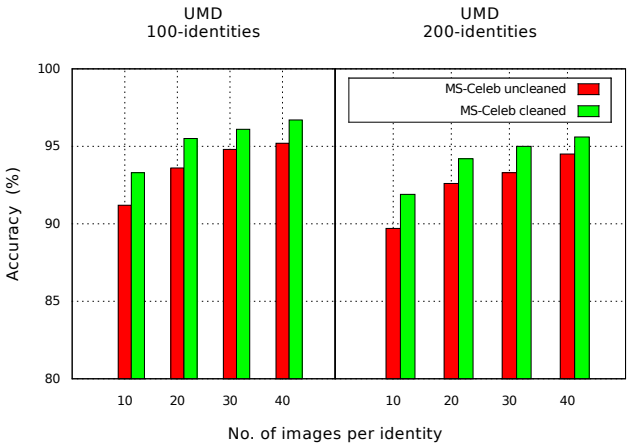

Fig. 8. Comparison of results obtained using the *Inception ResNet-v1* trained on both cleaned and uncleaned versions of MS-Celeb dataset. Independently from the number of identities used to train the face recognition classifier, the feature extraction model trained on a cleaned version of the dataset creates more discriminative embeddings, leading to better classification accuracy.

performed embedding extraction for each image and then applied the DBSCAN clustering algorithm. We calculated accuracy, precision and recall considering the cleaning process as a binary classification problem with positive as the class of images of the selected identity and negative for the remaining classes.

Table I shows performance values, where each number is the summation of all images of a specific identity assigned to a cluster. Our pipeline has a strong impact against the noise in datasets. In fact, a great amount of images (32.90%) have noisy labels and our cleaning process is able to eliminate them almost completely, leaving only a small percentage (2.19%) of noise into the clusters.

B. Evaluation of the face recognition phase

We are interested in measuring performance 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 trained on 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 images. The extracted embeddings were fed to an SVM, trained to classify all identities. Analyzing plots in Figs. 5 and 6, we see that the best model is the one trained on VGGFace2 dataset. We can also conclude that a CNN trained and tested on still images performs better than a CNN trained on still images and tested with video frames, as demonstrated in [17]. This happens because, usually video frames are slightly blurred compared to still images. 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. 6. These results indicate that we can still increase the recognition performance by increasing the number of images per identity. We conducted another experiment to find out the number of images for each identity needed to obtain the best results. We tested on UMDFaces dataset because it is considered a deeper dataset, it contains a higher number of images per identity. The result shown in Fig. 7 indicates that the highest accuracy is achieved with 40 images per identity.

Finally, we evaluated the overall accuracy of the small scale application developed. We performed an experiment on the 7Pixel-Face dataset, achieving an accuracy of 93.6% in face recognition task. Comparing with the result obtained from the YouTube Faces dataset, we can conclude that pose variability plays a significant role in face recognition.

VI. Conclusion

In this paper we proposed a generic pipeline for face recognition systems capable of creating and cleaning datasets. Moreover, the same pipeline can be used to recognize faces coming from video or image sources. We proposed a semi-supervised solution based on a CNN model with Center loss, that speeds-up the faces labeling process from a video containing a set of identities. With this approach, we showed that cleaning a dataset is a fully automatable process and improves the performance of a face recognition system. Features of videos used for training the recognition model are crucial: videos with low pose variability can lead to poor recognition performance. In the future we will create large scale faces datasets from videos, exploiting the proposed dataset creation pipeline.

References

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

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

[3] A. Calefati, M. K. Janjua, S. Nawaz, and I. Gallo, "Git loss for deep face recognition," *BMVC*, 2018.

[4] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa, "Umdfaces: An annotated face dataset for training deep networks," *arXiv preprint arXiv:1611.01484*, 2016.

[5] Y. Guo, L. Zhang, Y. Hu, X. He, and J. 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*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9907. Springer, 2016, pp. 87–102.

[6] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 343–347.

[7] C. Jin, R. Jin, K. Chen, and Y. Dou, "A community detection approach to cleaning extremely large face database," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[8] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. 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*, ser. KDD'96, 1996, pp. 226–231.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

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

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

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

[15] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity." in *CVPR*. IEEE Computer Society, 2011, pp. 529–534.

[16] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[17] A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa, "The do's and don'ts for cnn-based face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2545–2554.