Transfer Learning for Facial Attributes Prediction and Clustering

Chapter · November 2019

DOI: 10.1007/978-981-15-1301-5_9

Paola Barra
Università degli Studi di Salerno
8 PUBLICATIONS **8** CITATIONS

Silvio Barra
Università degli studi di Cagliari
41 PUBLICATIONS **272** CITATIONS

Fabio Narducci
Parthenope University of Naples
38 PUBLICATIONS **262** CITATIONS

Transfer Learning for Facial Attributes Prediction and Clustering

Luca Anzalone[2], Paola Barra[2][0000−0002−7692−0626], Silvio Barra[1][0000−0003−4042−3000], Fabio Narducci[3][0000−0003−4879−7138], and Michele Nappi[2][0000−0002−2517−2867]

[1] University of Cagliari, Department of Mathematics and Computer Science, Cagliari, ITALY silvio.barra@unica.it

[2] University of Salerno, Department of Computer Science, Salerno, ITALY {pbarra, mnappi}@unisa.it, l.anzalone2@studenti.unisa.it

[3] University of Naples "Parthenope", Department of Science and Technology, Naples, ITALY fabio.narducci@uniparthenope.it

Abstract. Notwithstanding the enhancement obtained in the last decade researches, the recognition of facial attributes is still today a trend. Besides the mere face recognition, the singular face features, like mouth, nose and hair, are considered as soft biometrics; these can be useful for human identification in cases the face is partially occluded, and only some regions are visible. In this paper we propose a model generated by transfer learning approach for the recognition of the face attributes. Also, an unsupervised clustering model is described, which is in charge of dividing and grouping faces based on their characteristics. Furthermore, we show how clusters can be evaluated by a compact summary of them, and how Deep Learning models should be properly trained for attribute prediction tasks.

Keywords: attribute clustering · k-means · face attributes · transfer learning · cluster summary

1 Introduction

Recognizing and grouping facial attributes is a very important task since it may result very useful in different applications, like verification and identification. Notwithstanding, this task becomes very challenging in the cases in which the subject is not collaborative, or not aware that he is being acquired. This usually happens in environments like smart cities, sensitive places, like banks and airports [5], but also for smart devices applications [6, 7] and learning platforms [8], which aims at granting the identity of the user [4]. In fact, whereas in collaborative scenarios many face variations, like illumination and pose, can be eliminated or greatly reduced, in an unconstrained ones, even the detection of the face becomes a very challenging activity.

Common face variations encountered in face-related tasks are the so-called PIE-issues, following explained:

- **Occlusions**: objects like hats, eyeglasses and scarfs tent to cover relevant regions of the face, eventually hiding the underneath features;
- **Pose**: the pose is probably one of the most challenging issue to address, since the face results deformed and the some of the features may be hidden;
- **Illumination**: high or low degrees of brightness may create noise. on the face;
- **Expressions**: occur when a change in expression alters the face features, thus reducing the change of a correct analysis.

In most cases, a face captured in unconstrained environments need to be normalized before being further analyzed.
Some approaches for pose estimation are based on a mixture of trees model [27], while other are able to detect even occlusions by performing landmarks estimation [3, 1]; at the state of the art, the most effective and robust approaches rely on training based methods like convolutional neural networks (CNNs) [14, 16, 26].
The approach here proposed relies on CNNs. In this way the model can learn robust features (compared to hand-made features like HOG, Haar, LBP), allowing it to infer facial attributes.

When dealing with smartcities, it may result very useful to apply face recognition based on a collection of soft biometrics, like the single facial features are. This is particularly true in the cases in which the normal pose of a subject may not permit an exhaustive capturing of the whole face. In this paper, a clustering and a recognition approach is faced, basing on the estimation of the facial features, in order to be used in a following operation of recognition. In particular, the main contributions of this paper are the following:

1. A model heavily based on the Transfer Learning approach, that estimates facial attributes; along with some training insights.
2. According to the detected attributes, we evaluate different clustering methods in order to discover and visualize the best grouping technique.

2 Related works

Although the recognition of face characteristics and the related grouping using Clustering techniques have been extensively analyzed in the literature, the clustering of facial images is a less discussed topic. In this section, the related works in the topics of the face attricute prediction and the clustering methods are shown.

2.1 Attribute prediction

In [14], the authors proposed an approach for face attribute prediction, by training. a model composed by two Convolutional Neural Networks: LNet and ANet; each of them pre-trained in a different way: the former on massive categories of general objects (for face localization), and the latter on large number of facial identities for attribute prediction. Then, either are trained and fine-tuned jointly with attributes tags.

Pre-training ANet on face identity allows the model to manage complex face variations thanks to the learnt face features.

Another possible way to deal with this problem is to learn a discriminative face representation. Researches in [20, 13], proposed a mode synthesising a face into a compact representation, called face embedding, that, being properly trained, is able to take apart faces belonging to the same subject (identity).

During the training process, the model implicitly learns enough features to distinguish face identities (the embeddings related to the same identity will lie on the same hyperplane). However the resulting face embedding is hard to interpret because it hides the learnt facial features, consequently losing any relations among attributes.

Thus, the embedding-based approaches are extremely good at maximizing the performance of a single feature; in this case the feature is the face identity. Indeed these are one of the most effective way of performing accurate face recognition tasks (which are based on a single high-level concept: identity).

2.2 Clustering methods

For the proposed approach, three clustering techniques have been compared: K-means, Agglomerative Clustering, and DBSCAN.

The K-Means algorithm [2] clusters data by trying to separate samples in n groups of equal variance, minimising the inertia criterion (the average squared distance between points in the same cluster). This algorithm requires the number of clusters to be specified as an input parameter. It scales well to large number of samples and has been used across a large range of applicative domains.

The Agglomerative Clustering approach falls within the Hierarchical clustering family. Hierarchical methods works by building nested clusters, obtained through merging or splitting underneath clusters. This nesting process stops when a single cluster is constructed; so the hierarchy of clusters is usually represented as a tree or dendrogram.

Differently from the other two, the DBSCAN algorithm is able to discover clusters of arbitrary shape; so, it doesn't require the number of clusters as input parameter. Moreover, DBSCAN is designed to require a minimal knowledge of the domain data, together with a good efficiency on large databases.

Otto et al. [15] faced the problem of clustering millions of faces into thousands clusters of related identities. They proposed an approximate Rank-Order clustering algorithm that performs better than popular clustering algorithms (k-Means and Spectral) by achieving a better accuracy and run-time complexity.

Finally, among the various works analyzed, another interesting evaluation was carried out on the work done by Rosebrock [17]. In this case an algorithm has been developed which extracts an array of 128 real numbers from each face and creates clusters with these data.

Each cluster contains a set of faces with similar facial features. Obviously, using

a clustering approach based on DBSCAN, the number of the created clusters depends on the algorithm, which is structured to return an optimal number of clusters.
Even in this case it doesnt fit our needs perfectly, as another goal that has been set is the possibility of independently choosing the number of clusters to be displayed.

2.3 Transfer learning

As the research goes further, the complexity of the ML tasks increases. The resulting model architectures becomes even bigger too, and slower to train. This aspect is particularly verified for Convolutional Neural Networks [11], which require a huge amount of data and computational power.
Thanks to the ImageNet classification challenge [18], many effective models have been trained (like the groundbreaking AlexNet [11]). It turned out that models like VGG [21], InceptionNet [22], and ResNet [9] are very good for solving general ML tasks.
Thus, the Transfer Learning [12] techniques allows to recycle a model architecture in order to perform a different task from the one it was developed for.
In this work, this technique is exploited to fine-tune our model.

3 Our approach

In this section, more details are given about the proposed detection-clustering pipeline.

Basically, the entire workflow consists of three parts:

- **Attributes inference**: in which we fine-tune a pre-trained model on 37 facial attributes taken from the CelebA [14] dataset.
- **Attributes clustering**: the labels predicted by the proposed model are given as input to a commonly available clustering algorithm (such as K-Means) which computes the grouping of the input faces according to a criterion of closeness.
- **Visualization and analysis**: by simply computing the occurrences of the attributes (of the given clusters), it is shown that these are enough to evaluate quantitatively the accuracy of the resulting clustering. Moreover, we provide an even more concise way to graphically evaluate clusters.

3.1 The CelebA dataset

All experiments and studies have been conducted on the CelebA dataset [24]. Unlike other datasets of faces (like LFW [10]), CelebA is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations.

This dataset is very challenging and, at the same time, appealing since it contains face-images covering a large amount of pose, expression, age, and occlusion variations.
We found out that CelebA has heavily imbalanced attributes (figure 1): more than a third of attributes are extremely rare (with frequencies below 10%), and only a couple of them are very common (by occurring more than 70% of the times).
This unbalance has pointed out the weaknesses of widely adopted loss functions, when training a model on rare-occurring instances.

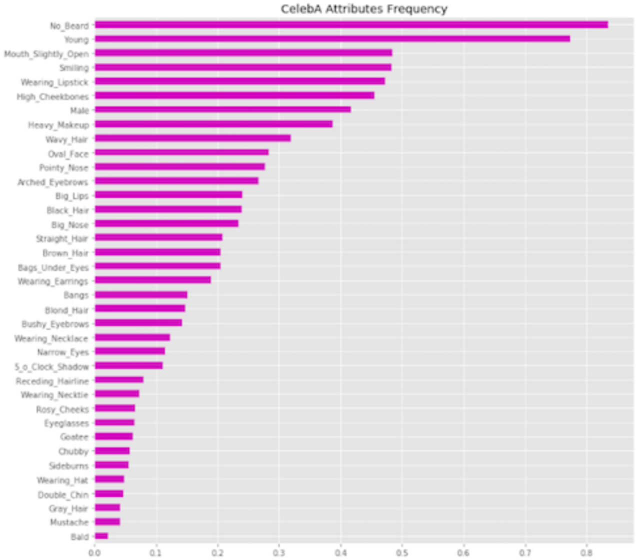
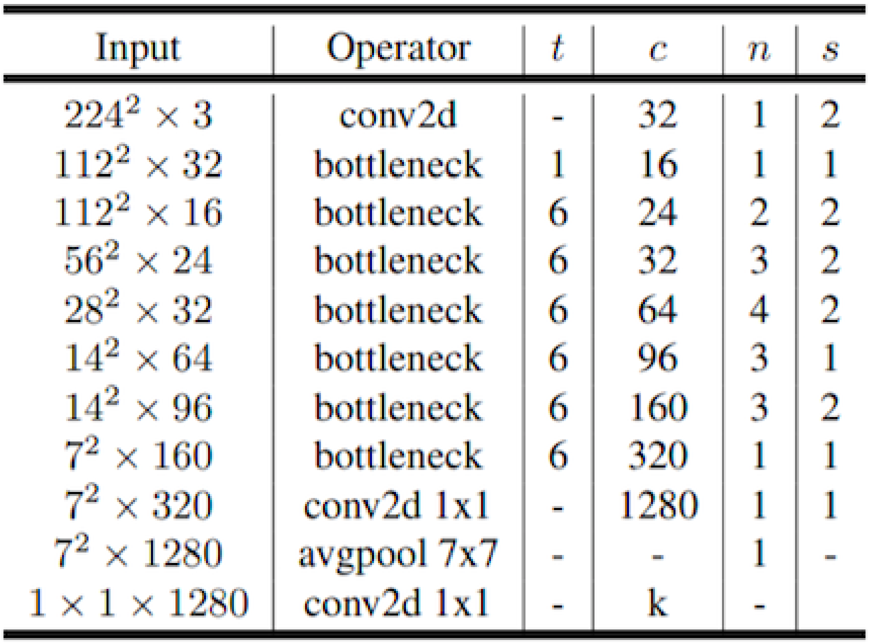


Fig. 1. Frequencies of CelebA attributes: the chart clearly shows the discrepacy about the occurrence of the attributes.

3.2 The fine-tuning of the model

We employ the Transfer Learning approach to fine-tune our model architecture, in order to achieve a faster training convergence.
Our baseline model is represented by the MobileNetV2 [19] architecture, but without the top classification layers (figure 2).

Our top layers (figure 3), simply consists of a Fully Connected layer, followed by a Batch Normalization operation before the final multi-label Dense layer that outputs the facial attributes of the input sample.
So, the model outputs a binary 37-d vector (we decided to drop three attributes: attractiveness, pale skin and blurry).



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Fig. 2. MobileNetV2: Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times.

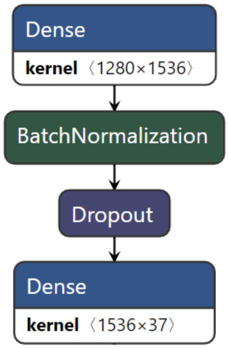


Fig. 3. Top Layers: A first Fully-Connected (Dense) Layer with 1536 neurons, normalized by Batch-Normalization and regularized by applying a dropout on 30% of the connections. The last Dense layer outputs the labels for every attribute (thus, requires 37 neurons *sigmoid*-activated).

In order to let the model generalize better, a Data Augmentation technique has been used. This method consists of augmenting the training images by applying some random modifications to them. In particular we apply the following augmentations:

- Rotation: the image is rotated by a maximum of 20 degrees.
- Shift: a translation (shift) on either width and height is applied, with a factor of 0.2.
- Shear: a random distortion is applied with an effect of 0.2.
- Zoom: the image is magnified by a maximum of 20% of their total dimension.
- Flipping: only horizontal flipping is applied to images.

In this way the number of training instances can be effectively increased: the model will see slightly different training samples during each epoch.

The common loss function like mean absolute error (MAE) and mean squared error (MSE), when used as the training objective, fails completely in their aim. These kind of loss function are not suitable for sparse binary multi-labeled data, when the position of a single bit is informative: different vectors but with the same amount of 1s (e.g. [0, 0, 1, 0] and [1, 0, 0, 0], versus [0, 0, 1, 0] and [0, 1, 0, 0]), produces the exactly same loss.
Training a model with these loss function leads to a completely dumb model: it predicts an array of 0's for whatever input instance regarding the belonging attributes. This scenario gets even worse when we deal with rare (in this case sparse) features.
The following example is provided for sake of clarity: if the model needs to predict whether a face belongs to a young person or not, and that feature is 1 for only 5% of the instances, a prediction of 0 (for every test instance) results in an accuracy of 95%. But the truth is that the model has not properly learned the feature.

For this reason, only relying on accuracy results may be misleading. In fact, we double-check the accuracy of our model with: qualitative results, and misprediction ratio.
Finally, a good loss function should understand the difference between two samples. Cosine Proximity is the loss function used for this purpose, and reported in the equation 1

$$L = -\frac{\mathbf{y} \cdot \hat{\mathbf{y}}}{\|\mathbf{y}\|_2 \cdot \|\hat{\mathbf{y}}\|_2} = -\frac{\sum_{i=1}^{n} y^{(i)} \cdot \hat{y}^{(i)}}{\sqrt{\sum_{i=1}^{n} (y^{(i)})^2} \cdot \sqrt{\sum_{i=1}^{n} \hat{y}^{(i)2}}} \tag{1}$$

where $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(n)}) \in \mathbf{R}^n$ and $\hat{\mathbf{y}} = (\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(n)}) \in \mathbf{R}^n$.
This function treats a binary array as a vector in a multidimensional space. The differences (or similarities) between the true and predicted labels are expressed as the arccos of the angle between them. When two vectors are orthogonal (theres a 90 angle between them), means that they are completely different (the loss value is at its maximum). Instead, when two vectors overlap (the angle is 0), they result the same (the loss value is at its minimum). The training process with cosine

proximity is effective: it allows the model to effectively learn patterns, used to infer the facial attributes.

In 1) the facial attributes recognition results are shown, along with the comparison against the LNet + ANet model proposed in [24].

Table 1. Performance comparison of attribute prediction models

	5 Shadow	Arched Eyebrows	Bags Under Eyes	Bald	Bangs	Big Lips	Big Nose	Black Hair	Blond Hair	Brown Hair	Bushy Eyebrows	Chubby	Double Chin	Eyeglasses	Goatee	Gray Hair	Heavy Makeup	High Cheekbones	Male
Our model	95	81	85	99	96	71	84	90	96	89	92	96	96	99	97	98	91	87	98
LNets+ANet	91	79	79	98	95	68	78	88	95	80	90	91	92	99	95	97	90	87	98

	Mouth Slightly Open	Mustache	Narrow Eyes	No Beard	Oval Face	Pointy Nose	Receding Hairline	Rosy Checks	Sideburns	Smiling	Straight Hair	Wavy Hair	Wearing Earrings	Wearing Hat	Wearing Lipstick	Wearing Necklace	Wearing Necktie	Young	**Average**
Our model	94	97	87	96	76	76	93	95	98	93	83	82	90	99	91	88	97	88	**91**
LNets+ANet	92	95	81	95	66	72	89	90	96	92	73	80	82	99	93	71	93	87	**87**

4 Experimental Results

As anticipated in the previous section, the dataset which has been used is CelebA dataset: for the training. phase, it has been divided in three partitions: *training*, *validation*, and *testing*; according to the partitioning suggested by the CelebA authors. For training, the model has been fed with the entire training partition ($160k$ samples resized to 224×224 and augmented) and validate each epoch on $20k$ of samples (validation-set). The model has been trained with a batch size of 64 images (mainly due to resource limitation), and let the AdaDelta [25] optimizer minimize the cosine proximity loss function. The AdaDelta optimizer has been selected since it requires less hyperparameter-tuning. Moreover, AdaDelta has a few interesting features:

- It's able to adapt the learning rate according to a moving window of gradient updates, preventing to stop learning after many iterations.

- It converges faster: thanks to the adaptive learning, it's able to accelerate when the direction is promising and to slow down when the loss starts to get worse.

4.1 Clustering

The clustering approach is probably the richest contribution of the paper, since it allows to group the faces, according to the facial features of each of them; in unconstrained environment, the possibility to associate an identity to a face, even if partially occluded is a big benefit, mainly in sensitive areas. Thank to the cluster syntetization, it is possible to group faces with the same features, so to simplify potential identikit recognition, very useful in forensics. The steps which characterize the here proposed clustering approach are the following:

- Choosing the number of clusters (no fixed values);
- Eventually, selecting a subset of the available attributes;
- Graphically visualizing the resulting clustering;
- Synthesizing a cluster into one face, which is the cluster's *eigenface*.

So, the target of the clustering method is to group together instances having similar attributes.

According to these requirements, it has been evaluated the goodness of different clustering algorithms by computing the *silhouette score* of each clustering. Furthermore, we give graphical insights by: attributes-occurrence plot, and the cluster's eigenface.

Considering that DBSCAN discovers the optimal number of clusters (for a given set of data), for a fair evaluation, we tested the performance of all the methods on the same number of clusters (the one discovered by DBSCAN), in order to analyze which one is the best (figure 4).

Subsequently, the performances of both K-Means and Agglomerative Clustering on a variable number of clusters has been compared with those of DBSCAN (figure 4).

In particular, as regards Agglomerative Clustering, various input parameters are analyzed in order to optimize performance. In fact, it processes by merging clusters together according to a linkage criteria:

- **Ward** minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the K-Means objective function but tackled with an agglomerative hierarchical approach.
- **Maximum or complete linkage** minimizes the maximum distance between observations of pairs of clusters.
- **Average linkage** minimizes the average of the distances between all observations of pairs of clusters.
- **Single linkage** minimizes the distance between the closest observations of pairs of clusters.

Also, the linkage criteria determines the metric used for the merge strategy: The latter can be euclidean, l1, l2, manhattan, or cosine. If linkage is ward, only euclidean is accepted.
The best combination of metric and linkage criteria is ”complete” and ”manhattan”, but despite this choice, the algorithm’s performance is still inferior to that of K-Means. Agglomerative Clustering reaches a silhouette score of 0.73, while K-Means achieves a value of 0.83.
Despite the better performances of DBSCAN, with a silhouette score of 0.87, we have decided to use K-means, since, as can be evaluated from the charts, it proves to be the best among the algorithms that allow you to choose the number of clusters.

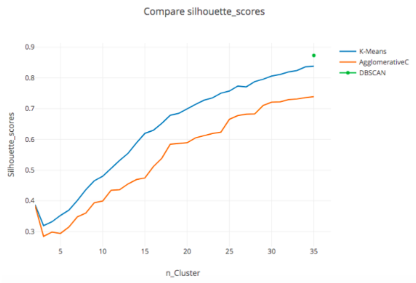


Fig. 4. Comparison of clustering algorithms: as we can see K-Means performs always better than Agglomerative Clustering.

More in details, the K-Means algorithm divides a set of N samples X into K disjointed clusters C, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster centroids; they are not, in general, points from X, although they live in the same space.
The K-Means algorithm aims at choosing centroids which minimize the inertia, or within-cluster sum-of-squares criterion:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2) \tag{2}$$

Following the clustering process, the results obtained by each cluster are analyzed, both in terms of characteristics and in terms of the corresponding eigenface.
The eigenfaces are computed (by standard dimensionality reduction methods, like PCA) in order to obtain an exhaustive and meaningful representation of each cluster.
Hand-checking the attributes of the faces in a given cluster is tidy and error-prone, especially for clusters with a large amount of faces.

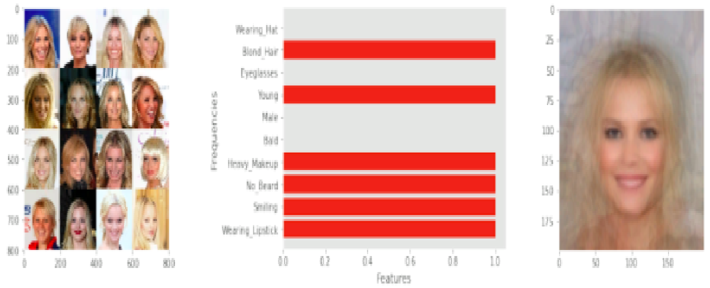


Fig. 5. Clustering results: the cluster (left), the attributes-occurence chart (middle), and the cluster's eigenface (right).

Regarding this, here two simple ways to summarize a cluster are shown:

1. By simply showing the chart of occurrences of the facial attributes within a given cluster, it is possible to understand the frequency of each attributes.
   With a chart like the one in figure 5, it is possible to determine which are the noisy attributes (those with low occurrence) and which are the prominent attributes (those with a very high occurrence); as regards the first, these outlier attributes are some sort of mistakes, made by the model and/or by the clustering method;

2. An Eigenface can be seen as a cluster representation, being representative and compact (it results in a single image).
   From a given cluster its eigenface is generated, by simply performing Principal Component Analysis (PCA) [23] on vectors obtained by flattening the images belonging to that cluster.
   The resulting eigenface is a face characterized by the prominent attributes of the given cluster. So, by simply observing the cluster eigenface is possible to determine what are the relevant attributes of that cluster.

In order to further improve the clustering process, a weight criterion is adopted in relation to the most frequent characteristics, or alternatively, to the less frequent characteristics.

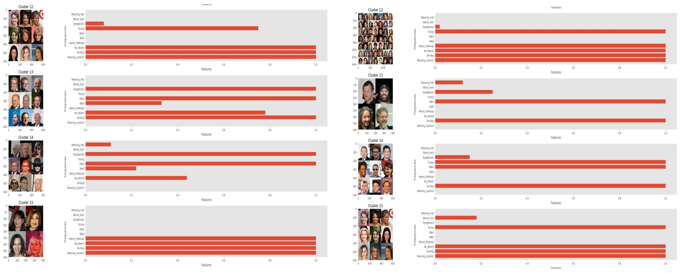


Fig. 6. 1. Clusters without weights 2. Clusters with weights

Thanks to this expedient, it is possible to drastically reduce the quantity of features that are only partially present within the cluster. The chart shown, in fact, describes the results following this operation.

Before focusing on the MobileNetV2 architecture, we have tried different model architectures even with 10x more parameters (our model has only 4.3M of parameters); each model achieves roughly the same accuracy, but at the cost of a much slower training.
Our model achieves 90.95% testing accuracy (table

5 Conclusion and Future enhancements

Biometric Recognition in unconstrained scenario is not always an easy practice; the light conditions, the pose and eventual occlusions, do not allow an accurate detection, with the consequent activity of subject recognition which results very challenging. Also, a precise recognition in such scenarios is not constantly ensured, and the guide of an operator is often needed for recognizing a subject and associating an identity to a face. In such cases, relying on soft biometrics, like facial features taken singularly, is an interesting and even more realistic decision. From this point of view, in this work it has been proposed a framework for facial features recognition by means of MobileNet-like convolutional neural network, and face clustering based on K-means and Eigenface for centroid definition. The results show that the recognition accuracy is higher than the net proposed by [24] and also the clustering return very interesting results. Qualitative and quantitative results have been proposed, also showing the advantages of the k-means with respect to DBSCAN and Agglomerative Clustering techniques.

Acknowledgments

A special thank goes to the students Luca Anzalone, Marialuisa Trere and Simone Faiella for having conducted the experiments and proposed the model.

References

1. Abate, A.F., Barra, P., Bisogni, C., Nappi, M., Ricciardi, S.: Near real-time three axis head pose estimation without training. IEEE Access **7**, 64256–64265 (2019). https://doi.org/10.1109/ACCESS.2019.2917451
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding (2006)
3. Barra, P., Bisogni, C., Nappi, M., Ricciardi, S.: Fast quadtree-based pose estimation for security applications using face biometrics. In: Au, M.H., Yiu, S.M., Li, J., Luo, X., Wang, C., Castiglione, A., Kluczniak, K. (eds.) Network and System Security. pp. 160–173. Springer International Publishing, Cham (2018)
4. Barra, S., De Marsico, M., Galdi, C., Riccio, D., Wechsler, H.: Fame: Face authentication for mobile encounter. In: 2013 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications. pp. 1–7 (Sep 2013). https://doi.org/10.1109/BIOMS.2013.6656140
5. Barra, S., Castiglione, A., Narducci, F., Marsico, M.D., Nappi, M.: Biometric data on the edge for secure, smart and user tailored access to cloud services. Future Generation Computer Systems (2019). https://doi.org/https://doi.org/10.1016/j.future.2019.06.019, `http://www.sciencedirect.com/science/article/pii/S0167739X19303188`
6. Fenu, G., Marras, M.: Leveraging continuous multi-modal authentication for access control in mobile cloud environments. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10590 LNCS**, 331–342 (2017). https://doi.org/10.1007/978-3-319-70742-$6_3$1
7. Fenu, G., Marras, M.: Controlling user access to cloud-connected mobile applications by means of biometrics. IEEE Cloud Computing **5**(4), 47–57 (2018). https://doi.org/10.1109/MCC.2018.043221014
8. Fenu, G., Marras, M., Meles, M.: A learning analytics tool for usability assessment in moodle environments. Journal of E-Learning and Knowledge Society **13**(3), 23–34 (2017). https://doi.org/10.20368/1971-8829/1388
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
10. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments (2008)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks (University of Toronto, 2012)
12. Link:: http://cs231n.github.io/transfer-learning/tf
13. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition (Georgia Institute of Technology,Carnegie Mellon University and Sun Yat-Sen University, 2017)
14. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset (Multimedia Laboratory, The Chinese University of Hong Kong, 2015)
15. Otto, C., Wang, D., Jain, K.: Clustering millions of faces by identity
16. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition (IEEE, 2017)
17. Rosebrock, A.: Face clustering with python (2018)
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2015)

19. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks (The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018)
20. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
22. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions (2014)
23. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis (1987)
24. Yang, S., Luo, P., Loy, C.C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
25. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
26. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks (IEEE, 2016)
27. Zhu, X., Ramann, D.: Face detection, pose estimation, and landmark localization in the wild