Correlation between Professional Domain and Facial Features based on Face Clustering

Bithiah Yuan

October 18, 2019

1 Introduction

A significant source of information and attributes can be derived from the human face by non-verbal communication [14]. As a result, facial features have been studied extensively in the social-science domain to predict success in reaching reputable leadership positions. In particular, studies have shown that certain facial features contribute to higher salaries and more prestigious employments for CEOs. In application, the relationship between facial characteristics and social attributes can provide an more powerful objective indicator for organizations to idenity and select effective leaders within their domain than broad facial cues such as attractiveness and competence. Results have shown that a human judge can identify business, military, and sports leaders from their faces with above-chance accuracy. However, these results are biased and do not imply the actual leadership qualities of a person [18].

Caused by behaviour experiments from human judgement, the the research of the social attributes and facial features in the social-sciences are limited in scalability, consistency, and generalization. For example, prior familiarity to the faces of the study and personal preferences can affect the results. Therefore, a growing number of social trait judgment studies have been extended and refined to computer vision and machine learning research due to the capability of using massive datasets and large-scale processing capacity [14].

Through a computational framework, [14] examined the relationship between facial traits and the social construction of leadership by a trained model that can predict the outcomes of political elections based on the perceived social attributes of a person's appearance. The results indicate that similar methods can be used to predict behavior in a broad range of human social relations, such as mate selection, job placement,and political and commercial negotiations [14].

Clustering analysis is an unsupervised learning technique that groups data points into clusters based on their similarities. It is useful in grouping a collection of unlabeled data with similar nature into clusters. [24] investigated clustering a large number of unlabeled face images into individual identities present in the data [24]. The workflow shown in Figure1. consists of obtaining face representations of a collection of unlabeled date by a deep neural network. The choice of clustering algorithm then groups the face images according to their identity.

Motivated by the researches in computational social trait judgment and [24], the following paper aims to examine the correlation between a person's profession based

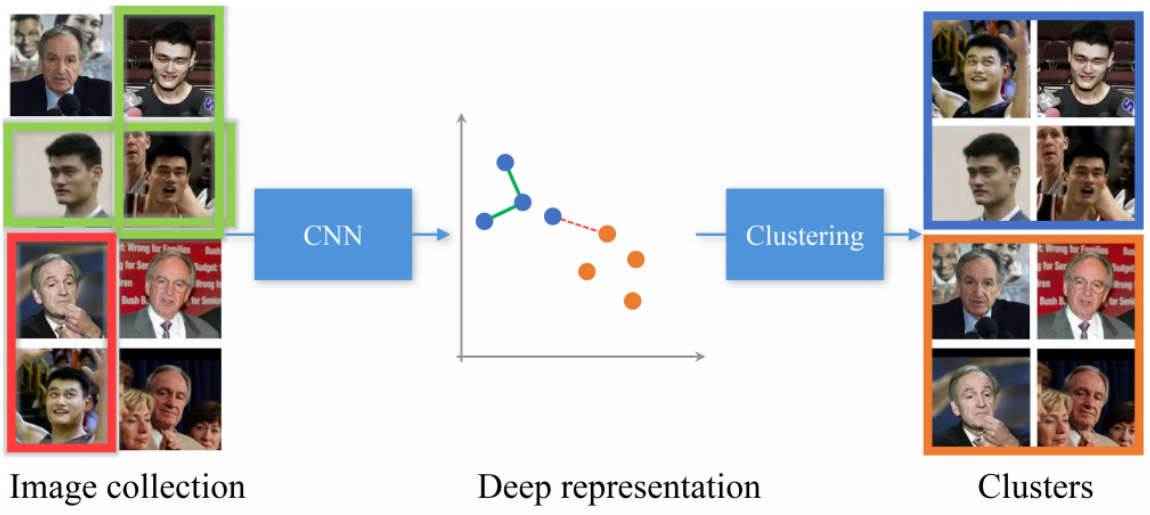

Figure 1: Face clustering workflow [24]

on their facial features through clustering face images. The clustering problem consists of the face representation and similarity metric of the face images and the choice of clustering algorithm [24]. Due to the importance of the underlying face representation in face clustering, this paper further compares different open-source state-of-the-art feature extraction methods based on deep learning.

2 Related Work

2.1 Face Recognition

Face recognition focuses on identifying or verifying the identity of subjects in images or videos [21].

Face recognition systems are usually composed of the following 4 steps [21]:

1. **Face Detection:**
   Detect the position of the faces in an image and returns the coordinates of a bounding box for each face as shown in Figure 2.

2. **Face Alignment:**
   Find a set of facial landmarks with the best affine transformation that fits a set of reference points located at fixed locations in the image. As shown in Figure 3, this step also includes resizing and cropping the image to the edges of the landmarks [5]. More specifically, as shown in Figure 4 given a set of mean landmark locations, the affine transformation makes the landmarks detected in the face image close to the mean [5].

3. **Face Representation:**
   Transform the pixel values of a face image into a low-dimensional discriminative feature vector, also known as an embedding.

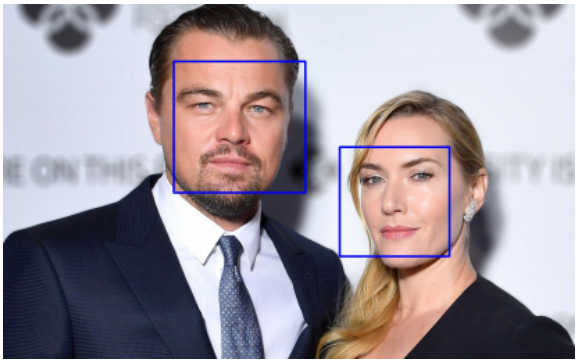

Figure 2: Face Detection [21]

Figure 3: Face Alignment [21]

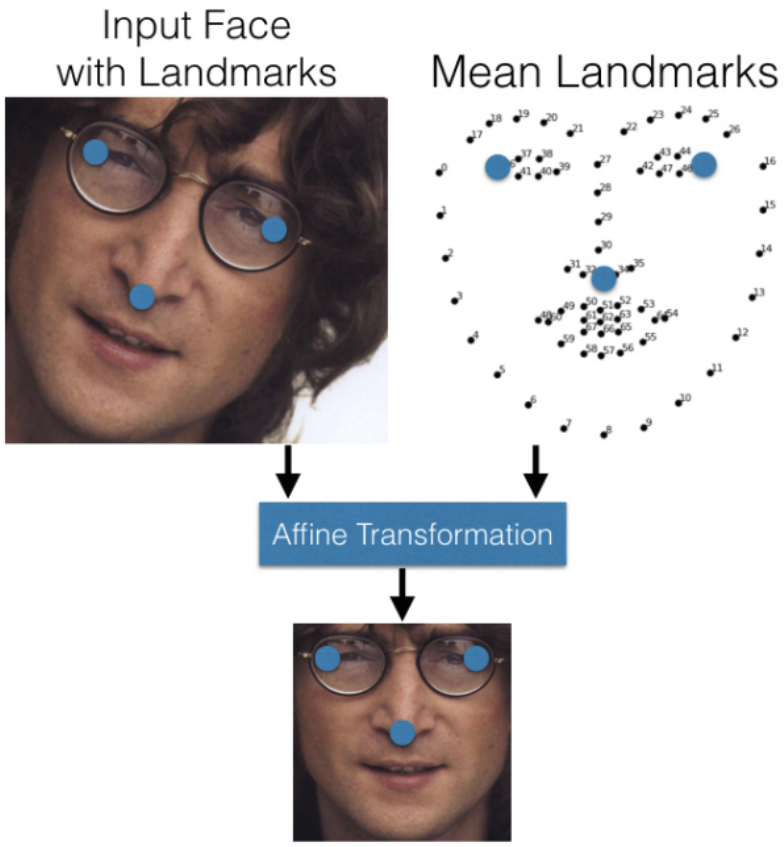

Figure 4: Applying affine transformation so that the face image is closer tothe set of mean landmarks. [5]

4. **Face Matching:**
   Compute similarity scores from feature vectors.

2.2 Face Detection

2.2.1 Histograms of Oriented Gradients

A traditional method for face detection is the Histograms of Oriented Gradients (HOG) descriptors, which utilizes the distribution of local intensity gradients or edge

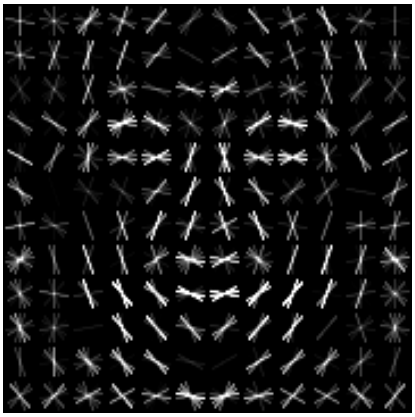

Figure 5: Trained HOG detector on multiple faces [21]

Figure 6: Hog representation of a face [10]

directions to characterize local object appearance and shape in an image. HOG divides the image into small grids, where each grid accumulates a histogram of gradient directions or edge orientations over the pixels of the cell. The combination of all the histogram in the cells form the face region. The cells are then normalized for better invariance to illumination, shadowing, and other variations. The normalized local histograms of image gradient orientations in a dense grid as features are then trained to classify the region of the face in an image [8]. When encountering a HOG representation of a new face image as shown in Figure 6, the part of the image that looks most similar to a trained HOG detector as shown in Figure 5 will form the region of the face.

2.2.2 Multi-task Cascaded Convolutional Networks

Another than using the tradition HOG representation and landmark detectors, face detection and alignment can also be done using CNNs. In particular, Multi-task Cascaded Convolutional Networks (MTCNN) is a widely used method to predict face and landmark location. The framework has a cascaded structure with three stages of deep CNNs shown in Figure 7. [25]

The image is first resized to different scales to build an image pyramid and is the input of the three stages:

1. **Proposal Network (P-Net):** Obtain candidates that will serve as potential positions of the bounding boxes [7].
2. **Refine Network (R-Net):** Using the image and the results of the first prediction of the bounding boxes, reduce false positives to get the final box boundaries [7].
3. **Output Network (O-Net):** Outputs five facial landmark positions [25].

Both the predicting the bounding box regression and facial landmark localization uses regression to minimize the Euclidean loss between the candidate positions of the bounding boxes, landmark coordinates and the ground truth. The ground truth of the bounding boxes are the left, top, height, width of the box. The ground truth of the facial landmarks are the coordinates of the left, right eye, nose, left and right mouth corner [25].

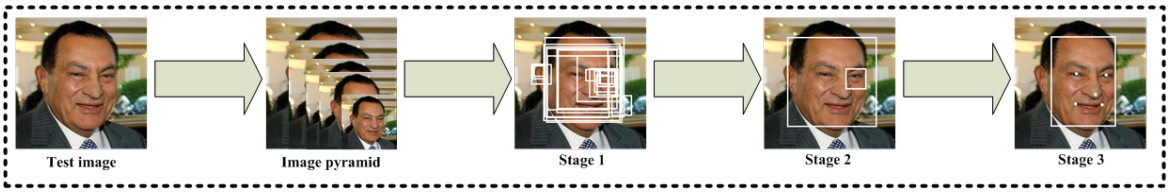

Figure 7: MTCNN: Cascaded structure with three stages of deep CNNs. [25]

Figure 8: Neural Network Traing Flow. [5]

2.3 Face Representation

Face representation is conceivably the most important component in the system. However, challenges occur in real world (in-the-wild) images due to variations ranging from head poses and illumination conditions to aging and facial expressions [21].

Traditional techniques include using statistical methods such as Principal Component Analysis (PCA) to represent faces as a combination of eigenvectors [5]. The top-performing face representation techniques use deep learning methods based on convolutional neural networks (CNNs) [5] since they are able to achieve very high accuracy by learning robust features due to the availability of large-scale faces in-the-wild datasets on the web [21].

2.3.1 Convolutional Neural Networks

As shown in Figure 8 a neural network feeds the input into many layers of function compositions followed by a loss function which measures how well the neural network models the data. Each layer is parameterized by a vector or matrix θ_i and the aim is to optimize the loss function iteratively by finding the optimal gradients $\delta L \ / \ \delta \theta_i$ which are computed with the backpropagation [5].

Residual networks (ResNets) is a popular network architecture for face recognition. ResNets introduces a shortcut connection to learn a residual mapping which contributes to information flow across layers and allows the training of much deeper architectures [21].

A common approach to training CNN models for face recognition is use a classification approach, where each face image in the training set corresponds to a class. When recognizing a new face image, the classification layer is discarded and the features of the previous layer are used as face representations. The downsides of this approach is that it doesn't generalize well to new face faces and that the representation size per face is large and inefficient [23].

Another approach is to learn the features for face representation directly by optimizing the distance between pairs or triplets of faces, in which the distances measure the similarity between faces [21] [23].

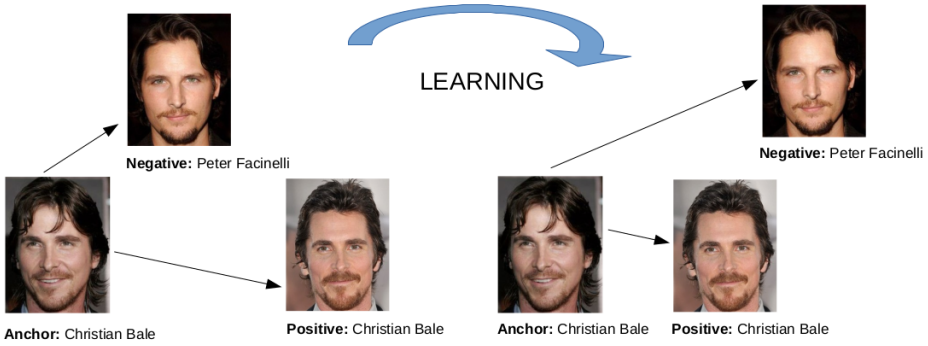

Figure 9: The loss of identical faces are minimized and the loss of distinct faces are maximized by the triplet loss function [1] [2].

2.3.2 Triplet Loss Function

When learning the face features directly, the choice of loss function has a great influence on the accuracy. One of the most used metric is the triplet loss function. The goal of the loss is to separate the distance between two aligned matching (positive) face images and a non-matching aligned (negative) face image by a distance margin. The result is a feature vector $f(x)$, known as embeddings, from a face image x to a compact Euclidean feature space in $\mathbb{R}^d$. The distance of the embeddings will be small if the faces are identical and large if the faces are distinct [23].

More specifically, as shown in the example in Figure 9. the distance between an anchor face image, x_i^a is minimized by the loss and will be closer to all other positive face images x_i^p than the negative face images x_i^n where the distance is maximized by the loss. For each triplet i, the following condition needs to be satisfied:

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

where α is a margin that from the positive and negative pairs [21].

For N possible triplets, the loss being minimized is:

$$L = \sum_i^N \Big[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\Big]_+$$

[23].

As shown in Figure 10 using a neural network, the triplet loss is computed and it's gradient is backpropagated through the network to the unique images [5].

3 Face Representation Methods

The following section examines open-source state-of-the-art face feature extraction methods.

3.1 FaceNet

FaceNet is a method that uses a deep CNN along with the GoogLeNet style Inception models and the triplet loss function to directly optimize the face embeddings.

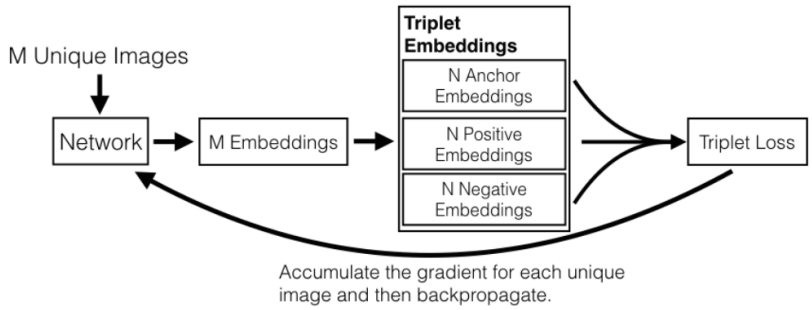

Figure 10: Learning the embeddings by optimizing the gradients of the triplet loss function [5].

The faces of images are first detected and aligned with MTCNN [22], the resulting face images are 160 x 160 pixels and serve as the input for the FaceNet model. The structure of FaceNet consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training as shown in Figure 11 [23].

Between 100 to 200 million face images consisting of about 8 million different identities were used for training. The large dataset of labelled faces consist of various poses, illuminations, and other variations. [23].

The pre-trained model (20180402-114759) of open-source FaceNet implementation [22] used in the experiment from the next section was trained using the VGGFace2 dataset. The faces of the dataset was detected and aligned using MTCNN . The dataset contains 3.31 million images of 9131 identities, with an average of 362.6 images for each person. The images were downloaded from Google Image Search and have large variations in pose, age, illumination, ethnicity and profession. The model uses the Inception ResNet v1 architecture and has an accuracy of 99.65% on the Labeled Faces in the Wild (LFW) benchmark [22]. [22]'s implementation results in a 512-dimensional feature vector.

3.2 Dlib-ml

[11] built a face recognition method using Dlib-ml, which is an open source library for developing machine learning software in C++ [16]. [11] used the HOG face detector from Dlib, which the HOG representation was trained with a linear classifier (SVM) [15]. The face representation model uses the ResNet architecture with 29 convolution layers and the triplet loss function to learn the embeddings [17]. The resulting feature embedding is a 128-dimensional vector [17].

A dataset of about 3 million faces and 7485 unique identities from a combination of the face scrub and VGG dataset as well as a large number of other images scraped from the internet was used for training. The pre-trained model has an accuracy of 99.38% on the LFW benchmark [17].

3.3 OpenFace

OpenFace uses Dlib for face detection and alignment. The input images after alignment are 96 x 96 pixels. OpenFace was trained with 500,000 images from a combination

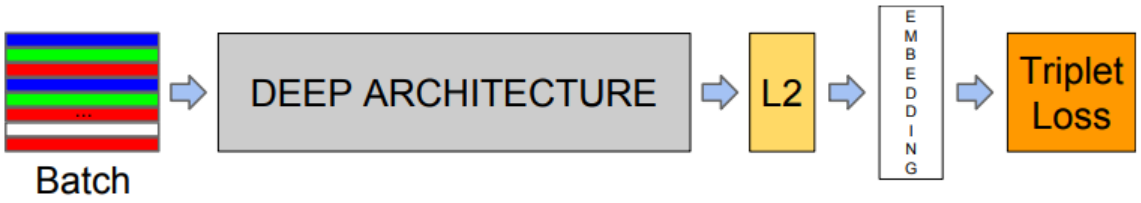

Figure 11: FaceNet model structure [5]

of CASIA-WebFace and FaceScrub. The face representation is obtained using a modification of FaceNet's architecture, which the number of parameters are reduced. The resulting feature embedding is a 128 dimensional vector [5].

3.4 ArcFace

[9] introduced a new loss function, additive angular margin (ArcFace), that uses a geometric interpretation for learning the discriminative features for face representation [9]. After applying MTCNN for face detection and alignment [13] get the 112 x 112 face input images, ArcFace further adjusts a face image by rotating an image to a straight face as shown in the comparsion in Figure 12. Consequently, the positions of eyebrows, eyes, nose, and mouth in different images are consistent and increases the effective when computing the similarities between the embeddings. The resulting embedding is a 512-dimensional vector [9].

A downside of ArcFace is that it cannot compute face representations of images with the high intensity of light reflection [7] as shown in Figure 13.

The model is trained on MS1MV2 which is a refinement of the MS-Celeb-1M dataset. The training dataset contains about 10 million images of 100,000 top celebrities selected from one million celebrities in terms of their web appearance frequency [9]. The pre-trained model (LResNet100E-IR) used in the experiment in the next section was trained on the MS1MV2 dataset using the ResNet100 architecture which achieved an accuracy of 0.9982 on the LFW benchmark [13].

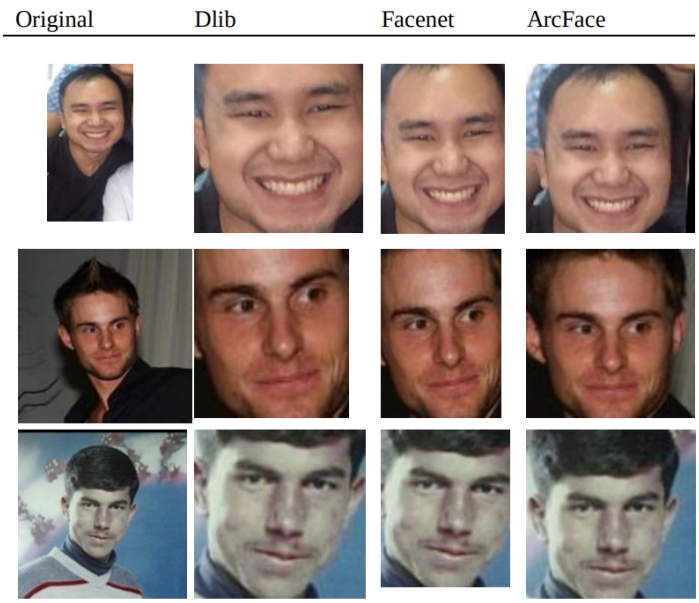

Figure 12: Comparsion of Face Detection between Dlib, FaceNet, and ArcFace [7]

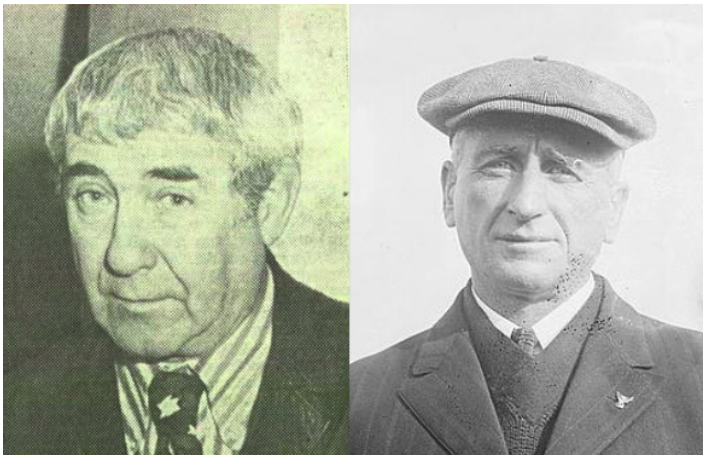

Figure 13: Images with high intensity of light reflection in which ArcFace is unable to compute the embeddings for [7]

4 Clustering Algorithms

4.1 K-Means

4.2 Hierarchical The Agglomerative

4.3 Spectral

4.4 Birch

5 Experiment

5.1 Dataset

5.2 Evaluation

The ROC curve shows the tradeoffs between the TPR and FPR.The perfect ROC curve would have a TPR of 1 everywhere, which is where todays state-of-the-artindustry techniques are nearly at. [5]

6 Conclusion

References

[1] `https://i.pinimg.com/originals/76/c3/ea/76c3ea5bcd34a4d7435320c05651d494.jpg`.

[2] `https://www.slovenskenovice.si/images/slike/2018/04/28/239717.jpg`.

[3] *2.3. Clustering*. `https://scikit-learn.org/stable/modules/clustering.html`. 2011.

[4] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. *OpenFace*. `http://cmusatyalab.github.io/openface/`. 2016.

[5] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. CMU-CS-16-118, CMU School of Computer Science, 2016.

[6] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: vol. 8. Jan. 2007, pp. 1027–1035. DOI: `10.1145/1283383.1283494`.

[7] Adulwit Chinapas et al. "Personal Verification System Using ID Card and Face Photo". In: *International Journal of Machine Learning and Computing* 9 (Aug. 2019), pp. 407–412. DOI: `10.18178/ijmlc.2019.9.4.818`.

[8] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)* 2 (June 2005).

[9] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: (Jan. 2018).

[10] *Face Recognition with Deep Learning*. `https://www.hackevolve.com/face-recognition-deep-learning/`. 2017.

[11] Adam Geitgey. *Face Recognition*. https://github.com/ageitgey/face_recognition. 2017.

[12] Adam Geitgey. *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78. 2016.

[13] Jia Guo and Jiankang Deng. *InsightFace: 2D and 3D Face Analysis Project*. https://https://github.com/deepinsight/insightface. 2019.

[14] J. Joo, F. F. Steen, and S. Zhu. "Automated Facial Trait Judgment and Election Outcome Prediction: Social Dimensions of Face". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3712–3720. DOI: 10.1109/ICCV.2015.423.

[15] Davis E. King. *Dlib 18.6 released: Make your own object detector!* http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html. 2014.

[16] Davis E. King. "Dlib-ml: A Machine Learning Toolkit". In: (July 2009).

[17] Davis E. King. *High Quality Face Recognition with Deep Metric Learning*. http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html. 2017.

[18] Christopher Olivola, Dawn Eubanks, and Jeffrey Lovelace. "The many (distinctive) faces of leadership: Inferring leadership domain from facial appearance". In: *The Leadership Quarterly* 25 (Oct. 2014). DOI: 10.1016/j.leaqua.2014.06.002.

[19] Charles Otto, Dayong Wang, and Anil Jain. "Clustering Millions of Faces by Identity". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (Apr. 2016). DOI: 10.1109/TPAMI.2017.2679100.

[20] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[21] Daniel Saez Trigueros, Li Meng, and Margaret Hartnett. "Face Recognition: From Traditional to Deep Learning Methods". In: (Oct. 2018).

[22] David Sandberg. *facenet*. https://github.com/davidsandberg/facenet. 2018.

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: June 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[24] Yichun Shi, Charles Otto, and Anil Jain. "Face Clustering: Representation and Pairwise Constraints". In: *IEEE Transactions on Information Forensics and Security* PP (June 2017). DOI: 10.1109/TIFS.2018.2796999.

[25] Kaipeng Zhang et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23 (Apr. 2016). DOI: 10.1109/LSP.2016.2603342.