A Comparison of Facial Feature Extraction Methods based on Professional Domain Clustering

Bithiah Yuan

Master Project
University of Freiburg - Department of Compute Science
Chair of Databases and Information Systems

Motivation

- **Question:** Are facial features correlated with a person's professional talents?
- **Problem:** Research in the social sciences are limited in scalability, consistency, and generalization
- **Solution:** Computational method based face clustering

Face Clustering

- Clustering: groups data points together based on their similarities
- Group similar faces together and evaluate based on profession
- The accuracy can determine if facial features are correlated with one's professional domain
- Face clustering is usually composed of 4 steps

Face Clustering

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

2. Face Alignment: Find a set of facial landmarks, resize and crop the image to the edges of the landmarks

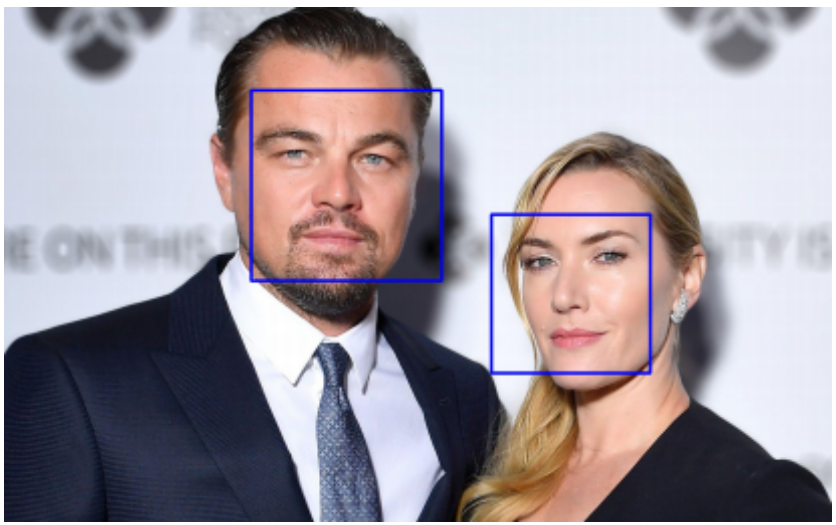


Figure: Face Detection [7]

Figure: Face Alignment [7]

Face Clustering

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

a.) Peter Dewald
Manager

b.) Reinhard Wolf
Manager

c.) Roberto Sanchez
Fighter

d.) Beneil Dariush
Fighter

Face Clustering

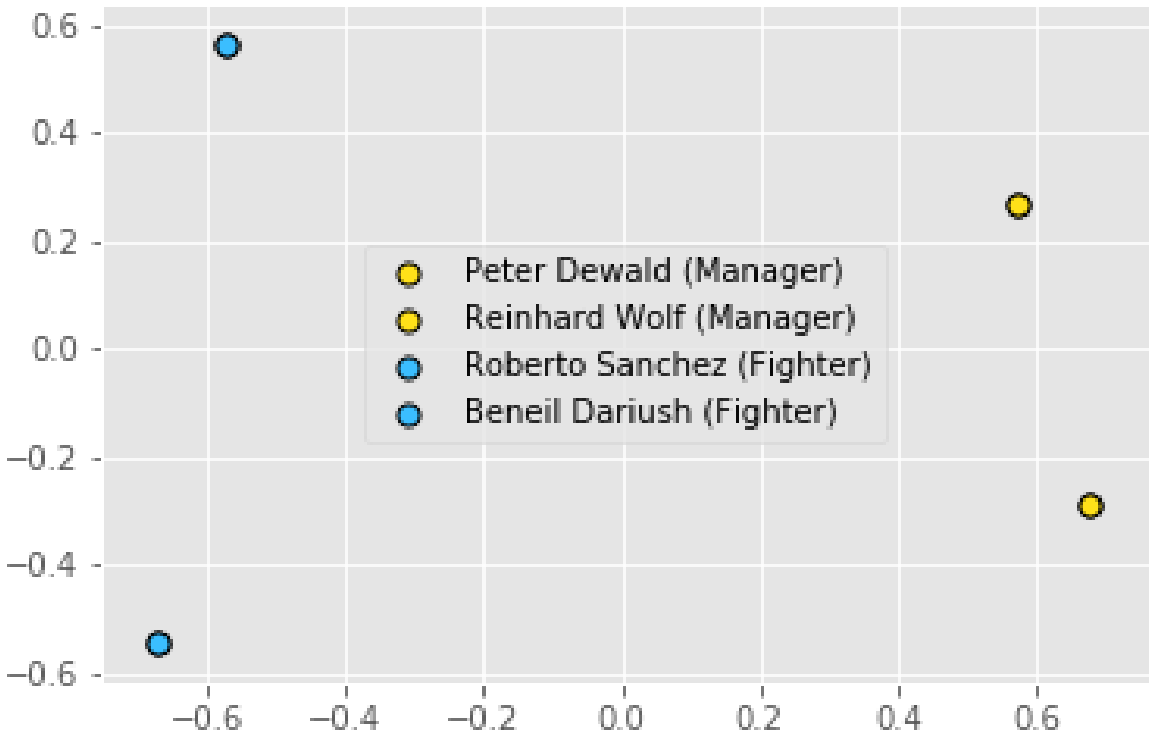


Figure: Professional Domain Clustering [7]

Face Detector: Histograms of Oriented Gradients (HOG)

- HOG divides the image into small grids
- Each grid accumulates a histogram of gradient directions over the pixels of the grid
- Trained to classify the region of the face in an image
- The part of the image that looks most similar to a trained HOG detector will be detected

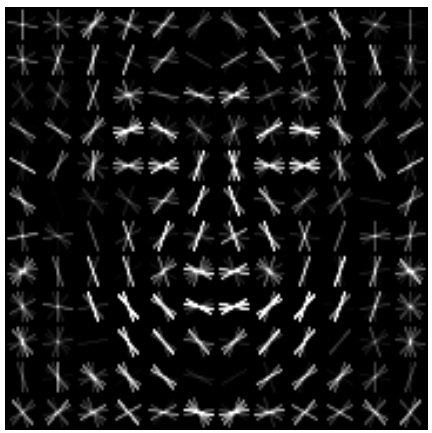


Figure: Trained HOG detector [7]

Figure: HOG representation [5]

Face Detector: Multi-task Cascaded Convolutional Networks (MTCNN)

1. **Proposal Network (P-Net):** Obtains candidates that will serve as potential positions of the bounding boxes
2. **Refine Network (R-Net):** Reduce false positives of the first prediction and get the final box boundaries
3. **Output Network (O-Net):** Outputs landmark positions

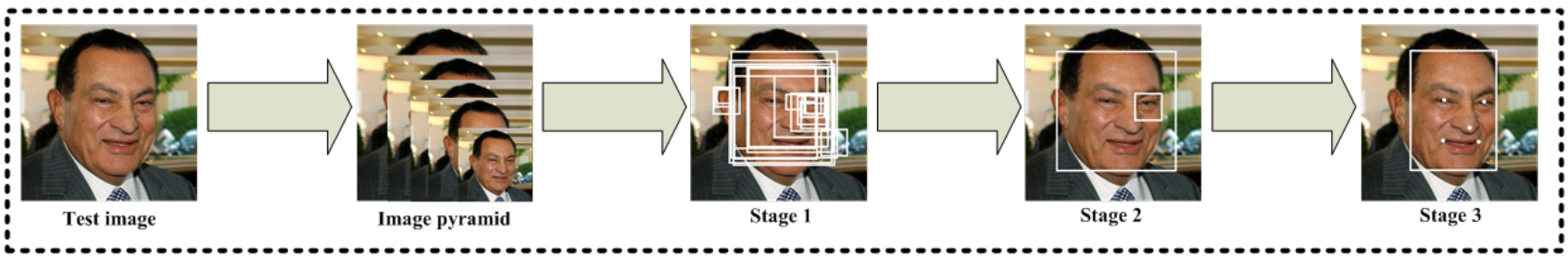


Figure: Cascaded structure with 3 stages of deep CNNs. [8]

Face Representation

- The top-performing face representation techniques use CNNs
- Learned robust features of large-scale in-the-wild face datasets directly
- CNN feeds the input into many layers of function compositions followed by a loss function
- The aim is to optimize the loss function and learn the embeddings directly

Triplet Loss Function

- Optimize the distance between two positive face images and a negative face image
- Result is a feature vector $f(x)$ from a face image x to a compact Euclidean feature space in $\mathbb{R}^d$.

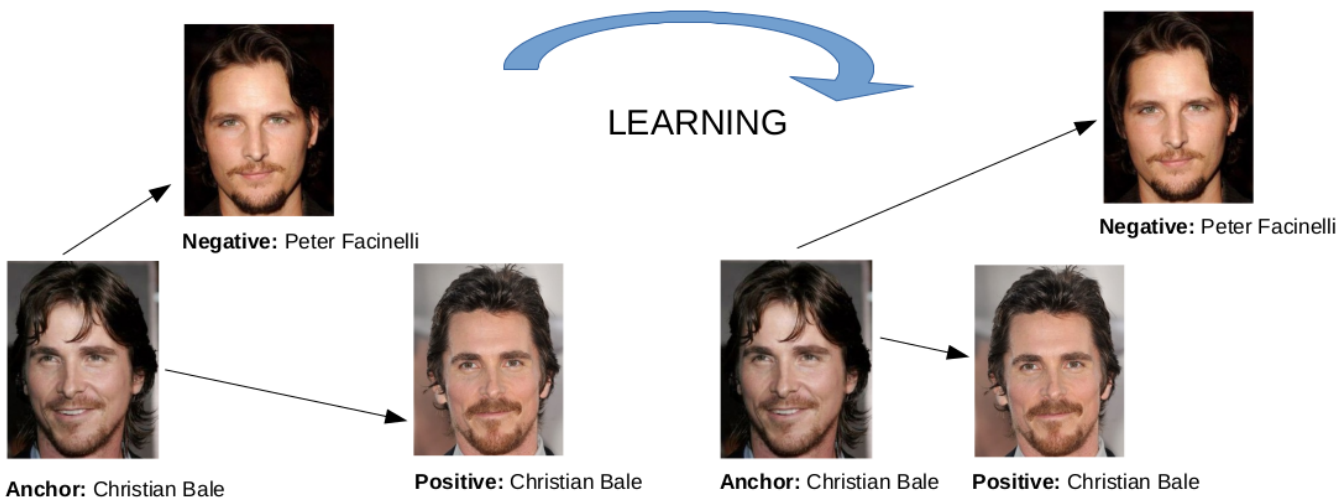


Figure: The loss of identical faces are minimized and the loss of distinct faces are maximized [1] [2].

Set-Up

- **Goal:** Cluster similar faces and evaluate the result by profession
- **Feature Extraction Methods:** FaceNet, Dlib, OpenFace, ArcFace
- Pre-trained models from each method was used
- **Clustering algorithms:** K-Means, Spectral, Hierarchical Agglomerative, EM, Birch
- Number of professions = Number of clusters

Dataset

- **Experiment 1:** $2,180$ unique images of five categories of athletes
- **Experiment 2:** $2,065$ unique images of five different categories of professions
- **Experiment 3:** $4,593$ unique images of 11 different categories of professions
- The majority of images from Experiment 2 and 3 were obtained from Wikidata using SPARQL query service

Experiment 1 Dataset

Figure: Five categories of athletes [3]

Experiment 2 Dataset

Figure: Five categories of professions [3]

Experiment 3 Dataset

Figure: In combination with the dataset from Experiment 2: 11 professions

Feature Extraction Methods

Method	Face Detector	Number of Features
FaceNet	MTCNN (160 x 160 px)	512
Dlib	HOG	128
OpenFace	HOG (96 x 96 px)	128
ArcFace	MTCNN (112 x 112 px)	512

Table: The face detection method used and the number of features of the embeddings extracted from each method.

Feature Extraction Methods

Method	LFW Accuracy	Training Dataset Size
FaceNet	0.9965	3.31M images, 9,121 identitie
Dlib	0.9938	3M images, 7,485 identities
OpenFace	0.9292	500k images
ArcFace	0.9982	10M images, 100k identities
Human-Level [4]	0.9753	

Table: Accuracy based on the LFW benchmark and training data size of the pre-trained models.

Evaluation: Pairwise F-Measure

- Actual Clusters $L = \{\{A1, A2, A3\}, \{B1\}, \{U1, U2\}\}$
- Cluster Output $C = \{\{A1, A2, B1\}, \{A3, U1, U2\}\}$

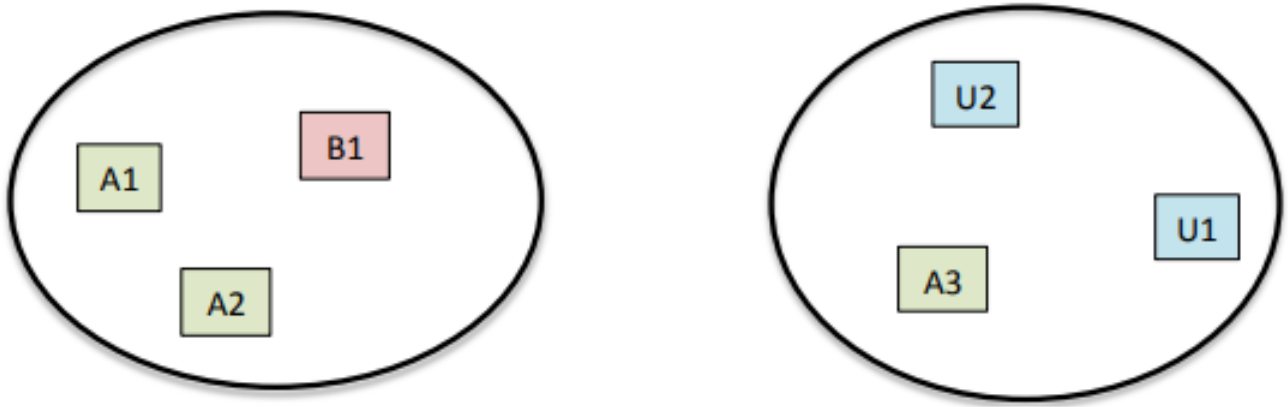
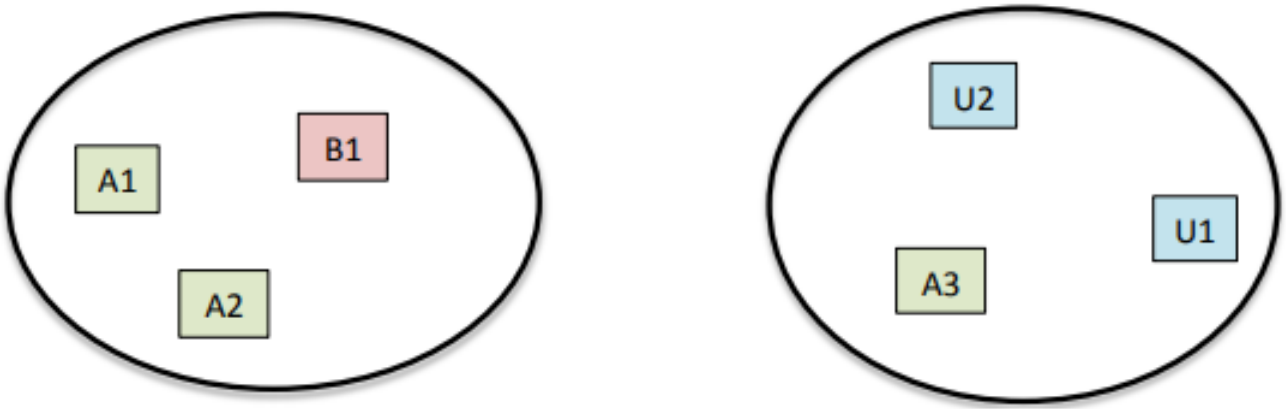


Figure: Example of a possible clustering output. Six data points are grouped into 2 clusters. A1, A2, and A3 have the same label, B1 has it's own label, and U1 and U2 have the same label [6].

Evaluation: Pairwise F-Measure

- **True Positives (TP):** The number of face pairs (i, j) that are correctly clustered into the same cluster.

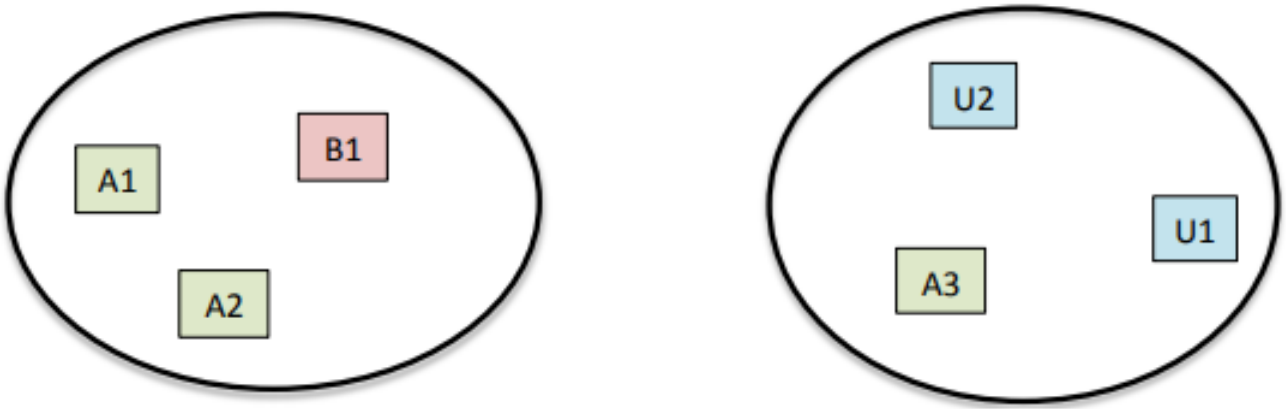
$$TP = |\{(A1, A2), (U1, U2)\}| = 2$$



Evaluation: Pairwise F-Measure

- **False Positives (FP):** The number of face pairs (i, j) that are incorrectly clustered to the same cluster.

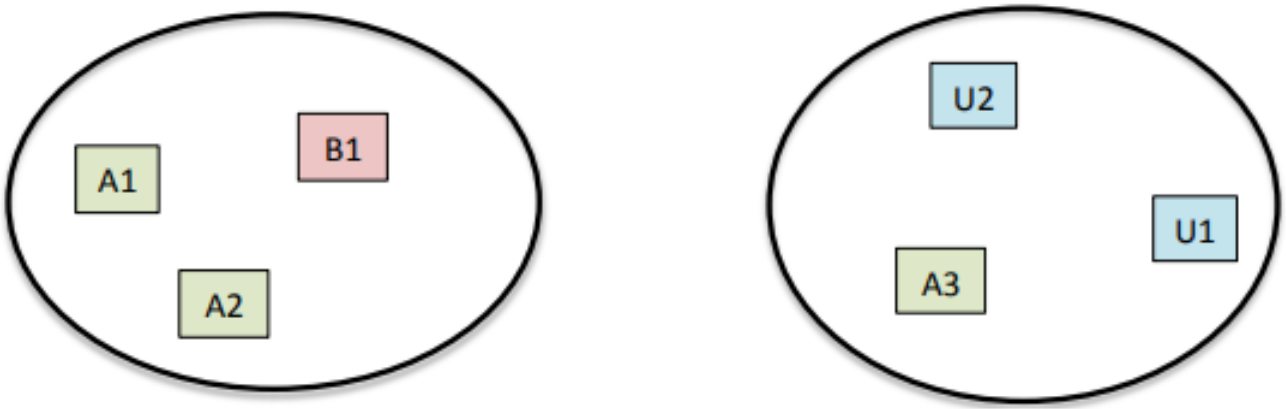
$$FP = |\{(A1, B1), (A2, B1), (A3, U1), (A3, U2)\}| = 4$$



Evaluation: Pairwise F-Measure

- **False Negatives (FN)**: The number of face pairs that are clustered to a different cluster.

$$FN = |\{(A1, A3), (A2, A3)\}| = 2$$



Evaluation: Pairwise F-Measure

$$Pairwise\ Precision = \frac{TP}{TP + FP}$$

$$Pairwise\ Recall = \frac{TP}{TP + FN}$$

$$Pairwise\ F\text{-}Measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Experiment 1 Results

Method	FaceNet	Dlib	OpenFace	ArcFace
K-Means	0.478	0.391	0.382	**0.516**
Spectral	0.445	0.361	0.345	**0.467**
HAC	**0.447**	0.362	0.391	0.413
EM	0.449	0.394	0.376	**0.512**
Birch	0.442	0.404	0.331	**0.453**

Table: F-Measure obtained by each feature extraction and clustering method

Experiment 1 True Positives

Figure: **True Positives** using FaceNet and K-Means. Clusters of Soccer Players and Fighting Champions.

Experiment 1 False Positives and False Negatives

Figure: **False Positives**

Figure: **False Negatives**

Experiment 2 Results

Method	FaceNet	Dlib	OpenFace	ArcFace
K-Means	**0.426**	0.405	0.306	0.294
Spectral	**0.392**	0.367	0.295	0.272
HAC	0.384	**0.392**	0.310	0.333
EM	**0.443**	0.411	0.310	0.277
Birch	**0.399**	0.383	0.294	0.304

Table: Experiment 2 F-Measure Comparison

Experiment 2 True Positives

Figure: Experiment 2: **True Positives** using FaceNet and K-Means. Cluster of Managers and Politicians

Experiment 3 Results

Method	FaceNet	Dlib	OpenFace	ArcFace
K-Means	**0.203**	0.185	0.141	0.182
Spectral	**0.204**	0.166	0.143	0.159
HAC	0.177	**0.188**	0.131	0.174
EM	**0.210**	0.191	0.146	0.179
Birch	0.182	**0.193**	0.149	0.151

Table: Experiment 3 F-Measure Comparison

Experiment 3 True Positives

Experiment 3 False Positives and False Negatives

Figure: **False Positives**

Figure: **False Negatives**

Analysis

- Experiment 1: The highest F-Measure was 0.516 using **ArcFace and K-Means clustering**
- Experiment 2: The highest F-Measure was 0.443 using **FaceNet and EM clustering**
- Experiment 3: The highest F-Measure was 0.210 using **FaceNet and EM clustering**

Analysis

Method	Experiment 1	Experiment 2	Experiment 3
FaceNet	**00:15:06**	**00:25:27**	00:57:44
Dlib	00:28:19	00:50:44	01:52:57
OpenFace	00:16:01	00:27:59	**00:53:30**
ArcFace	00:16:04	00:42:44	01:34:38

Table: Runtime to align and extract faces

Conclusion

- ArcFace is a good choice for images with high-quality head shots
- FaceNet is a better choice in terms of consistency, accuracy, and efficiency for in-the-wild images
- K-Means and EM Clustering provided the highest accuracy for the experiments
- Experiments 1 and 2 show a positive correlation between facial features and the person's professional domain

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

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

[3] Soumitra Agarwal. https://github.com/SoumitraAgarwal?tab=repositories.

[4] 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.

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

[6] 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.

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

[8] 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.