Fuzzy Face Clustering For Forensic Investigations

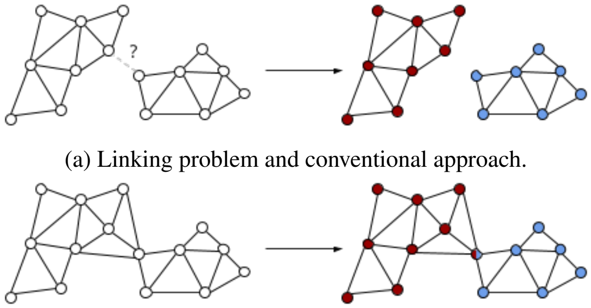
Hendrik J. Klip
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, Delft, The Netherlands
h.j.klip@student.tudelft.nl

Abstract

The amount of personal imagery kept on (mobile) devices is increasing by the day. Analysis and organization of these large collections of data are becoming increasingly important in the field of digital forensics, as they can aid in the search for legal evidence. The grouping of faces based on their identity is an important aspect as it provides an overview of the person in question and their connection with scenes, objects and other people. In this work, we propose a fuzzy approach to the hard partitioning problem of face clustering for the specific field of forensic investigations. We constructed a pipeline consisting of deep models for face detection and feature extraction, a method for transforming the resulting feature vectors to a graph representation and a graph-based clustering algorithm for the final partitioning. Focusing on the clustering step, we propose to assign face images to identity clusters using confidence values (rather than a hard cutoff) based on the average similarity with images present in the cluster relative to other clusters. Compared to existing methods, the approach is not only fuzzy but also embraces naïve linking, and instead of transitively merging the links it uses a graph-based algorithm to produce the clusters. Furthermore, we propose an adapted version of the MaxMax algorithm because the original method only returned fuzzy results if weights were exactly equal. However, similarities between images are continuous, making it unsuitable for the case of face clustering. Evaluation of the performance on the Labeled Face in the Wild (LFW) dataset and the challenging IARPA JANUS Benchmark B (IJB-B) shows promising results comparable with state-of-the-art face clustering algorithms.

1. Introduction

Forensic investigations have increasingly larger amounts of data at their disposal originating from sources such as surveillance footage and seized devices. Organizing the visual information contained in this data is crucial for the swiftness of investigations (take for example the Boston Marathon bombing [20, 36]). One desirable aspect is the grouping of faces based on their similarities to produce identity clusters. This information helps investigators to gain insight into the individuals present in the data and to form an overview of networks around suspects, be it criminal or social. Subsequently, this overview can provide evidence for the participation of the suspect or their accomplices in the investigated activities. Therefore, the correctness of the clustering is of utmost importance and it is preferable to append confidence values to the assignments of face images to identity clusters. These probability values lead to the possibility of a face belonging to multiple clusters. This results in a balance between clusters consisting of images of only one identity but this identity has possibly images in other clusters as well and clusters that are more diverse but get more images of the same identity together in one cluster. The latter is preferred as investigators prefer not to miss photos of a subject in a cluster with the sacrifice of also including photos of other subjects. However, the idea of multiple assignments of face images is counter-intuitive since a face can only belong to one person. To the best of our knowledge this has not been done before. So far, conventional methods use hard partition algorithms resulting in



(a) Linking problem and conventional approach.

(b) Our method.

Figure 1: The linking problem current methods try to solve (top) and our method using a simple metric for linking and a graph based algorithm (bottom).

non-overlapping clusters.

Advances in deep learning techniques have boosted the performance of face clustering methods considerably. By switching from handcrafted features to features extracted by a convolutional neural network [23], face representations are more discriminative than ever before. Commonly employed clustering algorithms such as K-means [26] and Agglomerative Hierarchical Clustering (AHC) [12] already show the superiority of these features. This evolution also shows the two fronts of development in the field of face clustering: the feature extraction network and the clustering algorithm.

While on the one hand developments are made on the side of neural network architectures [14, 33, 8] , on the other hand specifically designed algorithms are being proposed trying to solve the clustering problem [28, 25, 35, 24, 44, 41]. This problem is often seen as a linking problem where the question is which images should be linked together and therefore belong to the same identity. Linking two images is based on the similarity between them which can be defined in different ways. Basic notions include Euclidean and cosine similarities where more complex measures require nearest neighbor sets or another deep learning model.

In this work, instead of solving the linking problem directly by defining our own similarity measure, a simple metric (cosine similarity) with a cut-off threshold is used to construct a graph. Thereafter, a graph-based clustering algorithm, which is either Fuzzy Chinese Whispers [5] or an adapted version of the MaxMax algorithm dubbed α-MaxMax, is used to produce the partitioning (see Figure 1). The proposed method achieves results comparable with state-of-the-art crisp methods while outperforming other fuzzy approaches.

To summarize, the following contributions are made:

- A new approach for clustering face images using naïve linking with a cut-off threshold for constructing a graph representation of the data and a graph-based clustering algorithm providing the partitioning.
- The approach is also fuzzy, which is suitable for the field of digital forensics.
- A variant of the graph-based MaxMax algorithm called α-MaxMax, which can handle continuous weight values by setting a margin to cover a range of weight values instead of a single value.
- Fuzzy evaluation methods adapting homogeneity and completeness, which are appropriate for evaluating fuzzy solutions for both crisp and fuzzy problems.

The remainder of this paper is organized as follows. Section 2 covers the related work in the fields of face clustering and fuzzy clustering. Section 3 introduces the proposed method including the rest of the pipeline. Section 4 describes the datasets and metrics used for the evaluation of the method. Section 5 presents the experimental results and in Section 6 different elements of this work and the field of face clustering are discussed.

2. Related Work

Since this work combines two disciplines of clustering approaches, this section will cover the existing works in both fields.

2.1. Face Clustering

The rich variety in pose, illumination, occlusion of face images, the often unknown amount of identities and the varierty of cluster sizes render classical clustering algorithms like K-means [26] and Spectral Clustering [34] unsuitable for the situation. Not only do these algorithms require a pre-defined number of clusters, but they also have rigid assumptions on data distribution. Although AHC [12] also needs the number of clusters, it is capable of handling complex data distributions. For this reason, Lin et al. extended the algorithm to Proximity-Aware Hierarchical Clustering (PAHC) [25] by exploiting neighborhood similarities based on linear SVMs that classify positive and negative local instances. Improving the handling of large variations in cluster sizes, Lin et al. switched to density-based similarity with their Deep Density Clustering method [24]. They apply SVDD [38] to encapsulate local neighborhoods and iteratively merge them based on density similarities.

Also focusing on local neighborhoods, Otto et al. proposed Approximate Rank-Order clustering (ARO) [28]. First, for every sample an approximate nearest neighbor list is constructed. Next, pairwise distances between each face image are computed based on the presence/absence of shared nearest neighbors. Finally, pairs with a distance below a certain threshold are transitively merged. Shi et al. [35] took a different approach by treating the clustering problem as a Conditional Random Field (CRF) model. Their Conditional Pairwise Clustering (ConPaC) method tries to directly estimate the adjacency matrix by maximizing its posterior probability. Additionally, side information about whether pairs of images should be linked or not can be incorporated as pairwise constraints which allows for a semi-supervised approach. Finally, Wang et al. [41] used a graph convolutional network [18] to reason about instance pivot sub-graphs which are based on k-hop nearest neighbors and infer the likelihood of linkage between pairs in the sub-graphs.

2.2. Fuzzy Clustering

The notion of class membership was first introduced by Zadeh in his work about fuzzy sets [46]. A fuzzy set, being a class of objects with a grade of membership per object for the class, is characterized by a membership function defining this grade. Adopting the idea of fuzzy sets, Ruspini [32] proposed fuzzy c-partitions creating a fundamental basis for fuzzy clustering algorithms, such as Fuzzy C-means [9, 4].

2.2.1 FCM-type clustering

The Fuzzy C-means algorithm (FCM) was first developed by Dunn [9] and later improved by Bezdek [4]. FCM is a fuzzy variant of the well-known K-means algorithm [26] but instead of iteratively updating the cluster centers and the assignments of data points based on distance alone, the FCM algorithm involves the grade of membership as well. In this method, a parameter $m \geq 1$ determines the fuzziness of the results. The higher the value of m the fuzzier the membership assignments will become. On the other hand, if $m = 1$, a hard partitioning matrix will be produced with membership values of either 0 or 1.

The main issues of FCM and its variants [17, 22, 29, 30, 19] are the requirement to set the fuzziness index and to determine the number of clusters beforehand. For the former issue, Winkler et al. [43] propose a fuzziness index that is dependent on the dimensionality of the data setting it to be $m = \frac{2+d}{d}$, where d is the number of dimensions. They also concluded that without well initialized cluster centers FCM only works effectively in spaces of 5 or less dimensions.

Going one step further, Yang and Nataliani removed the need for all parameters of FCM in their Robust-Learning Fuzzy C-means (RL-FCM) algorithm [45]. The idea is to consider all data points as initial clusters with cluster weight: $\alpha_j = \frac{1}{n}$. Then, iteratively update cluster memberships and weights and discard clusters that satisfy $\alpha_j \leq \frac{1}{n}$.

2.2.2 Graph-based clustering

Beside FCM-type algorithms which iteratively update the cluster centers and recalculate membership values, another category of fuzzy methods are graph-based algorithms. These algorithms take an undirected weighted graph $G = (V, E)$ and cluster the nodes based on their adjacent edges.

Inspired by the eponymous children's game, Biemann proposed the graph-based clustering algorithm Chinese Whispers [5]. This algorithm starts by assigning all nodes their own cluster such that the initial number of clusters matches the number of nodes. Secondly, it iterates over the nodes in a random order and assigns them the cluster label that has the highest rank in its neighbourhood. The neighbourhood of a node v_i consists of all its adjacent nodes excluding itself ($v_i \neq v_j$) and is formally defined as:

$$K_i = \{v_j | (v_i, v_j, w_{ij}) \in E \vee (v_j, v_i, w_{ij}) \in E\}. \quad (1)$$

The rank of a cluster c in the neighbourhood of v_i is calculated as the total weight of neighbors carrying the cluster label:

$$r_{ic} = \sum_{C(v_j)==c, v_j \in K_i} w_{ij}, \quad (2)$$

where $C(v_j)$ is the cluster label of v_j. The algorithm stops after a certain number of iterations or if there are no changes in cluster labels between iterations. Note that, the random order makes the algorithm non-deterministic.

Algorithm 1 Chinese Whispers

```
# Initialize clusters:
for $v_i \in V$ do
    $v_c \leftarrow i$
end for
# Loop:
while label changes do
    for $v_i \in V$, randomized order do
        $v_c \leftarrow \max_c r_{ic}$
    end for
end while
```

So far, the described method leads to a crisp partitioning. To change it to a fuzzy one, Biemann proposed to normalize the rankings per node in the final step and let the results be the cluster memberships. This second version will be used in this work and further be called Fuzzy Chinese Whispers (FCW).

Another graph-based clustering algorithm is MaxMax [15]. This inherently fuzzy algorithm starts by transforming the undirected weighted graph G to a directed unweighted graph G' by treating the adjacent edge(s) with the highest weight as an incoming edge for every node and finally removing the remaining undirected edges. Also, all nodes are initially marked as *root*. Next, the algorithm loops over the nodes marked as *root* and mark its descendants as ¬*root*. Node v_j is said to be a descendant of v_i if there exists a directed path from v_i to v_j. In the end, clusters are identified by the remaining *root* nodes and its descendants. The order in which the nodes are looped determines the *root* nodes but does not change the produced clusters.

Algorithm 2 MaxMax

```
# Construct directed graph $G' = (V, E')$:
for $v_i \in V$ do
    $E' \leftarrow E' \cup \{(v_j, v_i)\}$ iff $(v_j, v_i, w_{ij}) \in E$ and
        $w_{ij} = \max_j w_{ij}$
    $v_i^{root} \leftarrow$ true
end for
# Determine root nodes:
for $v_i \in V$, $v_i^{root} =$ true do
    $v_j^{root} \leftarrow$ false if $v_j \in$ descendants$(v_i)$
end for
```

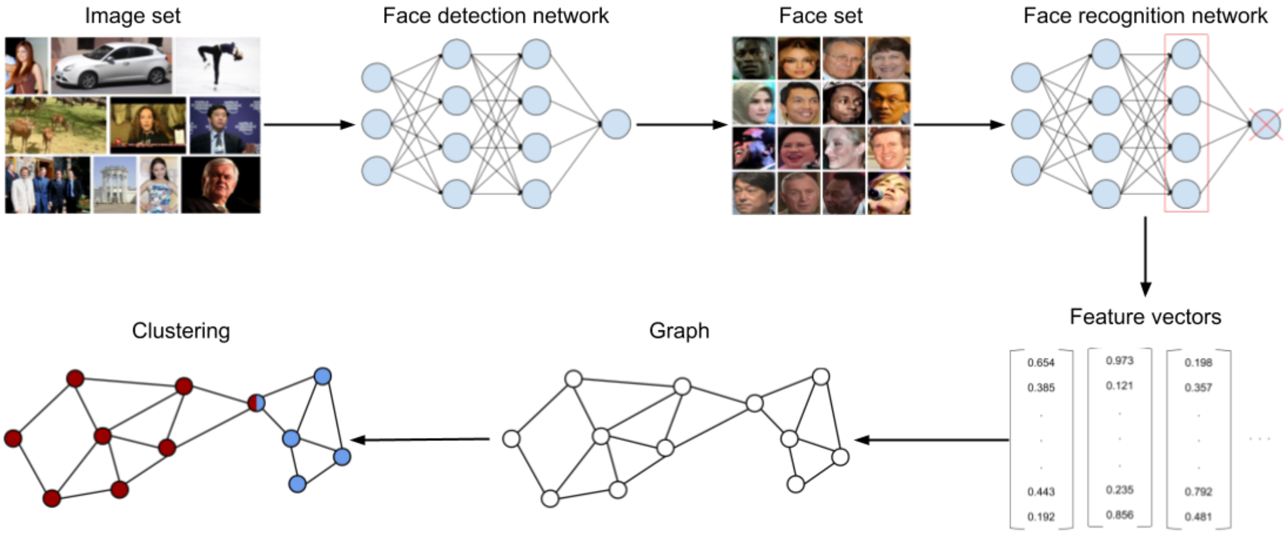


Figure 2: Visualization of our pipeline. Images contained in the illustration are from the IJB-B dataset [42].

3. Methodology

Given a set of images $O = [o_1, ..., o_m]$, the goal is to detect and partition the present faces in such a way that every group contains images of only one person. This way each group represents an identity. This process consists of three consecutive steps. In the first step, a deep learning model is used to detect the faces present in the images which are cropped and aligned for better comparison. This results in face image set $I = [i_1, ..., i_n]$ where n is the number of detected faces. Thereafter, features are extracted from the aligned face images by a second deep learning model. The returned feature vectors $X = [x_1, ..., x_n] \in \mathbb{R}^{n \times d}$ where d is the dimension of the feature vectors, are then transformed to a graph $G = (V, E)$. Finally, a graph-based algorithm is used to deliver a fuzzy partition matrix U where $u_{ij} \in U$ is the probability of x_i belonging to cluster c_j.

3.1. Face Detection and Alignment

The deep learning model responsible for the simultaneous detection and alignment of faces is the Multi-task Cascaded Convolutional Network (MTCNN) [47]. The model is a cascade of three deep convolutional neural networks (CNNs) forming a three-stage process. Before the first stage, an image pyramid is created of the input and fed to the Proposal Network. This network generates candidate windows for the detection of faces and is calibrated by bounding box regression vectors. Thereafter, non-maximum suppression is employed to merge highly overlapping windows. In the second stage, another CNN, called the Refine Network, rejects more windows as false candidates and refines the coordinates of the remaining ones. Again, non-maximum suppression is used afterwards. The third stage is similar to the second stage, but is focused on describing the face with facial landmarks, which is the output of this Output Network together with the bounding box and the probability of classifying it as a face. The resulting bounding boxes and facial landmarks are used to crop the original images and return the aligned dataset I.

3.2. Feature Extraction

Now that the images consist of aligned faces, a second deep learning model is used for the feature extraction. For that we take a pre-trained face recognition network and remove the ultimate layer to use the rest of the network as a feature extractor. Two different models are used in this work. One is the Inception-ResNet-v1 [37] which is trained on the VGGFace2 dataset [6] consisting of 3.31 million images of 9131 subjects using softmax loss. The other is the ArcFace network [8] which is trained on the union set of MS-Celeb-1M [13] which consists of 10 million images of 100k subjects, and VGGFace2 using additive angular margin loss. Taking image set I as input, either network returns feature vectors $x_i \in X$, which are 512 dimensional vectors.

3.3. Clustering

In the final phase of our pipeline the clustering takes place using a graph-based algorithm. Since the data is presented in a 512 dimensional geometric space, the first step is to transform it to an undirected weighted graph. Given this graph either Fuzzy Chinese Whispers (FCW) or α-MaxMax is used. The former is used as explained in section 2.2.2 and the latter will be explained in section 3.3.2.

3.3.1 Graph Construction

Prior to the graph construction, a $n \times n$ weight matrix W is formed representing the similarities between each feature vector x_i. Since the feature extraction network is trained with softmax loss, the similarities between vectors are based on their angular difference. Therefore, the cosine similarity measure is used which is defined as:

$$w(x_i, x_j) = \frac{x_i \cdot x_j}{||x_i||||x_j||}. \tag{3}$$

Due to normalization, the magnitudes of the face representation vectors are equal to 1, so only calculating the dot product is sufficient:

$$w(x_i, x_j) = w_{ij} = x_i \cdot x_j. \tag{4}$$

Now, graph $G = (V, E)$ is constructed by treating each feature vector x_i of face image i as a node while edges between nodes are only added if their weight w_{ij} is higher than z:

$$V = \{x_1, x_2, ..., x_n\}, \tag{5}$$

$$E = \{(x_i, x_j, w_{ij}) | i \neq j \text{ and } w_{ij} > z\}. \tag{6}$$

The threshold z determines the density of the network and has impact on the performance of the clustering algorithm which will be analysed in section 5.1.

3.3.2 α-MaxMax

We propose to adapt the original MaxMax algorithm as it currently does not include cluster membership values. Nodes that are assigned to multiple clusters are considered to be a part of the clusters equally. Moreover, it only focuses on the maximum weight per node (see Figure 3). However, in the case where weights are based on similarities between feature vectors, weights are rarely exactly equal, making the original MaxMax in the face clustering setting a hard partitioning algorithm.

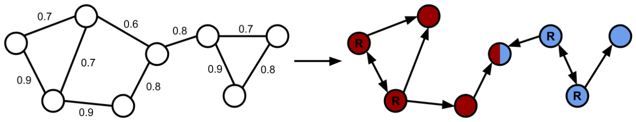


Figure 3: Example of the original MaxMax algorithm. The letter ”R” marks possible root nodes of the solution.

Our modified version includes membership values by keeping the weights in graph G'. The assignment of *root* nodes is the same as in the original algorithm (see Algorithm 2) but in α-MaxMax an extra step is taken to update the cluster memberships (see Algorithm 3). The value of u_{ji} is the probability of x_j belonging to cluster c_i which is the cluster with *root* node x_i. In the end, the fuzzy partition matrix U is normalized to satisfy $\sum_{c_i} u_{ij} = 1$ for every x_j.

Algorithm 3 α-MaxMax

```
# Construct directed weighted graph $G' = (V, E')$:
for $x_i \in V$ do
    $E' \leftarrow E' \cup \{(x_j, x_i, w_{ij})\}$ iff $(x_j, x_i, w_{ij}) \in E$
        and $w_{ij} \geq \alpha \cdot \max_j w_{ij}$
    $x_i^{root} \leftarrow$ true
end for
# Determine root nodes:
for $x_i \in V$, $x_i^{root} =$ true do
    for $x_j \in$ descendants$(x_i)$ do
        $x_j^{root} \leftarrow$ false
        $u_{ji} \leftarrow u_{ji} + w_{kj}$ where $x_k \in$ predecessors$(x_j)$
            and $x_k \in$ descendants$(x_i)$
    end for
end for
normalize$(U)$
```

Additionally, it introduces an α parameter to handle continuous weights as this is the case with cosine similarities. Multiple edges per node are transformed to incoming ones by first taking the maximum weight value $\max_j w_{ij}$ per node x_i. Next, the edges where $w_{ij} \geq \alpha \cdot \max_j w_{ij}$, are added as incoming edges, while the remaining ones are removed as in the original MaxMax algorithm. This modified version of MaxMax is called *α-MaxMax*, in which α determines the density of the directed graph.

Note that the addition of the variable α did not change the deterministic property of the algorithm. The transformation to a directed graph is always the same and even though the determination of the *root* node of a cluster can differ due to the order in which nodes are visited, the cluster itself stays the same. Also, the weights do not change resulting in equal fuzzy partition matrices when ran multiple times.

4. Evaluation data and metrics

For evaluating the performance of the proposed approach the following datasets and evaluation metrics are used.

4.1. Datasets

The Labeled Faces in the Wild dataset (LFW) [16] is a well-known and commonly used dataset for unconstrained face recognition and clustering. It consists of 13233 images in total capturing 5749 subjects and contains a large variety in images per subject, ranging from 1 to 530. Due to this characteristic and its size, it forms a ideal dataset for fine-tuning parameters of the algorithms. The model of section 3.1 is used for the detection and, more importantly, the alignment of the images.

Following its predecessor (IJB-A), the IARPA JANUS

Benchmark B (IJB-B) [42] defines eight challenges including protocols for detection and clustering and only clustering. Since this work is focused on clustering, the former protocol is not used. In contrast to the LFW dataset, the clustering protocol provides bounding boxes for the faces, removing the need for the detection and alignment module. This protocol consists of seven subtasks differing in both the number of images per subject and the total number of subjects. The benchmark is accompanied with a set of 67000 face images, 7000 face videos, and 10000 non-face images, where only the face images are used in these experiments. The face images are sampled as individual images and as frames from the face videos. Most of these face images display extreme poses and vary in image quality, making it a more challenging dataset than LFW.

(a) LFW

(b) IJB-B

Figure 4: Example images of Labeled Faces in the Wild (LFW) [16] and IARPA JANUS Benchmark B (IJB-B) [42].

4.2. Evaluation Metrics

There is no universally agreed upon performance metric for cluster analysis resulting in numerous methods testing different aspects of the clustering solution. To evaluate the clustering algorithms two measures are adopted: BCubed F-measure [2] and V-measure [31].

4.2.1 BCubed F-measure

Pairwise precision, recall and F-measure look at whether the prediction of placing pairs together in the same cluster was correct with respect to their true clusters. Precision in this case is the fraction of correctly placed pairs in the same cluster over the total number of pairs of that cluster, recall is the fraction of correctly placed pairs in the same cluster over the total number of pairs of that identity, and F-measure is the harmonic mean between precision and recall. Since the number of pairs grow quadratically with the cluster size, more emphasis is given to larger clusters in the pairwise F-measure calculation. To address this issue, Bagga and Baldwin came up with the BCubed F-measure [2], which defines precision and recall as pointwise measures. Precision becomes the fraction of points in a cluster that belong to the same identity and recall becomes the fraction of points of an identity that are in the same cluster. Intuitively, precision gives preference to clusters of the same identity while recall focuses on getting all images of an identity in the same cluster.

As defined by Aimgó et al. [1], for a feature vector x_i, $C(x_i)$ and y_i denote its predicted cluster and true cluster respectively. The correctness of a pair of feature vectors x_i and x_j is given by:

$$\mathrm{Corr}(x_i, x_j) = \begin{cases} 1 & \text{if } C(x_i) = C(x_j) \text{ and } y_i = y_j \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

Using equation (7), BCubed precision, recall and F-measure are defined as:

$$\text{Precision} = \mathrm{Avg}_{x_i}\left(\mathrm{Avg}_{x_j : C(x_i) = C(x_j)}\left(\mathrm{Corr}(x_i, x_j)\right)\right), \quad (8)$$

$$\text{Recall} = \mathrm{Avg}_{x_i}\left(\mathrm{Avg}_{x_j : y_i = y_j}\left(\mathrm{Corr}(x_i, x_j)\right)\right), \quad (9)$$

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (10)$$

4.2.2 V-measure

Rosenberg and Hirschberg came up with a conditional entropy-based evaluation measure: the V-Measure [31]. This measure consists of two complementary parts: homogeneity and completeness, just as the F-measure has precision and recall. The idea is that homogeneity is satisfied if all clusters only contain data points of the same class and completeness is satisfied when all data points of a class are in the same clusters. The two measures are roughly in opposition. For example, a solution where all data points are in the same clusters, such that there is only one cluster, completeness would be completely satisfied while homogeneity is totally neglected. The other way around would be a solution where every data points is its own cluster. Homogeneity and completeness use the same intuitions as precision and recall but differ as they use entropy and calculate directly over the whole solution instead of averaging over the pointwise scores. Formally, homogeneity and completeness are defined as:

$$\text{Hom} = \begin{cases} 1 & \text{if } H\big(C(X), Y\big) = 0 \\ 1 - \frac{H\big(Y|C(X)\big)}{H(Y)} & \text{otherwise} \end{cases}, \quad (11)$$

$$\text{Com} = \begin{cases} 1 & \text{if } H\big(C(X), Y\big) = 0 \\ 1 - \frac{H\big(C(X)|Y\big)}{H\big(C(X)\big)} & \text{otherwise} \end{cases}, \quad (12)$$

where $H(\cdot)$ is the marginal entropy, $H(\cdot|\cdot)$ is the conditional entropy and $H(\cdot,\cdot)$ is the joint entropy. Furthermore, $C(X)$ and Y are the predicted partitioning and the ground truth respectively, defined as:

$$C(X) = \{C_1, C_2, ..., C_n\}, \quad (13)$$

where $C_i = \{c_{i1}, c_{i2}, ...c_{ik}\}$ is the set of labels assigned to x_i, which is only one label ($C_i = \{c_i\}$) in the crisp case, and

$$Y = \{Y_1, Y_2, ..., Y_n\}, \quad (14)$$

where $Y_i = \{y_i\}$ is the set containing the true label of x_i. Again, V-Measure is the harmonic mean of the two:

$$\text{V-measure} = \frac{2 \cdot \text{Hom} \cdot \text{Com}}{\text{Hom} + \text{Com}}. \quad (15)$$

Another commonly used metric for (face) clustering is the Normalized Mutual Information score (NMI) which is equivalent to the V-measure as shown by Becker [3]. This allows for comparison between previously reported NMI scores with the V-measure scores produced in this work.

4.2.3 Fuzzy V-measure

The BCubed F-measure and the V-measure evaluate crisp partitionings which is satisfactory for most clustering problems. Even when looking at fuzzy algorithms these evaluation measures often suffice due to the fact that fuzzy algorithms are commonly used for crisp partitioning problems, where the cluster assignment per object is chosen by its highest membership value. However, for evaluating fuzzy partitionings another measurement is required. Adapting the original V-Measure Utt et al. proposed the Fuzzy V-Measure [39]. This evaluation works particularly for fuzzy problems, where the true membership is divided over its true classes, so point x can have labels y_1 and y_2 both with a membership value of 0.5. The idea of fuzzy face clustering is unfortunately a different kind of problem because the true partitioning is crisp but the methodology returns a fuzzy one. Therefore, to adjust for the current situation the evaluation input is adapted to compute new scores of homogeneity and completeness.

Due to the fuzziness of the solution there are more label assignments than data points resulting in a difference in size with the set of ground truth labels, so $|C(X)| > |Y|$ due to the possibility that $C(x_i)$ can contain multiple labels ($|C(x_i)| \geq 1$). To compensate for the gap, homogeneity and completeness modify the input of the predicted and true labels, both in their own way (see Figure 5). Homogeneity fills the set of truth labels with duplicates of data points that have multiple assignments in the predicted labels set:

$$Y^{hom} = \{Y_1^{hom}, Y_2^{hom}, ..., Y_n^{hom}\}, \quad (16)$$

where

$$Y_i^{hom} = \{(y_i)_{\times|C_i|}\}. \quad (17)$$

The set of predicted labels stays the same, $C^{hom}(X) = C(X)$. Completeness on the other hand reduces the size of the set of predicted labels by removing the multi-assignments to only match the ground truth or a random one if the truth label is not among the predicted ones.

$$C^{com}(X) = \{C_1^{com}, C_2^{com}, ..., C_n^{com}\}, \quad (18)$$

where

$$C_i^{com} = \begin{cases} Y_i & \text{if } y_i \in C_i \\ \{C_{ij} | j \sim U(1, |Ci|)\} & \text{otherwise} \end{cases}. \quad (19)$$

The set of true labels stays the same, $Y^{com} = Y$. This modification is consistent with the desire of not missing an image in an identity cluster regardless of its presence in other clusters. These metrics are further to be called *fuzzy homogeneity* and *fuzzy completeness* together with their harmonic mean: *fuzzy V-measure*.

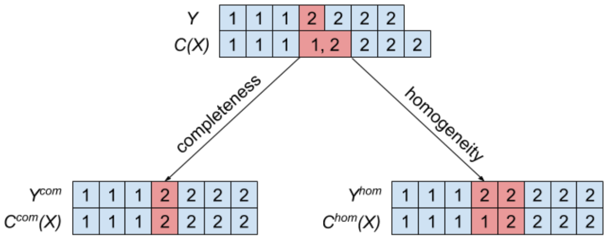


Figure 5: Conversion of the true and predicted labels for fuzzy homogeneity and fuzzy completeness.

Optionally, a membership threshold can be set to determine the minimum membership value for an image to be part of a cluster. Especially for the Fuzzy C-means algorithm, which gives every data point membership with almost every cluster, this threshold prevents evaluating all assignments.

5. Experiments

In this section, experiments are reported evaluating the proposed approach. First, different similarity threshold values used for forming the graph are analysed and results on the LFW dataset are used to choose a threshold value z for both Fuzzy Chinese Whispers (FCW) and α-MaxMax. Thereafter, the algorithms are evaluated on the IJB-B dataset and compared with baseline methods. The algorithms are also evaluated using another face representation to even out the input of the clustering algorithms and compared with state-of-the-art methods. In the end, the resulting clusters are explored to perform error analysis. It should be noted that if not explicitly stated, crisp evaluation methods are used. Only Table 2 contains fuzzy evaluation results. Also, results for non-deterministic algorithms such as Fuzzy C-means (FCM) are averaged over 10 runs.

5.1. Threshold analysis

In order to compare the clustering algorithms, parameters should be tuned first. One of the parameters is the threshold determining the minimum similarity value between images to be added as edges to the graph used by both FCW and α-MaxMax. Therefore, different values for the threshold z are compared using the V-measure.

An important aspect of the LFW dataset is the varying number of images per subject of which a large portion consists of singleton classes. Out of the 5749 people present in the dataset, only 1680 have two or more distinct photos. Such a distribution in the data is a realistic occurrence in forensic investigations due to detection of background faces. People detected in the background often appear only once and are insignificant for the investigation. Therefore, the performance on the LFW dataset with and without singleton clusters is analysed. Moreover, since the methodology of FCW and α-MaxMax differs, the threshold is set individually.

The graphs presented in Figure 6 show the V-measure scores of the FCW algorithm with threshold values between 0.5 and 0.9. Focusing on the singleton clusters, z should be set to a higher value compared to the situation when singleton clusters are removed. A higher threshold value means less edges in the graph resulting in more smaller clusters explaining the shift to the right on LFW with singleton clusters. However, in the case of forensic investigations the evaluation of singleton clusters is assumed to be insignificant. Therefore, the choice for z is based on the scores on LFW without singleton clusters and is set to 0.7 for the remainder of this paper.

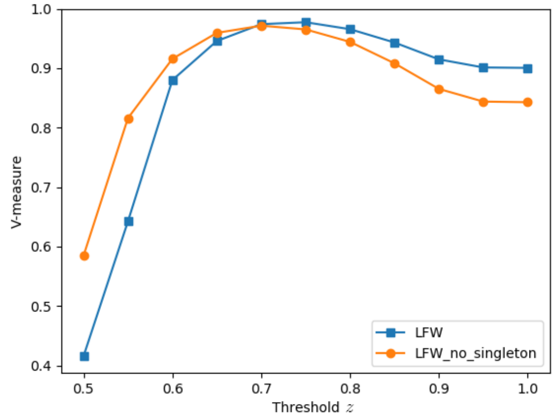


Figure 6: Performance of the FCW algorithm on the LFW dataset with and without singleton clusters.

In the case of α-MaxMax, two parameters need to be set for the optimal performance. One is the same threshold z for constructing the required similarity-based graph. The other is the α value determining the fuzziness of the algorithm. As can be seen from Figure 7, there is an optimum around $z = 0.7$ regardless of the α value. Also, the highest V-measure scores are returned by setting α to 0.95 leading to the choice of setting z to 0.7 and α to 0.95 for the remainder of this paper. Again, this choice was made on the set without singleton clusters as this will also be the set used in further experiments as the LFW dataset.

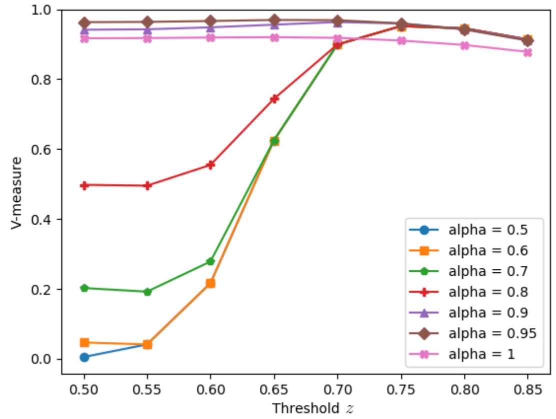


Figure 7: Performance of the α-MaxMax algorithm on the LFW dataset without singleton clusters.

5.2. Comparison with baseline methods

The BCubed F-measure and V-measure scores of both graph-based algorithms and different baseline clustering

	IJB-B-512		IJB-B-1024		IJB-B-1845	
	F	V(NMI)	F	V(NMI)	F	V(NMI)
AHC [12]	0.617	0.826	0.596	0.835	0.586	**0.840**
AP [11]	0.419	0.814	-	-	-	-
Spectral [34]	0.459	0.697	-	-	-	-
DBSCAN [10]	0.593	0.810	0.587	0.815	0.592	0.800
FCM [9, 4]	0.554	0.818	0.547	0.831	0.523	0.833
RL-FCM [45]	0.015	0.000	-	-	-	-
FCW [5]	**0.647**	**0.846**	**0.643**	**0.845**	**0.643**	**0.840**
α-MaxMax	0.575	0.831	0.556	0.833	0.565	0.835

Table 1: BCubed F-measure scores and V-measure scores of baseline methods and fuzzy graph-based methods evaluated on three subtasks of the IJB-B. The missing scores of AP and Spectral clustering could not be produced due to computational restrictions and the ones for RL-FCM were not computed due to its ineffectiveness.

	Crisp hom	Crisp com	Crisp V	Fuzzy hom	Fuzzy com	Fuzzy V
FCM [9, 4]	0.844	0.794	0.818	0.794	0.805	0.800
FCW [5]	0.863	0.830	0.846	0.821	0.844	0.832
α-MaxMax	0.878	0.788	0.831	0.881	0.793	0.835

Table 2: Crisp and fuzzy evaluation on the IJB-B dataset with 512 subjects. FCM is evaluated with a membership threshold of 0.1 and for FCW and α-MaxMax the threshold is set to 0.

methods are presented in the top part of Table 1. The generally used baseline methods are Agglomerative Hierarchical Clustering (AHC) [12] with average linkage, Affinity Propagation (AP) [11] with a damping factor of 0.5, Spectral Clustering [34] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [10] where the maximum distance between two samples to be considered as in the same neighborhood (ϵ) is set to 0.2. K-means is not evaluated due to the similarity to its fuzzy variant Fuzzy C-Means (FCM). AHC, Spectral Clustering and FCM all require a number of clusters parameter which is set to the true number of clusters per dataset. Every baseline method is evaluated using the same feature vectors as described in section 3.2 and calculate similarities based on cosine distance. AHC preforms relatively well and makes no assumptions about the data distribution, suggesting a rich variety in the distribution. On the other hand, FCW outperforms AHC while the used threshold assumes there is an optimal connectivity to use which is based on density. Also explaining the good performance of DBSCAN since this method requires the data to have similar density. Spectral clustering needs balanced data and a predefined number of clusters making the algorithm unfit for the presented face clustering problem. Furthermore, FCW consistently outperforms α-MaxMax and is therefore considered the better graph-based algorithm for crisp face clustering.

5.3. Comparison with fuzzy methods

Fuzzy C-means (FCM) [9, 4] and its Robust-Learning variant (RL-FCM) [45] are compared with the graph-based methods in the middle part of Table 1. Again, the same feature vectors as used by the graph-based methods are taken. However, both algorithms require dimensionality reduction to be done beforehand since they can not handle data with a high number of dimensions [43]. Therefore, the data is first reduced to 5 dimensions using the Uniform Manifold Approximation and Projection (UMAP) technique [27]. Although RL-FCM does not need any parameters to be set, the algorithm is incapable of handling unbalanced data and returns only one cluster. The introduced cluster weight α_j with its constraint $\sum_{j=1}^{c} \alpha_j = 1$ forces clusters to be roughly of the same size, making the algorithm unsuitable for face clustering in an unconstrained environment. Contrarily to RL-FCM, the original FCM algorithm shows decent results with V-measure scores close to those of FCW and α-MaxMax.

Beside the crisp evaluation, FCM is also compared based on the fuzzy homogeneity, completeness and V-measure as described in section 4.2.3. Table 2 presents the results on the IJB-B dataset with 512 true clusters. The idea of evaluation with fuzzy measures is to gain insight in the difference between fuzzy and crisp clusters produced by the algorithms. The expectation is that completeness would increase while sacrificing homogeneity. There are two scenarios affecting the scores. One is when data that was assigned to the wrong cluster in the crisp solution is assigned to multiple clusters, including its correct one, in the fuzzy solution. This increases completeness and even homogeneity if the extra assignment only includes the correct cluster. The second scenario is when data that is already correctly clustered

	IJB-B-512		IJB-B-1024		IJB-B-1845	
	F	V(NMI)	F	V(NMI)	F	V(NMI)
ARO [28]	0.763	0.898	0.758	0.908	0.755	0.913
PAHC* [25]	-	-	0.639	0.890	0.610	0.890
ConPaC* [35]	0.656	-	0.641	-	0.634	-
DDC [24]	0.802	0.921	0.805	0.926	0.800	0.929
GCN [41]	**0.833**	**0.936**	**0.833**	**0.942**	**0.814**	**0.938**
FCW$_{0.45}$ [5]	0.811	0.910	0.810	0.915	0.810	0.916
α-MaxMax$_{0.45,0.85}$	0.759	0.892	0.757	0.889	0.765	0.901

Table 3: BCubed F-measure scores and V-measure scores of (crisp) state-of-the-art methods and fuzzy graph-based methods evaluated on three subtasks of the IJB-B using the ArcFace features[1]. Methods marked with an asterisk (*) denote results from the original papers and the other top part results are from [41]. The subscript numbers of the graph-based algorithm represent the used values for z and α.

is assigned to other clusters. This decreases homogeneity without gaining completeness. Since the algorithms show good performance on the IJB-B dataset, it is assumed that the second scenario occurs more often than the first one, which results in a greater loss in homogeneity compared to the gain in completeness.

The initial expectation is confirmed by the results of Table 2 for FCM and FCW since both algorithms suffer a relatively high loss in homogeneity while gaining small amounts of completeness, resulting in a decrease in the V-measure score. Surprisingly, α-MaxMax gains in both homogeneity and completeness, leading to a higher V-measure score adn even surpassing that of FCW. This suggests a higher occurrence of the first scenario, making the fuzzy results of α-MaxMax better than its crisp ones. Both graph-based algorithms outperform Fuzzy C-means and do not require a number of clusters parameter, making them ideal candidates for the current use case. However, FCM produces membership values for all images with all clusters while FCW and α-MaxMax only return membership values for the edge cases.

5.4. Comparison with state-of-the-art

As mentioned in the introduction, the deep learning model used for feature extraction is partly responsible for the performance of the complete face clustering process. For this reason, another face representation produced by the ArcFace network [8] is used to compare the state-of-the-art algorithms with FCW and α-MaxMax. The results are showcased in Table 3 and are significantly higher than the ones in Table 1, showing the superiority of the ArcFace network. The scores of Proximity Aware Hierarchical Clustering (PAHC) [25] and Conditional Pairwise Clustering (ConPaC) [35] are taken from their original work and make use of another feature extraction network which explains their inferiority to the other algorithms. The scores of α-MaxMax are close to the rest of the state-of-the-art methods but is again outperformed by FCW. Deep Density Clustering (DDC) [24] produces better V-measure scores but worse BCubed F-scores compared to FCW and the Graph Convolutional Network (GCN) [41] outperforms every method.

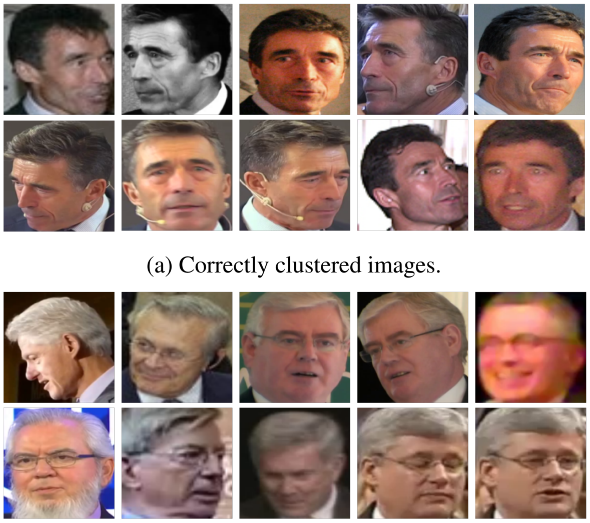
5.5. Error analysis

Exploring the results of the graph-based algorithms (see Figure 8), some interesting comments can be made. First of all, the method is greatly capable of handling invariance in illumination, pose and resolution of the image. Grayscale and color images are often correctly clustered together and difference in gender, race and age is correctly recognised as well. The method mainly focuses on distinctive features such as hair color, hair style, skin color and, the presence or absence of glasses, hats and facial hair. This is, for example, the reason for the incorrect cluster of Figure 8b. All people in this cluster have gray/white hair and most of them wear glasses. Furthermore, a large part of the incorrect clusters consists of people of Asian or African ethnicity. This can be explained by the low racial diversity in the training data of the feature extraction network. Although, the VGGFace2 dataset [6] on which the feature network is trained, contains a rich variety in pose, age and illumination, it still presents a gap in ethnicity. As presented by Wang et al. [40], the number of people of Caucasian ethnicity dominate this set with 74.2% of the amount of images. Additionally, Krishnapriya et al. [21] conclude that images of African-American people are often poorly lit, making the subjects harder to recognize. Both characteristics support the lack of performance on subject of Asian and African ethnicity.

6. Discussion and Conclusion

In this section three points of discussion are raised. The first one argues if the presented method is effective for the case of forensic investigations. The second one considers the theoretical need for the threshold used in the graph construction. The third point focuses on the influence of the

[1] https://github.com/Zhongdao/gcn_clustering



(a) Correctly clustered images.

(b) Incorrectly clustered images.

Figure 8: Example clusters produced by FCW after running on IJB-B-512.

feature extraction network on the total performance. Thereafter, future work is proposed and the final conclusions are made.

6.1. Forensic investigations

This work was constructed for the purpose of aiding in forensic investigations as automatically organizing large amounts of media can be beneficial in time-sensitive cases. Additionally, probability values are added to the assignments of images to clusters to provide insight in the confidence of the allocation. Therefore, a fuzzy clustering approach was proposed to satisfy the desired properties. Furthermore, the approach was designed to boost the completeness of the solution by increasing the possibility of assigning an image to its correct cluster while possibly sacrificing homogeneity. Lastly, the proposed method does not require the number of clusters to be known beforehand. This would be an obstacle as it is unknown how many people are contained in the images on someones device.

6.2. Threshold effect

One disadvantage of the presented methodology is the setting of threshold z which determines the density of the graph. Using this cut-off threshold compared to retaining a fully connected graph has computational benefits as both FCW and α-MaxMax have a run time linear in the number of edges. Therefore, the question arises if the threshold is necessary beside its computational time reduction. FCW and α-MaxMax use edge weights to determine node labels and therefore one could expect little change in the results when lowering the threshold, since the nodes' closest neighbors will keep dominating the choice for its label. However, both algorithms are affected by the choice for the threshold as can be seen in Figure 6 and 7. Since they are affected in different ways, the algorithms will be discussed separately.

6.2.1 Fuzzy Chinese Whispers

In FCW, the label of a node is determined by the strongest present label in its local neighborhood. This presence is measured by the total weight of adjacent nodes carrying this label. A denser graph means more neighbors per node and since the dominance of a label is counted as the total weight, a large and relatively far away cluster could become the strongest present label. This results in smaller clusters being swallowed by the larger ones, which has a domino effect because the large clusters become even larger and swallow more and more smaller clusters. In the end, there will be an equilibrium when the remaining clusters are so far away from each other that they do not have enough influence to flip the label of a node in one of the other clusters. For example, on the LFW dataset, the number of clusters that remained is 5. Regarding the computational time, it is increased in both the number of neighbors per node that need evaluation as well as the number of iterations until no label changes.

6.2.2 α-MaxMax

Lowering the threshold influences α-MaxMax less as the algorithm depends on the maximum weight per node which only changes for the singleton clusters since they did not have adjacent edges before this point. This is also the reason why lowering z is capped after a certain value as can be seen from Figure 7. The α value has no lower bound and has more influence on the amount of edges in the graph than z. Also, computational-wise, it has a greater effect due to it being related to the number of edges.

6.3. Influence of deep model

The deep learning model responsible for feature extraction has a great impact on the performance of the complete method as is shown by comparing the results displayed in Table 1 with those in Table 3. Furthermore, the results of the state-of-the-art methods on the same feature set are relatively close to each other, providing more evidence for its large share in improving performance. This makes sense since better (trained) networks project data in a more separable way, which makes it easier for all clustering methods to correctly cluster. Unfortunately, the performance differs per ethnicity which can be a sensitive point, especially in law enforcement. This can be solved by training the feature extraction network on more balanced datasets such as the one proposed by Wang et al. [40].

6.4. Future work

We identify three points for future work: applying end-to-end training, improving scalability and extending to multi-view.

6.4.1 End-to-end training

The impact of the feature network and its combined strength with the clustering algorithm give rise to the idea of an end-to-end training pipeline. So far, every developed method either tries to improve the performance of the feature extraction network or the clustering algorithm in which case the network is always trained in the supervised face recognition setting. Incorporating clustering in the training process in an end-to-end matter by using the final results as feedback for the network could prove useful. The work of Wang et al. [41] already shows the effectiveness of end-to-end learning with attention aggregation in which they train their Graph Convolutional Network with the clustering results. It is therefore expected that training the feature extraction network specifically with the output of the clustering algorithm would increase the global performance of the method.

6.4.2 Scalability

The implemented method leaves room for scalability improvements. Similarity matrix W used with graph construction grows quadratically with the number of images possibly requiring large amounts of memory. The matrix could be created as a combination of submatrices as shown by equation (20) reducing the intermediate memory required and using them subsequently for forming the graph.

$$W = \begin{bmatrix} W_{00} & W_{01} & \dots & W_{0s} \\ W_{10} & W_{11} & \dots & W_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ W_{t0} & W_{t1} & \dots & W_{ts} \end{bmatrix} \tag{20}$$

However, the size of the graph increases as well, especially with a low threshold z. Fortunately, the graph is a set of subgraphs which are not necessarily connected and could therefore be clustered separately without changing the results. The way to correctly construct the subgraphs using the submatrices is left for future work.

6.5. Multi-view clustering

Besides images, data from seized devices could also contain audiovisual material adding extra information for identification. The combination of visual and auditory data is an example of multi-view data. Wang et al. [41] used the VoxCeleb2 [7] to evaluate their method in the multi-view setting. A same approach could be taken to evaluate the clustering of the concatenated features by the proposed method. We expect that the rigidness of the threshold is going to be problematic here as different networks are needed for the two sources of data (audio and video), which probably require different threshold values just as the Inception-ResNet-v1 and ArcFace models required different values. In this case a network trained simultaneously on the visual and auditory data would be the ideal feature extractor since this network again has an optimal value for z.

6.6. Conclusion

In this paper, a fuzzy face clustering method is proposed for the purpose of assisting in forensic investigations. The approach deviates from the linking problem conventional methods try to solve and instead uses the cosine similarity as a rough estimate for link prediction and a fuzzy graph-based clustering algorithm to return the final partitioning. Additionally, an adaption of clustering algorithm MaxMax is proposed, called α-MaxMax and new evaluation measures are introduced fitting for the fuzzy face clustering case. Experiments show the effectiveness of the approach which is superior to other fuzzy algorithms and comparable with state-of-the-art crisp face clustering methods. The proposed α-MaxMax achieves 87.8% homogeneity and 78.8% completeness on the IJB-B-512 dataset for the crisp results. The scores even increase to 88.1% and 79.3% for homogeneity and completeness respectively when evaluating the fuzzy results. Meaning that the fuzzy clusters will contain more photos of the same person without including too much photos of other people compared to the crisp ones.

Acknowledgements

This research was commissioned by Delft University of Technology (TU Delft) in cooperation with the Netherlands Forensic Institute (NFI) and I would like to thank my supervisors Marius Staring (LUMC) and Huub Hoofs (NFI) for their guidance and council on both the project as well as the management of it. Additionally, I would like to thank the team of Forensic Big Data Analysis at the NFI for their ever support and advise during the project.

References

[1] Enrique Amigó, Julio Gonzola, and Javier Artiles. A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints Technical Report. *Information Retrieval*, 12(4):461 – 486, 2009.

[2] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 79–85, 1998.

[3] Hila Becker. Identification and Characterization of Events in Social Media. Technical report, Columbia University, 2011.

[4] James C Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[5] Chris Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Processing on the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, 2006.

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, pages 67–74, 2018.

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxceleB2: Deep speaker recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1086–1090, 2018.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *CVPR*, 2018.

[9] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 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*, pages 226–231, 1996.

[11] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[12] Chidananda K. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978.

[13] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *European Conference on Computer Vision*, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] David Hope and Bill Keller. MaxMax: A graph-based soft clustering algorithm applied to word sense induction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 368–381, 2013.

[16] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical report, University of Massachusetts, Amherst, 2007.

[17] Nicolaos B. Karayiannis. MECA: maximum entropy clustering algorithm. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, 1:630–635, 1994.

[18] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, pages 1–14, 2017.

[19] Frank Klawonn and Franko Höppner. What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In *Advances in intelligent data analysis V. 5th international symposium on intelligent data analysis*, pages 254–264, 2003.

[20] Joshua C. Klontz and Anil K. Jain. A case study of automated face recognition: The Boston marathon bombings suspects. *Computer*, 46(11):91–94, 2013.

[21] K S Krishnapriya, Kushal Vangara, Michael C King, V Albiero, and Kevin Bowyer. Characterizing the Variability in Face Recognition Accuracy Relative to Race. *arXiv:1904.07325*, 2019.

[22] Raghu Krishnapuram and James M. Keller. A Possibilistic Approach to Clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[23] Yann LeCun, Patrick Haffner, Lon Bottou, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. *Shape, Contour and Grouping in Computer Vision*, 1681:319–345, 1999.

[24] Wei-An Lin, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa. Deep Density Clustering of Unconstrained Faces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8128–8137, 2018.

[25] Wei An Lin, Jun Cheng Chen, and Rama Chellappa. A Proximity-Aware Hierarchical Clustering of Faces. In *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, pages 294–301, 2017.

[26] Stuart P Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.

[27] Leland McInnes and John Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426*, 2018.

[28] Charles Otto, Dayong Wang, and Anil K. Jain. Clustering Millions of Faces by Identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):289–303, 2018.

[29] N.R. Pal, K. Pal, and J.C. Bezdek. A mixed c-means clustering model. In *IEEE International Conference on Fuzzy Systems*, pages 11–21, 1997.

[30] Nikhil R. Pal, Kuhu Pal, James M. Keller, and James C. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 13(4):517–530, 2005.

[31] Andrew Rosenberg and Julia Hirschberg. V-Measure : A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, number June, pages 410–420, 2007.

[32] Enrique H Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, 1969.

[33] Swami Sankaranarayanan, Azadeh Alavi, Carlos D. Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *2016 IEEE 8th International*

Conference on Biometrics Theory, Applications and Systems, BTAS 2016, pages 1–8, 2016.

[34] Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 22(8):888–905, 2000.

[35] Yichun Shi, Charles Otto, and Anil K. Jain. Face Clustering: Representation and Pairwise Constraints. *IEEE Transactions on Information Forensics and Security*, 13(7):1626–1640, 2018.

[36] B. Scott Swann. FBI Video Analytics Priority Initiative. In *17th Annual Conference & Exhibition on the Practical Application of Biometrics*, 2014.

[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI*, pages 4278–4284, 2017.

[38] David M.J. Tax and Robert P.W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, 1999.

[39] Jason Utt, Sylvia Springorum, Maximilian Köper, and Sabine Schulte. Fuzzy V-Measure An Evaluation Method for Cluster Analyses of Ambiguous Data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 581–587, 2014.

[40] Mei Wang, Weihong Deng, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. *arXiv:1812.00194*, 2018.

[41] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage Based Face Clustering via Graph Convolution Network. *CVPR*, 2019.

[42] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K. Jain, James A. Duncan, Kristen Allen, Jordan Cheney, and Patrick Grother. IARPA Janus Benchmark-B Face Dataset. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:592–600, 2017.

[43] Roland Winkler, German Aurospace, and Frank Klawonn. Fuzzy C-Means in High Dimensional Spaces. *International Journal of Fuzzy System Applications*, 1(1):1–16, 2011.

[44] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to Cluster Faces on an Affinity Graph. *CVPR*, 2019.

[45] Miin Shen Yang and Yessica Nataliani. Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. *Pattern Recognition*, 71:45–59, 2017.

[46] Lotfi A. Zadeh. Fuzzy Sets, 1965.

[47] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

Supplementary Materials: Fuzzy Face Clustering For Forensic Investigations

A. Dimensionality Reduction

The returning vectors of the deep learning model responsible for feature extraction are fairly high (512 dimensions) making clustering in this high dimensional space computational costly and certain algorithm such as Fuzzy C-means (FCM) [9, 4] can not handle more than 10 dimensions [43].

In the field of clustering analysis subspace clustering is a commonly used tactic for clustering in high dimensional spaces. Subspace clustering is a way to project data to lower dimensional subspaces and cluster in the according spaces. Thereafter the cluster results are combined to return the found clusters in the original space. The idea comes from the assumption that some features contain little information and are less discriminative than others. Moreover, it is argued that some features are more relevant for certain features than others. However, since the embeddings are the result of a neural network trained for discriminated people based on facial appearance, it is assumed that every feature carries equal weight.

A.1. Principal Component Analysis

Beside subspace clustering there are also more general approaches, such as basic feature selection and feature extraction methods like Principal Component Analysis (PCA) [9]. Feature selection uses the same assumption as subspace clustering by choosing a subset of features seen as relevant and use only these features to cluster. On the other hand, PCA which aims to find the projection with the highest variance in the data, also has a philosophy of unequal importance of features.

The behaviour of PCA is analysed on a subset of the Labeled Faces in the Wild dataset (LFW) [16] consisting of all subjects (143) with at least 10 photos in Figure 1. In the figure two distinct clusters can be identified but the rest is too overlapping to logically partition. Adding extra principal components results in more separable clusters but reduces the desired effect of dimensionality reduction.

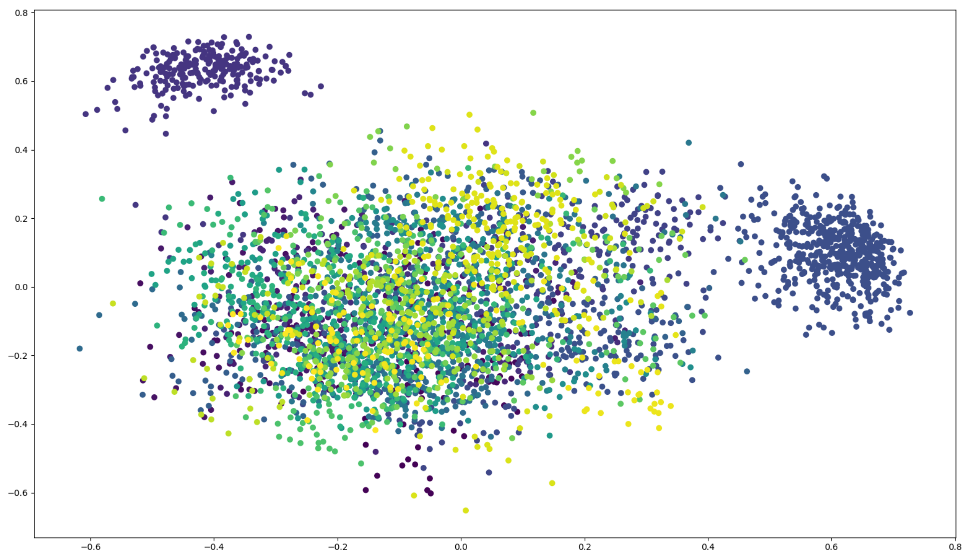


Figure 1: Visualization of the first two principal components of PCA analysis on the extracted features of a subset of LFW.

A.2. t-SNE and UMAP

Instead of PCA two other dimensionality reduction techniques were used to examine the clustering effect. A popular approach is the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [11]. t-SNE focuses on local relationships between data points and is able to capture non-linear structures. It creates a probability distribution (Gaussian) defining the relationships in the high dimensional space and recreates it in a low (2 or 3) dimensional one following the probability distribution. In the second step the algorithm uses a Student t-distribution, explaining the name of the approach. The embeddings are optimized using gradient descent based on the KL-divergence between the probability distributions of the original high dimensional space and the newly formed low dimensional space. A note to make is that the loss function is non-convex making the technique non-deterministic.

The other approach for dimensionality reduction is to use the Uniform Manifold Approximation and Projection (UMAP) technique [27]. With a strong mathematical foundation the algorithm is competitive with t-SNE and has superior run time performance. Also, UMAP scales better with the amount of data allowing the embedding of larger data sets than are feasible for t-SNE. Finally, UMAP can embed to spaces larger than 3 dimensions which t-SNE can not. Without going into the details of the algorithm, UMAP starts by creating a local fuzzy simplicial set per data point. Such a set is constructed by taking the k approximate nearest neighbours, calculate their normalized distance and translate the metric space to a simplicial set through the exponential of the negative distance. The union of the sets forms a topological representation of the high dimensional data. A similar process can be used to construct an equivalent topological representation of the target lower dimensional data. UMAP optimizes in a same way as t-SNE does, only in this case the goal is to minimize the cross-entropy between the topological representations by adjusting the layout of the lower dimensional data representation.

As can be seen from Figure 2 both mappings of t-SNE and UMAP present well-separated clusters, where they mainly differ in density. If dimensionality is applied in the pipeline it should be mentioned and passed to the clustering algorithm that similarity in the new lower dimensional space is measured as Euclidean distance instead of cosine.

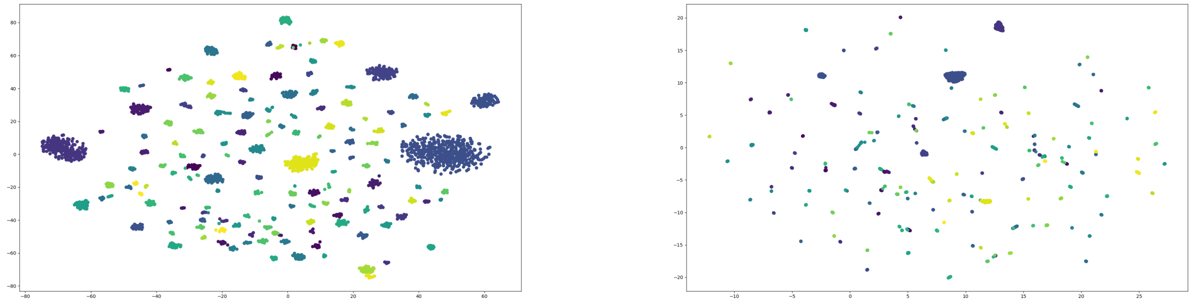


Figure 2: Visualization of t-SNE (left) and UMAP (right) results on the same set as used in Figure 1.

A.3. Results

To choose the most effective dimensionality reduction technique required for the FCM algorithm results using either t-SNE or UMAP with different amounts of dimensions are compared using the BCubed F-measure [2] and the V-measure [31]. The scores of FCM algorithm are shown in Table 1 with $c = c_{true} = 5749$ and $m = \frac{2+d}{d}$ where d is the number of dimensions after reduction.

		FCM		FCW		α-MaxMax	
		F	V(NMI)	F	V(NMI)	F	V(NMI)
Original data	512d	-	-	**0.904**	**0.974**	**0.887**	**0.973**
t-SNE	2d	**0.706**	0.914	0.770	0.938	0.792	0.949
	3d	0.699	0.917	0.702	0.912	0.768	0.929
UMAP	2d	0.627	0.891	0.632	0.908	0.672	0.915
	3d	0.640	0.908	0.649	0.917	0.675	0.922
	5d	0.682	**0.919**	0.776	0.938	0.789	0.934
	10d	0.690	0.915	0.787	0.940	0.802	0.937

Table 1: BCubed F-measure and V-measure scores of the fuzzy algorithms evaluated on the complete LFW dataset.

The scores are relatively close to each other and although the reduction with t-SNE to 2 dimensions results in the highest BCubed F-score the choice is made for the reduction with UMAP to 5 dimensions. This choice is also based on the computational time since reduction with UMAP is around 6 times faster for the complete LFW dataset and scales better as well in both amount of data and number of dimensions.

The effect of dimensionality reduction on the performance of graph-based algorithm is evaluated as well. Scores for Fuzzy Chinese Whispers (FCW) [5] and the adapted MaxMax [15] algorithm called α-MaxMax are reported in Table 1 alongside the scores for FCM. For both algorithms the optimal values for z and α are taken which differ per reduction. However, the scores are significantly worse compared to the ones in the original space. For this reason and that dimensionality reduction takes time the graph-based algorithms do not make use of this technique.

B. MaxMax vs α-MaxMax

The proposed α-MaxMax is an adapted version of its original MaxMax [15] algorithm. During the transformation of undirected input graph G to directed graph G' the α value is used as a margin to increase the number of incoming edges per node. In the original version of the algorithm, only the edge with the maximum weight is transformed to a directed edge which make sense in the field of Natural Language Processing since the weights often represent co-occurrences of words. However, in the case of face clustering weights represent the similarities between images which are continuous often in the range between 0 and 1. Therefore, applying MaxMax would result in every node having a single incoming edge as there is one maximum weight per node. Subsequently, every node only belongs to one cluster, namely the same as its predecessor resulting in a hard partitioning output.

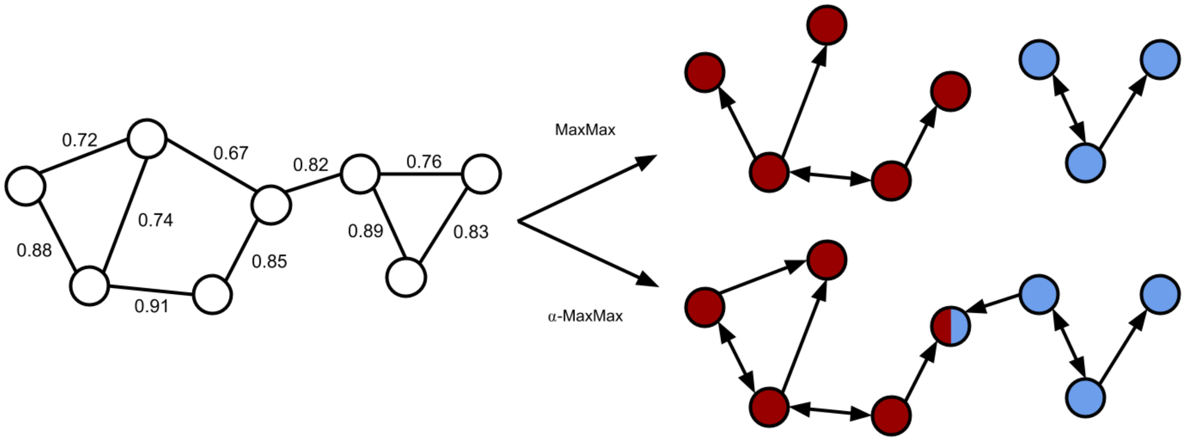


Figure 3: Toy example of a graph and the results produced by MaxMax and α-MaxMax with $\alpha = 0.95$.

The α parameter addresses this issue by adding a margin to increase the range in which weights are transformed to directed edges. Instead of being the maximum weight ($\max_j w_{ij}$) an edge is added if its weight is greater than $\alpha \cdot \max_j w_{ij}$. Figure 3 shows an example graph with continuous weights and the solution produced by MaxMax and α-MaxMax. As can be seen in the top right part of the figure, MaxMax transforms the graph in such a way that every node has one incoming edge which eventually leads to a crisp partitioning. The lower right part demonstrates the effect of the α parameter resulting in the possibility of multiple incoming edges per node, leading to a fuzzy partitioning.

The α parameter controls the density of the graph which impacts both the performance as well as the computational time of the algorithm. When α is decreased the computational time increase since MaxMax' complexity is linear in the number of edges [15] and a lower α means more incoming edges per node raising the total number of edges. The performance behaves differently as can be seen from Table 2 and suggests there is an optimum around 0.95 which is also presented by Figure 7 from the paper.

	α	F	V(NMI)
MaxMax	1	0.770	0.944
α-MaxMax	0.95	**0.887**	**0.973**
	0.9	0.878	0.967
	0.8	0.796	0.915

Table 2: BCubed F-measure and V-measure scores of MaxMax and α-MaxMax with different α values on the complete LFW dataset. The threshold z is set to 0.7 for both algorithms.

MaxMax is developed as a fuzzy algorithm and the fact that the scores of $\alpha = 0.95$ are higher than those of $\alpha = 1$ shows its intended use. Instead of equal weights, as is the case in the field of word sense induction for example, weights in the case of face clustering are rarely exactly equal. Therefore, α-MaxMax treats weights that are remotely close to each other as ”equal”. With the right α value a graph can be constructed for which the algorithm is intended. However, setting α too low results in a denser graph and consequently results in less clusters. For example, setting α to 0.5 would result in one cluster when taking the toy example graph of Figure 3.

Concluding, MaxMax was developed for a different field requiring the introduction of the α parameter to transfer it to the field of face clustering and mimic its intended use.

C. IARPA JANUS Benchmark C

For this research the IARPA JANUS Benchmark B is used for evaluating the proposed approach. The choice for this dataset beside its challenging nature was made for the ease of comparing it with state-of-the-art methods since these methods use the same dataset. However, Maze et al. [7] improved upon their previous version and released the IARPA JANUS Benchmark C advancing the goal of robust unconstrained facial analysis. The new set not only contains more data and variability but also introduces end-to-end protocols, combining face detection and 1:N identification. The clustering protocol is redistributed into 4 subtasks with 32, 1021, 1839 and 3531 subjects.

For the purpose of comparability with future work the graph-based algorithms used in this work are evaluated on the clustering protocol of this new dataset. The BCubed F-measure [2] and V-measure [31] are used again and the scores are reported in Table 3.

	IJB-C-32		IJB-C-1021		IJB-C-1839		IJB-C-3531	
	F	V(NMI)	F	V(NMI)	F	V(NMI)	F	V(NMI)
FCW	0.802	0.896	0.697	0.864	0.682	0.852	0.664	0.836
α-MaxMax	0.802	0.897	0.621	0.860	0.609	0.814	0.587	0.795

Table 3: BCubed F-measure and V-measure scores of Fuzzy Chinese Whispers (FCW) and α-MaxMax on the different IJB-C clustering subtasks.

Based on the results displayed in Table 3 FCW is preferred over α-MaxMax as it outperforms α-MaxMax in both evaluation measures. However, looking at the fuzzy scores of Table 4 on the IJB-C-1021 dataset, α-MaxMax returns a better fuzzy partitioning even outperforming the crisp scores of FCW.

	Crisp hom	Crisp com	Crisp V	Fuzzy hom	Fuzzy com	Fuzzy V
FCW	0.857	0.872	0.864	0.368	0.876	0.518
α-MaxMax	0.889	0.833	0.860	0.892	0.839	0.865

Table 4: Fuzzy homogeneity, completeness and V-measure scores of Fuzzy Chinese Whispers (FCW) and α-MaxMax on the IJB-C-1021 clustering subtask.

References

[1] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 79–85, 1998.

[2] James C Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[3] Chris Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Processing on the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, 2006.

[4] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[5] David Hope and Bill Keller. MaxMax: A graph-based soft clustering algorithm applied to word sense induction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 368–381, 2013.

[6] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical report, University of Massachusetts, Amherst, 2007.

[7] Brianna Maze, Jocelyn Adams, James A. Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K. Jain, W. Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA janus benchmark-C: Face dataset and protocol. *Proceedings - 2018 International Conference on Biometrics, ICB 2018*, pages 158–165, 2018.

[8] Leland McInnes and John Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426*, 2018.

[9] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[10] Andrew Rosenberg and Julia Hirschberg. V-Measure : A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, number June, pages 410–420, 2007.

[11] Laurens J. P. van der Maaten and Geoffrey E. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[12] Roland Winkler, German Aurospace, and Frank Klawonn. Fuzzy C-Means in High Dimensional Spaces. *International Journal of Fuzzy System Applications*, 1(1):1–16, 2011.