University of Groningen

Research internship

A comparison of clustering algorithms for face clustering

Author:
A.F. Bijl *(2581582)*

Supervisors:
A.Sobiecki
Dr M.H.F.Wilkinson

Groningen, the Netherlands,
July 24, 2018

Abstract

Video surveillance methods become increasingly widespread and popular in many organizations, including law enforcement, traffic control and residential applications. In particular, the police performs investigations based on searching specific people in videos and in pictures. Because the number of such videos is increasing, manual examination of all frames becomes impossible. Some degree of automation is strongly needed.

Face clustering is a method to group faces of people into clusters containing images of one single person. In the current study several clustering algorithms are described and applied on different datasets. The five clustering algorithms are: k-means, threshold clustering, mean shift, DBSCAN and Approximate Rank-Order. In the first experiments these clustering techniques are applied on subsets and the whole Labeled Faces in the Wild (LFW) dataset. Also a dataset containing faces of people appearing in videos of ISIS is tested to evaluate the performance of these clustering algorithms.

The main finding is that threshold clustering shows the best performance in terms of the f-measure and amount of false positives. Also DBSCAN has shown good performance during our experiments and is considered as a good algorithm for face clustering. In addition it is discouraged to use k-means and for large datsets Approximate Rank-Order when clustering faces.

CONTENTS

1 INTRODUCTION

Video surveillance methods become increasingly widespread and popular in many organizations, including law enforcement, traffic control and residential applications. In particular, the police performs investigations based on searching specific people in videos and in pictures. Because the number of such videos is increasing, manual examination of all frames becomes impossible. Some degree of automation is strongly needed.
Clustering is a technique to divide a set of objects in different groups or clusters resulting in each cluster having identical objects and different clusters contain objects with different characteristics. Face clustering is the process of grouping the faces of people present on a set of photos or videos. Ideally this process results in each person having his/her own cluster containing images of this particular person. By doing this we can answer certain questions such as who and how many different people were present in a particular photo or video?
As shown in the next chapter face clustering is an active field of research. Different approaches to face clustering exist using different models and cluster techniques. Various representations of a face can be obtained by using different models. Applying different cluster algorithms on these models can result in numerous clusterings. Each cluster algorithms has its own specifications and different parameters which must be given in order to cluster the data. Comparing these different cluster techniques and evaluating their performances is needed to obtain the best possible clustering of a set of faces. In the current study different clustering algorithms as threshold clustering, k-means, mean shift, DBSCAN and Approximate Rank-Order are applied and compared on the well-known model FaceNet. Besides a quantitative evaluation also a qualitative evaluation is obtained by an exhibition of the falsely clustered images. These evaluations help to gain more insight in clustering algorithms in order reach a good clustering of a set of faces which is applicable in real world applications.
The following project compares the performance of a number of clustering techniques. This is done by applying these techniques on different datasets and evaluating their performances. The research goal is to evaluate this performance of different clustering algorithms by different experiments. The following section describes the pipeline of face clustering and further describes state of the art clustering techniques. In section 3 the set-up of the experiments that compare different clustering algorithms is discussed. Section 4 discusses the data sets used in the experiments. The findings of the experiments are summarized and discussed in section 5. A conclusion on the findings is provided in section 6.

2 RELATED WORK

In this chapter state of the art techniques and methods of face clustering are described. First a short and general introduction about the visual pattern recognition pipeline is given. Then several state of the art clustering techniques are described.

2.1 *Visual pattern recognition pipeline*

Ideally the process of face clustering starts with a set of images which must be clustered. Then faces of these images must be extracted, recognized and clustered to a cluster containing all images of that person. A common approach to implement this clustering is using a visual pattern recognition pipeline. The pipeline for face clustering is shown in figure 1 and consists of three steps:

Segment data: every face that appears on an image must be segmented and separately send to the following step. Faces are extracted from the original image and must be separately processed in the following step. This first step is often called face detection because it mainly involves the detection of faces on an image.

Extract features: a mathematical representation of an image can be extracted in order to classify the faces. This mathematical structure is called a embedding and there are several methods to create these embeddings. A face embedding is a multidimensional numerical vector representation of a face.

Classify feature vector: in this step the actual classification or clustering is performed. On each embedding different techniques can be applied to cluster faces to clusters.

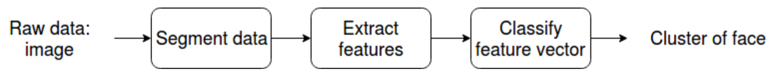


Figure 1: Visual pattern recognition pipeline

2.2 *MTCNN Face Detection*

One method to detect faces from images is called Multi-Task Cascaded Convolutional Networks (MTCNN) as proposed by Zhang et al. [1]. MTCNN consists of three stages and uses a neural network for each stage. The input of these pipeline is an image and the output is a part of this image where the face is detected. Zhang et al. compare the performance of MTCNN against state of the art face detection models and show better performance than their competitors.

2.3 *FaceNet*

A recent paper [2] proposed a model called FaceNet and was introduced by Schroff, Kalenichenko, and Philbin. Facenet creates a 128 dimensional long embedding containing coordinates in Euclidean space. The goal of this model is to have a minimal distance between faces of the same person and to have a maximal distance between faces of different persons. The model is trained by a deep convolutional network using a triplet-based loss function. The triples loss function considers three images: an anchor image ($f(x_i^a)$), an image of the same person as the anchor (positive: $f(x_i^p)$) and an image of a different person than the anchor (negative: $f(x_i^n)$). Given these three images one wants to satisfy the following rule:

$$jjf(x_i^a) \quad f(x_i^p)jj_2^2 + a < jjf(x_i^a) \quad f(x_i^n)jj_2^2,$$

$$8f(x_i^a), f(x_i^a), f(x_i^a)\ 2\ t$$

where a is a margin that is enforced between positive and negative pairs. t is the set of all possible triplets in the training set and has cardinality N. In other words, the distance between the anchor and a positive image incremented with a margin a must be smaller than the distance between the anchor and negative image for all combinations of images.
Satisfying this property for all triplets would lead to a time consuming job, therefore a subset of triplets is chosen. This is done by selecting the triplets that violate the triplet constraint given above. In order to ensure fast convergence a batch of images is selected and for this batch the violations are selected.

2.4 *Clustering techniques*

As mentioned in the introduction clustering is the process of grouping objects with the same characteristics into one group and with different characteristics into another group. As a result, objects within the same cluster have approximately identical characteristics, meaning that each single person has its own cluster. When an image of a person that already has its own cluster is evaluated it must be clustered to that specific cluster. Clustering is applied on the mathematical (or vectorial) representations of each face. From these embeddings the difference can be obtained in terms of the distance between two face embeddings. The distance can be calculated by taking the Euclidean distance between two embeddings. The Euclidean distance between two vectors $p = fp_1, p_2, \ldots, p_{128}g$ and $q = fq_1, q_2, \ldots, q_{128}g$ both having 128 values is given by:

$$d(p,q) = \sqrt{(q_1 \quad p_1)^2 + (q_2 \quad p_2)^2 + \quad + (q_{128} \quad p_{128})^2}$$

For clustering multiple approaches with varying performance exist. The performance can strongly vary when different parameters are given to the algorithms. The goal of this research is to evaluate different cluster algorithms which are described below.

2.4.1 *Threshold clustering*

The clustering approach suggested in [2] is threshold clustering. The embedding of a new face is evaluated against already clustered faces. The distance between two faces is calculated by the Euclidean distance given above. When the distance between a new face and its nearest neighbor in the set that already is clustered is smaller than a threshold specified by the user; the face is added to the existing cluster. If a face does not have a distance below the threshold; there is no match and a new cluster must be created.
The value of the threshold plays an important role in this approach. On one hand specifying a high threshold results in many false positives: a pair of faces having a distance below the threshold but obtained from different persons. On the other hand, specifying a low threshold results in many false negatives: a pair of faces having a distance above threshold but from the same person. Therefore during the experiment it is crucial to specify this parameter with care in order to obtain the desired output.

2.4.2 *k-means*

Another clustering method is that of *k*-means originally published by Lloyd [3] where new examples are assigned to certain prototypes. Each of these

k prototypes represents a cluster and is randomly initialized. K-means performs multiple iterations where oduring each iteration all examples are assigned to the closest prototype and updated to the mean of their clusters. The algorithm stops when prototypes do not change significantly anymore. K-means depends strongly on the amount of prototypes k specified by the user. A large k reduces the average distance between examples and their assigned prototypes but is time-consuming and results in a large amount of clusters. A large amount of clusters introduces false negatives. A small k is fast converging but has a large distance between examples and their assigned prototypes. A small k therefore introduces false positives.

2.4.3 *Mean shift*

Another approach is a clustering technique described by Comaniciu and Meer [4]. In this approach each embedding is represented in the Euclidean space. The underlying distribution is estimated by an approach called kernel density estimation. This works by placing a kernel on each point in the data set and moving each point towards its direction of change. Given a candidate sample x_i the update rule for iteration t is:

$$x_i^{t+1} = x_i^t + m(x_i^t)$$

where $m()$ is the mean shift vector that is computed for each sample that points towards a region of the maximum increase in the density of points. This mean shift vector is computed by the following equation:

$$m(x_i) = \frac{\sum_{x_j e N(x_i)} K(x_j \quad x_i) x_j}{\sum_{x_j e N(x_i)} K(x_j \quad x_i)}$$

where $N(x_i)$ is the neighborhood of samples within a given distance around x_i.
The advantage of this approach is that it is a non-parametric algorithm because it does not make assumptions about the data. For example (in contrast with k-means) the amount of clusters (or prototypes) is not specified.

2.4.4 *DBSCAN*

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jrg Sander and Xiaowei Xu in 1996 [5]. A characteristic of DBSCAN is that it is a non-parametric approach as well. One does not have to specify the exact amount of clusters in advance. Instead, given a set of datapoints (or embeddings), DBSCAN groups together points that lay close to each other based on the Euclidean distance. The DBSCAN algorithm requires two parameters: the minimum distance between two points that can be grouped together and the minimum points to form a dense region. In case of face clustering the minimum points to form a dense region must be 1, by setting this, a face with no close neighbors can be a cluster on its own. The distance between two points that can be grouped together can vary and the best value must be obtained during the experiments.

2.4.5 *Approximate Rank-Order*

The Rank-order algorithm proposed by Zhu, Wen, and Sun [6] is a form of agglomerative hierarchical clustering, using a nearest neighbor based distance measure. The algorithm starts with each embedding in its own cluster and starts to merge the two closest clusters. The distance between two clusters is considered as the minimum distance between any two samples in the clusters.
The distance metric used in Rank-order clustering is given by:

$$d(p,q) = \sum_{i=1}^{O_p(q)} O_q f_p(i)$$

where $f_p(i)$ is the i-th face in the neighbor list of p, and $O_q(p)$ gives the rank of face p in face qs neighbor list. This asymmetric distance function is then used to define a symmetric distance between two faces, p and as:

$$D(p,q) = \frac{d(p,q) + d(q,p)}{min(O_p(q), O_q(p))}$$

This symmetric distance is low for two faces that have a high value on each others list and when they have neighbors in common. Clustering is then performed by initializing each embedding to its own cluster and then merging all clusters between a given threshold. In the following iteration the distances are updated for the merged clusters and the procedure continues till no further clusters can be merged.

The rank-order algorithm is in complexity $O(n^2)$ which is undesirable. Therefore an approximation of this algorithm is proposed by Otto, Wang, and Jain [7]. In the approximation rank-order algorithm only the top k-neighbours are taken into account rather than the complete list of neighbors. This approach makes the actual rank of neighbors irrelevant because the importance is shifted towards the presence/absence of shared nearest neighbors. Therefore the distance function becomes:

$$d_m(p,q) = \mathop{\text{å}}\limits_{i=1}^{min(O_p(q),k)} = I_p(O_q(f_p(i)), k)$$

where $I_q(x,k)$ is an indicator function with a value of 0 if face x is in face q's top k nearest neighbors, and 1 otherwise. The combined modified distance measure is defined as:

$$D_m(p,q) = \frac{d_m(p,q) + d_m(q,p)}{min(O_p(q), O_q(p))}$$

We have presented state of art clustering techniques, each having its own approach and characteristics. In the experiments we select the parameters for these clustering techniques and test the performance of these. The next section will present the set-up of these experiments.

3 METHODS

In order to evaluate the cluster techniques introduced in the previous section experiments are conducted. In this section the set-up and framework of these experiments are described.

3.1 *Set-up*

In our experiments faces of people are clustered by identity using different clustering algorithms. These clustering algorithms are applied on the vectorial representation of these faces, the so-called embeddings. The first step is to translate images of faces to these embeddings. As described in the previous section MTCNN is a face detection method that indicates faces on images. FaceNet is a model that maps these faces to an embedding vector. A combination of these two techniques is chosen in the experiments to extract the embeddings. On top of these embeddings multiple clustering algorithms are implemented.

3.2 *Framework*

A well-known public available implementation of FaceNet is published by David Sandberg [1]. This program is written in Python and Tensorflow and enables one to train a model on a given dataset. Also pre-trained models are available and are used for this experiment. A shared-memory Dell R815 Rack Server with four 16-core AMD opteron processors and RAM memory of 512GB was used in the tests. The program works as follows:

> Load images: a folder containing images to be clustered must be given, this folder is then loaded into the program.
>
> Detect faces: faces on the images in the folder must be detected. This is done by the MTCNN implementation in FaceNet based on the model of Zhang et al. [2].
>
> Create embeddings: for each detected face an embedding is calculated by an existing model. In the experiment a pre-trained model '20180402-114759' is used. This model is trained on another dataset [8] having 3.31 million images of 9131 subjects. As shown by David Sandberg this model results in a higher precision than existing models and therefore it is used during our experiment. Instead of vector of 128 this model calculates an embedding of 512 values. To illustrate these embeddings a Principal Component Analysis (PCA) plot [9] in 2d is shown in figure 2. This plot shows the embeddings of randomly selected images of 3 persons. PCA is a method to reduce the dimensionality of large vectors but preserving the variance in each dimension. In figure 2 it can be observed that the variance between different persons is preserved enabling the clustering of these persons.
>
> Create distance table: a distance table is created by calculating the Euclidean distance between two embeddings. In the distance table each row contains the distance from a single face to all other extracted faces.

The next step is to perform the clustering of the embeddings. The algorithms described in the previous section are applied on the embeddings resulting in a cluster for each image. The program assigns a cluster to each face in the image folder and move this image to its corresponding cluster. The result is a folder containing the resulting clusters with their images of faces.

[1] `https://github.com/davidsandberg/facenet`
[2] `https://github.com/kpzhang93/MTCNN_face_detection_alignment`

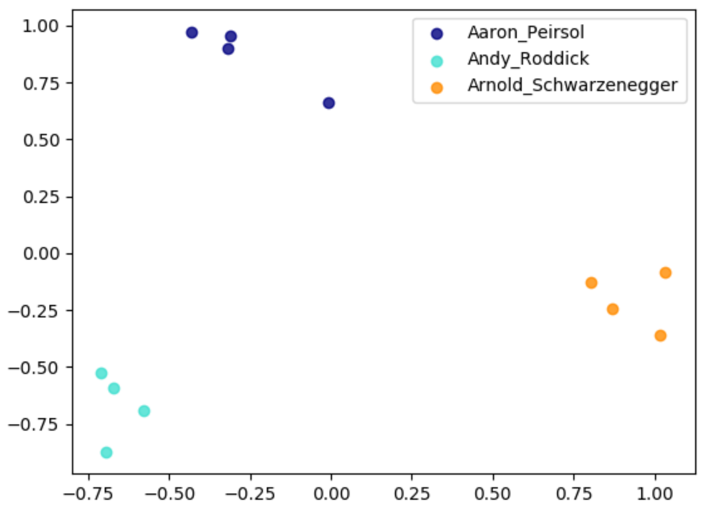


Figure 2: A PCA plot of 12 images

4 DATA

In the experiments clustering algorithms are tested on several datasets. In the following section these datasets are described.

4.1 *Labeled Faces in the Wild*

The Labeled Faces in the Wild (LFW) [10] is a dataset collected by researches of the University of Massachusetts. This dataset enables one to test methods on a labeled database. Labeling means that the identity of the person on an image is known. An example is given in figure 3 where we have an image of Aaron Peirsol with as file name `Aaron_Peirsol_0001.jpg`. The LFW dataset contains 13233 images of 5749 people where 1680 persons have 2 or more images. This makes the LFW suitable for evaluating different cluster algorithms.

Figure 3: An image in the LFW dataset of Aaron Peirsol

In our experiment multiple subsets of the LFW are evaluated in order to find potential differences in size of dataset. Therefore each dataset contains a different amount of images:

> The first (relatively) small subset contains all people whose name starts with an A. This subset contains 1052 images of 432 persons, making this dataset useful to indicate an initial performance of the clustering algorithms.
>
> For the second experiment a larger subset of the LFW is used containing 4936 images of 2013 persons. This dataset contains the images of people whose name start with a letter between A and G in the alphabet. By applying the clustering algorithms on this subset we have additional information on the performance of each clustering algorithm.
>
> The third experiment is applied on the whole LFW dataset. As described above the LFW dataset contains 12233 images of 5749 distinct persons. This experiments results in the final measures of the performance on this dataset.

4.2 *ISIS dataset*

In our last experiment the clustering algorithms are applied on a dataset containing faces of ISIS members. These images are extracted from Youtube videos published by ISIS and therefore have a lower resolution. The dataset contains 55 images of 11 persons. Each person has 5 images which makes the dataset balanced. Below the dataset is shown by figure 4. The evaluation can be done in the same manner as described above because the dataset is (manually) labeled.

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

(11) (12) (13) (14) (15) (16) (17) (18) (19) (20)

(21) (22) (23) (24) (25) (26) (27) (28) (29) (30)

(31) (32) (33) (34) (35) (36) (37) (38) (39) (40)

(41) (42) (43) (44) (45) (46) (47) (48) (49) (50)

(51) (52) (53) (54) (55)

Figure 4: Dataset containing images of ISIS members

5 results

In this section the outcomes of the experiment are described. In order to compare the algorithms the evaluation of the resulting clusters is discussed. Then results on the three datasets from the Labeled Faces in the Wild (LFW) are discussed. Increasing the length of an embedding is a possible improvement and is measured and evaluated in both of the rst datasets. The section ends with describing results of applying clustering algorithms on the ISIS dataset.

5.1 Evaluation

To express the performance of different clustering algorithms various evaluation scores can be obtained. One such a measure of the quality of clusters is the pairwise F-measure and is used in our experiments.

5.1.1 F-measure

In order to de ne the F-measure it is necessary to state several other de nitions. Consider two sets of labels: L and C. The set $L = \{l_1, l_2, ..., l_n\}$ contains the actual labels for each face used in the clustering. Set $C = \{c_1, c_2, ..., c_n\}$ is the output of the clustering algorithm for each face. With these de nitions we can make the following de nitions:

The amount of true positives (TP) or sensitivity is compromised of face pairs (i, j) that are correctly clustered to the same cluster. The amount of true positives is given by:

$$TP = |(i, j)| \text{ where } c_i = c_j \text{ and } l_i = l_j$$

The amount of false positives (FP) is compromised of face pairs (i, j) that are incorrectly clustered to the same cluster. The number of false positives is given by:

$$FP = |(i, j)| \text{ where } c_i = c_j \text{ and } l_i \neq l_j$$

The amount of true negatives (TN) is compromised of pairs that are correctly clustered to a different cluster. The number of true negatives is given by:

$$TN = |(i, j)| \text{ where } c_i \neq c_j \text{ and } l_i \neq l_j$$

The amount of false negatives (FN) is compromised of face pairs (i, j) that are incorrectly clustered to different clusters. The number of false negatives is given by:

$$FN = |(i, j)| \text{ where } c_i \neq c_j \text{ and } l_i = l_j$$

The pairwise precision (P) is de ned as the fraction of pairs that are correctly clustered to the same cluster (TP) over all pairs that were actually clustered to the same cluster by the clustering algorithm ($TP + FP$). Therefore the pairwise precision is given by:

$$P = \frac{TP}{TP + FP}$$

The pairwise recall (R) is de ned as the fraction of pairs that are correctly clustered to the same cluster (TP) over all pairs that are of the same cluster ($TP + FN$). The pairwise recall is given by:

$$R = \frac{TP}{TP + FN}$$

Finally the F-measure can be de ned as:

$$F = 2 \ \frac{P \ \ R}{P + R}$$

The F-measure effectively quali es the nal clustering performed by a clustering algorithm. If a clustering algorithm creates a single cluster for each single face the precision is high but the recall extremely low. As a consequence the F-measure results in poor performance. In case that a clustering algorithm creates a single cluster containing all faces, the recall is high but the precision is low. The F-measure also indicates a poor performance in this case.
One goal of our clustering is to cluster with care, meaning that we want to cluster faces only when we are strongly convinced that images contain the same person. An undesirable effect of the f-measure is that is can possibly increase when the amount of false positives increases. Instead we want to reduce the amount of false positives. Therefore a limit is set on the amount of false positives; this can not be higher than 1% of the amount of pictures in the dataset.

As described above the f-measure is obtained by evaluating true or false positives and true or false negatives. With the fact that the datasets are labeled this can be done in an simple manner. Each image has a le name containing the name of the person and a number which makes the le name unique. We can obtain all le names by scanning the image folder to a list of images. By removing the unique numbers in the le names, these only consist of the name of a person. By this manipulation of le names we can evaluate each single pair of faces by checking the le names and the assigned labels. The outcomes of each combination are shown in table 1.

	Same cluster	Different cluster
Same le name	TP	FN
Different le name	FP	TN

Table 1: Outcomes of different combinations of le names and clusters

5.1.2 ROC curve

A plot to visualize the performance of a cluster method is called Receiver Operating Characteristic (ROC) curves. These plots show the performance for a varying parameter resulting in different performance for each clustering algorithm. For example the performance of k-means over a varying amount of k can be measured. An example of a ROC curve is given below in Figure 5. A 2D ROC curve has two axis with a range from 0 to 1, the x-axis contains the false positive rate (FPR) and the y-axis contains the true positive rate (TPR). These rates are given by:

$$TPR = R = \frac{TP}{TP + FP}$$

$$FPR = \frac{FP}{FP + FN}$$

The best result in the ROC space is a point in the top left corner because this represents a perfect clustering: a high value for the true positive rate and a low value for the false positive rate. In the example of gure 5 the red and blue line correspond to the performance of two cluster algorithms. As a reference the black line is given corresponding to randomly guessing in a binary classi cation (only two classes). This is useless in our methods because we do not have two clusters but many more.

5.2 Results on Labeled Faces in the Wild

The results on different subsets of the LFW are described below.

Figure 5: An example of a ROC curve

5.2.1 Small subset

Several state of the art clustering techniques are applied on the subset of the LFW. Each clustering algorithms has a speci c parameter which can vary resulting in different outcomes of the clustering. In this rst experiment we evaluate for each algorithm the actual process and the value of parameters.

threshold clustering

Threshold clustering calculates the distance between a pair of images and assigns it to the same cluster when this distance is lower than a given threshold. The algorithm has different performance over various values of this threshold. Figure 11a contains the f-measure on the LFW subset with the threshold from 0 to 2 showing a peak at approximately 0.8. At the left side of this peak we see a low and strict threshold clustering each image to its own cluster. Here no distance between a pair of images has a value below the threshold. In gure 12a we can see these points at the bottom left, no clusters have more images than 1 resulting in a false positive rate and true positive rate of 0. Increasing the threshold makes larger clusters which can be seen as the peak in the f-measure and the top left values in the ROC. A threshold having a value at the right of the peak in the f-score shows the extreme case where only one cluster exists. This results in the highest values of true positive and false positive rates in the ROC. A look at at the maximum shows that the highest f-measure is given by a threshold of 0.83 but has an amount of 120 false positives. To satisfy the requirement that the amount of false positives cannot be larger than 1% of the database the threshold of 0.73 is chosen which gives a f-measure of 0.87.

k-means

The k-means algorithm initializes several prototypes which are updated to their nearest neighbors. The k amount of prototypes can vary and the f-measure over the amount of k is plotted in gure 11b. One can see that the maximum f-measure of 0.74 is obtained for $k = 80$. The ROC curve in gure 12b shows the performance over this varying k. In the top right corner we see the value for only one cluster with a maximum of true positives and false positives. The value of the false positive rate decreases for a higher k but at the end this will result in each picture having its own cluster which is also not a desirable output. Therefore a value in between has the best possible clustering. Unfortunately k-means has a bad performance in terms of the amount of false positives, this amount is not smaller than 10 for this dataset with $k > 1$ and therefore the value of the best f-score is taken as the best amount.

mean shift

For the mean shift algorithm the bandwidth of the kernel can be speci ed. Figure 11c shows the f-measure as a result of a bandwidth range from 0.1 till 1.0. The maximum f-measure of 0.93 is obtained for a bandwidth of 0.76 having an amount of false positives above 10. Therefore the value of

0.73 is reported resulting in a f-measure of 0.92 and only 6 false positives. Looking at the ROC curve in gure 12c one can see that this varies extremely. Therefore we can conclude that changing the bandwidth has a high impact on the performance of mean shift.

dbscan

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) has two parameters which are speci ed as follows: the distance between two examples that can be considered as a close region and the amount of examples that can be considered as a close region. The LFW subset has several folders containing only one face as described in the related work, therefore the amount of examples that can be considered as a close region is 1. The distance can vary and gure 11d shows the f-measure as a result of a range of this distance. The highest f-measure is obtained for distance is 0.74 resulting in a f-score of 0.90 but also an amount of false positives higher than 10. A distance of 0.73 has only 1 false positive and also a value of 0.90 therefore this value is taken to evaluate. In gure 12d the ROC curve can be seen. Values at the bottom left correspond to a low value of the distance and each face has its own cluster. With increasing the distance, faces are clustered together. Around the value of 0.9 the top left values can be seen. Increasing the distance makes even less and large clusters where in the extreme only one cluster exists. This case can be seen in the top right.

approximate rank -order

The Approximate Rank-Order algorithm has two parameters which can be speci ed. The rst is amount of neighbors that are used in subsets of nearest neighbors. The second is the threshold corresponding to the percentage of overlapping neighbors. In order to approach the best values for both parameters, we rst see how different amount of neighbours affect the f-score. We set the threshold on 0.5, meaning that 50% of both subsets must overlap. The f-score of this experiment is shown in gure 6, the maximum f-measure is obtained for k = 47, meaning that both lists contain at most 47 closest neighbors. After k = 47 increasing k does not have a signi cant better f-score. Therefore this k is taken to nd the value of a threshold that yields the highest f-score. The f-score for k = 47 and the ROC curve are respectively shown in gure 11e and 12e. We can see that the highest value is obtained for 0.5. This value can be seen in the top left of the ROC curve. Unfortunately, the amount of false positives is high for these values but having an amount below 10 gives a large drop in the f-measure. Therefore the highest f-measure is reported, but it is interesting to nd out whether for a larger dataset the amount of false positives will drop. If this is not the case Approximate Rank-Order does not satisfy a low amount of false positives.

Figure 6: F-measure by Approximate Rank-Order with varying k

To summarize the performance of the clustering algorithms table 2 is given by listing the maximal value of the f-measure for each clustering. The rst observation that can be made is that non-parametric clustering algorithms such as mean shift, DBSCAN and threshold clustering have the best performance on this subset. These algorithms cluster faces together based on distances between their embeddings which are successive on this subset. The Approximate Rank-Order algorithm has a similar performance to the non-parametric clustering algorithms. Potentially an interesting observation can be made to compare the performance on a larger dataset to this performance. K-means has the worst performance, this was expected because the LFW subset is unbalanced.

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	432	0	1.00	1.00
Threshold clustering	0.87	598	1	0.99	0.77
k-means	0.74	80	4205	0.64	0.87
Mean shift	0.92	548	6	0.99	0.85
DBSCAN	0.90	572	1	0.99	0.82
Approximate Rank-Order	0.90	420	739	0.91	0.89

Table 2: Performance of different clustering algorithms on the small LFW subset with an embedding of length 512

possible improvement

A possible improvement to the previous experiment is to use larger embeddings. A method to increase the embeddings is to ip each image horizontally, extract embeddings from these and to add these embeddings to the original embeddings of this image. The result is having double sized embeddings, increasing the distance between images. The shapes of the plots for each method stay roughly the same so therefore only the table is shown below. The parameters for each algorithm resulting in the best performance and shown in the table are the following:

threshold clustering: the threshold is 1.07

k-means: amount of k is 98

Mean-shift: the distance is 1.02

DBSCAN: the bandwidth is 1.06

Approximate Rank-Order: amount of neighbors is 47 and threshold is 0.5

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	432	0	1.00	1.00
Threshold clustering	0.88	573	4	0.99	0.79
k-means	0.74	98	3152	0.68	0.80
Mean shift	0.91	561	1	0.99	0.84
DBSCAN	0.90	556	9	0.99	0.82
Approximate Rank-Order	0.88	414	673	0.92	0.85

Table 3: Performance of different clustering algorithms on the small LFW subset with an embedding of length 1024

Comparing table 3 to table 2 shows similar behaviour on this dataset but certain observations can be made. The rst observation is that an increase in size of the embeddings results in a slightly worse f-score for mean shift, DBSCAN and Approximate Rank-Order. These algorithms had the highest f-scores for the embeddings with size 512 and are negatively effected by an increase in length of the embeddings. This is not the case for threshold clustering, comparing both f-scores shows an increase for embeddings with size 1024. The second observation which can be made is that an increase in size reduces the false positives for k-means and Approximate Rank-Order, which still have a large amount of false positives. Also intuitively this makes sense because adding more features in a vector increases the distance between two vectors. Therefore the distance between two different faces is increased and clustered in a later stadium. With these two observations taken into account we cannot say which embedding size performs better, therefore no conclusions are drawn and both performances are tested on the following dataset.

qualitative evaluation

Apart from the quantitative analysis above, a qualitative analysis of the clusters is useful to gain more insight in the clustering process. The k-means and Approximate Rank-Order algorithms have shown bad quantitative performance in the previous sections and therefore the following algorithms are taken into account: threshold clustering, DBSCAN and mean shift. The number of false positives are counted above but nding out which images are falsely clustered together helps to indicate the performance of these algorithms. The requirement to not have more than 10 false positives is set which makes the amount of false positive relatively low. With evaluating the mentioned three algorithms for both 512 and 1024 values long embeddings we have only have three common mistakes.

1. The cluster shown in gure 7 is a cluster created by each algorithm. The reason for this is that these images contain the same person instead of two different persons as indicated by the labels. Also the LFW indicates this as an error on their webpage. As a result each clustering method clusters these images together which adds a false positive in the evaluation. Looking at the performance tables we indeed see each clustering method have one false positive which is this known error in the dataset.

(a) Andrew Gilligan 1 (b) Andrew Caldecott 1

Figure 7: A cluster containing images of Andrew Gilligan and Andrew Caldecott

2. The cluster shown in gure 8 is a cluster that is clustered by mean shift with an embedding of size 512 and DBSCAN with an embedding of size 1024. The cluster contains images of two different persons: ve images of Alan Greenspan and and one of Alain Ducasse. The most likely reason that these images are clustered together is simply because these two persons look like each other due to the fact that both persons have a light skin, grey hear and wear round glasses. As a consequence both embeddings or feature vectors can have similar values which make the embeddings close to each other and likely to be clustered.
3. Figure 9 shows a cluster created by the algorithms of by threshold clustering and DBSCAN on embeddings of size 1024. This clusters

(a) Alan Greenspan 1 (b) Alan Greenspan 2 (c) Alan Greenspan 3

(d) Alan Greenspan 4 (e) Alan Greenspan 5 (f) Alain Ducasse 1

Figure 8: A cluster containing images of Alan Greenspan and Alain Ducasse

contains three images of Anna Kournikova and one of Akiko Morigami who are both tennis players. The images are taken from both tennis players in action wearing a white cap during the game. Therefore the reason is again the similarity in embeddings of these images, resulting in a small distance between these images which makes them likely to cluster.

(a) Anna Kournikova 3 (b) Anna Kournikova 7 (c) Anna Kournikova 8

(d) Akiko Morigami 1

Figure 9: A cluster containing images of Anna Kournikova and Akiko Morigami

5.2.2 Large subset

Applying the clustering techniques on the small dataset enabled us to gain insight in each clustering technique. In order to achieve the best clustering on different subsets we increase the amount of images for this second experiment. This subset contains 4936 different images of 2013 persons. The plots extracted on the dataset above show suf cient understanding of what really happens behind the clustering, therefore only tables containing values of performance are shown for this dataset. The k-means algorithm has shown a lower f-score than the other clustering techniques and is not taken into account on this dataset. Therefore the results includes four clustering techniques: threshold clustering, mean shift, DBSCAN and Approximate Rank-Order. In the previous section no solid conclusions were drawn from increasing the embedding size, therefore the algorithms are applied on both embedding sizes 512 and 1024. With applying the clustering algorithm on this dataset there is a large difference between a high f-measure and low amount of false positives. A low amount of false positives is chosen as the primary goal because we prefer to cluster less but certain images instead of many and possibly uncertain images.

512 embeddings

The following parameters are used to obtain the values in table 4 :

Threshold clustering: the threshold is 0.53.

Mean shift: the bandwidth is 0.42.

DBSCAN: the distance is 0.42.

Approximate Rank-Order: the most strict threshold already includes too many false positives for k = 47, therefore the lowest possible value of 0.01 is used.

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	4935	0	1.00	1.00
Threshold clustering	0.27	3842	10	0.99	0.15
Mean shift	0.24	4447	8	0.99	0.14
DBSCAN	0.24	4515	8	0.99	0.14
Approximate Rank-Order	0.06	3596	242	0.96	0.03

Table 4: Performance of different clustering algorithms on the large LFW subset with an embedding of length 512

1024 embeddings

The following parameters are used to obtain the values in table 5:

Threshold clustering: the threshold is 0.69.

Mean shift: the bandwidth is 0.58.

DBSCAN: the distance is 0.59.

Approximate Rank-Order: amount of neighbors is 47 and threshold is 1.0.

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	4935	0	1.00	1.00
Threshold clustering	0.08	4474	50	0.99	0.05
Mean shift	0.12	4701	10	0.99	0.06
DBSCAN	0.14	4708	10	0.99	0.08
Approximate Rank-Order	0.49	1730	26020	.074	0.37

Table 5: Performance of different clustering algorithms on the large LFW subset with an embedding of length 1024

Comparing tables 4 and 5 provides more information about the embedding size. We can conclude that increasing the embedding size does not result in better clustering. Every clustering algorithm results in a higher f-measure using embeddings with size 512 than embeddings with size 1024. Therefore we reject the possible improvement of increasing the embedding size shown in table 5.
From table 4 we can conclude that threshold clustering performs results in the highest f-measure. Another striking observation is that the algorithms

mean shift and DBSCAN show identical performance. A reason could be that both algorithms follow a comparable approach and therefore cluster faces in a same manner. The results of Approximate Rank-Order has the highest f-measure for this dataset but also an extremely high amount of false positives. A small amount of false positives below 1% of the dataset could not be obtained, therefore the best found f-measure was reported.

qualitative evaluation

Also for this dataset a qualitative look at the actual clusters gains more insight in the performance of the clustering algorithms. Looking at the false positives resulted from a threshold of 0.53 shown in gure 10 one can see that no actual false positive exist for this threshold. Each image labeled as a false positive is a consequence of an incorrect label or another person on the image. Therefore we consider this value of a threshold as a safe clustering which does not result in actual false positives. Potentially the value of this threshold can even be increased.

(a) Chok Tong Goh 1
(b) George W Bush 477
(c) Emmy Rossum 1
(d) Eva Amurri 1

(e) Gonzalo Sanchez de Lozada 10
(f) Carlos Savedra 1
(g) Doc Rivers 1
(h) Glenn Rivers 1

(i) Dai Chul Chyung 1
(j) Chyung Dai Chul 1
(k) Doug Duncan 1
(l) Charles Moose 3

(m) George W Bush 108
(n) Colin Powell 47
(o) Colin Powell 49
(p) Colin Powell 189

Figure 10: False positives clustered by threshold clustering

5.2.3 Whole dataset

To obtain the whole set of embeddings, the dataset was splitted in batches with a maximum size of 5000 images. All embeddings were attached in the same manner to enable the clustering. As described in the previous experiments, results with a higher f-meaurse were obtained but included also many false positives. The following parameters are used to obtain the values in table 6:

Threshold clustering: the threshold is 0.49.

Mean shift: the bandwidth is 0.38.

DBSCAN: the distance is 0.40.

Approximate Rank-Order: the most strict threshold already includes too many false positives for k = 50, therefore the lowest possible value of 0.01 is used.

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	4935	0	1.00	1.00
Threshold clustering	0.21	11578	130	0.99	0.12
Mean shift	0.14	12759	47	0.99	0.08
DBSCAN	0.18	12652	130	0.99	0.09
Approximate Rank-Order	0.13	9642	702	0.96	0.07

Table 6: Performance of different clustering algorithms on the whole LFW dataset with an embedding of length 512

As shown in table 6 the highest f-measure is obtained by threshold clustering. A signi cant different between mean shift and DBSCAN is also obtained: DBSCAN has a higher f-measure of 0.04 than mean shift. Although these algorithms are much more time consuming compared to threshold clustering. Approximate Rank-Order has a large amount of false positives so is considered as a bad clustering on the whole dataset.
A qualitative analysis on the whole dataset includes an analysis of too many false positives and clusters. Therefore it is not included in this research.

5.3 ISIS dataset

In our last experiment clustering algorithms are applied on the ISIS dataset. This dataset contains images of poor resolution and quality which could have different performance by the clustering algorithms. As a result of the poor performance in the rst two experiments k-means and Approximate Rank-Order are excluded from this experiment. Making only threshold clustering, mean shift and DBSCAN applied on this dataset. Also only embeddings with size 512 are calculated because the horizontal ipping the images did not show better results
When creating the embeddings from the original images the MTCNN model could not detect faces from certain images. These images were removed from the dataset because otherwise the program would crash. Four images were removed namely image 22-25 from gure 4. The reason that the model could not detect faces on the image could be too low resolution or that the picture is taken from a large angle making it hard to locate a face. The clustering algorithms are applied on the remaining 51 images and the scores are obtained from these 51 images.
The quantitative results of the three algorithms are shown in table 7. The actual clustering by each algorithm is shown by gures 13, 14 and 15 in the appendix. It can be concluded that on this dataset mean shift results in a perfect clustering. Each cluster contains the original ve images of this person. Also DBSCAN performs well resulting in an almost perfect clustering. A likely reason for this could be that the dataset is highly balanced, which is better for clustering techniques as mean shift and DBSCAN. Threshold clustering results also in a high f-measure but is lower than the other two algorithms. The values used in each algorithm are the following:

Threshold clustering: the threshold is 0.88

Mean shift: the bandwidth is 0.76

DBSCAN: the distance is 0.77

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	11	0	100	1.00
Threshold clustering	0.94	14	0	100	0.89
Mean shift	1.00	11	0	100	1.00
DBSCAN	0.99	12	0	100	0.97

Table 7: Results on ISIS dataset

6 conclusion

In the experiment multiple clustering techniques are applied on the Labeled Faces in the Wild (LFW) dataset and on the ISIS dataset. The main ndings for each clustering method are described below:

k-means: this relatively simple approach to cluster faces has shown poor performance on the rst experiment. Compared to the other algorithm k-means resulted in the lowest f-measure. Also the fact that k-means needs the amount of clusters as an input is not desirable in a set of faces where the amount of different persons is unknown. In this light k-means was not further included for the other experiments and we consider this algorithm as a bad choice to cluster faces.

Threshold clustering: this is the most simple and fastest algorithm because only a distance has to be calculated and compared to a threshold. In the rst experiment this algorithm gives a high f-measure but still better f-measures by other algorithms were obtained. In the second experiment threshold clustering resulted in the highest f-measure and as shown in the qualitative evaluation has no actual false positives. Therefore the value of threshold can even be increased to reduce the amount of clusters. In the third and largest experiment we obtained a slightly lower value of threshold. Therefore we conclude that the value depends on the amount of images. For a large dataset as all the images in the LFW dataset the value of threshold is 0.49. But as shown in the second experiment on the large LFW dataset these images still could be valid results. Therefore we consider 0.50 as a very safe threshold which could be increased when looking at the false positives. This all makes threshold clustering the most encouraged algorithm to cluster a set of faces of the LFW dataset. On the ISIS dataset threshold clustering performed less than other algorithms. A possible reason could be that threshold clustering has worse performance on images with lower resolution. But based on this research no actual conclusions can be drawn about this.

Mean shift: this clustering algorithm has shown the highest f-measure in the rst and last experiments. In the second experiment applied on the larger dataset both DBSCAN and threshold clustering resulted in a better f-measure. Although DBSCAN resulted in only hardly noticeable better results. In the third experiment on the whole LFW dataset, DBSCAN has a signi cant higher f-measure. Therefore we can conclude that mean shift is a proper method to cluster small datasets as shown in the rst experiment and in the experiment on the ISIS dataset. The value of the bandwidth which in both experiments resulted in the best clusterings is 0.76. Still on larger dataset mean shift results in less performance compared to DBSCAN and threshold clustering. Despite the fact that the running time is not taken into account, this would be a disadvantage because it takes considerably more time to perform a mean shift approach compared to threshold clustering and even DBSCAN.

DBSCAN: this clustering algorithm has shown comparable performance to the mean shift approach. In particular on the large dataset used in experiment 2 the performance is similar to mean shift, where DBSCAN clustering is better in terms of running time. In the third experiment DBSCAN has shown a better f-measure than mean shift. On the ISIS dataset DBSCAN resulted in an almost perfect clustering. With this all in mind, it is concluded that DBSCAN is a clustering technique that performances well on face clustering. Although on the large dataset threshold clustering results in better performance. The value of the minimum distance strongly depends on the amount of images. In the rst and last experiments using small datasets this value is much larger than in the experiments on the larger LFW subsets.

Approximate Rank-Order: in all experiments the actual performance described by Otto et al. [7] was not obtained. Approximate Rank-Order has similar f-measures for the small dataset but an extremely worse f-measure for the large datasets. A possible reason for this could be the model used to create the embeddings. Also the Approximate Rank-Order algorithm resulted in many false positives which is highly undesirable when clustering unknown faces. Therefore the Approximate Rank-Order in combination with the used model to create embeddings is not encouraged to use for face clustering.

Potential future and additional work could be to measure performance using different models. Models could use different manners of creating embeddings resulting in other performance which can be new and even better than results obtained in this research. Also examining the performance of Approximate Rank-Order to approach the results in the original paper by Otto et al. [7] can be potential work. Although Approximate Rank-Order by de nition is more complex than simple and fast threshold clustering encouraged in this research.

7 appendix

(a) Clustering using a threshold

(b) k-means

(c) Mean shift

(d) DBSCAN

(e) Approximate rank order

Figure 11: The f-scores of the different clustering algorithms on a small dataset containing 1052 images

(a) Clustering using a threshold

(b) k-means

(c) Mean shift

(d) DBSCAN

(e) Approximate rank order

Figure 12: The ROC curves of the different clustering algorithms on a small dataset containing 1052 images

(1) (2) (3) (4) (5)

(6) (7) (8) (9) (10) (11) (12)

(13) (14) (15) (16) (17) (18) (19) (20) (21)

(22) (23) (24) (25) (26) (27) (28) (29)

(30) (31) (32) (33) (34) (35) (36)

(37) (38) (39) (40) (41) (42) (43) (44) (45) (46)

(47) (48) (49) (50) (51)

Figure 13: Clusters of the ISIS dataset clustered by threshold clustering

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

(11) (12) (13) (14) (15) (16) (17) (18) (19) (20)

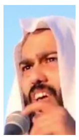
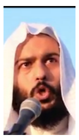
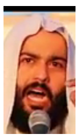
(21) (22) (23) (24) (25) (26) (27) (28) (29) (30)

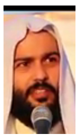
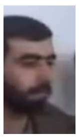
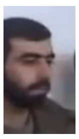
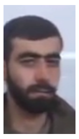
(31) (32) (33) (34) (35) (36) (37) (38) (39) (40)

(41) (42) (43) (44) (45) (46) (47) (48) (49) (50)

(51) (52) (53) (54) (55)

Figure 14: Clusters of the ISIS dataset clustered by mean shift

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

(11) (12) (13) (14) (15) (16) (17) (18) (19) (20)

(21) (22) (23) (24) (25) (26) (27) (28) (29) (30)

(31) (32) (33) (34) (35) (36) (37) (38) (39) (40)

(41) (42) (43) (44) (45)

(46) (47) (48) (49) (50) (51) (52) (53) (54) (55)

Figure 15: Clusters of the ISIS dataset clustered by DBSCAN

REFERENCES

[1] K. Zhang et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23.10 (Oct. 2016), pp. 1499–1503. ISSN: 1070-9908. DOI: 10.1109/LSP.2016.2603342.

[2] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CoRR* abs/1503.03832 (2015). arXiv: 1503.03832. URL: http://arxiv.org/abs/1503.03832.

[3] Stuart P. Lloyd. "Least squares quantization in pcm". In: *IEEE Transactions on Information Theory* 28 (1982), pp. 129–137.

[4] Dorin Comaniciu and Peter Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.5 (May 2002), pp. 603–619. ISSN: 0162-8828. DOI: 10.1109/34.1000236. URL: http://dx.doi.org/10.1109/34.1000236.

[5] Martin Ester et al. "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231. URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

[6] Chunhui Zhu, Fang Wen, and Jian Sun. "A Rank-order Distance Based Clustering Algorithm for Face Tagging". In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 481–488. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995680. URL: http://dx.doi.org/10.1109/CVPR.2011.5995680.

[7] Charles Otto, Dayong Wang, and Anil K. Jain. "Clustering Millions of Faces by Identity". In: *CoRR* abs/1604.00989 (2016). arXiv: 1604.00989. URL: http://arxiv.org/abs/1604.00989.

[8] Qiong Cao et al. "VGGFace2: A dataset for recognising faces across pose and age". In: *CoRR* abs/1710.08092 (2017). arXiv: 1710.08092. URL: http://arxiv.org/abs/1710.08092.

[9] Jonathon Shlens. "A Tutorial on Principal Component Analysis". In: *CoRR* abs/1404.1100 (2014). arXiv: 1404.1100. URL: http://arxiv.org/abs/1404.1100.

[10] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.