Article

Comparative Analysis of Supervised and Unsupervised Approaches Applied to Large-Scale “In The Wild” Face Verification

Tomasz Hachaj * and Patryk Mazurek

Institute of Computer Science, Pedagogical University of Krakow, 2 Podchorążych Ave, 30-084 Krakow, Poland; patryk.mazurek@up.krakow.pl

* Correspondence: tomekhachaj@o2.pl

Received: 29 September 2020; Accepted: 2 November 2020; Published: 5 November 2020

Abstract: Deep learning-based feature extraction methods and transfer learning have become common approaches in the field of pattern recognition. Deep convolutional neural networks trained using triplet-based loss functions allow for the generation of face embeddings, which can be directly applied to face verification and clustering. Knowledge about the ground truth of face identities might improve the effectiveness of the final classification algorithm; however, it is also possible to use ground truth clusters previously discovered using an unsupervised approach. The aim of this paper is to evaluate the potential improvement of classification results of state-of-the-art supervised classification methods trained with and without ground truth knowledge. In this study, we use two sufficiently large data sets containing more than 200,000 “taken in the wild” images, each with various resolutions, visual quality, and face poses which, in our opinion, guarantee the statistical significance of the results. We examine several clustering and supervised pattern recognition algorithms and find that knowledge about the ground truth has a very small influence on the Fowlkes–Mallows score (FMS) of the classification algorithm. In the case of the classification algorithm that obtained the highest accuracy in our experiment, the FMS improved by only 5.3% (from 0.749 to 0.791) in the first data set and by 6.6% (from 0.652 to 0.718) in the second data set. Our results show that, beside highly secure systems in which face verification is a key component, face identities discovered by unsupervised approaches can be safely used for training supervised classifiers. We also found that the Silhouette Coefficient (SC) of unsupervised clustering is positively correlated with the Adjusted Rand Index, V-measure score, and Fowlkes–Mallows score and, so, we can use the SC as an indicator of clustering performance when the ground truth of face identities is not known. All of these conclusions are important findings for large-scale face verification problems. The reason for this is the fact that skipping the verification of people’s identities before supervised training saves a lot of time and resources.

Keywords: clustering; deep learning; face classification; facenet; Fowlkes–Mallows score; Silhouette Coefficient

1. Introduction

Mobile devices supply users with the possibility of instantly taking photos and uploading them to social media platforms. Every day, millions of new photos depicting everyday situations become available on the Internet. Among them are images containing human faces of unknown identity. The human face is a widely used biometric modality for revealing the identity of a person. In spite of a great deal of research on face recognition, it remains a challenging issue [1]. In real-life scenarios, due to a large amount of data, face verification systems deal with unlabeled images in which identities

of the people in the images are initially unknown. The typical approach to solve the problem of face verification is to train a classification algorithm that can “learn” to assign an appropriate class label (identity) to a given face. The most popular and effective classification algorithms, such as neural networks, support vector machines, and k-nearest neighbors require ground truth data for the training procedure. If the ground truth of training data is not known, we can undertake one of two possible solutions to find it: we can manually or semi-manually group persons in training data by identity or we can use unsupervised approaches based on clustering. In case of supervised classifier training, the manually or semi-manually labeled data set is commonly considered more reliable than a data set which is labeled by an unsupervised method; however, depending on the problem we are dealing with, the efficiency might differ.

1.1. Background

In this subsection, we discuss what is already known about the subject, how it is related to this paper, and the open problems which we wish to solve.

1.1.1. State-of-the-Art

Convolutional Neural Networks (CNN) are now the state-of-the-art approach for generating numerical vectors that represents faces (so-called embeddings), which are later used as input for clustering and classification algorithms. The role of CNN-based features is to supply a computer system with a real-valued vector-based discriminative face representation. Although the process of training of such a network is a supervised procedure [2] (i.e., the input data need to have labeled identities), novel papers have introduced some heuristics that allow this important limitation to be partially overcome [3]. After generating a face embedding, face verification systems utilize classification algorithms to assign faces to identities. Among most common classification approaches are k-nearest neighbors (KNN) [4,5], fully connected Neural Networks (NN) [6], and Support Vector Machines (SVM) [7].

Researchers have also addressed problems of domain adaptation networks for face recognition “in the wild” (i.e., in a non-laboratory environment) [8–11]. Some researchers have recommended adding additional face normalization by applying unsupervised face normalization (especially face frontalization) [12]. Training deep convolutional neural networks (CNNs) is often time demanding, as it requires the optimization of a large number of network parameters. To overcome this limitation, attempts have been made to perform additional preprocessing of data based on computing predefined convolution kernels from training data [13]. Researchers have also reported the application of Principal Components Analysis (PCA) in the role of unsupervised dimensionality reduction algorithm for face recognition [14]. In this context, “unsupervised” means that, contrary to Linear Discriminant Analysis (LDA), PCA does not require prior knowledge about the identities of people in the training data set. PCA has also been used, for example, to learn a filter bank for the convolutional layer [1].

Performance of a face recognition system may be conditioned by the quality of images in the data set. In [15], the authors proposed a quality assessment method aimed at estimating the suitability of a face image for recognition. Due to the complexity of CNN models and a lack of understanding of deep image features, there is still ongoing research into other feature discriminative methods [16,17].

Complex unsupervised systems that enable face identification have been already proposed in the literature. The solution proposed in [18] employs Deep Convolutional Neural Networks to extract features and an online clustering algorithm to determine face IDs. In [19], a graph-based unsupervised feature aggregation method for face recognition was proposed. The method uses the inter-connection between face pairs with a directed graph approach to refine the pairwise scores.

Large surveys on various aspects of deep learning applications for face recognition can be found in [20–23]. The survey [24] addressed the problems of occlusion, single sample per subject, and expression, while [25] addressed age invariant face recognition and [26] discussed face recognition under morphing attacks.

1.1.2. Motivation of This Paper

As can be seen in Section 1.1.1, researchers have addressed the problems of training face verification systems with and without knowledge about the ground truth; however, to the best of our knowledge, there has not been a complex study on the influence of knowledge about the ground truth on the efficiency of the trained classifier. The aim of this paper is to evaluate the potential improvement of classification results of state-of-the-art supervised classification methods trained with and without knowledge about ground truth and to answer the question “Is ground truth data required to train an effective face verification system?” This issue is very important in practice: manual or even semi-manual face image labelling is very time-consuming, as a face data set might consist of hundreds of thousands of images. Furthermore, we also propose a method to estimate the quality of a clustering algorithm for unlabeled data and its influence on the effectiveness of the classifier. In our research, we use two sufficiently large data sets that contain more than 200,000 “taken in the wild” images, each with various resolutions, visual quality, and face poses which, in our opinion, guarantees the statistical significance of our results.

1.2. Overview

Deep learning-based feature extraction methods and transfer learning have become common approaches in the field of pattern recognition. Deep convolutional neural networks trained using triple-based loss functions allow for the generation of face embeddings, which can be directly applied to face verification and clustering. Knowledge about the ground truth of face identities might improve the effectiveness of the final classification algorithm; however, it is also possible to conduct supervised training utilizing knowledge about data clusters discovered by an unsupervised approach. The aim of this paper is to evaluate the potential improvement of classification results of a state-of-the-art supervised classification method trained with and without knowledge of the ground truth.

We used an unsupervised clustering solution similar to that in [18], in order to discover groups of images with same identities; however, the algorithm in [18] has a slightly different purpose than ours: it is devoted to video data with a much smaller number of identities but has a similar image processing pipeline to that in our approach. In this paper, we propose many important improvements. Contrary to the work in [18], among other possible clustering algorithms, we evaluate HDBSCAN instead of DBSCAN, as HDBSCAN requires less adaptation parameters. Further, instead of video data, we evaluate our method on a data set which has about 6.3 times more identities than the YouTube data set used in [18]. We also propose the use of PCA-based dimension reduction for deep facial images features, which not only simplifies the computation by limiting the number of parameters but may also improve the face recognition results.

We evaluate the most popular supervised pattern recognition algorithms, applied to face embedding classification, namely, KNN [4], NN [6], and SVM [7], with various adaptive parameter values. We evaluated all algorithm results in terms of the clustering result quality measures, as it is not possible to obtain an accuracy of classification directly when the classifier is trained on labels generated by an unsupervised method. In this paper, we also show that it is possible to accurately estimate the quality of the clustering algorithm for unlabeled data, as there is a moderate positive correlation between measures utilizing and not utilizing knowledge about the ground truth. This is a very important find, as it allows for estimation of the algorithm efficiency in real-world scenarios. To the best of our knowledge, large-scale comparison of cluster quality measurements between classifiers trained on ground truth data and without this knowledge applied to state-of-the-art deep learning face recognition algorithms has not yet been published, and therefore we believe that results presented in this paper will be useful to the applied computer science community.

In the following sections, we present the details of our evaluation procedure and we describe the training and validation data sets, which we use to perform the experiments. We evaluate various possible clustering and classification algorithms. In this research, we use two sufficiently large data sets containing more than 200,000 “taken in the wild” images each, with various resolutions, visual quality,

and face poses. The first data set contains celebrity images, each with 40 attribute annotations. Both data sets contain images that cover large pose variations and background clutter. More details about the data can be found in Section 2.5. All of this, in our opinion, guarantees the statistical significance of the obtained results. Furthermore, both the source code and the data sets are available for download and our results can be reproduced.

2. Materials and Methods

In this section, we present the proposed methods, the data set we used for testing, and the scoring metrics we used for evaluation.

2.1. Face Verification without Knowing the Ground Truth

The following pipeline can be used for face verification without knowing the ground truth. An overview of the proposed method is presented in Figure 1. The system should be able to operate in a certain environment (e.g., a given social network), in which it can collect images that potentially contains face photos. At the “cold start”, the system needs to gather a sufficiently large training data set of images containing faces. Depending on the environment it operates within, its size might differ. For example, when we consider a social network of several thousand people, we should gather at least several photos published by each user. Each of those photos might contain faces of certain individuals that might appear on several other photos. It is also possible that an image does not contain any face; in this case, it has to be removed from the later processing. As described in Section 2.5, we trained this algorithm on a training data set containing about 100,000 facial images of about 10,000 individuals; however, there are no obstacles to utilizing even more images, as the clustering and classification algorithms we use are scalable. We do not have to make any assumption about how many images of certain individuals are present in the data set. A Figure 2 presents a block diagram of the proposed research.

The training data set of images, after initial processing, is used as the training data set for a classification algorithm. The preprocessing step consists of face detection, deep feature generation from facial images (which might be followed by PCA dimension reduction [27]), and unsupervised clustering. After clustering, each image has a cluster label to which it has been assigned. These labels, together with the deep features, are used as input data for supervised training of the classification algorithm. After the classifier has been trained, the system is ready to operate and moves to the working phase. In this phase, the pipeline works as follows; face detection, deep feature generation from facial images (which might be followed by PCA dimension reduction), and, finally, classification. PCA and classifier parameters are learned during the training phase. The trained classification algorithm assigns class labels that were discovered by clustering. Classifiers allow for the assignment of new facial images to classes faster than with the application of unsupervised clustering each time a new image is discovered. Furthermore, depending of the classification algorithm, it might also allow for generalization of the obtained results. The system might be retrained/adapted after certain amount of images has been recognized or after the scoring parameters described in Section 2.6.2, when the training data set enhanced by newly acquired data drops below a certain amount (in comparison to the initial value). Retraining of the system basically consists of repeating the training phase. The training data set should contain all of the images that have been acquired so far. The exact retraining “trigger” is highly dependent on the characteristics of the image data source, which is not in the scope of this paper.

Figure 1. An overview of the pipeline for face verification without knowing the ground truth. The top row presents the training procedure, while the bottom row shows the working phase. The system may be continuously improved by repeating the training procedure and adding data that was gathered so far into initial training data set.

Figure 2. A block diagram of the proposed research. (A) Face verification without known ground truth. (B) Face verification with known ground truth.

2.2. Face Detection and Feature Generation

After an image has been acquired, we need to detect the region of interest that might contain a face. Most approaches to date also require that the face detection method performs additional alignment of the face. Alignment is responsible for positioning the face on the output image in such a way that we should expect that all facial images aligned by the same algorithm should have a similar spatial positioning of certain parts of the face. Face detection and alignment is challenging, due to the various poses, illuminations, and occlusions involved. For our purposes, we adapted a deep cascaded multi-task framework with three stages of deep convolutional networks which predict face and landmark location in a coarse-to-fine manner (MTCNN), proposed in [28]. We used a pre-trained model implemented in Keras (<https://github.com/ipazc/mtcnn>).

After detecting and aligning a face, we can perform feature generation (embedding). Feature generation is among most important steps in the image classification pipeline, as the incorrect choices of features make cause the classification problem to become unsolvable. As already mentioned in Section 1.1.1, there exist many methods that can be used to generate features from facial images. Face embeddings should have similar feature vectors when we compare photos of the same person and different vectors when we compare photos of two different persons. The similarity might be defined as the Euclidean distance between m -dimensional feature vectors. Furthermore, feature generation methods should be robust to the lighting conditions in the photo, the pose of the persons (e.g., the direction of the head), facial expression, hair style, clothing, and so on. Among the most popular methods that satisfy these needs are deep learning-based methods that utilize neural network architectures. In this research, we chose the facenet architecture, initially described in [2]. Facenet is a convolutional neural network trained using a supervised Triplet Loss approach. Triplet Loss minimizes the distance between an anchor (image) and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. We used a pretrained facenet model (<https://github.com/nyoki-mtl/keras-facenet>) with over 22M trained parameters, as implemented in Keras and trained on the MS-Celeb-1M data set (<https://www.microsoft.com/en-us/research/project/ms-celeb-1m-challenge-recognizing-one-million-celebrities-real-world/>). The input layer of the model has size $160 \times 160 \times 3$ (as it uses color images), while the output layer is a real-valued vector with 128 dimensions. Facenet in recent years is among most popular face embedding methods [29–33].

We can reduce the dimension of the problem by applying a dimension reduction method. As in a real-life scenario we will not know the identities of objects in the data set, we might apply the Principal Components Analysis (PCA) [34] to estimate the amount of variance explained by the generated components. Reduction of the number of dimensions can reduce the computational burden and size of the model without affecting the overall classification effectiveness.

2.3. Clustering

In a real-world scenario, it is hardly possible to estimate how many classes (identities) of faces are present in a given data set; thus, there is no point in using methods such as k -means clustering which require prior knowledge about the number of classes. Due to this, we take into account methods that do not require such knowledge. Each clustering algorithm has some parameters that govern its performance. The clustering scorings we used are discussed in the following Section 2.6. We evaluated two clustering algorithms that use different approaches to cluster structures. The first one was Agglomerative (Hierarchical, also called Tree-based) Clustering, which makes the assumption that clusters have a concentric structure. We chose the ward cluster linkage and evaluated various distance threshold values on linkage.

The second clustering algorithm we tested was Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which is a density-based method [35]. It performs DBSCAN clustering over varying epsilon (spatial distance threshold) values and integrates the results to find a clustering that gives the best stability over epsilon. Due to this, HDBSCAN may find clusters of

varying densities and is more robust to input parameter selection than DBSCAN. The parameter of HDBSCAN is the minimum size of clusters (i.e., the minimal number of objects in the cluster).

2.4. Classification

Many classification algorithms are commonly used for face verification problems. Due to the massive amount of data and large number of classes (in comparison to the number of objects) in the data set, there are several algorithms that are used more often than others. Among them are the K-nearest neighbors approach (KNN), the linear SVM method, and neural networks.

K-nearest neighbors is among most basic but, at the same time, effective methods of classification. The main drawbacks of this method are its poor generalization ability and the necessity of keeping the whole data set in the memory. The search for the nearest elements can be sped up by the application of k-d trees for hierarchical decomposition of the space along different dimensions [36].

Without application of a kernel trick, SVM is a linear classifier. In the case of large problems, linear SVM requires less parameters than the kernel-based approach and its coverage and final classification performance is faster. At present, multi-class SVM optimization problems are generally formulated as sequential dual methods [37]. Recent findings have allowed for the parallelization of coordinate descent methods, which further speeds up the coverage of this classifier [38,39]. Forward-only sequential neural networks are popular and well-established classifiers. Such networks are often composed of a set of dense (fully connected) neuron layers, positioned one after another, which play the role of partitioning the feature space by decision hyperplanes. Each unit (neuron) of a dense layer takes a linear combination of the input vector and the neuron's weight which, then, is an argument for an activation function:

$$y = A(x \circ w + bias), \quad (1)$$

where x is an input vector, w is a vector of weights, $\circ$ is the dot product, and A is an activation function. Among the most commonly used activation functions, we can mention the Rectified Linear Unit (ReLU), which is defined as $\max(x, 0)$, where x is an input vector. Although this activation function seems very basic, studies have reported that some network architectures with ReLUs consistently learn several times faster than their equivalents using saturating neurons (e.g., with $\tanh$ as the activation function) [40]. The assignments to the classes are usually determined through a softmax transformation. The softmax object is assigned to a class, the index of which refers to highest value of softmax-transformed coordinate of the input vector,

$$\sigma(x_p) = \frac{e^{x_p}}{\sum_{r=1}^g x_r}, \quad (2)$$

where g is number of dimensions of the vector x . In this study, we trained the network using the Adam optimizer [41], which is first-order gradient-based optimization method using a categorical cross-entropy loss function.

2.5. Data Sets

In order to perform our experiment, we selected the Large-scale CelebFaces Attributes (CelebA) Data set (<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>) which is among the largest and most popular "in the wild" image sets containing facial images of people with information about their identities. The original data set contained 202,599 images. We performed face detection and aligning using the previously mentioned algorithm [28]. MTCNN detected faces in 202,039 images; however, some aligned images did not contained faces. We manually removed those images and the final data set we used contained 201,804 objects with 10,177 unique identities (classes). The number of removed images was below 0.4% of overall data, which should not disturb the algorithm's evaluation process. The number of images of the same person differed, throughout the data set, from 1 to 35. In Figure 3,

we present a histogram that summarizes the quantity of identities that have a certain number of images in the data set.

Figure 3. Histogram that summarizes the quantity of identities that have a certain number of images in the CelebA data set.

We randomly split this data set into two halves: A training data set and a validation data set. The training data set contained 100,902 objects with 10,021 identities (classes), while the validation data set had 100,902 images with 10,004 identities (classes). Figures 4 and 5 present histograms that summarize the quantity of identities that had a certain number of images in the training and validation data sets.

Figure 4. Histogram that summarizes the quantity of identities that have a certain number of images in the training data set.

Figure 5. Histogram that summarizes the quantity of identities that have a certain number of images in the validation data set.

The use of random selection did not guarantee that each identity was represented in both the training and validation data sets. We did it on purpose, however, as this situation better represents the real-life scenario. With this experimental set-up, it is virtually impossible to obtain 100% recognition accuracy.

In the next step, we generated facenet features for all images in both data sets. In Figure 6, we visualize a fragment of the data set, selecting only those objects that have at least 31 instances in the data set. The visualization was done using the Gephi 0.9.2 software. As this application is a graph visualization software, we have represented this fragment of the data set as a fully connected graph, the nodes of which are face objects and the undirected edges have weights equal to the Euclidean distances between vectors representing the pairs of faces that they connect. To make the graph layout more clear, we also removed edges with weights above 11 and filtered out all nodes that have degree below 3. To generate the layout, we used the Force Atlas 2 algorithm [42], which is a force-directed method. As can be seen in Figure 6, faces with the same identity seem to create clusters, as was expected.

We also used another large face data set, namely, CASIA-WebFace (Chinese Academy of Sciences) [43], to perform our experiment. We used about 40% of the original data (with data ordered by identity id) in order to make this data set a similar size as the CelebA dataset. After this selection, the CASIA-WebFace data set contained 206,458 objects with 3374 identities (classes). We performed face detection and alignment using the previously mentioned algorithm [28]. MTCNN detected faces in 205,312 images. The number of removed images was below 0.6% of overall data, which should not disturb the algorithm's evaluation process. We also randomly split this data set into two halves: a training data set and a validation data set. The training data set contained 102,656 objects with 3374 identities (classes), while the validation data set contained 102,657 images with 3373 identities (classes).

Figure 6. Visualization of a fragment of CelebA data set in the form of a graph (edges are hidden) with layout produced by the Force Atlas 2 algorithm. Images with the same identity create spatial clusters.

2.6. Evaluation of Clustering Results

Let us assume that we have an object set $S = \{O_1, \dots, O_n\}$ and suppose that $U = \{u_1, \dots, u_l\}$ and $V = \{v_1, \dots, v_k\}$ are two different partitions of S defined in the following way,

$$\begin{cases} \bigcup_{i=1}^l u_i = S = \bigcup_{j=1}^k v_j; \\ u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}, \end{cases} \quad (3)$$

for $1 \leq i \neq i' \leq l$ and $1 \leq j \neq j' \leq k$. We can evaluate the performance of the partitioning done by the clustering algorithm using several indexing/scoring methods.

2.6.1. The Ground Truth Is Known

Let us assume that U is a ground truth and V is the partition we want to evaluate. The Rand Index is given by

$$RI = \frac{a + b}{C_n^2}, \quad (4)$$

where a is the number of element pairs that are in same set in U and V , b is the number of element pairs that are in different sets in U and V , and C_n^2 is the number of all possible pairs. In order to make the scoring (4) values close to zero when the assignment is random, the Adjusted Rand Index [44] is defined as follows,

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}, \quad (5)$$

where $E[RI]$ is the expected value of (4), while $\max(RI)$ is the maximal value of (4). ARI is scaled into the range $[-1, 1]$ and measures the similarity of U and V , ignoring permutations.

The homogeneity score is defined as [45]

$$\begin{cases} HS = 1 & \text{when } H(U) = 0 \\ HS = 1 - \frac{H(U|V)}{H(U)} & \text{when } H(U) \neq 0 \end{cases} \quad (6)$$

where $H(U|V)$ is the conditional entropy of the class distribution given the proposed clustering and $H(U)$ is a maximal reduction in entropy the clustering information can provide. $H(U) = 0$ means that there is only a single class. The homogeneity score has the highest value (of 1) when each cluster contains only members of a single class.

The completeness score is defined as [45]

$$\begin{cases} CS = 1 & \text{when } H(V) = 0 \\ CS = 1 - \frac{H(V|U)}{H(V)} & \text{when } H(V) \neq 0 \end{cases} \quad (7)$$

where $H(V|U)$ is the conditional entropy of the proposed cluster distribution given the class of components datapoints and $H(V)$ is a maximum reduction in entropy the class information can provide. $H(V) = 0$ means that there is only a single cluster. The completeness score has highest value (of 1) when all members of all classes are assigned to same cluster.

The V-measure score is defined as [45]

$$VMS = \frac{(1 + \beta) \cdot HS \cdot CS}{\beta \cdot HS + CS}. \quad (8)$$

The Fowlkes–Mallows score is defined as [46]

$$FMS = \frac{TP}{\sqrt{(TP + FP) \cdot (TP + FN)}}, \quad (9)$$

where TP is the number of members of the same classe that were assigned to the same clusters, FP is the number of members of the same classe that were not assigned to the same clusters, and FN is the number of same clusters that do not belong to the same classes. $FMS = 1$ that means that U and V are equal (with or without permutation).

The discovered clusters number ratio is

$$CNR = 1 - \left\| \frac{l - k}{l + k} \right\|. \quad (10)$$

2.6.2. The Ground Truth Is Unknown

In the situation when there is no information about the ground truth, we have to use the V partition to evaluate it on itself. The Silhouette Coefficient for a single object O_g is given as [47]

$$SC1(O_g, V) = \frac{mdnnc(O_g, V) - mdsc(O_g, V)}{\max(md nnc(O_g, V), mdsc(O_g, V))}, \quad (11)$$

where $g \in [1, n]$, $mdnnc(O_g, V)$ is the mean distance between an object O_g and all objects in the next nearest assignment, and $mdsc(O_g, V)$ is the mean distance between an object O_g and all other objects in the same assignment.

The Silhouette Coefficient for an assignment is defined as

$$SC = \frac{1}{n} \cdot \sum_{g=1}^n SC1(O_g, V). \quad (12)$$

The Calinski–Harabasz Index [48] is defined as

$$CHS = \frac{tr(B_k)}{tr(W_k)} \cdot \frac{n - k}{k - 1}, \quad (13)$$

where $tr(B_k)$ is a trace between the group dispersion matrix and $tr(W_k)$ is a trace of the within-cluster dispersion matrix.

3. Results

Our experiments were implemented in Python 3.6. Among the most important packages that were used are Tensorflow 2.1 for machine learning with configured GPU support, in order to speed up network training, mtcnn 0.1.0 for face detection and segmentation, the deep neural networks (DNN) Keras 2.3.1 library, the sklearn package for KNN, and OpenCV-python 4.2.0.32 for general purpose image processing. For algorithm training and evaluation, we used a PC computer with an Intel i7-9700F 3.00 Ghz CP, 64 GB RAM, and an NVIDIA GeForce RTX 2060 GPU on Windows 10 OS. All source code can be downloaded from our GitHub repository (<https://github.com/browarsoftware/adlcc>). We performed evaluation on data sets introduced in Section 2.5. In those particular data sets, the ground truth is known. We did not use this knowledge for unsupervised algorithm training purposes.

We performed the following calculations,

- face detection and embedding, with PCA analysis of face embedding;
- face embedding clustering with HDBSCAN and Agglomerative Clustering;
- supervised training of KNN, SVM, and NN on face clusters discovered by the unsupervised algorithm;
- analysis of linear relationships between cluster quality measurements;
- supervised training of KNN, SVM, and NN on ground truth data; and
- comparison of cluster quality measurements between classifiers trained on ground truth data and without this knowledge.

3.1. Face Detection and Embedding, PCA Analysis

After deep feature generation (see Section 2.2), each face was represented by a 128-dimensional real-valued vector. We performed PCA analysis of these values. Table 1 and Figure 7 present the number of PCA components and cumulative % of variance explained by them for the CelebaA data set. Results for the CASIA-WebFace data set are presented in Table 2. As can be seen for CelebaA, 21 components were required to explain over 51% of the variance, 36 components to explain over 75% of the variance, 48 components to explain over 90% of the variance, 53 components to explain over 95%, and 62 components to explain over 99% of the variance. In the case of CASIA-WebFace, the results were nearly identical: 21 components were required to explain over 50% of the variance, 36 components to explain over 75% of the variance, 48 components to explain over 90% of the variance, 53 components to explain over 95%, and 62 components to explain over 99% of the variance. As can be seen, only half of the components were required to explain nearly all of the variance present in the data sets. We attempted to take advantage of this by evaluating not only the original

128-dimensional vectors, but also the data projected onto lower-dimensional space by PCA. In further analysis, eigenvectors calculated with PCA on training data sets were also used to project the validation data sets to lower dimensional space. The dimensionality limitation of the problem may reduce its complexity; for example, reducing the number of coefficients that are required to be calculated during training of the classification model.

Table 1. Number of Principal Components Analysis (PCA) components and cumulative % of variance explained by them in the training data set (CelebA).

Number of Components	Cumulative % of Variance Explained
21	51.1
36	75.8
48	90.6
53	95.2
62	99.9

Table 2. Number of PCA components and cumulative % of variance explained by them in the training data set (CASIA-WebFace).

Number of Components	Cumulative % of Variance Explained
21	50.6
36	75.6
48	90.8
53	95.5
62	99.9

Figure 7. Number of PCA components and cumulative % of variance explained by them in the training data set.

3.2. Clustering with HDBSCAN and Agglomerative Clustering

We have examined two clustering algorithms, namely, Agglomerative Clustering (Hierarchical clustering) and HDBSCAN [35,49], with various parameter values. In the case of Agglomerative Clustering, an adaptive parameter was the distance threshold, which is the linkage distance threshold; above which, clusters will not be merged. In the case of HDBSCAN, the parameter was minimal cluster size: single linkage splits that contain fewer points than this will be considered points “falling out” of a cluster, rather than a cluster splitting into two new clusters. We chose those two particular clustering

approaches as they cover both centroid-based and density-based approaches. We chose various ranges for the clustering parameter values and also used both the 128 features set and the features set projected to lower dimensionality. Evaluation of the clustering results was done using the scores described in Section 2.6, as presented in Table 3. In the first columns, the number beside to clustering algorithm's name is its parameter. The string PCA and number informs that PCA dimensional reduction has been applied and how many dimensions persisted in the data set. The parameter β in (8) was set as 1. In Tables 3–10 bold font indicates best results of certain scoring metrics in experiment.

Table 3. Results of evaluation of various clustering algorithms on the CelebA training data set.

Method	ARI	HS	CS	VMS	FMS	SC	CHS	CNR
Agglomerative Clustering 2	0.001	1.000	0.788	0.881	0.017	0.004	121.138	0.181
Agglomerative Clustering 4	0.013	1.000	0.791	0.883	0.078	0.023	24.178	0.187
Agglomerative Clustering 6	0.102	0.999	0.810	0.895	0.227	0.064	12.934	0.222
Agglomerative Clustering 8	0.304	0.997	0.848	0.917	0.416	0.112	11.377	0.308
Agglomerative Clustering 10	0.532	0.991	0.891	0.939	0.592	0.165	12.881	0.465
Agglomerative Clustering 12	0.691	0.982	0.923	0.952	0.714	0.211	16.369	0.666
Agglomerative Clustering 14	0.765	0.975	0.941	0.957	0.774	0.229	19.466	0.815
Agglomerative Clustering 16	0.786	0.967	0.948	0.958	0.789	0.228	21.780	0.918
Agglomerative Clustering 18	0.783	0.961	0.952	0.956	0.783	0.223	23.189	0.983
Agglomerative Clustering 20	0.766	0.953	0.953	0.953	0.766	0.216	24.194	0.967
Agglomerative Clustering 22	0.740	0.946	0.953	0.950	0.741	0.209	24.910	0.925
Agglomerative Clustering 24	0.706	0.937	0.954	0.945	0.708	0.199	25.511	0.884
Agglomerative Clustering 16, PCA 21	0.489	0.880	0.912	0.896	0.496	0.119	36.759	0.817
Agglomerative Clustering 16, PCA 36	0.732	0.950	0.943	0.946	0.733	0.204	25.415	0.996
Agglomerative Clustering 16, PCA 48	0.777	0.963	0.948	0.955	0.779	0.225	22.998	0.946
Agglomerative Clustering 16, PCA 53	0.784	0.965	0.949	0.957	0.786	0.228	22.469	0.934
Agglomerative Clustering 16, PCA 62	0.785	0.967	0.948	0.958	0.788	0.229	21.784	0.918
HDBSCAN 2	0.004	0.823	0.930	0.873	0.037	0.144	12.300	0.858
HDBSCAN 3	0.003	0.777	0.949	0.855	0.034	0.152	16.447	0.909
HDBSCAN 4	0.002	0.716	0.948	0.816	0.027	0.123	15.901	0.833
HDBSCAN 5	0.001	0.661	0.946	0.778	0.023	0.093	15.335	0.769
HDBSCAN 2, PCA 21	0.000	0.549	0.901	0.682	0.014	-0.062	10.280	0.880
HDBSCAN 2, PCA 36	0.002	0.726	0.937	0.818	0.026	0.105	14.711	0.919
HDBSCAN 2, PCA 48	0.003	0.770	0.945	0.849	0.033	0.144	15.913	0.926
HDBSCAN 2, PCA 53	0.003	0.777	0.947	0.853	0.034	0.148	16.000	0.930
HDBSCAN 2, PCA 62	0.004	0.823	0.930	0.873	0.037	0.144	12.310	0.858
HDBSCAN 2, PCA 128	0.004	0.823	0.930	0.873	0.037	0.144	12.300	0.858
HDBSCAN 3, PCA 21	0.000	0.454	0.905	0.605	0.013	-0.113	10.781	0.708
HDBSCAN 3, PCA 36	0.001	0.656	0.936	0.771	0.021	0.072	14.460	0.822
HDBSCAN 3, PCA 48	0.002	0.708	0.944	0.810	0.026	0.115	15.584	0.843
HDBSCAN 3, PCA 53	0.002	0.718	0.946	0.816	0.027	0.122	15.771	0.846
HDBSCAN 3, PCA 62	0.003	0.777	0.949	0.854	0.034	0.152	16.452	0.908
HDBSCAN 3, PCA 128	0.003	0.777	0.949	0.855	0.034	0.152	16.447	0.909
HDBSCAN 4, PCA 21	0.000	0.383	0.905	0.538	0.012	-0.156	10.775	0.597
HDBSCAN 4, PCA 36	0.001	0.592	0.934	0.725	0.018	0.036	13.962	0.745
HDBSCAN 4, PCA 48	0.001	0.652	0.942	0.770	0.022	0.085	14.925	0.781
HDBSCAN 4, PCA 53	0.001	0.662	0.944	0.778	0.023	0.093	15.104	0.787
HDBSCAN 4, PCA 62	0.002	0.716	0.947	0.816	0.027	0.122	15.905	0.833
HDBSCAN 4, PCA 128	0.002	0.716	0.948	0.816	0.027	0.123	15.901	0.833
HDBSCAN 5, PCA 21	0.000	0.317	0.906	0.470	0.012	-0.195	10.974	0.489
HDBSCAN 5, PCA 36	0.001	0.531	0.932	0.676	0.017	0.001	13.514	0.671
HDBSCAN 5, PCA 48	0.001	0.592	0.941	0.726	0.019	0.053	14.369	0.712
HDBSCAN 5, PCA 53	0.001	0.603	0.942	0.735	0.020	0.061	14.577	0.718
HDBSCAN 5, PCA 62	0.001	0.660	0.946	0.777	0.023	0.093	15.334	0.768
HDBSCAN 5, PCA 128	0.001	0.661	0.946	0.778	0.023	0.093	15.335	0.769

In the case of CelebA, the highest ARI and FMS scores were obtained for Agglomerative Clustering with threshold parameter of 16, with ARI = 0.786 and FMS = 0.789; however, the ARI value for Agglomerative Clustering with same threshold on the data set projected onto 53 dimensions differed by only 0.002 and on 62 dimensions by only 0.001 (in the case of FMS by 0.03 and 0.01, respectively). As the ARI value was positive and close to 1, we can assume that the clustering was carried out successfully. ARI is an important indicator, as it does not make any assumption about the cluster structure. The highest value of HS (of 1) was obtained under Agglomerative Clustering with a threshold of 2; however, this situation likely occurred as each cluster contained very few elements (for example, 1) and, due to this, the HS obtained its highest possible value. The highest value of CS was obtained for Agglomerative Clustering 24. In this case, the algorithm generated very large clusters, which contained all members of objects of a single class; however, these might also contain many objects of different classes. Due to this, HS and CS should not be considered separately but using VMS.

The highest value of VMS appeared in Agglomerative Clustering 16 (0.958) and Agglomerative Clustering 16/PCA 62. This is a very high value; however, we have to remember that this scoring is not normalized with regards to random labeling and, due to the large number of clusters that were present in our data set, this value might not be as meaningful as the ARI.

FMS reached its highest value for Agglomerative Clustering 16 (0.789); however, for Agglomerative Clustering 16/PCA 62, the scoring differed by only 0.001. This is a very good result, assuring us that the labeling corresponded with the real classes.

SC reached its highest value for Agglomerative Clustering 14 and Agglomerative Clustering 16/PCA 62, which was equal to 0.229. Very similar results were obtained for Agglomerative Clustering 16 and Agglomerative Clustering 16/PCA 53, where the FMS differed by only 0.001. A magnitude higher than the other values of CHS was obtained by Agglomerative Clustering 2. As can be seen, a degenerate case like this, where we have many one-element clusters, $ARI = 0.001$, $HS = 1$, and $SC = 0.004$, is a strong indicator that this metric is not meaningful and does not correspond well with other measures. Due to this, we omit the CHS in further evaluation and discussion. CNR had the highest value for Agglomerative Clustering 16/PCA 36. The CNR for Agglomerative Clustering 16 was above 0.9, meaning there was not much difference between the actual number of classes and clusters.

In the case of the CASIA-WebFace data set, we evaluated only Agglomerative Clustering. The highest ARI and FMS scores were obtained for Agglomerative Clustering with threshold parameters 34, equal to 0.643 and 0.649, respectively. ARI and FMS values for Agglomerative Clustering with same threshold on data set projected onto 62 dimensions differed by only 0.005. The ARI value was positive and close to 1; thus, we can assume that the clustering was carried out successfully. The highest value of HS was equal to 0.996, which was obtained for Agglomerative Clustering with threshold 8; however, this situation probably occurred as nearly all clusters contained very few elements (i.e., close to 1) and, due to this, the HS obtained its highest possible value.

The highest value of VMS appeared in Agglomerative Clustering 22 (0.875) and Agglomerative Clustering 24.

SC reached its highest value for Agglomerative Clustering 28 (0.126). Very similar results were obtained for Agglomerative Clustering 30, 32, and 34, where the FMS differed by only 0.001. The CNR had the highest value for Agglomerative Clustering 30 (0.973). In the case of Agglomerative Clustering 34, the CNR was 0.871, which means there was not much difference between the actual number of classes and clusters.

From Table 3, we can clearly see that all meaningful scorings indicate that Agglomerative Clustering performed better than HDBSCAN. Due to this, we did not evaluate HDBSCAN in Table 4. In case of the CelebA data set, in all but one case, the best results were obtained when threshold parameter was set as 16. Due to this, for further evaluation on the CelebA data set, we chose assignments that were generated with Agglomerative Clustering 16 and its variations with varying numbers of PCA-projected dimensions. In the case of the CASIA-WebFace data set, the highest values of ARI, CS, and FMS were obtained for Agglomerative Clustering 34. Due to this, for further evaluation on CASIA-WebFace data set, we chose assignments that were generated with Agglomerative Clustering 34 and its variations with varying numbers of PCA-projected dimensions.

Table 4. Results of evaluation of Agglomerative Clustering on CASIA-WebFace training data set.

Method	ARI	HS	CS	VMS	FMS	SC	CHS	CNR
Agglomerative Clustering 8	0.063	0.996	0.701	0.823	0.166	0.072	10.262	0.099
Agglomerative Clustering 9	0.086	0.993	0.716	0.832	0.202	0.082	10.044	0.121
Agglomerative Clustering 10	0.114	0.987	0.732	0.840	0.237	0.089	10.541	0.154
Agglomerative Clustering 11	0.145	0.979	0.746	0.847	0.270	0.094	11.726	0.200
Agglomerative Clustering 12	0.178	0.972	0.759	0.852	0.303	0.097	13.255	0.250
Agglomerative Clustering 14	0.249	0.959	0.783	0.862	0.367	0.100	16.803	0.361
Agglomerative Clustering 16	0.317	0.946	0.801	0.868	0.420	0.104	20.826	0.478
Agglomerative Clustering 18	0.388	0.936	0.816	0.872	0.473	0.110	24.871	0.588
Agglomerative Clustering 20	0.447	0.926	0.828	0.874	0.516	0.116	28.639	0.684
Agglomerative Clustering 22	0.495	0.916	0.837	0.875	0.549	0.120	32.303	0.772
Agglomerative Clustering 24	0.536	0.907	0.844	0.875	0.578	0.123	35.508	0.844
Agglomerative Clustering 26	0.568	0.899	0.850	0.873	0.600	0.125	38.535	0.909
Agglomerative Clustering 28	0.596	0.890	0.854	0.872	0.618	0.126	41.453	0.970
Agglomerative Clustering 30	0.608	0.881	0.856	0.868	0.624	0.125	44.401	0.973
Agglomerative Clustering 32	0.628	0.872	0.859	0.866	0.638	0.125	47.145	0.922
Agglomerative Clustering 34	0.643	0.864	0.862	0.863	0.649	0.125	50.028	0.871
Agglomerative Clustering 36	0.639	0.855	0.862	0.859	0.643	0.122	52.611	0.827
Agglomerative Clustering 38	0.632	0.846	0.862	0.854	0.633	0.117	55.521	0.780
Agglomerative Clustering 40	0.623	0.837	0.862	0.849	0.623	0.113	58.056	0.742
Agglomerative Clustering 50	0.553	0.793	0.861	0.826	0.559	0.092	72.054	0.564
Agglomerative Clustering 34, PCA 21	0.457	0.741	0.798	0.768	0.459	0.063	112.339	0.559
Agglomerative Clustering 34, PCA 36	0.594	0.825	0.847	0.836	0.595	0.106	67.829	0.743
Agglomerative Clustering 34, PCA 48	0.629	0.852	0.859	0.855	0.632	0.12	55.048	0.829
Agglomerative Clustering 34, PCA 53	0.635	0.859	0.86	0.859	0.639	0.123	52.093	0.853
Agglomerative Clustering 34, PCA 62	0.638	0.864	0.86	0.862	0.644	0.124	49.884	0.873

3.3. Supervised Training of KNN, SVM, and NN on Face Clusters Discovered by Unsupervised Algorithm

In the next step of evaluation, we used clusters obtained by Agglomerative Clustering to perform training of the classification algorithm using a supervised approach. We have to remember that, during training, we do not use any knowledge about the ground truth of classes, only the results of the previous unsupervised approach. We selected the nearest neighbor approach (KNN) with k-d trees for hierarchical decomposition, linear support vector machine (SVM) [50], and an artificial neural network (NN) with a single hidden fully-connected layer with ReLU activation and softmax output. Classifiers are trained with a supervised algorithm, in which we use the same deep features as before.

After training, classifiers were used to assign classes to the validation data sets described in Section 2.5. As we cannot directly calculate the accuracy of each approach due to the unknown mapping relation between ground truth and clusters index assignment, we have to once again perform evaluation using the scoring described in Section 2.6. The results are presented in Tables 5 and 6. In the first column, the number beside KNN indicates the number of neighbors considered in the classification and the number beside NN is number of neurons in hidden layer. If PCA was applied, the number indicates how many dimensions were used.

We can see, from Table 5, that the highest ARI score was obtained for SVM PCA 62 (at 0.769). A similar ARI value of 0.768 was obtained with the SVM. The HS had peak values for SVM and SVM/PCA 62. In the case of CS, there were several classifiers that obtained the same value of 0.955: KNN 1, KNN 5, KNN 1/PCA 53, KNN 1/PCA 62, KNN 3/PCA 62, KNN 5/PCA 53, and KNN 5/PCA 62. The highest value of VMS (0.957) was reached for KNN 1 and KNN 1/PCA 62. Similar VMS values were obtained by SVM and SVM/PCA 62 (0.955). The FMS reached its highest value with SVM/PCA 62 (0.769). This is a very good result that, together with the ARI value, assures us that the classification corresponded well to the real classes. According to the SC, the most dense and well-separated clusters are obtained by KNN 1. The maximal value of CNR was for KNN 5/PCA 53 (0.998); however, nearly every method had this parameter equals at least 0.95, which means that there was nearly the same number of classes to which elements were assigned by the classifiers as ground truth.

In Table 6, the highest ARI, HS, VMS, FMS, and CNR scores were obtained by KNN 1/PCA 62 (0.652, 0.875, 0.878, 0.654, and 0.983, respectively). This is also a very good result: the high CNR and ARI values assure us that classification corresponds with the real classes. KNN had the highest values of CS (0.882) and VMS (0.878).

Table 5. Results of evaluation of classification methods trained on clustering results of Agglomerative Clustering 16/Agglomerative Clustering 16 PCA for the CelebA data set.

Method	ARI	HS	CS	VMS	FMS	SC	CNR
KNN 1	0.749	0.958	0.955	0.957	0.749	0.196	0.950
KNN 3	0.708	0.953	0.954	0.954	0.711	0.194	0.968
KNN 5	0.697	0.951	0.955	0.953	0.701	0.194	0.982
KNN 1, PCA 36	0.643	0.940	0.946	0.943	0.649	0.170	0.992
KNN 1, PCA 48	0.712	0.953	0.954	0.953	0.714	0.192	0.969
KNN 1, PCA 53	0.724	0.956	0.955	0.955	0.726	0.194	0.961
KNN 1, PCA 62	0.738	0.958	0.955	0.957	0.739	0.195	0.950
KNN 3, PCA 36	0.529	0.933	0.945	0.939	0.552	0.168	0.980
KNN 3, PCA 48	0.663	0.948	0.953	0.951	0.670	0.190	0.986
KNN 3, PCA 53	0.641	0.950	0.954	0.952	0.652	0.194	0.977
KNN 3, PCA 62	0.662	0.952	0.955	0.953	0.669	0.194	0.969
KNN 5, PCA 36	0.509	0.932	0.946	0.939	0.538	0.171	0.971
KNN 5, PCA 48	0.638	0.946	0.954	0.950	0.649	0.192	0.998
KNN 5, PCA 53	0.618	0.948	0.955	0.951	0.633	0.193	0.990
KNN 5, PCA 62	0.641	0.950	0.955	0.953	0.652	0.192	0.981
SVM	0.768	0.959	0.952	0.955	0.768	0.190	0.942
SVM, PCA 36	0.591	0.936	0.943	0.939	0.602	0.170	0.992
SVM, PCA 48	0.717	0.954	0.952	0.953	0.719	0.190	0.966
SVM, PCA 53	0.749	0.956	0.952	0.954	0.749	0.191	0.955
SVM, PCA 62	0.769	0.959	0.952	0.955	0.769	0.190	0.944
NN 1024	0.700	0.947	0.944	0.946	0.701	0.180	0.952
NN 2048	0.706	0.946	0.944	0.945	0.707	0.178	0.953
NN 4096	0.671	0.941	0.941	0.941	0.672	0.175	0.957
NN 8192	0.655	0.938	0.939	0.938	0.657	0.171	0.960
NN 1024, PCA 21	0.341	0.840	0.884	0.862	0.359	0.059	0.814
NN 1024, PCA 36	0.644	0.929	0.934	0.931	0.646	0.158	0.991
NN 1024, PCA 48	0.716	0.947	0.945	0.946	0.716	0.181	0.969
NN 1024, PCA 53	0.730	0.949	0.946	0.947	0.730	0.183	0.961
NN 1024, PCA 62	0.727	0.951	0.947	0.949	0.727	0.185	0.948
NN 2048, PCA 21	0.358	0.844	0.885	0.864	0.373	0.059	0.815
NN 2048, PCA 36	0.624	0.927	0.933	0.930	0.626	0.154	0.991
NN 2048, PCA 48	0.693	0.943	0.943	0.943	0.693	0.178	0.970
NN 2048, PCA 53	0.713	0.947	0.945	0.946	0.713	0.181	0.963
NN 2048, PCA 62	0.723	0.949	0.945	0.947	0.723	0.181	0.950
NN 4096, PCA 21	0.341	0.840	0.884	0.862	0.359	0.057	0.815
NN 4096, PCA 36	0.609	0.924	0.931	0.928	0.612	0.151	0.989
NN 4096, PCA 48	0.686	0.941	0.942	0.942	0.687	0.176	0.974
NN 4096, PCA 53	0.676	0.942	0.942	0.942	0.677	0.177	0.964
NN 4096, PCA 62	0.697	0.945	0.943	0.944	0.698	0.178	0.953
NN 8192, PCA 21	0.337	0.839	0.883	0.861	0.355	0.054	0.813
NN 8192, PCA 36	0.583	0.921	0.930	0.925	0.587	0.149	0.987
NN 8192, PCA 48	0.633	0.934	0.937	0.935	0.635	0.170	0.975
NN 8192, PCA 53	0.644	0.936	0.939	0.937	0.646	0.171	0.969
NN 8192, PCA 62	0.654	0.938	0.939	0.938	0.656	0.171	0.959

Table 6. Results of evaluation of classification methods trained on clustering results of Agglomerative Clustering 34/Agglomerative Clustering 34 PCA for CASIA-WebFace data set.

Method	ARI	HS	CS	VMS	FMS	SC	CNR
KNN 1	0.650	0.874	0.882	0.878	0.651	0.109	0.891
KNN 1, PCA 36	0.587	0.833	0.864	0.849	0.588	0.100	0.763
KNN 1, PCA 48	0.623	0.861	0.877	0.869	0.624	0.114	0.849
KNN 1, PCA 53	0.628	0.868	0.879	0.873	0.629	0.117	0.873
KNN 1, PCA 62	0.652	0.875	0.881	0.878	0.654	0.118	0.893
SVM	0.647	0.862	0.863	0.863	0.650	0.122	0.891
SVM, PCA 36	0.578	0.824	0.852	0.838	0.579	0.118	0.763
SVM, PCA 48	0.638	0.855	0.864	0.859	0.639	0.124	0.849
SVM, PCA 53	0.640	0.859	0.864	0.861	0.642	0.124	0.873
SVM, PCA 62	0.647	0.863	0.863	0.863	0.652	0.122	0.893
NN 1024	0.604	0.847	0.854	0.851	0.605	0.121	0.891
NN 2048	0.598	0.847	0.855	0.851	0.599	0.109	0.891
NN 4096	0.604	0.848	0.855	0.852	0.605	0.108	0.891
NN 8192	0.601	0.847	0.855	0.851	0.602	0.108	0.891
NN 1024, PCA 21	0.404	0.719	0.783	0.750	0.410	0.053	0.579
NN 1024, PCA 36	0.549	0.807	0.837	0.822	0.549	0.095	0.763
NN 1024, PCA 48	0.586	0.837	0.851	0.844	0.587	0.107	0.849
NN 1024, PCA 53	0.595	0.843	0.853	0.848	0.596	0.109	0.873
NN 1024, PCA 62	0.604	0.849	0.854	0.852	0.606	0.110	0.893
NN 2048, PCA 21	0.417	0.722	0.784	0.752	0.422	0.051	0.579
NN 2048, PCA 36	0.539	0.805	0.836	0.821	0.539	0.094	0.763
NN 2048, PCA 48	0.594	0.839	0.853	0.846	0.594	0.107	0.849
NN 2048, PCA 53	0.588	0.842	0.852	0.847	0.589	0.109	0.873
NN 2048, PCA 62	0.595	0.848	0.855	0.851	0.597	0.109	0.893
NN 4096, PCA 21	0.412	0.722	0.784	0.751	0.417	0.052	0.579
NN 4096, PCA 36	0.541	0.806	0.836	0.821	0.542	0.092	0.763
NN 4096, PCA 48	0.589	0.837	0.852	0.844	0.589	0.105	0.849
NN 4096, PCA 53	0.589	0.842	0.853	0.847	0.590	0.107	0.873
NN 4096, PCA 62	0.601	0.849	0.855	0.852	0.602	0.106	0.893
NN 8192, PCA 21	0.415	0.722	0.785	0.752	0.421	0.051	0.579
NN 8192, PCA 36	0.534	0.804	0.835	0.819	0.535	0.090	0.763
NN 8192, PCA 48	0.581	0.835	0.851	0.843	0.582	0.105	0.848
NN 8192, PCA 53	0.595	0.844	0.854	0.849	0.596	0.107	0.873
NN 8192, PCA 62	0.585	0.844	0.851	0.847	0.586	0.104	0.893

3.4. Analysis of Linear Relationships between Cluster Quality Measurements

In the situation where the ground truth values of image classes are not known, we need to check whether it is possible to estimate the performance of the proposed solution using clustering quality measures that do not require knowledge of the ground truth. We performed correlation analysis in order to investigate the linear relationship between the pairs of values presented in Tables 3 and 4. The correlation matrix is presented in Tables 7 and 8 and is visualized in Figures 8 and 9.

Table 7. Correlation matrix representing the linear relationship between pairs of values from Table 3 (CelebA).

	ARI	HS	CS	VMS	FMS	SC	CHS	CNR
ARI	1.000	0.676	0.184	0.679	0.997	0.649	0.122	0.268
HS		1.000	−0.216	0.963	0.707	0.778	0.361	−0.040
CS			1.000	0.036	0.140	0.424	−0.449	0.910
VMS				1.000	0.700	0.913	0.253	0.203
FMS					1.000	0.654	0.112	0.223
SC						1.000	0.021	0.512
CHS							1.000	−0.318
CNR								1.000

Table 8. Correlation matrix representing the linear relationship between pairs of values from Table 4 (CASIA-WebFace).

	ARI	HS	CS	VMS	FMS	SC	CHS	CNR
ARI	1.000	−0.792	0.981	0.289	0.994	0.739	0.702	0.958
HS		1.000	−0.735	0.329	−0.737	−0.184	−0.985	−0.640
CS			1.000	0.397	0.992	0.785	0.625	0.946
VMS				1.000	0.381	0.840	−0.457	0.454
FMS					1.000	0.796	0.638	0.970
SC						1.000	0.0522	0.835
CHS							1.000	0.553
CNR								1.000

Figure 8. Color-coded visualization of correlation matrix from Table 7.

As can be seen in Table 7, the ARI and FMS are strongly positively correlated, and there was a moderately positive correlation between those two scorings and the SC (0.649 and 0.654, respectively). There was also a strong positive correlation between the VMS and SC, equal to 0.913. This is very important information, as we can use the SC (which does not require ground truth knowledge) to estimate the ARI, FMS, and VMS.

The results in Table 8 confirm the previous results. The ARI and FMS were strongly positively correlated, and there was a strong positive correlation between those two scorings and SC (0.739 and 0.796, respectively). Furthermore, there was a strong positive correlation between the VMS and SC (0.840). It seems that, in both CelebA and CASIA-WebFace, there are similar linear relationships between cluster quality measurements.

Figure 9. Color-coded visualization of correlation matrix from Table 8.

3.5. Supervised Training of KNN, SVM, and NN on Ground Truth Data

In the last part of the experiment, we compared the effectiveness of previously trained classifiers with pattern recognition methods that were trained using ground truth data. We used the same training and evaluation data sets as above; however, this time, we utilized knowledge about the ground truth. We used the same classifier configurations as in [4] (KNN-1), [6] (NN with single fully connected layer with 1024 neurons and softmax), and [7] (linear SVM). As the data set projected by PCA onto 62-dimensional space obtained satisfactory results in our previous experiment, we evaluated the classifiers on both the original and projected data. The results are presented in Table 9 for CelebA and in Table 10 for CASIA-WebFace.

In the case of the CelebA data set, the highest accuracy for the classification algorithm trained on the ground truth data was obtained using the KNN-1 and SVM methods (equals 0.875). Application of PCA dimensionality reduction does not affect the overall accuracy positively: It either remains the

same as for the 62-dimensional data set or has a smaller value. The accuracy of the NN classifier is slightly smaller and equal to 0.851 or 0.853, depending on whether PCA was used.

In the case of the CASIA-WebFace data set, the highest accuracy for classification algorithm trained on ground truth data was obtained for SVM, equal to 0.838. Application of PCA dimensionality reduction slightly improved the accuracy in the case of NN (by 0.001). In other cases, it either remained the same as it was for the 62-dimensional data set or had a smaller value.

Table 9. Comparative analysis of training with known ground truth and with class labels discovered by Agglomerative Clustering 16 or Agglomerative Clustering 16 PCA in the CelebA data set. GT means that ground truth data was used in training. U means that class labels were calculated by Agglomerative Clustering 16 or Agglomerative Clustering 16 with PCA projection onto 62 dimensions (PCA 62).

Method	ARI	HS	CS	VMS	FMS	SC	CNR	Accuracy
GT KNN 1	0.791	0.958	0.961	0.959	0.791	0.184	0.985	0.875
GT KNN 1, PCA 62	0.791	0.958	0.961	0.959	0.791	0.184	0.985	0.874
U KNN 1	0.749	0.958	0.955	0.957	0.749	0.196	0.950	–
U KNN 1, PCA 62	0.738	0.958	0.955	0.957	0.739	0.195	0.950	–
GT NN	0.725	0.946	0.953	0.950	0.727	0.178	0.977	0.851
GT NN, PCA 62	0.732	0.948	0.955	0.951	0.733	0.179	0.979	0.853
U NN 1024	0.700	0.947	0.944	0.946	0.701	0.180	0.952	–
U NN 1024, PCA 62	0.727	0.951	0.947	0.949	0.727	0.185	0.948	–
GT SVM	0.742	0.956	0.961	0.959	0.745	0.186	0.980	0.875
GT SVM, PCA 62	0.705	0.955	0.961	0.958	0.712	0.187	0.979	0.875
U SVM	0.768	0.959	0.952	0.955	0.768	0.190	0.942	–
U SVM, PCA 62	0.769	0.959	0.952	0.955	0.769	0.190	0.944	–

Table 10. Comparative analysis of training with known ground truth and with class labels discovered by Agglomerative Clustering 34 or Agglomerative Clustering 34 PCA in the CASIA-WebFace data set. GT means that ground truth data was used in training. U means that class labels were calculated by Agglomerative Clustering 34 or Agglomerative Clustering 34 with PCA projection onto 62 dimensions (PCA 62).

Method	ARI	HS	CS	VMS	FMS	SC	CNR	Accuracy
GT KNN 1	0.697	0.881	0.887	0.884	0.697	0.095	0.989	0.825
GT KNN 1, PCA 62	0.696	0.881	0.887	0.884	0.697	0.068	0.979	0.825
U KNN 1	0.650	0.874	0.882	0.878	0.651	0.109	0.891	–
U KNN 1, PCA 62	0.652	0.875	0.881	0.878	0.654	0.118	0.893	–
GT NN	0.670	0.869	0.881	0.875	0.672	0.091	0.979	0.811
GT NN, PCA 62	0.670	0.869	0.882	0.876	0.672	0.069	0.977	0.812
U NN 1024	0.604	0.847	0.854	0.851	0.605	0.121	0.891	–
U NN 1024, PCA 62	0.604	0.849	0.854	0.852	0.606	0.110	0.893	–
GT SVM	0.716	0.891	0.897	0.894	0.718	0.115	0.979	0.838
GT SVM, PCA 62	0.690	0.889	0.895	0.892	0.692	0.091	0.979	0.835
U SVM	0.647	0.862	0.863	0.863	0.650	0.122	0.891	–
U SVM, PCA 62	0.647	0.863	0.863	0.863	0.652	0.122	0.893	–

3.6. Comparison of Cluster Quality Measurements between Classifiers Trained on Ground Truth Data and without This Knowledge

As we were unable to calculate the actual accuracy of classification algorithms trained without the ground truth knowledge, we compared their effectiveness with pattern recognition methods trained with this knowledge using metrics that we used before to evaluate the quality of clustering. As can be seen in Tables 9 and 10, there was not much difference between those two groups of methods. Among the available metrics, the Fowlkes–Mallows score (FMS) seems to be most informative, in terms of accuracy. In the case of the CelebA data set, application of PCA dimensionality reduction on the data set had a minimal impact on the overall accuracy. In the case of KNN-1, the accuracy was reduced by 0.1%; in the case of NN, it increased by 0.2% and had no influence on the SVM. This was also true for the CASIA-WebFace data set: in the case of SVM, it reduced the accuracy by 0.3%. In the case of the NN, it increased by 0.1% and had no influence on the KNN-1.

In the CelebA data set, the highest FMS score was obtained for KNN-1 (0.791). In this case, the FMS had the same value as the original data set and that with PCA-reduced dimensionality. When KNN-1 was trained using data clustered by the Agglomerative clustering method with a threshold value of

16, its FMS score was reduced by 5.3%. In the case of NN, this reduction was equal to 3.6% (0.8% for PCA-processed data). In the case of SVM, all clustering parameters beside CS, VMS, and CNR were increased when the algorithm was trained without knowledge of the ground truth.

In the CASIA-WebFace data set, the highest FMS score was obtained for SVM (0.718). When the SVM was trained using data clustered by Agglomerative clustering with a threshold value of 34 and reduced dimensionality, its FMS score was reduced by 6.6%. In the case of NN, this reduction was equal to 6.7% (6.6% for PCA-processed data). In the case of KNN-1, this reduction was equal to 4.6% (4.3% for PCA-processed data).

4. Discussion

It was expected that we would obtain better clustering results with a centroid-based classification approach than with a density-based approach. This was because the deep feature optimization method using the Triplet Loss algorithm resulted in the creation of centroid-based clusters of faces with the same identity. Basing on the results from Table 3 (CelebA data set), we can indicate, without any doubt, that Agglomerate Clustering with a threshold of 16 maximizes the most important clustering quality measures. We also obtained very good results for Agglomerative Clustering with a threshold of 16 and projection of the data set onto 62-dimensional space with PCA. As can be seen from Table 1, the 62-dimensional projection explained over 0.999 of overall variance present in the data set and, at the same time, reduced the dimensionality of the problem by over two times. According to Table 5, the best results were obtained with KNN 1 and SVM/PCA 62; however, very similar results were found in case of SVM without PCA. Although training of a linear SVM classifier is usually more time-consuming than KNN, SVM has two very important advantages over KNN: SVM allows for generalization of the problem and can better deal with outliers.

These results were confirmed by experiments on the CASIA-WebFace data set. In that case, however, Agglomerate Clustering with threshold 34 maximized the most important clustering quality measures (see Table 4). According to Table 6, the best results were obtained with KNN 1 and SVM/PCA 62; however, very similar results were obtained in the case of SVM without PCA.

As was shown in Tables 7 and 8 SC was positively correlated with ARI, VMS, and FMS. We can use SC as an indicator of clustering quality performance.

We can clearly see, from Tables 9 and 10, that the clusters discovered by unsupervised methods can be safely used to train a classifier without knowing the ground truth, as lack of this knowledge does not deteriorate the overall clustering assignments much. This conclusion is especially important in real-world scenarios in which ground truth can be obtained only by manual assignment. Manual assignment is often very time-consuming and expensive for large data sets. We can conclude that, beside highly secure systems in which face verification is a key component, face identities discovered by unsupervised approaches can be safely used for training supervised classifiers. We can also observe that the CASIA-WebFace data set was more difficult to classify than CelebA: both scoring parameters described in Section 2.6.2 and the accuracy given in Tables 9 and 10 had slightly lower values in the case of CASIA-WebFace. This may have been caused by the fact that the pictures in CelebA data set have a visually better image quality.

5. Conclusions

We found that knowledge about ground truth data improves the Fowlkes–Mallows score only by 5.3% (from 0.749 to 0.791) for the classification algorithm with the highest accuracy (namely, KNN-1 in the CelebA data set) and by 6.6% (from 0.652 to 0.718) for the SVM classifier in the CASIA-WebFace data set. As we have already mentioned, beside highly secure systems in which face verification is a key component, the face identity clusters discovered by an unsupervised method can be safely used to train a classifier. Furthermore, we found that the Silhouette Coefficient (SC) of unsupervised clustering was positively correlated with the Adjusted Rand Index, V-measure score, and Fowlkes–Mallows score and, so, we can use the SC as an indicator of clustering performance when the ground truth of

face identities is not known. All of these conclusions are important findings relating to large-scale face verification problems. The reason for this is the fact that skipping the necessity of verification of identities before supervised training saves a lot of time and resources.

Author Contributions: T.H. was responsible for conceptualization, proposed methodology, software implementation, and writing the original draft; P.M. was responsible for data curation and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Pedagogical University of Krakow.

Conflicts of Interest: The author declares no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Alahmadi, A.; Hussain, M.; Aboalsamh, H.; Zuair, M. PCAPool: Unsupervised feature learning for face recognition using PCA, LBP, and pyramid pooling. *Pattern Anal. Appl.* **2019**, *23*, 673–682. [\[CrossRef\]](#)
2. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
3. Datta, S.; Sharma, G.; Jawahar, C.V. Unsupervised Learning of Face Representations. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; pp. 135–142.
4. Menezes, A.G.; Sá, d.C.J.M.D.; Llapa, E.; Estombelo-Montesco, C.A. Automatic Attendance Management System based on Deep One-Shot Learning. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 3–5 June 2020; pp. 137–142.
5. Glowacz, A.; Glowacz, Z. Recognition of images of finger skin with application of histogram, image filtration and K-NN classifier. *Biocybern. Biomed. Eng.* **2016**, *36*. [\[CrossRef\]](#)
6. Li, J.; Chai, W.; Hu, J.; Deng, W. AF-Softmax for Face Recognition. In Proceedings of the 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), Guiyang, China, 22–24 August 2018; pp. 358–362.
7. Gallo, I.; Nawaz, S.; Calefati, A.; Piccoli, G. A Pipeline to Improve Face Recognition Datasets and Applications. In Proceedings of the 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ), Auckland, New Zealand, 19–21 November 2018; pp. 1–6.
8. Hong, S.; Ryu, J. Attention-Guided Adaptation Factors for Unsupervised Facial Domain Adaptation. *Electron. Lett.* **2020**, *56*, 816–818. [\[CrossRef\]](#)
9. Luo, Z.; Hu, J.; Deng, W.; Shen, H. Deep Unsupervised Domain Adaptation for Face Recognition. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; pp. 453–457.
10. ElSayed, A.; Kongar, E.; Mahmood, A.; Sobh, T. Unsupervised face recognition in the wild using high-dimensional features under super-resolution and 3D alignment effect. *Signal Image Video Process.* **2018**, *12*, 1353–1360. [\[CrossRef\]](#)
11. Hong, S.; Ryu, J. Unsupervised Face Domain Transfer for Low-Resolution Face Recognition. *IEEE Signal Process. Lett.* **2020**, *27*, 156–160. [\[CrossRef\]](#)
12. Qian, Y.; Deng, W.; Hu, J. Unsupervised Face Normalization With Extreme Pose and Expression in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
13. Zhang, M.; Khan, S.; Yan, H. Deep eigen-filters for face recognition: Feature representation via unsupervised multi-structure filter learning. *Pattern Recognit.* **2019**, *100*, 107176. [\[CrossRef\]](#)
14. Kumar, V.; Kalitin, D.; Tiwari, P. Unsupervised learning dimensionality reduction algorithm PCA for face recognition. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 32–37.
15. Terhorst, P.; Kolf, J.N.; Damer, N.; Kirchbuchner, F.; Kuijper, A. SER-FIQ: Unsupervised Estimation of Face Image Quality Based on Stochastic Embedding Robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.

16. Pal, D.K.; Juefei-Xu, F.; Savvides, M. Discriminative Invariant Kernel Features: A Bells-and-Whistles-Free Approach to Unsupervised Face Recognition and Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
17. Oszust, M.; Piórkowski, A.; Obuchowicz, R. No-reference image quality assessment of magnetic resonance images with high-boost filtering and local features. *Magn. Reson. Med.* **2020**, *84*. [\[CrossRef\]](#)
18. Wang, Y.; Shen, J.; Petridis, S.; Pantic, M. A real-time and unsupervised face Re-Identification system for Human-Robot Interaction. *Pattern Recognit. Lett.* **2018**, *128*, 558–569. [\[CrossRef\]](#)
19. Cheng, Y.; Li, Y.; Liu, Q.; Yao, Y.; Sai Vijay Kumar Pedapudi, V.; Fan, X.; Su, C.; Shen, S. A Graph Based Unsupervised Feature Aggregation for Face Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Korea, 27 October–2 November 2019.
20. Li, S.; Deng, W. Deep Facial Expression Recognition: A Survey. *IEEE Trans. Affect. Comput.* **2020**, *99*. [\[CrossRef\]](#)
21. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* **2019**, *189*, 102805. [\[CrossRef\]](#)
22. Jayaraman, U.; Gupta, P.; Gupta, S.; Arora, G.; Tiwari, K. Recent Development in Face Recognition. *Neurocomputing* **2020**, *408*, 231–245. [\[CrossRef\]](#)
23. Safaa El-Din, Y.; Moustafa, M.N.; Mahdi, H. Deep convolutional neural networks for face and iris presentation attack detection: Survey and case study. *IET Biom.* **2020**, *9*, 179–193. [\[CrossRef\]](#)
24. Lahasan, B.; Lutfi, S.; San-Segundo, R. A survey on techniques to handle face recognition challenges: occlusion, single sample per subject and expression. *Artif. Intell. Rev.* **2017**, *52*. [\[CrossRef\]](#)
25. Sawant, M.; Bhurchandi, K. Age invariant face recognition: A survey on facial aging databases, techniques and effect of aging. *Artif. Intell. Rev.* **2018**, *52*. [\[CrossRef\]](#)
26. Scherhag, U.; Rathgeb, C.; Merkle, J.; Breithaupt, R.; Busch, C. Face Recognition Systems Under Morphing Attacks: A Survey. *IEEE Access* **2019**, *7*, 23012–23026. [\[CrossRef\]](#)
27. Arozi, M.; Caesarendra, W.; Ariyanto, M.; Muna, M.; Setiawan, J.; Glowacz, A. Pattern Recognition of Single-Channel sEMG Signal Using PCA and ANN Method to Classify Nine Hand Movements. *Symmetry* **2020**, *12*, 541. [\[CrossRef\]](#)
28. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [\[CrossRef\]](#)
29. Kwon, H.; Kwon, O.; Yoon, H.; Park, K. Face Friend-Safe Adversarial Example on Face Recognition System. In Proceedings of the 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2–5 July 2019; pp. 547–551. [\[CrossRef\]](#)
30. Yeh, C.Y.; Chen, R.H. A Smart Reminder for Social Purposes Using a Deep Learning-Based Face Recognition Technique. *IEEE Trans. Electr. Electron. Eng.* **2020**, *1*–5. [\[CrossRef\]](#)
31. Dua, M.; Shakshi, Singla, R.; Raj, S.; Jangra, A. Deep CNN models-based ensemble approach to driver drowsiness detection. *Neural Comput. Appl.* **2020**, *1*–14. [\[CrossRef\]](#)
32. Lin, F.C.; Ngo, H.H.; Dow, C.R. A cloud-based face video retrieval system with deep learning. *J. Supercomput.* **2020**, *1*–21. [\[CrossRef\]](#)
33. Sreenu, G.; Durai, M.A. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *J. Big Data* **2019**, *6*, 48. [\[CrossRef\]](#)
34. Jolliffe, I.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [\[CrossRef\]](#)
35. Campello, R.; Moulavi, D.; Sander, J. *Density-Based Clustering Based on Hierarchical Density Estimates*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7819; pp. 160–172. [\[CrossRef\]](#)
36. Otair, M. Approximate K-Nearest Neighbour Based Spatial Clustering Using K-D Tree. *Int. J. Database Manag. Syst.* **2013**, *5*. [\[CrossRef\]](#)
37. Keerthi, S.; Sellamanickam, S.; Chang, K.W.; Hsieh, C.J.; Lin, C.J. A sequential dual method for large scale multi-class linear SVMs. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 408–416. [\[CrossRef\]](#)
38. Chiang, W.L.; Lee, M.C.; Lin, C.J. Parallel Dual Coordinate Descent Method for Large-scale Linear Classification in Multi-core Environments. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1485–1494. [\[CrossRef\]](#)

39. Zhuang, Y.; Juan, Y.; Yuan, G.X.; Lin, C.J. Naive Parallelization of Coordinate Descent Methods and an Application on Multi-core L1-regularized Classification. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 1103–1112. [\[CrossRef\]](#)
40. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Inf. Process. Syst.* **2012**, *25*. [\[CrossRef\]](#)
41. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
42. Jacomy, M.; Venturini, T.; Heymann, S.; Bastian, M. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS ONE* **2014**, *9*, e98679. [\[CrossRef\]](#)
43. Yi, D.; Lei, Z.; Liao, S.; Li, S. Learning Face Representation from Scratch. *arXiv* **2014**, arXiv:1411.7923.
44. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [\[CrossRef\]](#)
45. Rosenberg, A.; Hirschberg, J. 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), Prague, Czech Republic, 28–30 June 2007; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp. 410–420.
46. Fowlkes, E.B.; Mallows, C.L. A Method for Comparing Two Hierarchical Clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [\[CrossRef\]](#)
47. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [\[CrossRef\]](#)
48. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.* **1974**, *3*, 1–27. [\[CrossRef\]](#)
49. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*. [\[CrossRef\]](#)
50. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874. [\[CrossRef\]](#)

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).