Deep Density Clustering of Unconstrained Faces

Wei-An Lin Jun-Cheng Chen Carlos D. Castillo Rama Chellappa
University of Maryland, College Park
walin@umd.edu pullpull@cs.umd.edu carlos@cs.umd.edu rama@umiacs.umd.edu

Abstract

In this paper, we consider the problem of grouping a collection of unconstrained face images in which the number of subjects is not known. We propose an unsupervised clustering algorithm called Deep Density Clustering (DDC) which is based on measuring density affinities between local neighborhoods in the feature space. By learning the minimal covering sphere for each neighborhood, information about the underlying structure is encapsulated. The encapsulation is also capable of locating high-density region of the neighborhood, which aids in measuring the neighborhood similarity. We theoretically show that the encapsulation asymptotically converges to a Parzen window density estimator. Our experiments show that DDC is a superior candidate for clustering unconstrained faces when the number of subjects is unknown. Unlike conventional linkage and density-based methods that are sensitive to the selection operating points, DDC attains more consistent and improved performance. Furthermore, the density-aware property reduces the difficulty in finding appropriate operating points.

1. Introduction

Given a collection of unseen face images, humans have the capability of grouping and summarizing how many distinct subjects are present by exploiting previously learned knowledge about essential components of a face and possible variations of faces from the same person. In computer vision research, this corresponds to the task of grouping visual data into clusters with targeted semantics. Most existing unsupervised algorithms group data into visually similar clusters, unaware of the underlying semantics. The success in clustering handwritten digits or faces appearing in consecutive video frames is mainly based on the fact that images that belong to the same category are visually similar. For visual data that have extreme intra-class variations, these methods may not be applicable. In this work, we focus on clustering unconstrained face images without prior knowledge of the number of distinct subjects. Visual variations caused by nuisance factors such as pose, illumination and expressions may be larger than variations between subjects. To our knowledge, few previous works have addressed this challenging problem. Recent works on face clustering first extract feature vectors using deep neural networks (DNNs), and then group data directly in the feature space. Face clustering based on deep features generally has advantages over other unsupervised methods due to side information present in the training data. However, since clustering algorithms generally deal with *unseen* data, these methods will suffer from the shift in data distribution across different domains. Therefore, the underlying structure should be considered to prevent performance degradation.

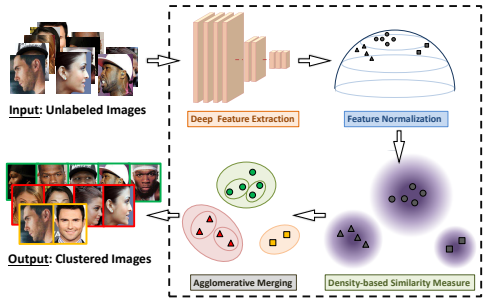


Figure 1: We introduce Deep Density Clustering (DDC) for unconstrained face images. DDC is a density-based clustering algorithm, which exploits the local structure of deep features for improved similarity measure.

To tackle the challenges discussed above, we propose a clustering framework, named Deep Density Clustering (DDC) that exploits the neighborhood structure of deep representations. DDC consists of three main steps: extracting deep features, computing density-based similarity, and merging clusters. The novelty is mainly in the second step: DDC first associates each data point with an ϵ-neighborhood. Points inside the neighborhood are then represented by a minimal covering sphere which encapsulates local information. Finally, DDC computes pairwise similarity by evaluating data points on the functionals defined by

the spheres.

To summarize, we make the following contributions:

- A new approach for characterizing a collection of data points that encapsulates sufficient structural information in the deep feature space.
- A new method, DDC algorithm, for clustering unconstrained face images without prior knowledge of the number of subjects. We argue that information about local structures should be included in the linkage criterion, and propose a novel similarity measure based on local density levels. We theoretically show that the similarity measure is asymptotically a Parzen window density estimator.

The remainder of this paper is organized as follows: We first discuss the related works in unsupervised deep clustering, unconstrained face clustering, and deep representation. Then we introduce the proposed face clustering algorithm. Finally, we detail our experiments and discuss the impact of the proposed method.

2. Related Works

In this section, we briefly introduce recent advances in unsupervised clustering and unconstrained face clustering using deep representations.

2.1. General Clustering Algorithms

Conventional clustering algorithms typically rely on the absolute distance defined in the embedded space. Several recent clustering algorithms, however, have shown that in addition to point-to-point topology, high-level structure could be incorporated for improved clustering performance. For example, sparse subspace clustering (SSC) [3] exploits the underlying linear subspace structure within data. Several different extensions of the SSC algorithm [19, 18, 35, 20] have yielded impressive results on MNIST [13] and Extended Yale B [6] datasets. However, the SSC algorithm relies on the assumption that the given dataset can be well-approximated by a union of low-dimensional subspaces, which may not be true for unconstrained face images.

2.2. Deep Unsupervised Clustering Algorithms

Recently, deep neural networks (DNNs) are extensively used to learn representation and clusters. In [33], a recurrent framework that successively updates representations and clusters is proposed. Although good results are achieved, it requires tuning a large number of hyperparameters and repeated training of deep networks. In [31, 32, 7] encoder-decoder structures are used to learn low-dimensional embeddings and cluster assignments. Xie *et al*. [31] proposed to first learn deep representations using a stacked autoencoder. Cluster assignments are then iteratively refined by minimizing the KL divergence between the soft assignments and the target distribution. Yang *et al*. introduce a joint dimensionality reduction and clustering approach that learns a clustering-friendly latent representations. Dizaji *et al*. [7] proposed an end-to-end clustering framework, named DEPICT. They derived a regularized relative entropy loss function to encourage balanced clusters. In addition, the joint framework avoids layer-wise training and is computationally more efficient. Ji *et al*. [11] proposed the deep subspace clustering network which uses a novel self-expressive layer to mimic the self-expressiveness property. One major drawback of this method is that the number of parameters for the self-expressive layer scales quadratically with the number of images.

While successful in some applications, these methods generally require exact knowledge of the number of categories [33, 31, 32, 7, 11], layer-wise pretraining [31, 32, 11], and tuning network structures [33, 31, 32, 11]. Furthermore, it is not clear whether clustering based on the encoder-decoder structure could be scaled to datasets with a large number of categories. In fact, the evaluations of these approaches are limited to number of clusters that are less than a hundred. The proposed DDC algorithm, on the other hand, does not require the number of categories as a prior, and is also evaluated on challenging unconstrained datasets that have more than one thousand categories.

2.3. Unconstrained Face Clustering

Otto *et al*. [17] developed an efficient algorithm called the approximated rank-order clustering that measures pairwise similarity based on the number of shared nearest neighbors. The approach of capturing the high-level structure is efficient when most of the identities have only a few instances. However, when the dataset contains more large clusters, the loss of original point-to-point topology would adversely affect the performance. Lin *et al*. [14] proposed the proximity-aware hierarchical clustering (PAHC) which exploits neighborhood similarity based on linear SVMs that separates local positive instances and negative instances. While improved results are achieved on unconstrained face datasets, it was applied to group faces with balanced cluster size. Unlike PAHC, the proposed method can be applied to face images with large variations in cluster sizes. Shi *et al*. [25] proposed the ConPaC algorithm in which the clustering problem is formulated as a conditional random field model. By maximizing the posterior probability of the adjacency matrix, improved performance is achieved on the recently released IJB-B dataset. However, their approach does not scale well in speed. The proposed DDC algorithms, on the other hand, could run significantly faster than the ConPaC algorithm. Jin *et al*. [12] proposed the Erdős-Rényi clustering algorithm for joint face detection and clustering in videos. The algorithm is based on the rank-1 count simi-

larity which requires a reference set. In their work, a collection of frames are sampled as the reference set that are likely to have similar distribution to the target distribution. However, collecting such reference set for general face clustering is not an easy task. Unlike the Erdős-Rényi clustering algorithm, our approach does not require a domain-specific reference set.

2.4. Deep Face Representations

Deep convolutional neural networks (DCNNs) have been widely used for face classification [26, 24, 15]. A DCNN trained on labeled face images is able to separate faces from distinct identities in the embedded feature space. In this work, we use deep face representations for *unseen* face images to retain sufficient amount of semantic information for distinguishing different identities.

3. Proposed Method

For an unlabeled dataset $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$, the goal of unsupervised clustering algorithm is to find proper cluster assignment for each data point, such that data of the same state-of-the-nature identity are grouped together. In this work, we consider X as a collection of unconstrained face images with unknown number of subjects. We adopt the basic average-linkage clustering approach, in which pairs of face images are grouped according to (1) the distance measure in the embedded space and (2) the average linkage criterion that measures the dissimilarity between two groups of face images.

For unconstrained face images, within-subject variations could be larger than between-subject variations. To capture sufficient amount of semantic information for distinguishing different subjects, face images are first projected into the embedded space using a DNN $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^d$. Recent works [15] on deep representations have shown that for DNNs trained with softmax loss, label prediction is mainly determined by angular similarities to each class. Therefore, we consider cosine distance as the distance measure in the feature space. Without loss of generality, $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{S}^{d-1}$ is used to represent a DNN, where $\mathbb{S}^{d-1}$ is a unit hypersphere.

3.1. Key Observations

In this section, we first show that point-to-point distance measurement might be insufficient, and then describe the motivation for the proposed method.

In Figure 2, the average linkage between two groups of points C_i and C_j determines whether they should be merged together. By definition, if the distance measure is d, the average linkage is calculated by

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{u}\in C_i, \boldsymbol{v}\in C_j} d(\boldsymbol{u}, \boldsymbol{v}). \qquad (1)$$

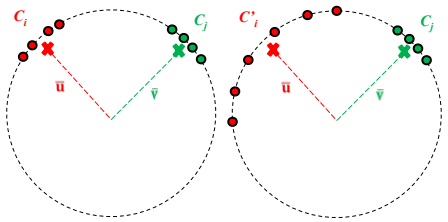


Figure 2: Linkage computation for two groups of data points on a circle. It is clear that after averaging, $\bar{u}$ and $\bar{v}$ fail to represent whether the original group of points are sparsely or densely distributed.

For data points that lie on a unit hypersphere S^{d-1}, (1) equals to $1-\bar{\boldsymbol{u}}^T\bar{\boldsymbol{v}}$, which is equivalent to the cosine distance between arithmetic averages $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$. Note that local information about C_i is not retained in $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$. One can find another sparsely distributed C_i' with the same $\bar{\boldsymbol{u}}$. However, merging C_i' and C_j is less desirable since the cluster $C_i' \cup C_j$ is less homogeneous than $C_i \cup C_j$. We argue that neighborhood information should be aggregated during linkage computation in order to differentiate merging C_i or C_i' with C_j. Specifically, when measuring the distance between two points, their neighboring points should also be considered. Following this observation, we propose a new similarity measure based on the following steps: (1) building a nearest-neighbor graph for the entire dataset, which will be described in Section 3.2, (2) representing each neighborhood in a compact form, which will be discussed in Section 3.3, and (3) computing a density-based similarity, which will be described in Section 3.4. We name the proposed method Deep Density Clustering since the similarity measure is asymptotically a Parzen window density estimator as will be proved in Section 3.4.

3.2. Nearest-Neighbor Graph Construction

We can view a set of data points as a union of local neighborhoods. Namely, we can write $X = \bigcup_{m=1}^{N} V(\boldsymbol{x}_m)$, where $V(\boldsymbol{x}_m)$ consists of neighboring points of $\boldsymbol{x}_m$ measured in the feature space. Common approaches to constructing local neighborhoods include k-nearest neighbors and ϵ-neighborhood. We construct $V(\boldsymbol{x}_m)$ based on the ϵ-neighborhood approach since it is more robust to density variations, and as $N \rightarrow \infty$, $|V(\boldsymbol{x})| \rightarrow \infty$ holds, which achieves the asymptotic property that will be discussed in subsequent sections. However, proper selection of ϵ is not trivial and usually depends on the representation. In this work, we propose to select ϵ as the maximum likelihood (ML) estimator of the cosine distance between *matched pairs* (image pairs belong to the same subjects). Formally,

$$\epsilon = \operatorname*{argmax}_d \mathbb{E}_c[p(d\,|c)], \qquad (2)$$

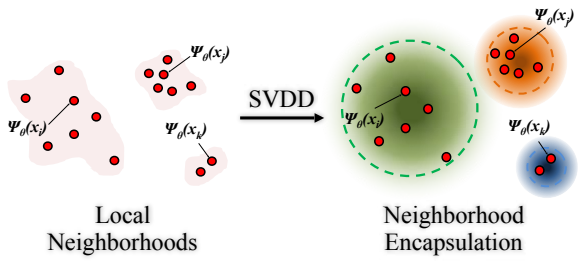


Figure 3: Neighborhood encapsulation. (left) Pink regions are the local neighborhoods of the points x_i, x_j, and x_k in feature space. (right) Encapsulations are learned by solving (3). The encapsulation is density-aware. In the figure, regions closer to the centers of the spheres have higher density.

where c is the subject label. The matched pairs can be sampled from the training data or an external dataset. Details about the selection of ϵ will be presented in Section 4.1.

3.3. Local Neighborhood Encapsulation

A trivial way of characterizing points in a neighborhood is to store all the points, however, this representation will not be useful. We propose to encapsulate each local neighborhood in a hypersphere which retains information about local structure. This is inspired by the SVDD algorithm [28] that describes a collection of data by finding a sphere that covers all the target data while including no superfluous space. Instead of the entire dataset, we apply SVDD to all the local neighborhoods. For each $V(x_m)$, we solve for its encapsulation using the following optimization:

$$\begin{aligned} \min_{\boldsymbol{c}_m, \bar{R}_m, \boldsymbol{\xi}_m} \quad & \bar{R}_m + \frac{1}{\nu \cdot n_V} \sum_{\boldsymbol{z} \in V(\boldsymbol{x}_m)} \xi_m(\boldsymbol{z}) \\ \text{s.t.} \quad & \|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}_m\|^2 \leq \bar{R}_m + \xi_m(\boldsymbol{z}), \\ & \xi_m \geq 0, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}_m), \end{aligned} \tag{3}$$

where $\bar{R}_m = R_m^2$ is the squared radius and n_V is the size of $V(\boldsymbol{x}_m)$. Note that in (3), instead of minimizing over R_m, we aim to solve for optimal $\bar{R}_m^*$ since the original formulation in [28] is not convex. Readers are referred to [2] for more details. After solving (3) for $m = 1, \cdots, N$, the resulting collection of spheres $\{(\boldsymbol{c}_m^*, R_m^*)\}_{m=1}^N$ minimally covers each local neighborhood as demonstrated in Figure 3. In what follows, when there is no confusion possible, we will drop the subscript m in (3) for more compact notations.

3.3.1 Relation to One-Class SVM

One-class SVM (OC-SVM) was first proposed in [23] to build a representational model for a given dataset. Suppose we choose the set as $V(\boldsymbol{x})$, then OC-SVM aims to solve the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, \rho, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu \cdot n_V} \sum_{\boldsymbol{z} \in V(\boldsymbol{x})} \xi_{\boldsymbol{z}} - \rho \\ \text{s.t.} \quad & \boldsymbol{w}^T \Psi_\theta(\boldsymbol{z}) \geq \rho - \xi_{\boldsymbol{z}}, \\ & \xi_{\boldsymbol{z}} \geq 0, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}). \end{aligned} \tag{4}$$

The optimal hyperplane separates the data with the origin in feature space and maximizes the distance from the hyperplane to the origin. We present the following Lemma showing equivalence between the formulations in (3) and (4). The proof for the lemma is provided in the supplementary material.

Lemma 1. *If $1/n_V < \nu \leq 1$, the SVDD formulation in (3) is equivalent to the OC-SVM formulation in (4) when the evaluation functions for the two are given by*

$$h_{SVDD}(\boldsymbol{x}) = \bar{R}^* - \|\Psi_\theta(\boldsymbol{x}) - \boldsymbol{c}^*\|^2, \tag{5}$$

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \boldsymbol{w}^{*T} \Psi_\theta(\boldsymbol{x}) - \rho^*, \tag{6}$$

with the correspondence $\boldsymbol{w}^* = \boldsymbol{c}^*$, and $\rho^* = \boldsymbol{c}^{*T} \Psi_\theta(\boldsymbol{x}_s)$, where $\boldsymbol{x}_s$ is a support vector in (3) that lies on the learned enclosing sphere.

Intuitively, the evaluation functions (5) and (6) measures the closeness to the neighborhood $V(\boldsymbol{x})$.

3.4. Density-based Similarity Measure

Our goal is to associate each pair of points with a similarity measure. We first use the following theorem to show the evaluation function defined in (5) is a local density estimator. The detailed proof is provided in the supplementary material.

Theorem 1. *If $1/n_V < \nu \leq 1$ and $\boldsymbol{c}^{*T} \Psi_\theta(\boldsymbol{x}_s) \neq 0$ for some support vector $\boldsymbol{x}_s$, $h_{SVDD}(\boldsymbol{x})$ defined in (5) is asymptotically a Parzen window density estimator in the feature space with Epanechnikov kernel.*

Proof. Given the condition, according to Lemma 1, $h_{SVDD}(\boldsymbol{x})$ is equivalent to $h_{OC\text{-}SVM}(\boldsymbol{x})$ with $\rho^* \neq 0$. From the results in [21] and the fact that $\sum \alpha_i = 1$ in the dual formulations of (3) and (4), it can be shown that

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \frac{8}{3} \sum_{i=1}^{n_V} \alpha_i K_E \left(\frac{\|\Psi_\theta(\boldsymbol{x}) - \Psi_\theta(\boldsymbol{x}_i)\|}{2} \right) - \rho^* - 1,$$

where $K_E(u) = \frac{3}{4}(1 - u^2)$, $|u| \leq 1$ is the Epanechnikov kernel. As a consequence of Proposition 4 in [21] and the proof of Proposition 1 in [22], when $n_V \to \infty$, the fraction of support vector is ν, and the fraction of points with $0 < \alpha_i < 1/(\nu \cdot n_V)$ vanishes. Therefore, either $\alpha_i = 0$ or

$\alpha_i = 1/(\nu \cdot n_V)$. By introducing the notation $\bar{S} = \{i \mid \alpha_i = 1/(\nu \cdot n_V)\}$, it can be shown that

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \frac{2^{d+3}}{3} \hat{f}\left(\Psi_\theta(\boldsymbol{x})\right) - \rho^* - 1, \tag{7}$$

where $\hat{f}(\boldsymbol{z}) = \frac{1}{\nu \cdot n_V \cdot 2^d} \sum_{s \in \bar{S}} K_E\left(\frac{\|\boldsymbol{z}_s - \boldsymbol{z}\|}{2}\right)$ is a density estimator. As a result, $h_{SVDD}(\boldsymbol{x})$ is equivalent to a Parzen window density estimator with Epanechnikov kernel of bandwidth 2. By scaling properly, Parzen window estimator with different bandwidths can be obtained. □

According to Theorem 1, we associate each neighborhood $V(\boldsymbol{x}_m)$ with a density estimator $\mathcal{E}_m : \mathbb{R}^D \to \mathbb{R}$:

$$\mathcal{E}_m(\boldsymbol{x}) = \bar{R}_m^* - \left\|\Psi_\theta(\boldsymbol{x}) - \boldsymbol{c}_m^*\right\|^2. \tag{8}$$

Data points that yield smaller $\mathcal{E}_m$ lie in low-density region of $V(\boldsymbol{x}_m)$ and are therefore less similar to the neighborhood $V(\boldsymbol{x}_m)$. This leads to a similarity measure between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, which is defined as:

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{2}\left[\frac{\sum_{\boldsymbol{z} \in V(\boldsymbol{x}_j)} \mathcal{E}_i(\boldsymbol{z})}{|V(\boldsymbol{x}_j)|} + \frac{\sum_{z \in V(\boldsymbol{x}_i)} \mathcal{E}_j(\boldsymbol{z})}{|V(\boldsymbol{x}_i)|}\right]. \tag{9}$$

The distance between pairs of data samples can be taken as a proper monotonically decreasing function of $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

3.5. Negative Set Mining

In [27], the authors proposed that when negative samples are available for SVDD, they can be incorporated to improve the description. Specifically, the enclosing sphere is refined by modifying the constraints in the following way:

$$\left\|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}\right\|^2 \leq \bar{R} + \xi, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}), \tag{10}$$

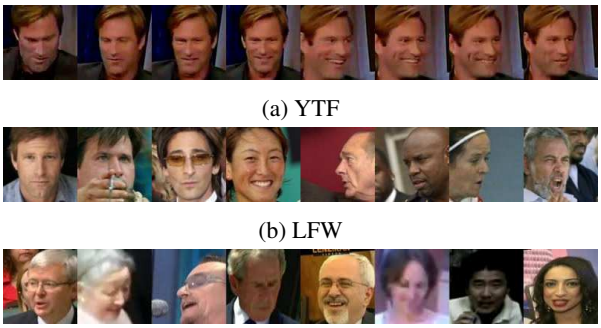
$$\left\|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}\right\|^2 \geq \bar{R} - \xi, \quad \forall \boldsymbol{z} \in V^-(\boldsymbol{x}), \tag{11}$$

where $V^-(\boldsymbol{x}) \subset X$ contains instances which are *hard negatives* of $\boldsymbol{x}$. However, since no label information about $\boldsymbol{x}$ is available, the general notion of negative samples is not well-defined. Instead, we view the selection of hard negative samples as finding a balance between the amount of false positives and the false negatives in binary hypothesis testing formulation. Specifically, points in $V^-(\boldsymbol{x})$ are sampled from $\{\boldsymbol{x}' : d(\boldsymbol{x}', \boldsymbol{x}) > \eta\}$, where η is chosen to minimize the misclassification rate. In other words, we assign the same risk function to the action of selecting false positives and false negatives. Details about the selection of η will be presented in Section 4.1.

To incorporate the negative samples, we make the following observations. From the equivalence of (3) and (4), when no negative samples are available, encapsulations are learned by separating the data points against the origin with a margin ρ. Note that the $-\rho$ in (4) encourages large separation with the origin. In the presence of negative samples, $-\rho$ is no longer required, and the hyperplane in (4) should target at separating positive and negative samples. We propose to learn the set of enclosing spheres such that positive and negative examples are separated by a margin Δ. Therefore, by the equivalence of (5) and (6) and the arguments given above, we formulate the algorithm with negative set mining as:

$$\begin{aligned} \min_{\boldsymbol{w}, \rho, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum \xi_{\boldsymbol{z}} \\ \text{s.t.} \quad & \boldsymbol{w}^T\Psi_\theta(\boldsymbol{z}) - \rho \geq \Delta - \xi_{\boldsymbol{z}}, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}), \\ & \boldsymbol{w}^T\Psi_\theta(\boldsymbol{z}) - \rho \leq -\Delta + \xi_{\boldsymbol{z}}, \forall \boldsymbol{z} \in V^-(\boldsymbol{x}), \\ & \xi_{\boldsymbol{z}} \geq 0, \qquad \forall \boldsymbol{z}. \end{aligned} \tag{12}$$

Note that Δ is not a hyperparameter since we can divide both sides of the constraints by Δ and obtain a large margin formulation with L_1 normalization.



(a) YTF

(b) LFW

(c) IJB-B

Figure 4: Sample images for the datasets.

4. Evaluation and Discussion

In this section, we evaluate the proposed clustering approach on YouTube Faces Database (YTF), Labeled Faces in the Wild (LFW) and IARPA JANUS Benchmark B (IJB-B) datasets. The datasets are briefly described as follows:

- **YouTube Faces Database (YTF) [30]:** The dataset contains 3,425 videos of 1,595 different people. We choose the first 41 subjects from the YTF dataset as in [33, 7].
- **Labeled Faces in the Wild (LFW) [10]:** It is a well-known and standard dataset for unconstrained face recognition which contains 13,233 images of 5749 subjects. Note that 4169 subjects of the dataset have only one image. We evaluate the proposed approach using the entire dataset.

- **IARPA JANUS Benchmark B (IJB-B) [29]:** The IJB-B dataset contains 1,845 subjects with 11,754 images, 55026 video frames and 10,044 nonface images. It contains a clustering protocol, which consists of seven subtasks. These subtasks differ in the number of distinct identities and the number of face images. Many face images are in extreme poses or of low quality, making the dataset more challenging than YTF and LFW. We evaluate clustering algorithms on four subtasks with number of identities 128, 256, 1024 and 1845. The results for the remaining three subtasks are presented in the supplementary materials.

Dataset	# Samples	# Subjects
YTF	10,000	41
LFW	13,233	5,749
IJB-B-128	5,224	128
IJB-B-256	9,867	256
IJB-B-1024	36,575	1,024
IJB-B-1845	68,195	1,845

Table 1: Datasets used in the experiments.

4.1. Implementation Details

Deep Face Representation. We adopt the network architecture presented in [36]. The network is first trained on the CASIA-WebFace dataset [34] using SGD for 750K iterations with a standard batch size 128 and momentum 0.9. Then, the model is finetuned for 230K iterations using the MSCeleb-1M dataset [9]. The inputs to the networks are 100×100×3 RGB images. Data augmentation is performed by randomly cropping and horizontally flipping face images. Given a face image, the deep representation is extracted from the `pool5` layer with dimension 320. The training and preprocessing details are provided in the supplementary materials.

Parameter Selection. There are two main hyperparameters in the proposed approach: ϵ for constructing neighborhoods and η for mining hard negatives. To select ϵ, we follow (2) by randomly sampling 100 subjects from the training dataset and computing cosine distance between all matched pairs. The red curve in Figure 5 represents the fitted distribution. The ML estimate is therefore $\epsilon \approx 0.23$. The green curve in Figure 5 represents the distribution of the cosine distance between mismatched pairs among the 100 subjects. From Figure 5, it is clear that $\eta \approx 0.40$ minimizes the Bayesian risk of selecting false positives and false negatives.

We use the default parameters provided with the code [1] when solving (3) or (12).

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/

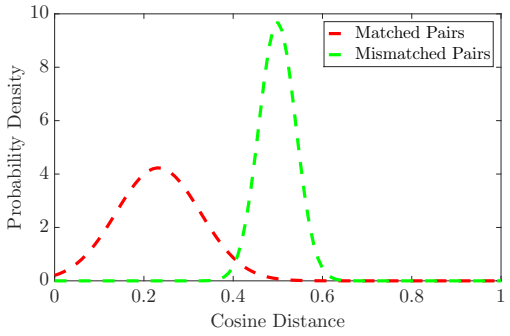


Figure 5: Distribution of cosine distance from the training dataset.

4.2. Evaluation Metrics

To evaluate clustering algorithms, we adopt two measures: normalized mutual information (NMI) and BCubed F-measure [1].

NMI is a widely used metric that measures the normalized similarity between the ground truth labels and the labels decided by the clustering algorithms. NMI is suitable for evaluation when the number of clusters is assumed to be a known quantity. However, when the number of clusters is unknown or is the quantity we are trying to estimate, NMI may fail to penalize algorithms that yield over-clusterings. We use NMI mainly for comparing with other state-of-the-art unsupervised image clustering methods.

BCubed F-measure [1] is the harmonic mean of BCubed precision and BCubed recall. BCubed precision calculates the fraction of points in the same cluster that belong to the same class. BCubed recall calculates the fraction of points in the same class that are assigned to the same cluster. Formally, for an item e, $C(e)$ and $L(e)$ are used to denote its cluster and ground truth label, respectively. For a pair of items e and e', the relation $\text{Correct}(e, e')$ is defined as:

$$\text{Correct}(e, e') = \begin{cases} 1, & \text{if } C(e) = C(e') \text{ and } L(e) = L(e'), \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The BCubed Precision, BCubed Recall, and BCubed F-measure are defined as:

$$\text{Precision} = \text{Avg}_e[\text{Avg}_{e':C(e')=C(e)}[\text{Correct}(e, e')]], \tag{14}$$

$$\text{Recall} = \text{Avg}_e[\text{Avg}_{e':L(e')=L(e)}[\text{Correct}(e, e')]]. \tag{15}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{16}$$

BCubed Precision and Recall can be used to evaluate clustering algorithms that yield different number of clusters.

Methods		DDC-NEG	DDC	PAHC	DBSCAN	AHC
YTF	0.70	1.000	1.000	0.168	1.000	1.000
	0.75	1.000	1.000	0.144	1.000	1.000
	0.80	0.999	1.000	0.120	1.000	0.999
	0.85	0.958	0.966	0.098	0.990	0.948
LFW	0.70	0.994	0.992	0.976	1.000	1.000
	0.75	0.991	0.991	0.956	0.995	1.000
	0.80	0.991	0.991	0.919	0.936	0.994
	0.85	0.990	0.990	0.822	0.664	0.990
IJB-B-128	0.70	**0.966**	**0.960**	0.431	0.842	0.947
	0.75	**0.913**	**0.857**	0.253	0.705	**0.913**
	0.80	**0.786**	0.504	0.172	0.461	**0.679**
	0.85	**0.411**	0.225	0.156	**0.275**	0.253
IJB-B-256	0.70	**0.937**	0.901	0.169	0.725	**0.915**
	0.75	**0.893**	0.760	0.132	0.592	**0.868**
	0.80	**0.620**	0.396	0.102	0.395	**0.524**
	0.85	**0.181**	0.126	0.079	**0.230**	0.139
IJB-B-1024	0.70	**0.798**	0.616	0.087	0.485	**0.735**
	0.75	**0.459**	0.210	0.053	**0.347**	0.307
	0.80	**0.105**	0.101	0.038	**0.241**	0.055
	0.85	0.050	**0.066**	0.022	**0.157**	0.025
IJB-B-1845	0.70	**0.771**	0.610	0.059	0.492	**0.690**
	0.75	**0.341**	0.204	0.045	**0.350**	0.235
	0.80	**0.083**	0.081	0.031	**0.233**	0.052
	0.85	**0.068**	0.051	0.018	**0.151**	0.019

Table 2: BCubed precision evaluated at different BCubed recall values. The best performance is reported using **bold red**, and the second best is reported using **bold blue**.

They satisfy several formal constraints on evaluation metrics, and is shown to be more suitable than metrics based on set matching, pair counting, entropy or editing distance [1].

4.3. Baseline Methods

We compare the proposed DDC algorithm, DDC with negative set mining (DDC-NEG), with the following methods: Agglomerative Hierarchical Clustering (AHC) [8], K-means [16], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [4], Affinity Propagation (AP) [5], Sparse Subspace Clustering using Orthogonal Matching Pursuit (SSC-OMP) [35], Joint Unsupervised Learning of deep representations and clusters (JULE) [33], Deep Embedded Regularized Clustering (DEPICT) [7], Proximity-Aware Hierarchical Clustering (PAHC) [14], Approximate Rank-Order Clustering (ARO) [17], and Conditional Pairwise Clustering (ConPaC) [25].

Precision and Recall Comparisons. Table 2 shows the BCubed precision measured at different BCubed recall for methods that yields different number of clusters. On the YTF and LFW datasets, all methods except PAHC and DBSCAN attains near-perfect performance. On the more-challenging IJB-B dataset, the proposed approach performs the best across several subtasks. It should be noted that the proposed approach has consistent behavior across different operating points and dataset scales, while the basic AHC achieves degraded performance at higher recall regions for larger scale data, and DBSCAN is inferior at lower recall regions.

F-measure and NMI Comparisons. Table 4 reports the F-measure and NMI comparisons. Some experiments for SSC-OMP do not finish within the cut-off threshold of ten hours and are replaced by double dash marks (- -). Results reported from the original papers are marked by asterisks (*). As shown in the table, the proposed DDC and DDC-NEG outperforms other methods. Although AHC achieves high F-measure and NMI using the oracle supplied threshold, it is inferior at other operating points as discussed in the previous section.

Note that we view JULE, DEPICT and DDC as solving a complete different problem. Given a collection of unseen face images, it is not practical to assume the number of subjects to be a known quantity. Furthermore, the number of classes reflects the complexity of the data at hand. Without this information, methods such as JULE and DEPICT may suffer from tuning network structures. Therefore, the proposed algorithm is more suitable for applications in which the number of clusters is not known.

Discussion. We observe from the statistics in Table 1 that LFW contains a large number of singleton clusters and YTF consists of multiple large clusters. Since the AHC algorithm uses cosine similarity as the underlying measure, in LFW, it exploits the discriminative power of deep features in 1-1 comparisons (verification) and hence high performance is achieved. However, AHC exhibits inferior performance in YTF, since it ignores local structures as presented in Section 3.1.

Both DBSCAN and PAHC are aware of local neighborhoods with fixed sizes. DBSCAN attains improved performance for larger clusters, and PAHC performs well on template-based data [14]. However, since the neighborhood sizes are not adaptive to local density variations, DBSCAN has degraded performance on LFW, and PAHC does not achieve comparable performance with other methods. The proposed algorithm attains improved performance by balancing discriminative power and density-aware property.

Running Time Comparisons. We compare the running time performance using IJB-B-1024 and IJB-B-1845 subtasks which contain 36,575 and 68,195 faces respectively. The results are reported in Table 3.

Dataset	*IJB-B-1024*	*IJB-B-1845*
K-means [16]	00:00:17	00:01:00
AHC [8]	00:00:29	00:01:32
DBSCAN [4]	00:07:49	00:49:31
AP [5]	03:55:42	08:42:50
PAHC [14]	00:01:19	00:03:56
ARO [17]	00:00:37	00:00:73
ConPaC [25]	00:20:06	02:53:58
DDC	00:02:17	00:05:32
DDC-NEG	00:01:55	00:05:39

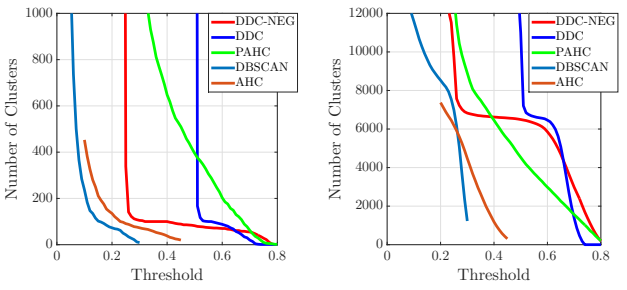
Table 3: Running Time Comparisons (HH:MM:SS).

Dataset	*YTF*		*LFW*		*IJB-B-128*		*IJB-B-256*		*IJB-B-1024*		*IJB-B-1845*	
	F	NMI	F	NMI	F	NMI	F	NMI	F	NMI	F	NMI
K-means [16]	0.815	0.915	0.688	0.922	0.628	0.835	0.585	0.838	0.551	0.851	0.532	0.854
AHC [8]	0.908	0.960	0.940	0.987	0.824	0.925	0.805	0.922	0.736	0.919	0.729	0.921
AP [5]	0.312	0.795	0.618	0.906	0.439	0.822	0.426	0.836	0.411	0.854	0.405	0.858
DBSCAN [4]	**0.923**	**0.967**	0.868	0.973	0.777	0.893	0.762	0.895	0.675	0.894	0.672	0.895
SSC-OMP [35]	0.142	0.174	- -	- -	0.177	0.476	0.136	0.483	- -	- -	- -	- -
JULE* [33]	-	0.848	-	-	-	-	-	-	-	-	-	-
DEPICT* [7]	-	0.802	-	-	-	-	-	-	-	-	-	-
PAHC [14]	0.360	0.734	0.857	0.958	0.695	0.863	0.648	0.865	0.639	0.890	0.610	0.890
ARO* [17]	-	-	0.870	-	0.482	-	0.423	-	0.352	-	0.317	-
ConPaC* [25]	-	-	0.922	-	0.563	-	0.493	-	0.452	-	0.429	-
DDC	0.906	0.960	0.943	0.988	0.810	0.918	0.788	0.916	0.723	0.913	0.725	0.919
DDC-NEG	0.919	0.965	**0.955**	**0.991**	**0.829**	**0.927**	**0.816**	**0.926**	**0.751**	**0.922**	**0.746**	**0.925**

Table 4: BCubed F-measure and NMI performance comparisons. For linkage-based approaches, scores are reported using optimal (oracle-supplied) threshold. The best performance is reported in **bold**.

4.4. Determining Operating Point

The reported performance on different operating points is obtained by thresholding the pairwise similarity matrix at different levels: large thresholds result in several tiny clusters which correspond to high precision and low recall operating points, while small thresholds result in a few gigantic clusters which correspond to low precision and high recall operating points. Neither of the two cases provide desirable clustering results. In real-world applications, we are often interested in generating high precision and recall clustering assignments and at the same time know the approximate number of distinct identities. This requires one to find proper operating points. In this section, we investigate the influences of different operating thresholds on the resulting number of clusters. Results on the YTF and LFW datasets are reported. From Figures 6a and 6b, we observe kinks and clear fall-offs from the proposed methods. The kinks provide hints to the number of distinct identities and reduce the dynamic range of generated number of clusters.



(a) YTF (b) LFW

Figure 6: Qualitative evaluations on YTF and LFW.

5. Conclusion

In this paper, we proposed a novel algorithm to cluster unconstrained face images without knowing the number of subjects. Based on a local compact representation and a density-based similarity measure, the proposed approach adaptively models the neighborhood structure for each sample and yield a more discriminative neighborhood similarity measure. We theoretically show that the representation is asymptotically a Parzen window density estimator. The proposed approach achieves improved performance than other state-of-the-art approaches on challenging face datasets. The results also show that the density-aware property reduces the difficulty of finding proper operating points in clustering.

6. Acknowledgments

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009. 6, 7

[2] W.-C. Chang, C.-P. Lee, and C.-J. Lin. A revisit to support vector data description (svdd), 2013. 4

[3] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 2

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. 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, pages 226–231, 1996. 7, 8

[5] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315, 2007. 7, 8

[6] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001. 2

[7] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 5, 7, 8

[8] K. C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978. 7, 8

[9] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016. 6

[10] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008. 5

[11] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering network. In *NIPS*, 2017. 2

[12] S. Jin, H. Su, C. Stauffer, and E. Learned-Miller. End-to-end face detection and cast grouping in movies using erdos-renyi clustering. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[13] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998. 2

[14] W.-A. Lin, J.-C. Chen, and R. Chellappa. A proximity-aware hierarchical clustering of faces. In *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 2017. 2, 7, 8

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[16] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 7, 8

[17] C. Otto, D. Wang, and A. K. Jain. Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99), 2017. 2, 7, 8

[18] D. Park, C. Caramanis, and S. Sanghavi. Greedy subspace clustering. In *Neural Information Processing Systems*, December 2014. 2

[19] V. M. Patel, H. Van Nguyen, and R. Vidal. Latent space sparse subspace clustering. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2

[20] C. Peng, Z. Kang, and Q. Cheng. Subspace clustering via variance regularized ridge regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[21] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), July 2001. 4

[22] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, May 2000. 4

[23] P. B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems (NIPS)*, pages 582–588. 2000. 4

[24] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[25] Y. Shi, C. Otto, and A. K. Jain. Face clustering: Representation and pairwise constraints. *CoRR*, abs/1706.05067, 2017. 2, 7, 8

[26] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014. 3

[27] D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004. 5

[28] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999. 4

[29] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother. Iarpa janus benchmark-b face dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 6

[30] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[31] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, 2016. 2

[32] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning (ICML)*, 2017. 2

[33] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 5, 7, 8

[34] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 6

[35] C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 7, 8

[36] J. Zheng, J.-C. Chen, N. Bodla, V. M. Patel, and R. Chellappa. Vlad encoded deep convolutional features for unconstrained face verification. In *IEEE International Conference on Pattern Recognition*, 2016. 6