Deep Adaptive Image Clustering

Jianlong Chang[1,2] Lingfeng Wang[1] Gaofeng Meng[1] Shiming Xiang[1] Chunhong Pan[1]
[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2] School of Computer and Control Engineering, University of Chinese Academy of Sciences
{jianlong.chang, lfwang, gfmeng, smxiang, chpan}@nlpr.ia.ac.cn

Abstract

Image clustering is a crucial but challenging task in machine learning and computer vision. Existing methods often ignore the combination between feature learning and clustering. To tackle this problem, we propose Deep Adaptive Clustering (DAC) that recasts the clustering problem into a binary pairwise-classification framework to judge whether pairs of images belong to the same clusters. In DAC, the similarities are calculated as the cosine distance between label features of images which are generated by a deep convolutional network (ConvNet). By introducing a constraint into DAC, the learned label features tend to be one-hot vectors that can be utilized for clustering images. The main challenge is that the ground-truth similarities are unknown in image clustering. We handle this issue by presenting an alternating iterative Adaptive Learning algorithm where each iteration alternately selects labeled samples and trains the ConvNet. Conclusively, images are automatically clustered based on the label features. Experimental results show that DAC achieves state-of-the-art performance on five popular datasets, e.g.*, yielding 97.75% clustering accuracy on MNIST, 52.18% on CIFAR-10 and 46.99% on STL-10.*

1. Introduction

Image clustering is an essential data analysis tool in machine learning and computer vision. Many applications such as content-based image annotation [19, 20, 22, 23] and image retrieval [12, 24, 34] can be viewed as different instances of image clustering. Technically, image clustering is the process of grouping images into clusters such that the images within the same clusters are similar to each other, while those in different clusters are dissimilar.

In the literature, much research has been dedicated to image clustering [9, 29, 32, 37, 38]. Traditionally, various clustering methods have been explored, including Kmeans [32], agglomerative clustering [9], and so on. In spite of their success in data clustering, traditional methods depend on predefined distance metrics which are difficult to identify on image datasets. Recently, efforts have focused on deep unsupervised feature learning methods, such as the auto-encoder [1] and the auto-encoding variational bayes [13], for learning the representations of images which are used for clustering images. Technically, they adopt a multi-stage pipeline that pre-trains deep neural networks with unsupervised methods firstly and employs traditional methods for clustering images as post processing. While the advances are observable, these representation-based approaches also have some intrinsic limitations. First, multi-stage image clustering paradigms are obviously cumbersome in practice. Second, the learned representations are fixed after the unsupervised feature learning. Consequently, in the clustering process, the representations can not be further improved to obtain better performance.

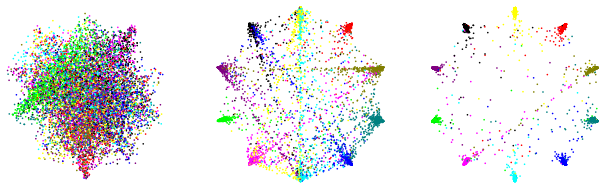

Figure 1. Clustering results on the MNIST [16] test set. Different colors represent different clusters, respectively. For clarity, we map the learned label features to the regular decagon in the two-dimensional space. The ten vertexes correspond to the ten one-hot vectors in the ten-dimensional space, respectively. The details of the mapping function can be found in the supplementary material.

In this paper, we introduce Deep Adaptive Clustering, a single-stage ConvNet-based method to cluster images. To this end, we consider the image clustering task as a binary pairwise-classification problem to judge whether pairs of images belong to the same clusters. Specifically, the image are represented by label features generated by a deep ConvNet, and similarities are measured by the cosine distance between label features. Furthermore, the learned label features tend to be one-hot vectors by introducing a constraint into DAC. Since the ground-truth similarities are unknown, we also develop an Adaptive Learning algorithm, an alternating iterative method, to optimize our model. During each iteration, pairwise images with the estimated similarities are

first selected based on the fixed ConvNet. Subsequently, DAC employs the selected labeled samples to train the ConvNet in a supervised way. The algorithm converges when all the samples are included for training and the objective function of the binary pairwise-classification problem can not be improved further. Finally, images are clustered by locating the largest response of label features. The visual results of DAC on the MNIST test set [16] are illustrated in Figure 1.

To sum up, the main contributions of this work are:

- The proposed DAC model adopts a binary pairwise-classification framework for image clustering, which benefits the feature learning in a "supervised" manner.
- The learned label features tend to be one-hot vectors by introducing a constraint into DAC. Thus we can perform clustering by locating the largest response of the learned label features, which can dramatically simplify the image clustering process.
- We introduce a single-stage method named Adaptive Learning algorithm to optimize our model, which can streamline the learning procedure for image clustering.

2. Related Work

Data Clustering. Much research has been devoted to data clustering methods [9, 29, 32, 37, 38]. Generally, existing methods can be roughly divided into three categories: distance-based, density-based and connectivity-based methods. Distance-based methods, such as the K-means [31, 32] and the agglomerative clustering (AC) [9], seek to find the relationship between data points based on various distance metrics. Density-based methods attempt to cluster data points via a proper density function, including the density-based spatial clustering of applications with noise [33]. Compared with the previous methods, connectivity-based methods cluster data points into a cluster if they are highly connected. The frequently used method is the spectral clustering (SC) [40]. The aforementioned ideas form the basis of a number of methods, such as the ensemble clustering [11], the non-negative matrix factorization (NMF) based clustering [3], and so on.

Image Representation. Image representation is one of the most important issues in image clustering. In the literature, several methods have been proposed. Clustering methods traditionally encode images according to low-level features, such as HOG [6], SIFT [17], and so on. While these feature descriptors may loose representations from messy variables (*e.g.*, rotation, luminance), they often suffer from appearance variations of scenes and objects.

Over the last decade, deep unsupervised feature learning has been explored to learn the informative representations of images. Technically, most deep unsupervised learning methods aim to learn the feature representations that are able to reconstruct the inputs themselves, such as the auto-encoder (AE) [1], the sparse auto-encoder (SAE) [18], the denoising auto-encoder (DAE) [30], the deconvolutional network (DeCNN) [39], the stacked what-where auto-encoder (SWWAE) [41], and so on. Additionally, deep generative models, including the auto-encoding variational bayes (AEVB) [13] and the generative adversarial network (GAN) [21], have been provided to encode visual information recently. However, clustering results can not be obtained immediately based on the generated representations by the aforementioned methods.

Combination. Recently, several methods have been proposed to combine feature learning with clustering into a single model. Inspired by the parametric t-SNE [28], Xie *et al*. [35] proposed deep embedded clustering (DEC), which can be used to learn cluster centers. There is a nuisance fact that the utilized deep networks require pre-training in advance. However, how to effectively pre-train deep networks is an open problem. Unlike DEC, the joint unsupervised learning (JULE) [36] guides agglomerative clustering and feature learning jointly based on the over-clustering initialized by KNN. Since the distances between different images are difficult to define, beginning with the over-clustering may degrade the performance of JULE, especially when image datasets are observably complicated.

Sample Selection. In machine learning, how to select training samples to learn more effective models is an active research topic. Primitively, boosting algorithm [8] randomly selects partial samples from training set to train a set of diverse models. And a single strong learner is created by integrating these models. Furthermore, by mimicking the cognitive process of humans, curriculum learning [2] uses the easy samples first and gradually provides the learning algorithm with more complex ones. To voluntarily select samples in training, Kumar *et al*. [15] presented self-paced learning that incorporates curriculum choosing into model training. Although such achievements are notable, these methods are purely working with the labeled data.

3. Deep Adaptive Clustering Model

To begin with, we assume that the relationship of pairwise images is binary. That is, each pair of images belong to either the same clusters or different clusters. Based on this assumption, we recast the image clustering task into a binary pairwise-classification model. Since the similarities between images are unknown, we adaptively select pairwise images to train the model by investigating the similarities. The flowchart of DAC is illustrated in Figure 2. More details are given in the following subsections.

3.1. Binary Pairwise-Classification for Clustering

Given the unlabeled image dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ and the predefined number of clusters k, where $\mathbf{x}_i$ indicates i-th images, we formulate the image clustering task as a binary pairwise-classification problem. Denote the training data

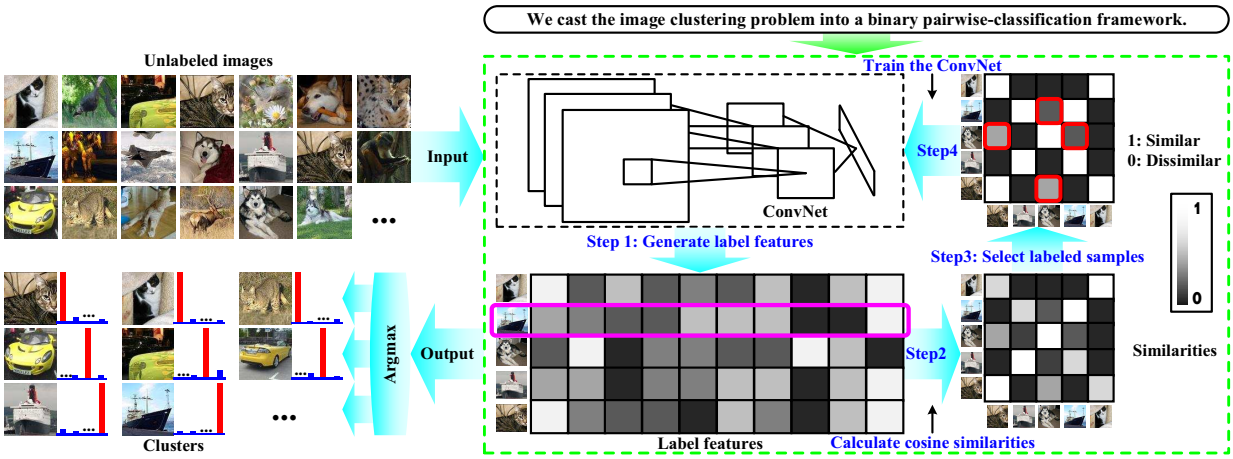

Figure 2. The flowchart of DAC. The input is a set of unlabeled images. Step 1 generates the label features (as shown in the pink box) of the images by using a ConvNet. Step 2 calculates the cosine similarities between images based on the label features. Step 3 selects training samples according to the cosine similarities, and the samples depicted in the red boxes represent the omitted samples in training procedure. Step 4 utilizes the selected samples to train the ConvNet based on the formulated binary pairwise-classification model. Iterate step 1 to step 4 until all the samples are considered for training. Conclusively, images are clustered by locating the largest response of label features.

as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j, r_{ij})\}_{i=1,j=1}^{n}$, where $\mathbf{x}_i,\ \mathbf{x}_j \in \mathcal{X}$ are the unlabeled images (which refer to the input) and $r_{ij} \in \mathcal{Y}$ is the unknown binary variable (which refer to the output). In this work, $r_{ij} = 1$ indicates that $\mathbf{x}_i,\ \mathbf{x}_j$ belong to the same cluster and $r_{ij} = 0$ otherwise. Accordingly, the objective function of DAC is defined as follows:

$$\min_{\mathbf{w}} \mathbf{E}(\mathbf{w}) = \sum_{i,j} L(r_{ij}, g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})), \tag{1}$$

where $L(r_{ij}, g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w}))$ is the loss between r_{ij} and the estimated similarity $g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})$, $\mathbf{w}$ represents the model parameters in function g. Formally,

$$L(r_{ij}, g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})) = \tag{2}$$
$$-r_{ij} \log(g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})) - (1 - r_{ij}) \log(1 - g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})).$$

Generally, two issues in Eq. (1) need to be addressed, *i.e.*, the clusters of $\mathbf{x}_i$ and $\mathbf{x}_j$ are unacquirable by only accessing to the estimated similarity $g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})$ and $\mathcal{Y}$ is unknown in the image clustering process. Section 3.2 and 3.3 focus on addressing these two issues, respectively.

3.2. Label Features under Clustering Constraint

To measure the similarities of image pairs, we introduce label features $\mathcal{L} = \{\mathbf{l}_i \in \mathbb{R}^k\}_{i=1}^{n}$, where $\mathbf{l}_i$ represents the k-dimensional label feature of the image $\mathbf{x}_i$ [10]. The similarity $g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})$ is defined as the cosine distance between two label features. Further, we impose a clustering constraint on the label features to learn more beneficial feature representations for clustering images, *i.e.*,

$$\forall\, i,\ \| \mathbf{l}_i \|_2 = 1,\ \text{and}\ l_{ih} \geq 0,\ h = 1, \cdots, k, \tag{3}$$

where $\| \cdot \|_2$ represents L_2-norm of a vector and l_{ih} is the h-th element of label feature $\mathbf{l}_i$. Due to $\forall\, i,\ \| \mathbf{l}_i \|_2 = 1$, the cosine similarity $g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w})$ can be formulated as:

$$g(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w}) = f(\mathbf{x}_i; \mathbf{w}) \cdot f(\mathbf{x}_j; \mathbf{w}) = \mathbf{l}_i \cdot \mathbf{l}_j, \tag{4}$$

where $f_{\mathbf{w}}$ is a mapping function that maps input images to label features and the operator "$\cdot$" represents dot product between two label features. By introducing the clustering constraint, the DAC model can be reformulated as:

$$\min_{\mathbf{w}} \mathbf{E}(\mathbf{w}) = \sum_{i,j} L(r_{ij}, \mathbf{l}_i \cdot \mathbf{l}_j),$$
$$\text{s.t.}\ \forall\, i,\ \| \mathbf{l}_i \|_2 = 1,\ \text{and}\ l_{ih} \geq 0,\ h = 1, \cdots, k. \tag{5}$$

The clustering constraint in Eq. (3) brings an interesting property for data clustering. Let $\mathbb{E}^k$ denote the standard basis of the k-dimensional Euclidean space, we have the following theorem (the proof of this theorem is reported in the supplementary material):

THEOREM 1. *If the optimal value of Eq. (5) is attained, for* $\forall\, i,\ j,\ \mathbf{l}_i \in \mathbb{E}^k, \mathbf{l}_i \neq \mathbf{l}_j \Leftrightarrow r_{ij} = 0$ *and* $\mathbf{l}_i = \mathbf{l}_j \Leftrightarrow r_{ij} = 1$.

Theorem 1 indicates that the learned label features are k diverse one-hot vectors ideally. That is, images can be automatically clustered based on the learned label features.

3.3. Labeled Training Samples Selection

In practice, a strategy for selecting labeled training samples is needed, since $\mathcal{Y}$ is unknown in image clustering. For ConvNets, specifically, we have two observations. First, if ConvNets are already trained, the high-level features of images can be generated. Second, for randomly initialized ConvNets, they can also capture the low-level features of images, since the randomly initialized filters act as edge detectors [27]. Based on these observations, we employ ALL-ConvNets [25] to implement $f_{\mathbf{w}}$ and select labeled training

samples based on the generated label features, *i.e.*,

$$r_{ij} := \begin{cases} 1, & \text{if } \mathbf{l}_i \cdot \mathbf{l}_j \geq u(\lambda), \\ 0, & \text{if } \mathbf{l}_i \cdot \mathbf{l}_j < l(\lambda), \\ \text{None}, & \text{otherwise}, \end{cases} \quad i,\ j = 1, \cdots, n, \tag{6}$$

where λ is an adaptive parameter for controlling the selection, $u(\lambda)$ and $l(\lambda)$ are the thresholds for selecting similar and dissimilar labeled samples, respectively. And "None" implies that the sample $(\mathbf{x}_i, \mathbf{x}_j, r_{ij})$ is omitted for training.

Inspired by curriculum learning [2], we attempt to control the clustering procedure such that the samples are increasingly selected. That is, "easy" samples with high likelihood are first selected as training samples to find rough cluster patterns. Then, as the clustering procedure progresses, the trained ALL-ConvNets can be utilized for extracting more effective label features and more samples will be gradually appended to find more refined cluster patterns. For this purpose, we control the parameter λ, $u(\lambda)$ and $l(\lambda)$ as follows. In the clustering process, λ is gradually increased. Furthermore, $u(\lambda) \propto -\lambda$, $l(\lambda) \propto \lambda$ and $l(\lambda) \leq u(\lambda)$ are permanently satisfied. And $u(\lambda) = l(\lambda)$ is satisfied if and only if all the samples are used for training.

So far we have addressed the two issues in Section 3.1. The DAC model can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{w},\lambda} \mathbf{E}(\mathbf{w}, \lambda) = \sum_{i,j} v_{ij} L(r_{ij}, \mathbf{l}_i \cdot \mathbf{l}_j) + u(\lambda) - l(\lambda), \\ &\text{s.t. } l(\lambda) \leq u(\lambda), \\ &\quad v_{ij} \in \{0, 1\},\ i,\ j = 1, \cdots, n, \\ &\quad \forall\ i,\ \| \mathbf{l}_i \|_2 = 1, \text{ and } l_{ih} \geq 0,\ h = 1, \cdots, k, \\ &\quad r_{ij} := \begin{cases} 1, & \text{if } \mathbf{l}_i \cdot \mathbf{l}_j \geq u(\lambda), \\ 0, & \text{if } \mathbf{l}_i \cdot \mathbf{l}_j < l(\lambda), \\ \text{None}, & \text{otherwise}, \end{cases} \quad i,\ j = 1, \cdots, n, \end{aligned} \tag{7}$$

where $\mathbf{v}$ is an indicator coefficient, *i.e.*,

$$v_{ij} := \begin{cases} 1, & \text{if } r_{ij} \in \{0, 1\}, \\ 0, & \text{otherwise}, \end{cases} \quad i,\ j = 1, \cdots, n, \tag{8}$$

where $v_{ij} = 1$ indicates that the sample $(\mathbf{x}_i, \mathbf{x}_j, r_{ij})$ is selected for training, and $v_{ij} = 0$ otherwise. Notice that $u(\lambda) - l(\lambda)$ is a penalty term for the number of training samples. By decreasing the penalty term, more samples will be selected for training until all the samples are included.

4. Deep Adaptive Clustering Algorithm

In this section, we present an optimization algorithm for the DAC model in Eq. (7) and a label inference method for image clustering.

4.1. Adaptive Learning

To optimize the model in Eq. (7), Adaptive Learning algorithm, an alternating iterative optimization scheme, is developed. The algorithm focuses on two issues, namely the clustering constraint and the iterative optimization.

Algorithm 1 Deep Adaptive Clustering

Input: Dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, $f_{\mathbf{w}}$, λ, $u(\lambda)$, $l(\lambda)$, η, m.
Output: Cluster label c_i of $\mathbf{x}_i \in \mathcal{X}$.

1: Randomly initialize $\mathbf{w}$;
2: **repeat**
3: **for all** $k \in \{1, 2, \cdots, \lfloor \frac{n}{m} \rfloor\}$ **do**
4: Sample batch $\mathcal{X}_k$ from $\mathcal{X}$; // m images per batch
5: Select training samples from $\mathcal{X}_k$; // Eq. (6)
6: Calculate the indicator parameter $\mathbf{v}$; // Eq. (8)
7: Update $\mathbf{w}$ by minimizing Eq. (10);
8: **end for**
9: Update λ according to Eq. (12);
10: **until** $l(\lambda) > u(\lambda)$
11: **for all** $\mathbf{x}_i \in \mathcal{X}$ **do**
12: $\mathbf{l}_i := f(\mathbf{x}_i; \mathbf{w})$;
13: $c_i := \arg\max_h(l_{ih})$;
14: **end for**

We establish a restraint layer to implement the clustering constraint in Eq. (3). The mapping functions of the restraint layer are formulated as:

$$L_h^{out} := \exp^{L_h^{in} - \max_h(L_h^{in})},\ h = 1, \cdots, k, \tag{9a}$$

$$L_h^{out} := \frac{L_h^{out}}{\| \mathbf{L}^{out} \|_2},\ h = 1, \cdots, k, \tag{9b}$$

where $\mathbf{L}^{in},\ \mathbf{L}^{out} \in \mathbb{R}^k$ are the input and output of the restraint layer, respectively. L_h^{in} and L_h^{out} represent the h-th element of $\mathbf{L}^{in}$ and $\mathbf{L}^{out}$, respectively. Note that all the elements of the output $\mathbf{L}^{out}$ are mapped into $[0, 1]$ by Eq. (9a) and the output $\mathbf{L}^{out}$ is simultaneously limited to unit vector by Eq. (9b). In our model, the ALL-ConvNets are always followed by the restraint layer. That is, $\forall\ i,\ \mathbf{l}_i$ invariably satisfies the clustering constraint in Eq. (3).

The optimization of $\mathbf{w}$ and λ is performed alternately. Once $\mathbf{r}$ and $\mathbf{v}$ are obtained and λ is fixed, the DAC model degenerates as follows:

$$\min_{\mathbf{w}} \mathbf{E}(\mathbf{w}) = \sum_{i,j} v_{ij} L(r_{ij}, f(\mathbf{x}_i; \mathbf{w}) \cdot f(\mathbf{x}_j; \mathbf{w})). \tag{10}$$

Since $\mathbf{r}$ and $\mathbf{v}$ are available, Eq. (10) is a supervised learning problem that the back-propagation learning algorithm can be utilized to update $\mathbf{w}$. Specifically, the storage complexity of $\mathbf{r}$ and $\mathbf{v}$ is $O(n^2)$ because the similarities of pairwise images need to be calculated. It is too high to deal with large datasets. To handle this issue, we randomly sample image batches from the original datasets and update $\mathbf{w}$ on each batch, as illustrated in the line 3 to line 8 of the Algorithm 1.

Similarly, our DAC model can be simplified as follows

when $\mathbf{w}$ is fixed,

$$\min_{\lambda} \mathbf{E}(\lambda) = u(\lambda) - l(\lambda). \tag{11}$$

According to the gradient descent algorithm, in each iteration, the update rule of λ can be written as:

$$\lambda := \lambda - \eta \cdot \frac{\partial \mathbf{E}(\lambda)}{\partial \lambda}, \tag{12}$$

where η is the learning rate of λ. Since $u(\lambda) \propto -\lambda$ and $l(\lambda) \propto \lambda$, $\frac{\partial \mathbf{E}}{\partial \lambda} = \frac{\partial u(\lambda)}{\partial \lambda} - \frac{\partial l(\lambda)}{\partial \lambda} \leq 0$ is always satisfied. This corresponds to our target scenario that all the samples are gradually added for training with the increasing of λ.

4.2. Label Inference for Image Clustering

The label features are ideally one-hot vectors according to Theorem 1. As a result, images can be clustered via:

$$c_i := \arg\max_{h}(l_{ih}),\ h = 1, \cdots, k, \tag{13}$$

where c_i is the cluster label of image $\mathbf{x}_i$. In practice, however, the label features may not be one-hot vectors strictly due to the following two reasons. First, it is hard to reach the global optima for training a ConvNet due to its strong non-convex property. Second, even it achieves a global optimum on the training data, it is almost impossible for it to achieve a global optimum for all data (including the unseen testing data). To address this issue, we label images by locating the largest response of label features, *i.e.*, Eq. (13) is implemented to cluster.

In summary, we illustrate the DAC algorithm in Algorithm 1. The Adaptive Learning algorithm optimize the DAC model iteratively. During each iteration, the algorithm alternately selects samples via the fixed ConvNet and trains the ConvNet based on the selected samples. When all the samples are considered for training and the objective function in Eq. (10) can not be improved further, the algorithm converges. Conclusively, images are clustered by locating the largest response of the label features.

5. Experiments

In this section, we apply the proposed DAC model to image clustering and evaluate the performance on several popular datasets with three frequently-used measures. Specifically, our core code [1] is released at `https://github.com/vector-1127/DAC`.

5.1. Datasets

We perform experiments on five popular image datasets, including MNIST [16], CIFAR-10 [14], CIFAR-100 [14], STL-10 [5] and ILSVRC2012 1K [7]. The number of images and clusters, and image size are listed in Table 1. As described in [35, 36], the training and testing images of each dataset are jointly utilized in our experiments. For the CIFAR-100 dataset, the 20 superclasses are considered in our experiments. Specifically, we randomly choose 10 subjects from the ILSVRC2012 1K dataset [7] and resize these images to $96 \times 96 \times 3$ to construct the ImageNet-10 dataset for our experiments. Furthermore, to compare the clustering methods on more complex dataset, we randomly select 15 kinds of dog images from ILSVRC2012 1K to establish the fine-grained ImageNet-Dog dataset.

Table 1. The image datasets used in our experiments.

Dataset	Images	Clusters	Image size
MNIST [16]	70000	10	28×28
CIFAR-10 [14]	60000	10	$32 \times 32 \times 3$
CIFAR-100 [14]	60000	20	$32 \times 32 \times 3$
STL-10 [5]	13000	10	$96 \times 96 \times 3$
ImageNet-10 [7]	13000	10	$96 \times 96 \times 3$
ImageNet-Dog [7]	19500	15	$96 \times 96 \times 3$

5.2. Evaluation Metrics

In our experiments, three popular measures in the literature are employed to evaluate the performance of clustering methods, including Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and clustering Accuracy (ACC). Specifically, these measures range in $[0, 1]$, and higher scores imply more accurate clustering results.

5.3. Compared methods

Several clustering algorithms are employed for comparison. Specifically, the traditional methods, including K-means [32], SC [40], AC [9] and the NMF based clustering [3], are adopted to compare with our model. For the representation-based clustering approaches, as described in [35], we employ some unsupervised learning methods, including AE [1], SAE [18], DAE [30], DeCNN [39], SWWAE [41], AEVB [13] and GAN [21], to learn feature representations of images and use K-means [32] to cluster images as post processing. We also compare DAC with DEC [35] and JULE [36] for a comprehensive comparison. To evaluate the capability of Adaptive Learning algorithm, we consider all the samples for training during each iteration, and this training strategy denoted by DAC*.

5.4. Experimental Settings

For the traditional clustering methods, following the previous work [35], we concatenate HOG feature [6] and a 8×8 color map as input when we experiment on STL-10, ImageNet-10 and ImageNet-Dog. For the remaining datasets and methods, the pixel intensities serve as inputs.

In our experiments, the ALL-ConvNet described in [25] is utilized in our model (the details of the devised ConvNets are listed in the supplementary material). Since the prior probability of image pairs belonging to different clusters is

[1] Relies on Keras [4] with the Theano [26] backend.

Table 2. The clustering results of various methods on six datasets. The best three results are highlighted in **bold**. DAC* represents that all the samples are considered for training in each iteration.

Dataset	MNIST [16]			CIFAR-10 [14]			CIFAR-100 [14]			STL-10 [5]			ImageNet-10 [7]			ImageNet-Dog [7]		
Metric	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
K-means [32]	0.4997	0.3652	0.5723	0.0871	0.0487	0.2289	0.0839	0.0280	0.1297	0.1245	0.0608	0.1920	0.1186	0.0571	0.2409	0.0548	0.0204	0.1054
SC [40]	0.6626	0.5214	0.6958	0.1028	0.0853	0.2467	0.0901	0.0218	0.1360	0.0978	0.0479	0.1588	0.1511	0.0757	0.2740	0.0383	0.0133	0.1111
AC [9]	0.6094	0.4807	0.6953	0.1046	0.0646	0.2275	0.0979	0.0344	0.1378	0.2386	0.1402	0.3322	0.1383	0.0674	0.2420	0.0368	0.0207	0.1385
NMF [3]	0.6082	0.4298	0.5447	0.0814	0.0338	0.1895	0.0791	0.0263	0.1175	0.0962	0.0458	0.1804	0.1316	0.0652	0.2302	0.0442	0.0155	0.1184
AE [1]	0.7257	0.6139	0.8123	0.2393	0.1689	0.3135	0.1004	0.0476	0.1645	0.2496	0.1610	0.3030	0.2099	0.1516	0.3170	0.1039	0.0728	0.1851
SAE [18]	0.7565	0.6393	0.8271	0.2468	0.1555	0.2973	0.1090	0.0436	0.1567	0.2520	0.1605	0.3203	0.2122	0.1740	0.3254	0.1129	0.0729	0.1830
DAE [30]	0.7563	0.6467	0.8316	0.2506	0.1627	0.2971	0.1105	0.0460	0.1505	0.2242	0.1519	0.3022	0.2064	0.1376	0.3044	0.1043	0.0779	0.1903
DeCNN [39]	0.7577	0.6691	0.8179	0.2395	0.1736	0.2820	0.0923	0.0378	0.1327	0.2267	0.1621	0.2988	0.1856	0.1421	0.3130	0.0983	0.0732	0.1747
SWWAE [41]	0.7360	0.6518	0.8251	0.2330	0.1638	0.2840	0.1034	0.0391	0.1472	0.1962	0.1358	0.2704	0.1761	0.1603	0.3238	0.0936	0.0760	0.1585
AEVB [13]	0.7364	0.7129	0.8317	0.2451	0.1674	0.2908	0.1079	0.0403	0.1517	0.2004	0.1464	0.2815	0.1934	0.1683	0.3344	0.1074	0.0786	0.1788
GAN [21]	0.7637	0.7360	0.8279	**0.2646**	**0.1757**	**0.3152**	0.1200	0.0453	0.1510	0.2100	0.1390	0.2984	0.2250	0.1571	0.3459	0.1213	0.0776	0.1738
JULE [36]	**0.9130**	**0.9270**	**0.9640**	0.1923	0.1377	0.2715	0.1026	0.0327	0.1367	0.1815	0.1643	0.2769	0.1752	0.1382	0.3004	0.0537	0.0284	0.1377
DEC [35]	0.7716	0.7414	0.8430	0.2568	0.1607	0.3010	**0.1358**	**0.0495**	**0.1852**	**0.2760**	**0.1861**	**0.3590**	**0.2819**	**0.2031**	**0.3809**	**0.1216**	**0.0788**	**0.1949**
DAC*	**0.9246**	**0.9406**	**0.9660**	**0.3793**	**0.2802**	**0.4982**	**0.1623**	**0.0776**	**0.2189**	**0.3474**	**0.2351**	**0.4337**	**0.3693**	**0.2837**	**0.5026**	**0.1815**	**0.0953**	**0.2455**
DAC	**0.9351**	**0.9486**	**0.9775**	**0.3959**	**0.3059**	**0.5218**	**0.1852**	**0.0876**	**0.2375**	**0.3656**	**0.2565**	**0.4699**	**0.3944**	**0.3019**	**0.5272**	**0.2185**	**0.1105**	**0.2748**

Table 3. The results of the traditional methods based on the DAC learned label features. The best results are indicated in **bold**.

Dataset	MNIST [16]			CIFAR-10 [14]			CIFAR-100 [14]			STL-10 [5]			ImageNet-10 [7]			ImageNet-Dog [7]		
Metric	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
K-means [32]	**0.9358**	0.9481	0.9761	0.3943	0.2993	0.5106	0.1850	0.0811	0.2251	**0.3689**	0.2542	0.4692	0.3937	**0.3023**	0.5224	0.2164	0.1074	0.2709
SC [40]	0.8830	0.8532	0.9348	0.3935	0.2991	0.5193	0.1843	**0.0880**	0.2361	0.3667	0.2524	0.4674	0.3913	0.3004	0.5212	0.2149	0.1101	0.2702
AC [9]	0.9347	0.9465	0.9753	0.3881	0.2601	0.5030	0.1824	0.0876	0.2302	0.3586	0.2340	0.4524	0.3874	0.2976	0.5102	0.2052	0.1038	0.2697
NMF [3]	0.9077	0.9153	0.9347	0.3761	0.2615	0.4993	0.1801	0.0726	0.2136	0.3498	0.2278	0.4454	0.3856	0.2934	0.5045	0.2007	0.0984	0.2631
DAC	0.9351	**0.9486**	**0.9775**	**0.3959**	**0.3059**	**0.5218**	**0.1852**	0.0876	**0.2375**	0.3656	**0.2565**	**0.4699**	**0.3944**	0.3019	**0.5272**	**0.2185**	**0.1105**	**0.2748**

higher than to the same clusters, we set $u(\lambda) = 0.95 - \lambda$ and $l(\lambda) = 0.455 + 0.1 \cdot \lambda$ for selecting similar and dissimilar samples, respectively. The adaptive parameter λ is initialized to 0 with the learning rate $\eta = 0.009$. In each iteration, we randomly select $m = 1000$ images to select training samples. In DAC*, we set $u(\lambda) = l(\lambda) = 0.95$ for the beginning, followed by an annealing phase which decreases linearly to $u(\lambda) = l(\lambda) = 0.5$. More details of training settings are reported in the supplementary material.

5.5. Image Clustering

In Table 2, we report the quantitative clustering results of these clustering methods. Note that DAC dramatically outperforms the others methods with significant margins on all the three clustering quality measures. Further observation, several tendencies can be observed in Table 2. First, the performance of representation-based clustering methods (*e.g.*, AE [1], AEVB [13]) is superior to the traditional methods (*e.g.*, K-means [32], SC [40]). This indicates that clustering methods have only a minor impact on performance, while representations are more important. It means that the representation learning plays a crucial role in image clustering. Second, although the effective representations can be learned by these unsupervised methods, the improvement is limited compared against our approach. This demonstrates that the end-to-end clustering scheme can observably improve the performance of image clustering. The reason is that our single-stage method can learn more excellent representations for image clustering. Thirdly, more distinct superiority is achieved by DAC on CIFAR-10, CIFAR-100, STL-10 and ImageNet. This verifies that DAC has enough capability to handle complex large-scale image datasets.

In Figure 3, we qualitatively analysis the label features learned by DAC on MNIST, STL-10 and a randomly chosen ImageNet-10. We observe that the same neurons will be distinctly activated in the label features if the images belong to the same clusters. That is, our method learns high-level features, rather than the simple combination of visual features. This is the reason why more complicated images, such as the airliner and airship images in ILSVRC2012 1K, can be distinguished by DAC. Furthermore, most of the label features of the failure modes appear reasonable. For example, in terms of car, only the other types of vehicle (e.g., truck) might be considered as plausible labels, rather than the clusters beyond vehicle. It implies that more interpretable features are leaned by DAC for image clustering.

5.6. Empirical Analysis

Effect of Adaptive Learning Algorithm. We compare the performance of DAC* with DAC to investigate the effect of the Adaptive Learning algorithm. From Table 2, we observe that DAC achieves better performance than DAC*. Further analysis, since ConvNets are initialized randomly, more noisy samples will be utilized for training in DAC*. Contrary to DAC*, DAC can select highly confident training data based on the Adaptive Learning algorithm. By using these selected samples, DAC can begin with more refined cluster patterns and improve the clustering performance consequently.

Effect of Clustering Tactics. In order to evaluate the effect of our clustering tactics in Eq. (13), we employ the traditional methods (*e.g.*, K-means [32]) to cluster images

Figure 3. The label features learned by DAC on MNIST, STL-10 and a randomly chosen ImageNet-10. For each dataset, the correct labels are written on the upward side, the label features are shown on the right side of images and the bottom line shows some failure modes.

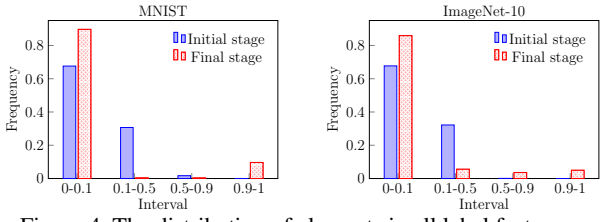

Figure 4. The distribution of elements in all label features.

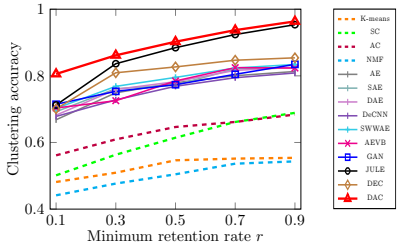

Figure 5. Clustering accuracy on imbalanced subset of MNIST.

based on the label features learned by DAC. The results are listed in Table 3. Note that our clustering tactics achieves better performance than the traditional methods. Furthermore, compared with these traditional methods, the clustering tactics is more terse since DAC just needs to locate the largest response of label features to cluster images.

Contribution of Clustering Constraint. To investigate the effect of the clustering constraint in Eq. (3), we report the distribution of the learned label features on MNIST and ImageNet-10 in different clustering stages in Figure 4. We count the elements of the learned label features in the four disjoint intervals, *i.e.*, $[0, 1)$, $[0.1, 0.5)$, $[0.5, 0.9)$ and $[0.9, 1]$. For the initial stage, the major elements of the label features locate in $[0, 1)$ and $[0.1, 0.5)$. By training our model, most elements move to $[0, 1)$ and $[0.9, 1]$ in the final stage. This implies that the learned label features are sparse and the non-zero elements in the label features tend to be 1. It corresponds to our target that DAC attempts to learn one-hot vectors to represent and cluster images.

Performance on Imbalanced Datasets. We perform additional experiments to study the performance of DAC on imbalanced datasets. Following the previous work [35], we randomly sample subsets of MNIST with various minimum retention rates. For the minimum retention rate r, data points of class 0 will be kept with probability r and class 9 with probability 1, with the other classes linearly in between. From Figure 5 we observe that DAC is more robust than the others methods on the various imbalanced datasets. A possible reason is that DAC executes image clustering

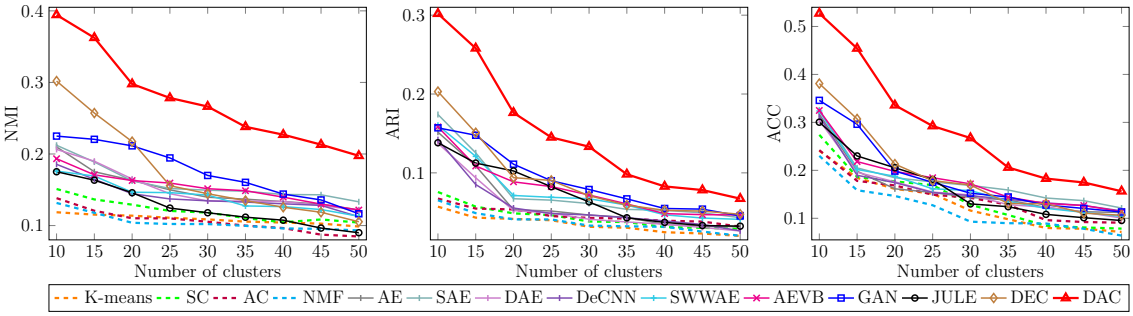

Figure 6. Comparison of clustering performance with increasing number of clusters on ILSVRC2012 1K (1300 images per cluster).

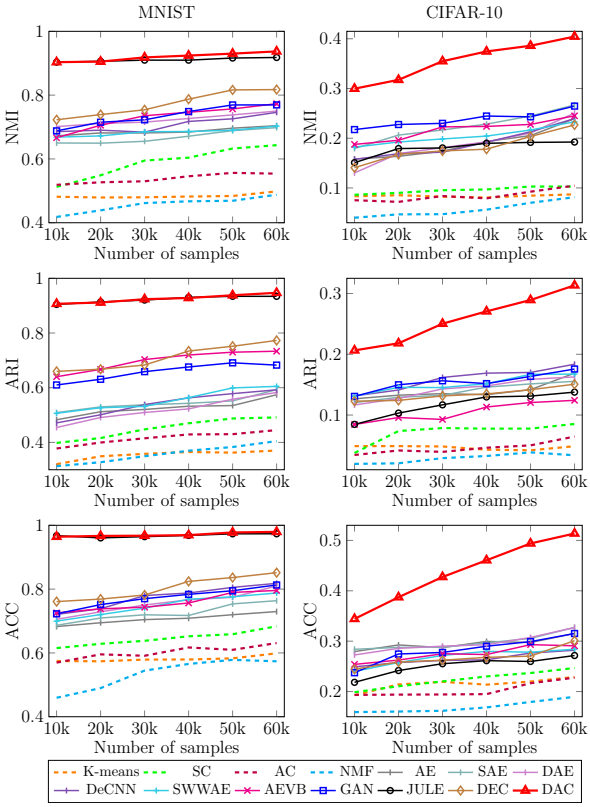

Figure 7. Comparison of clustering performance with increasing number of samples on MNIST (left) and CIFAR-10 (right).

based on similarities between images only, which can reduce the influence of the unbalancedness of datasets.

Performance on Various Number of Clusters. We also conduct experiment on the ILSVRC2012 1K dataset [7] to study the stabilities of these methods by varying the number of clusters. Intuitively, Figure 6 shows the clustering results when the number of clusters various between 10 and 50 with an interval 5. In summary, as the number of clusters increases, all the methods are generally degraded. This is because more uncertainty is triggered as the number of clusters increases. However, contrary to other methods, the superiority of DAC still holds with the various number of clusters. The results demonstrate that DAC possesses adequate capability to tackle various clusters.

Performance on Various Number of Samples. To observe the effect of the number of samples to these methods, we vary it between 10000 and 60000 with an interval 10000 on MNIST and CIFAR-10. Figure 7 visually shows that the performance of most methods improves with more samples. This indicates that more samples are beneficial for these methods. For the CIFAR-10 dataset, we observe that the performance of DAC increases rapidly when more samples are considered. Contrary to CIFAR-10, DAC reaches an saturation status by using less samples on MNIST. This is to be expected, since sufficient samples are essential for mapping more intricate images from the visual features to the label features. In particular, the performance of the JULE method [36] approximates to our method on the MNIST dataset. However, there is a conspicuous margin on CIFAR-10. This is because the initialization strategy of the JULE method is invalidated on intricate image datasets, which degenerates the performance of JULE. Compared with JULE, DAC alleviates the dependence on additional techniques by introducing the Adaptive Learning algorithm, which elevates the dependability of DAC.

6. Conclusion

We proposed a single-stage ConvNet-based method to cluster images. Our method is motivated from a basic assumption that the relationship between pairwise images is binary. Based on this assumption, a binary constrained pairwise-classification model is proposed by investigating the similarities between image pairs. We theoretically verified that our model can be guided to represent images via one-hot vectors that can be utilized for clustering images. In comparison with the existing approaches, the proposed method achieves superior performance on five challenging datasets. It shows that our method can deal with large-scale images, not merely limited to some simple image datasets.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (NSFC Nos. 91646207, 61403376, 61370039 and 91338202), the Beijing Nature Science Foundation under Grant No. 4162064, and the Youth Innovation Promotion Association CAS.

References

[1] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, 2006.

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[3] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, pages 1010–1015, 2009.

[4] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[5] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[8] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

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

[10] S. Hong, J. Choi, J. Feyereisl, B. Han, and L. S. Davis. Joint image clustering and labeling by matrix factorization. *T-PAMI*, 38(7):1411–1424, 2016.

[11] D. Huang, J. Lai, and C. Wang. Ensemble clustering using factor graph. *Pattern Recognition*, 50:131–142, 2016.

[12] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, pages 774–787, 2012.

[13] D. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[14] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's Thesis, Department of Computer Science, University of Torono*, 2009.

[15] M. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197, 2010.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] A. Ng. Sparse autoencoder. *CS294A Lecture notes*, 72:1–19, 2011.

[19] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang. Correlative multi-label video annotation. In *ACM MM*, pages 17–26, 2007.

[20] G. Qi, W. Liu, C. Aggarwal, and T. Huang. Joint intermodal and intramodal label transfers for extremely rare or unseen classes. *T-PAMI*, 2016.

[21] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[22] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[23] S. Sclaroff, M. Cascia, S. Sethi, and L. Taycher. Unifying textual and visual cues for content-based image retrieval on the world wide web. *CVIU*, 75(1-2):86–98, 1999.

[24] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, pages 114–127, 2012.

[25] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[26] Theano Development Team. Theano. `http://deeplearning.net/software/theano/`.

[27] Theano Development Team. Theano: Deep learning tutorials – convolutional neural networks. `http://www.deeplearning.net/tutorial/lenet.html#lenet`.

[28] L. van der Maaten. Learning a parametric embedding by preserving local structure. *JMLR*, 5:384–391, 2009.

[29] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[30] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11:3371–3408, 2010.

[31] J. Wang, J. Wang, Q. Ke, G. Zeng, and S. Li. Fast approximate k-means via cluster closures. In *CVPR*, pages 3037–3044, 2012.

[32] J. Wang, J. Wang, J. Song, X. Xu, H. Shen, and S. Li. Optimized cartesian k-means. *IEEE Trans. Knowl. Data Eng.*, 27(1):180–192, 2015.

[33] W. Wang, Y. Wu, C. Tang, and M. Hor. Adaptive density-based spatial clustering of applications with noise according to data. In *ICMLC*, pages 445–451, 2015.

[34] Y. Wu, Q. Tian, and T. Huang. Discriminant-em algorithm with application to image retrieval. In *CVPR*, pages 1222–1227, 2000.

[35] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[36] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, pages 5147–5156, 2016.

[37] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang. Image clustering using local discriminant models and global integration. *T-IP*, 19(10):2761–2773, 2010.

[38] J. Yu. General c-means clustering model and its application. In *CVPR*, pages 122–127, 2003.

[39] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, pages 2528–2535, 2010.

[40] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, pages 1601–1608, 2004.

[41] J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun. Stacked what-where auto-encoders. *CoRR*, abs/1506.02351, 2015.