A Decoder-Free Approach for Unsupervised Clustering and Manifold Learning with Random Triplet Mining

Oliver Nina
Air Force Research Lab
Dayton, OH
oliver.nina.1@us.af.mil

Jamison Moody
Brigham Young University
Provo, UT
jmm1995@byu.edu

Clarissa Milligan
Wright State University
Dayton, OH
milligan.26@wright.edu

Abstract

Unsupervised clustering is a very relevant open area of research in machine learning with many applications in the real world. Learning the manifold in which images lie and measuring the proximity distance of the sample points to the clusters in their latent space is non-trivial. Recent deep learning methods have proposed the use of autoencoders for manifold learning and dimensionality reduction in an effort to better cluster image samples. However, offline training of autoencoders is cumbersome and rather tedious to update. Moreover, trained autoencoders tend to be biased towards the training set and are impractical for performing data augmentation. In this paper, we introduce a novel method that uses a triplet network architecture in order to avoid the need of pre-trained autoencoders. Because our framework can be trained online, we can train our network with data augmented pairs which allows us to build a more robust encoder and improve accuracy. In contrast to other clustering methods that require nearest neighbor comparisons at every step, our method introduces a novel approach for selecting random training samples pairs with an adaptive metric distance which we call Random Triplet Mining. Our method remains competitive compared with other current methods while we obtain state of the art results on the Fashion-MNIST dataset.[1]

1. Introduction

Unsupervised clustering (UC) can be defined as a per-class assembling of groups or clusters of unlabeled data. UC is an essential problem in machine learning and artificial intelligence due to large amounts of labeled data that a supervised method otherwise would require for generalization over the entire training data.

Recent UC approaches avoid the need of labeled training data while attempting to learn the manifold of the space where the data lies and produce feature embeddings through dimensionality reduction. There are different ways in which such non-linear dimensionality reduction is obtained, and more recent methods apply deep learning frameworks to train autoencoders as a pre-process step in order to approximate the manifold of the data [34, 40, 12, 19, 13].

In general, autoencoders provide an approximation of the manifold and provide feature embeddings that could be used as starting point for measuring distances between training samples. Despite the moderate success of autoencoders, because they are separately trained in a pre-process step, they carry an inherent bias towards the data used during their training. The pre-training step of an autoencoder is also cumbersome and makes the pipeline difficult to train in an end-to-end way.

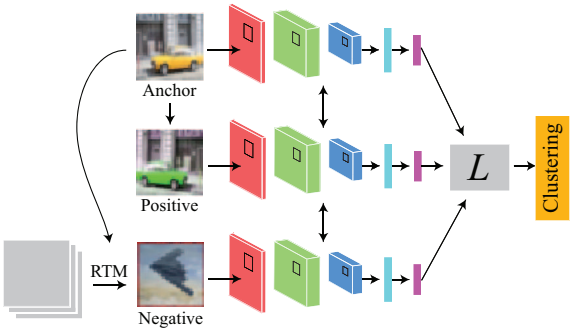


Figure 1: Outline of our method which consists of a triplet network. The first two streams take a positive or similar pair of images and the last takes a negative pair found through random triplet mining (RTM). ***L*** represents the loss of the system. The learned embeddings then can be clustered using any traditional clustering method.

[1]Our code is available at https://github.com/mood2jam/ARTM/

In order to improve upon the drawbacks of current UC methods, in this paper we introduce a novel method that employs a triplet network without the need to pre-train an autoencoder as an additional step. Because our method does not need a pre-trained autoencoder, we can train our method as a whole system in an end-to-end fashion. Furthermore, because we remove the need to pre-train autoencoders offline, our method allows us to perform data augmentation more succinctly which helps improve our results.

In order to choose the best pairs to train our triplet network at each iteration, we introduce a novel matching pair function. This function considers the euclidean distances of the pairs on latent space and selects the pairs with statistical significance that are difficult to find and important for the model to see repeatedly. We call this process *Random Triplet Mining* (RTM), and we show that RTM can help improve online training of our triplet network significantly.

Our method yields competitive and significant improvements over current methods, particularly on the Fashion-MNIST dataset where we achieve state of the art results.

2. Background

Unsupervised clustering via manifold learning has achieved significant progress in recent years. There are in the literature different types of approaches for clustering that range from non-linear dimensionality reduction techniques to more complex manifold learning techniques employing autoencoders and deep learning.

Earlier approaches on non-linear dimensionality reduction include [32] which uses a the kernel trick for principal component analysis of high dimensional data. Isomap [36] is another non-linear dimensionality reduction technique that attempts to preserve the intrinsic geometry of the data by using geodesic manifold distances. Local Linear Embedding (LLE) [31] exploits the local symmetries of linear reconstruction to learn the structure of non-linear manifolds. In [2], the data manifold is approximated by the adjacency graph obtained from the data points and the embedding maps of the data is approximated by the eigenmaps of the Laplacian of the graph. Diffusion maps [9] use eigenfunctions of Markov matrices that represent complex geometric structures of data sets.

Recently deep learning and neural networks have proven to have potential to perform dimensionality reduction [8]. Some more recent methods have used neural networks and more particularly, convolutional neural networks (CNN) to create an affinity matrix of the data. For instance, [3] utilizes CNNs to create hierarchical clustering and a Laplacian graph. Another use of CNNs is through a siamese network where a CNN with the same weights is run with two different inputs. For example, [21] employs a siamese net that is trained on augmented data in a supervised manner and reused for one-shot learning. A triplet network [16] is a variation of a Siamese net that has three instances of the same network instead of two. The network calculates distances for both a similar image and a different image and uses these to learn the dataset.

Other methods use autoencoders to perform dimensionality reduction such as [27], where diffusion map encodings are stacked in between neural networks to help train the networks. This allows the autoencoder to perform out of sample extension while also being robust to noise. To increase performance in higher dimensions, [25] utilized siamese nets to determine a distance metric that could be fed to a traditional nearest neighbors algorithm. Deep nets [7] use a deep learning algorithm to determine a local coordinate system for an unknown manifold without eigendecomposition. In [30] a convolutional Generative Adversarial Network (GAN) is developed in order to establish a more stable training architecture for unsupervised learning. In order to more effectively generalize CNNs to graphs, [10] combines CNNs with spectral graph theory. Then using the Graph Laplacian, spectral filters can be determined for the CNN instead of regular filters. Deep Isometric Manifold Learning (DIMAL) [29] is an unsupervised deep learning approach for computing distance-preserving maps. A siamese net is used to learn the geodesic distance between landmark points which, in theory, should be uniformly distributed on the manifold. In [40], a dual autoencoder is combined with mutual information estimation to increase discrimination before spectral clustering. This method's discrimination and robustness to noise allow it to achieve near state-of-the-art results for unsupervised clustering.

By using dimensionality reduction techniques, unsupervised and semi-supervised clustering is performed. Joint Unsupervised Learning [39] is a popular unsupervised clustering method that creates a CNN framework that utilizes the clustering algorithm in the forward pass and representation learning in the backward pass. The two processes benefit from being incorporated together allowing for more accurate clustering and better representations. In [20], the adjacency matrix is put in the loss function to generalize unseen points. Information Maximizing Self-Augmented Training (IMSAT) [17] employs data augmentation, deep neural nets, and stochastic gradient descent to achieve near state-of-the-art results in clustering and unsupervised hash learning. Deep Adaptive Image Clustering (DAC) [5] proposes a semi-supervised approach to image clustering that employs a CNN to generate label features that are then utilized in a pairwise constraint to determine if the images belong in the same cluster. Deep learning is applied to semi-supervised clustering in [41]. This method also experiments with different constraints, such as triplet constraints. In [12], an unsupervised clustering method called DEeP embedded regularIzed ClusTering (DEPICT) is proposed, incorporating a soft-max layer on top of convolutional au-

toencoder through a joint learning framework. Overclustering is prevented by reconstruction loss functions. Another recent method for unsupervised learning, called SpectralNet [34], utilizes spectral clustering approximated by deep learning. This method is useful because of its ability to scale to large data-sets, out-of-sample extension, and adept handling of non-convex clusters. The best results are achieved through the use of a siamese net and autoencoders to encode input data. In [11], a semi-supervised method is adapted from the mean teacher variant to work in a domain adaptation scenario.

Additionally, General Adversarial Networks (GAN) have been used in a wide variety of applications since their introduction in 2014, including with unsupervised clustering. For example, in [15] a GAN is combined with a convolutional encoder and discriminator network for unsupervised clustering. The encoder approximates the inverse of the GAN and learns disentangled images with useful unique characteristics. GANs are also utilized by [28] for unsupervised clustering by implementing an inverse network trained alongside sampled latent variables making clustering in the GAN's latent space possible. In [37], a convolutional siamese network is employed to extract vector features from image data. Invariant Information Clustering (IIC) [18] is an unsupervised clustering method that utilizes CNNs and fully connected layers to maximize information between unlabeled paired data. The method's entropy maximization component of mutual information makes the framework robust to degeneration while an auxiliary overclustering layer eliminates noisy data. The current state of the art unsupervised clustering method for MNIST is Associative Deep Clustering (ADC) [14]. It is a direct unsupervised clustering algorithm that employs a convolutional neural network and centroid variables (embedding-like variables that are part of the model) that are trained alongside the weights through a cost function.

There are also some survey papers on clustering methods such as [1] and [26] that the reader might find instructive.

3. Method

In this section we explain the main contributions of our method. In Section 3.1, we explain our triplet network and loss for training random image pairs. Section 3.2 explains our data augmentation step, and finally in Section 3.3, we discuss our random triplet mining (RTM) algorithm.

3.1. Triplet Network Encoder

In contrast to current methods where autoencoders are trained in order to approximate the weights of the encoder, our model is trained as a whole and from the ground up without the need of a pre-training step.

Our triplet network consists of three encoder streams with shared weights. Each stream takes as input a sample image. We define this sample image as either the anchor (α), positive (β) or negative (γ) image. The anchor and positive images correspond to different augmentations of an original image from our dataset. A negative image is selected in a particular way in order to maximize the likelihood that a chosen image belongs to a different class. We explain more about this procedure in Section 3.3.

Formally, given a training sample x, we define δ_β as the ℓ^2-norm (euclidean distance) between the anchor and positive embeddings and δ_γ the anchor and negative sample distance. We calculate the distance from each stream as:

$$\delta_i^\beta = \|f(x_i^\alpha) - f(x_i^\beta)\|_2^2, \tag{1}$$

$$\delta_i^\gamma = \|f(x_i^\alpha) - f(x_i^\gamma)\|_2^2, \tag{2}$$

$\forall(f(x_i^\alpha), f(x_i^\beta), f(x_i^\gamma)) \in \mathbb{T}$ where $\mathbb{T}$ is the set of all possible triplets i. Note that in practice we do not train on all possible triplets but rather on a subset of $\mathbb{T}$ (Section 3.3). Usually in the context of triplet networks [33], a loss function is defined as:

$$L = \sum_i^N \max(\delta_i^\beta - \delta_i^\gamma + m, 0) \tag{3}$$

where m is a parameter chosen during training.

One of the drawbacks of Eq. 3 is when $|\delta_i^\beta + m| < \delta_i^\gamma$, in such instance there is information loss because the $\max$ function will clip negative values as the whole term goes to zero.

To avoid any value clipping and to prevent such information loss, we modify the loss function of our triple network in the following way:

$$L = \frac{1}{N}\sum_i^N \delta_i^\beta + (m - \sqrt{\delta_i^\gamma})^2 \tag{4}$$

3.2. Online Data Augmentation

In the literature, some methods make use of data augmentation for unsupervised clustering such as in [13]. However, [13] still requires a denoising autoencoder and a preprocess training step to learn the weights of the encoder and decoder. In contrast, our method does not require an autoencoder and can be trained end-to-end. Furthermore, our method makes use of a triplet network to replace the pretrained encoder of an autoencoder-based unsupervised clustering paradigm. Because we do not require offline training of the autoencoder, we can speed up training time by focusing entirely on our triplet encoder training with our data augmentation step.

Traditionally, deep learning methods use data augmentation to create and increase the number of labeled training samples. In contrast, our method employs data augmentation to *create* triplet samples to train our model. Thus, our

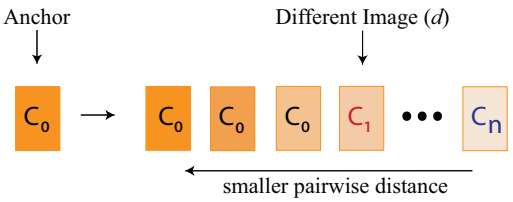


Figure 2: Diagram of Random Triplet Mining process. C_n corresponds to the number of classes or clusters to be considered. d defines the index of the 'closest' different image to the anchor image to be chosen to train at the next iteration.

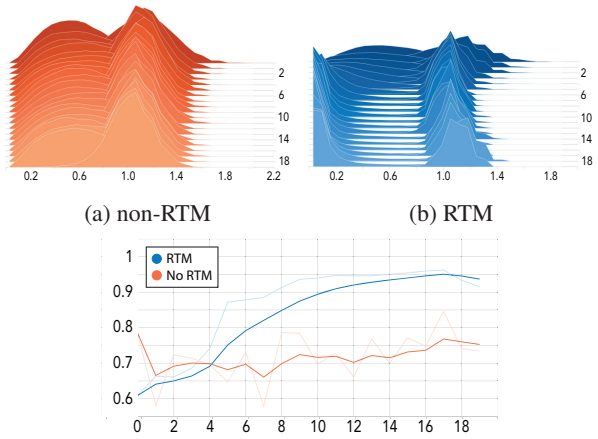
framework heavily depends on our data augmentation process to provide the samples to input into our network. In this way, we select an anchor image and a negative image via Random Triplet Mining (Section 3.3) and perform data augmentation on the anchor and negative image which then become the positive and negative images respectively on the given triplet. This data augmentation process is performed online which avoids the need of storing any extra images.

The augmentations performed include different image transformations such as: scaling, shearing, rotation, color adjustments (brightness, saturation, hue, contrast), horizontal and vertical flips, and random crops. We choose the augmentations depending on each dataset. At each iteration, the network learns from new augmentations which create diverse samples while avoiding the intrinsic bias of the training dataset. This process also helps the model learn from samples not present in the original dataset which then translates into improved clustering accuracy and delayed overfitting of the model.

3.3. Random Triplet Mining

Because the anchor and positive images are generated from the same image via data augmentation, it is paramount to always choose the correct negative image, which is defined in this context as an image of a different class out of the set of possible classes C. By choosing the *hard* samples to train at every iteration, we not only accelerate the learning process of the model but also improve its accuracy (see Figure 3). We call this process of selecting negative pair images, random triplet mining (RTM). Although a non-random form of triplet mining has been used before for supervised learning [33], our RTM method has been modified to be applied in a fully unsupervised way.

Concretely, RTM selects a negative image to train the encoder to distinguish between the positive and negative images. For each anchor image in the dataset, we select p comparison images randomly from the dataset. We then look at the pairwise euclidean distance between the anchor image and each of the comparison images and sort these pairwise



(a) non-RTM (b) RTM

(c) RTM vs. non-RTM

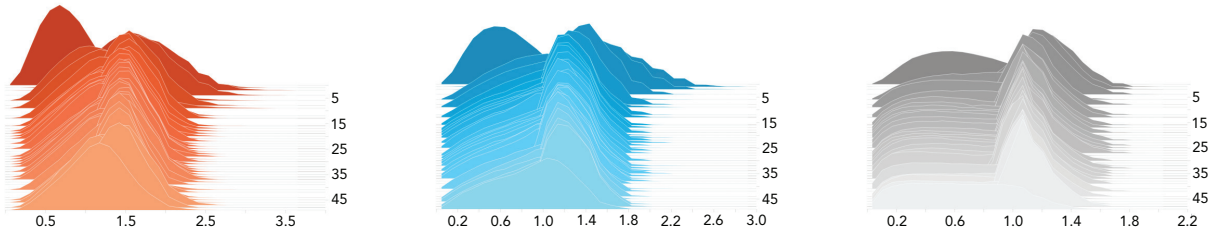
Figure 3: Comparison of our method with RTM and without RTM. Figures (a) and (b) show bi-modal distributions of pair distances of positive and negative pairs during training through time (after 20 epochs). Figure (c) shows the accuracy per epoch. Notice our RTM version (Blue) outperforms the non-RTM version by about 20 percent.

distances. To choose negative samples γ, we pick the index d that represents the d-th position in this sorted list. This index is chosen as a parameter based on the dataset. We can calculate the probability that γ will be a different pair and corresponds to an image in our original training set, and ideally we want this probability to greater than .9.

When calculating the probability our negative image is of a different class we make two important assumptions: 1) we assume that our network is producing embeddings whose distances can be correctly sorted in euclidean space (i.e. similar images from the same class are closer to the anchor image) and 2) we assume that each of the classes contain the same number of training samples. Hence, the approximate probability P that the d-th image is from a different class is given by the following formula:

$$P = \sum_{i=0}^{d} \frac{\binom{c}{i}\binom{c\cdot(n-1)}{p-i}}{\binom{n\cdot c}{p}} \tag{5}$$

where c corresponds to the data points per class, d represents the index of the negative pair γ, n is the number of classes, p is the number of pairs. Given Eq. 5, the goal is to select an index d for the negative pair with high confidence. In our experiments we found $P \approx .97$ to be a good target probability. Figure 3 shows an overlay of the distribution of distances between anchor and positive images (similar pairs) along with the anchor and negative images (different pairs). Note that without RTM the distribution of similar pair distances remains close to the distribution of different



(a) $d = 0$ (b) $d = 1$ (c) $d = 2$

(d) $d = 3$ (e) $d = 4$ (f) $d = 5$

Figure 4: Comparison of the separations of similar and different pair distance distributions based on selecting different d index values. In this case $d = 3$ is the optimal separation because the two distributions are not overlapping or too spread out.

pair distances. As we apply RTM at every iteration, the two distributions start separating which indicates that the model is learning to better discriminate between similar and different pairs. Figure 4 shows how the change of index d affects in different ways the separability of positive and negative pair distances' distributions. This shows that *mining* or choosing the correct negative image index significantly helps the model to learn more representative embeddings.

Finally, we use the embeddings obtained from our triplet network to perform clustering on our dataset. In our experiments we utilized the K-means clustering algorithm.

4. Results

In this section we describe the datasets we used in our experiments and give an overview of our results and a comparison with other state of the art methods.

We ran our method on four different image datasets: MNIST, Fashion-MNIST, CIFAR-10, and xView-10. When applicable, we utilized the full dataset (training and test). The MNIST dataset [24] consists of 70,000 gray-scale images of handwritten digits: zero through nine. The Fashion-MNIST dataset [38] is composed of 70,000 gray-scale images of apparel and accessories. The CIFAR-10 dataset [22] is made up of 60,000 color images that fall into one of ten categories, including airplanes, cars, dogs, and cats. The xView-10 dataset consists of color satellite images derived from the xView dataset [23]. The images are cropped from the existing bounding boxes to be 32x32, and the ten largest classes are combined and balanced to form a set of 60,000 images.

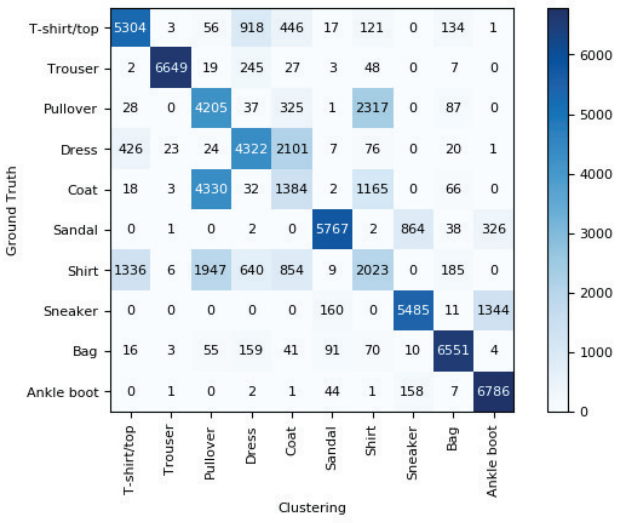


Figure 5: Confusion matrix after applying our method on Fashion-MNIST.

In order to measure the success of our method, we make use of the following metrics: accuracy (ACC) [34], Normalized Mutual Information (NMI) [4], and Adjusted Rand Index (ARI). All of these metric values lie between 0 and 1 where a higher number is better.

Table 2 shows the results of our method on the MNIST dataset. Our method shows competitive results against cur-

Table 1: Dataset Descriptions

Dataset	Size	Classes	Color	Description	Dimensions	Measure
MNIST [24]	70000	10	BW	Handwritten Numbers	(28x28)	EO
Fashion-MNIST [38]	70000	10	BW	Clothing/Shoes	(28x28)	EO
CIFAR-10 [22]	60000	10	RGB	Vehicles/Animals	(32x32)	EO

rent state of the art methods on unsupervised clustering on this dataset.

Table 3 shows a comparison of our method with other current methods on Fashion-MNIST. Our method shows significant improvements over the state of the art and a 4 percent improvement over the best performing method on this dataset [40]. Figure 5 shows the confusion matrix over the 10 classes of Fashion-MNIST.

Table 4 shows the results of our method on CIFAR-10. Notice that in this table our results over-perform other methods that are fully unsupervised. The only method in this table that is better than ours is [17]. However, this method is pre-training their model with ImageNet and thus their results are much better. In contrast, in our method we do not pre-train our model with any other dataset.

Table 2: Unsupervised Clustering Method Comparison for MNIST

Model	ACC	NMI	ARI
ADC 2018 [14]	98.7	-	-
DEC-DA [13]	98.6	96.2	-
IMSAT [17]	98.4	-	-
DAE Network [40]	97.8	94	-
SpectralNet [34]	97.1	92.4	-
BD InfoGAN [15]	96.6	-	-
DEPICT [12]	96.5	92	-
JULE [39]	96.4	91	93
CatGAN [35]	95.7	-	-
InfoGAN [6]	95.0	-	-
ClusterGAN [28]	95.0	89	89
VaDE [19]	94.5	-	-
Our Model	96.8	93.3	93.2

5. Acknowledgements

We would like to thank Dr. Olga Mendoza-Schrock for her mentoring and feedback, Dr. Scott Clouse for his support and input, Mr. Ed Zelnio for his mentoring and encouragement, Dr. Vincent Velten for his technical advice and mentoring, Washington Garcia for his help with fine-tuning our parameters and Ali Heydari for his input on improving our loss function.

Table 3: Fashion-MNIST dataset. Unsupervised Clustering Method Comparison.

Model	ACC	NMI	ARI
DAE Network [40]	66.2	64.5	-
ClusterGAN [28]	63.0	64.0	50
InfoGAN [6, 40]	61.0	59.0	-
DEC-DA [13]	58.0	65.2	-
VaDE [19, 40]	57.8	63.0	-
JULE [39, 40]	56.3	60.8	-
SpectralNet [34]	53.3*	55.2*	-
DEPICT [12, 40]	39.2	39.2	-
Our Model	**70.98**	**68.5**	**57.8**

(*) The autoencoder was pre-trained on the same dataset.

Table 4: CIFAR-10 dataset. Unsupervised Clustering Method Comparison.

Model	ACC	NMI	ARI
IMSAT (Pre-trained) [17]	45.6	-	-
JULE [39, 18]	27.2	-	-
ADC 2018 [14]	26.7	-	-
SpectralNet [34]	21.8*	-	-
Our Model	**30.9**	**19.7**	**11.5**

(*) The autoencoder was pre-trained on the same dataset.

This work was funded by the Air Force Office of Scientific Research. The documentation has been approved for public release by the U.S. Air Force 88th Air Base Wing with PA approval number 88ABW-2019-3718.

6. Conclusion

In this paper, we have presented a novel method for unsupervised clustering. Our method makes use of a triplet network trained on data augmented pairs chosen in a special and particular order by our Random Triplet Mining method to avoid overfitting the model.

Our method avoids the need for a separate autoencoder being trained offline. Our method allows us to train our model end-to-end to learn the manifold of the data and produce superior embeddings. Our method shows competitiveness by yielding state of the art results on Fashion-MNIST.

References

[1] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*, 2018.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[3] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, pages http–openreview, 2014.

[4] D. Cai, X. He, and J. Han. Locally consistent concept factorization for document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 23(6):902–913, 2010.

[5] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888. IEEE, 2017.

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[7] C. K. Chui and H. N. Mhaskar. Deep nets for local manifold learning. *Frontiers in Applied Mathematics and Statistics*, 4:12, 2018.

[8] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[9] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[11] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, number 6, 2018.

[12] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017.

[13] X. Guo, E. Zhu, X. Liu, and J. Yin. Deep embedded clustering with data augmentation. In *Asian Conference on Machine Learning*, pages 550–565, 2018.

[14] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers. Associative deep clustering: Training a classification network with no labels. In T. Brox, A. Bruhn, and M. Fritz, editors, *Pattern Recognition*, pages 18–32, Cham, 2019. Springer International Publishing.

[15] T. Hinz and S. Wermter. Inferencing based on unsupervised learning of disentangled representations. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2018.

[16] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[17] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1558–1567. JMLR. org, 2017.

[18] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018.

[19] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1965–1972. AAAI Press, 2017.

[20] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[21] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

[22] A. Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[23] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

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

[25] G. Lewis and W. Yang. Augmenting nearest neighbor-based algorithms with siamese neural networks. 2016.

[26] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018.

[27] G. Mishne, U. Shaham, A. Cloninger, and I. Cohen. Diffusion nets. *Applied and Computational Harmonic Analysis*, 47(2):259 – 285, 2019.

[28] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the... AAAI Conference on Artificial Intelligence*, 2019.

[29] G. Pai, R. Talmon, A. Bronstein, and R. Kimmel. Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 819–828. IEEE, 2019.

[30] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[31] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[32] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[34] U. Shaham, K. Stanton, H. Li, R. Basri, B. Nadler, and Y. Kluger. Spectralnet: Spectral clustering using deep neural networks. In *International Conference on Learning Representations*, 2018.

[35] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[36] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[37] D. J. Trosten and P. Sharma. Unsupervised feature extraction–a cnn-based approach. In *Scandinavian Conference on Image Analysis*, pages 197–208. Springer, 2019.

[38] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[39] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[40] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4066–4075, 2019.

[41] H. Zhang, S. Basu, and I. Davidson. Deep constrained clustering-algorithms and advances. *arXiv preprint arXiv:1901.10061*, 2019.