Augmented Data as an Auxiliary Plug-In Toward Categorization of Crowdsourced Heritage Data

Shashidhar Veerappa Kudari, Akshaykumar Gunari, Adarsh Jamadandi, Ramesh Ashok Tabib, and Uma Mudenagudi

1 Introduction

In this paper, we propose a novel training mechanism to mitigate problems such as data sparsity, high inter-class variance, and low intra-class variance which leads to poor clustering performance. Traditional clustering algorithms such as K-means, Gaussian mixture models (GMM) [3], and spectral clustering [8] rely largely on the notion of *distance*; for example, K-means [11] uses Euclidean distance to assign data points to clusters. Recent advances in deep learning have led to emergence of clustering techniques parameterized by deep neural networks [2, 13, 14, 17, 18] attempting to jointly learn representations, and perform clustering relying on tools like Stochastic Gradient Descent and backpropagation with a clustering objective function. This introduces challenges in choosing an appropriate neural network architecture, and a right clustering objective function. Recent methods [5, 16], attempt to circumvent these problems, limited literature show investigations on the effect of data sparsity and high intra-class variance, usually found in crowdsourced cultural heritage datasets. The apparent architectural differences arise due to data acquisition methods and cultural similarities might lead to assignment of false clusters. In this paper, we empirically demonstrate the use of different transformations such as random—

S. Veerappa Kudari (✉) · A. Gunari · A. Jamadandi · R. A. Tabib · U. Mudenagudi
KLE Technological University, Hubballi, India
e-mail: shashidharvk100@gmail.com

A. Jamadandi
e-mail: adarsh.cto@tweaklabsinc.com

R. A. Tabib
e-mail: ramesh_t@kletech.ac.in

U. Mudenagudi
e-mail: uma@kletech.ac.in

U. Mudenagudi et al. (eds.), *Proceedings of the Satellite Workshops of ICVGIP 2021*,
Lecture Notes in Electrical Engineering 924,
https://doi.org/10.1007/978-981-19-4136-8_4

scaling, rotation, and shearing as data augmentation techniques toward increasing the data density, yielding superior clustering performance.

Crowd-sourcing facilitates desired data at scale and involves task owners relying on a large batch of supposedly anonymous human resources with varying expertise contributing a diversified amount of data. In our case, we are interested in obtaining a large image corpus of Indian Heritage Sites with the hindsight of large scale 3D reconstruction toward digital archival and preservation. An essential step in this pipeline is to formulate an efficient deep clustering method toward mitigate the issues outlined above. Toward this-

- We propose a novel training strategy to circumvent the problem of poor clustering performance by
  - introducing data augmentation as an auxiliary plug-in for deep embedded clustering

    to densify data and facilitate better feature representation considering limited data.
    to address data with high intra-class and low inter-class variance.
    to augment data using affine transforms (rotation, scaling and shearing).

  - incorporating Consistency Constraint Loss (CCL) with Mean Squared Error (MSE) Loss to handle introduced transformations.
- We demonstrate our proposed strategy on a crowdsourced Indian heritage dataset and show consistent improvements over existing works.

In Sect. 2, we discuss contemporary works related to clustering. In Sect. 3, we propose a strategy to circumvent the problem of poor clustering performance. In Sect. 4, we discuss the experimental setup carried out on Indian Heritage Dataset. In Sect. 5, we demonstrate results through quantitative and qualitative metrics, and conclude in Sect. 6.

2 Related Works

In this section, we discuss contemporary works addressing clustering using deep features. Classical clustering techniques such as K-means [11], Gaussian Mixture Models (GMM) [3], and Spectral clustering [8] are limited by their distance metrics and perform poorly when the dimensionality is high. Toward this, recent techniques such as Deep Embedded Clustering (DEC) [16], Improved Deep Embedded Clustering [5] extract deep features toward categorization in lower dimension embedding space.

Recent advances in deep neural networks have ushered in a strategy of parameterizing clustering algorithms with neural networks. Deep Embedded Clustering (DEC),

proposed by authors in [16], pioneered the idea of using deep neural networks to learn representations and solve for cluster assignment jointly. The method involves using Stochastic Gradient Descent coupled with backpropagation to extract deep features while simultaneously learning the underlying representations. However, as authors in [5] point, the choice of clustering loss tends to distort the feature space, which consequently affects the overall clustering performance. To mitigate this, the authors propose an under-complete autoencoder to preserve the data structure, leading to improved clustering performance. Inspired by these works, we propose a method to improve Clustering performance by densifying the data distribution. We hypothesize that data distribution sparsity is a significant deterrent in clustering. The problem is further exacerbated when the data exhibits high intra-class variance. We empirically show that using data augmentation as an auxiliary plug-in helps in improving cluster performance. Extensive experiments on cultural heritage dataset show consistent improvements over existing methods.

3 Categorization of Crowdsourced Heritage Data

The crowdsourced heritage data arrives in the incremental fashion, where the number of classes and number of images belonging to class are obscure. More likely, we observe that images belonging to a particular class may arrive in large number while very few samples may arrive for some other classes. This brings the problems of class imbalance and data sparsity. Due to data sparsity, deep learning techniques used for the feature representation of the images fail in their task, making the clustering performance poor. Deep learning architectures like Convolutional Autoencoders (CAE) are sensitive to these problems. Toward this, we attempt to mitigate the data sparsity issue via data augmentation (Fig. 1).

We increase the density of the data, by performing three kinds of data transformations, i.e., random rotation, random shear, and random scaling on the original data, as this would generally make the model more robust in terms of learning. These techniques tend to provide more generic and genuine data. There are many other augmentation techniques as described in [1, 10, 12, 15] used to increase the data density and class imbalance [6, 19].

Convolutional Autoencoder (CAE) has proven to be effective in case of classification, clustering, and object detection. We combine the augmented data with original data to train CAE to generate embeddings toward clustering.

Considering x_i, $i \in \{1, ..., m\}$ as an image, the transformation t_j, $j \in \{1, .., s\}$ applied on x_i generates transformed image $x_i^{t_s}$, represented as $x_i^t = T(x_i)$. The total number of images after augmentation are N, where $N = s \times m$.

The traditional objective function used for training the CAE is Mean Squared Error(MSE) between the input x_i and decoder output $x_i^{'}$ which is as

$$MSE(x, x^{`}) = \frac{\sum_{i=1}^{N} (x_i - x_i^{`})^2}{N} \tag{1}$$

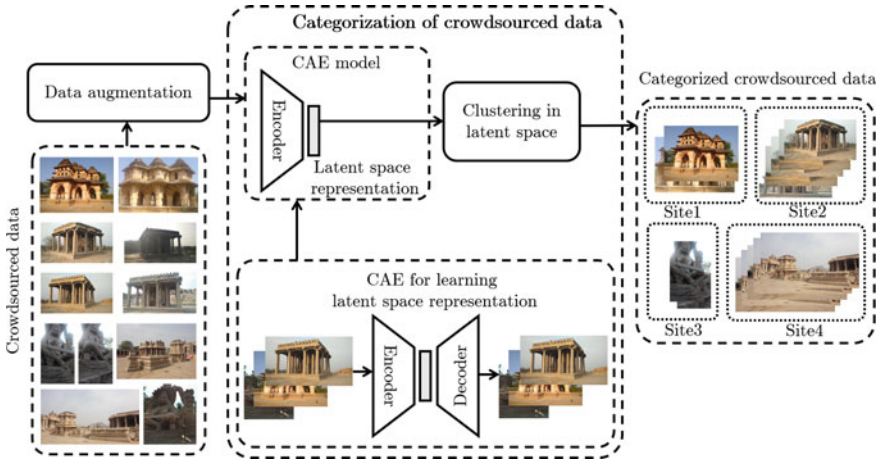

Fig. 1 Categorization of crowdsourced heritage data

The MSE loss for training CAE limits information about the relation between original data and the augmented data. To overcome this, we incorporate the Consistency Constraints [9]. The Consistency Constraints are seen to be effective in the Semi Supervised Learning (SSL). A Consistency Constraint Loss (CCL) can be incorporated by enforcing the predictions of a data sample and its transformed counterpart (which can be obtained by randomly rotating, shearing, or scaling the images) to be minimal. The CCL Loss is defined as follows:

$$L_{CCL} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \parallel p(k|i) - p^t(k|i) \parallel \tag{2}$$

where N represents the total number of data points and the K represents total number of clusters, $p(k|i)$ represents the probability of assignment of each image x_i, and $p^t(k|i)$ represents the probability of assignment of randomly transformed image x_i^t to cluster k. $p(k|i)$ is parameterized by assuming they follow the Student's T distribution as follows:

$$p(k|i) \propto (1 + \frac{\parallel z_i + \mu_k \parallel^2}{\alpha})^{-\frac{\alpha+1}{2}} \tag{3}$$

Here z_i is the feature representation of the image x_i, μ_k represents the cluster center of cluster k. If U represents the cluster centers then $U = \{\mu_k, k = 1....K\}$ which are initialized by K-means and tuned as the training progress.
The overall objective function of CAE is now defined as

$$Loss = MSE(x, x') + L_{CCL} \tag{4}$$

We use K-means technique to quantify our results and depict how augmentation can improve the performance of the clustering. We show how data augmentation can improve the performance of the existing state of art methods Deep Embedded Clustering (DEC) and Improved Deep Embedded Clustering (IDEC), where CAE is used as the initial feature extractor. We provide the extensive ablation study of these methods over the combinations of different CAEs trained.

4 Experiments

4.1 Dataset

We extensively experiment on crowdsourced Indian Digital Heritage (IDH) Dataset. The dataset is collected through a platform sourced by crowd. We consider 10 classes of this dataset with 150 images per class toward experimentation as these 10 classes consists of high intra-class and low inter-class variance. The considered dataset undergoes augmentation like random rotation, random shearing, and random scaling. Random rotation of images is performed over the range of 0–90°, random shear is performed over 50° of transformation intensity and random scaling is performed over the scale of 0.5–1.0. We generate around 6000 images through these transformations. The same dataset is used throughout the experimentation to maintain the uniformity in comparison of results in different experiments in different environments.

4.2 Training Setup

- Runtime Environment: Nvidia GP107CL Quadro P620
- Architecture: Autoencoder
  - Encoder:

    Contains 4 VGG Blocks
    VGG Block has 2 Convolution Layers followed by a maxpooling layer
    Batch normalization layer was used at the end of each layer before the activation function
    Activation Function: ReLU

  - Decoder:

    Decoder part of the model consists of convolution transpose layers with batch normalization layer at the end of each layer before the activation function
    Activation Function: ReLU, Sigmoid (Output Layer)

- Batch Size: 16
- Learning Rate: 0.001
- Number of Epochs: 500 (CAE), 2000 (DEC and IDEC)
- Optimizer: Adam

4.3 Evaluation Metrics

Toward evaluation of proposed strategy and comparison with state-of-the-art methods, we use Unsupervised Clustering Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI).

4.3.1 Unsupervised Clustering Accuracy (ACC):

It uses a mapping function m to find the best mapping between the cluster assignment output c of the algorithm with the ground truth y which can be defined as

$$ACC = max_m \frac{\sum_{i=1}^{N} 1\{y_i = m(c_i)\}}{N} \tag{5}$$

For the given image x_i, let c_i be resolved cluster label and y_i be the ground truth label, m is the delta function [7] that equals one if $x = y$ and zero otherwise. m maps each cluster label c_i to the equivalent label from the datasets. The best mapping can be found by using the Kuhn-Munkres algorithm [4].

4.3.2 Normalized Mutual Information (NMI)

It measures the mutual information $I(y, c)$ between the cluster assignments$c$ and the ground truth labels y and is normalized by the average entropy of both ground labels $H(y)$ and the cluster assignments $H(c)$, and can be defined as

$$NMI = \frac{I(y, c)}{\frac{1}{2}[H(y) + H(c)]} \tag{6}$$

4.3.3 Adjusted Rand Index (ARI)

It computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. It is defined as

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \tag{7}$$

5 Results and Discussions

In this section, we discuss the results of the proposed strategy toward categorization of crowdsourced Indian Heritage (IDH) dataset and compare the results with state-of-the-art methods.

We measure the clustering performance by reporting the unsupervised clustering accuracy, NMI and ARI. From Table 1 we observe, CAE trained with data augmentation yields better performance over CAE trained without augmentation. We see an improvement of **2.74%** when trained with MSE loss and an improvement of **15.21%** when trained with a combination of MSE and CCL, as CCL loss mainly depends on augmented data. This improvement is significant in the context of clustering data with high intra-class variance.

To discern the effect of individual augmentation techniques (rotate, scale and shear), we choose samples in the combination of—{ori, rot}, {ori, sher} and {ori, scal}. The results are presented in Table 2. We observe, set {ori, rot} shows poor performance compared to augmentations consisting of {ori, sher} and {ori, scal}. We hypothesize that the performance drop for {ori, sher} can be attributed to the fact that, the CAE is not equipped with appropriate symmetry inductive bias that enables it to learn rotation-invariant features.

Table 1 Performance of CAE trained with and without augmented data. CAE-WAug represents CAE model trained without augmented data and CAE-Aug represents CAE model trained with augmented data

	MSE			MSE + CCL		
	ACC	NMI	ARI	ACC	NMI	ARI
CAE-WAug	0.5424	0.4880	0.3377	0.4035	0.3752	0.1880
CAE-Aug	**0.5698**	**0.5327**	**0.4000**	**0.5566**	**0.4936**	**0.3336**

Table 2 Effect of augmentation on performance of the original data. Original, rotated, sheared, scaled data are represented as ori, rot, sher and scal, respectively

	CAE – MSE			CAE – (MSE + CCL)		
	ACC	NMI	ARI	ACC	NMI	ARI
ori + rot	0.4355	0.3168	0.2155	0.3491	0.2681	0.1493
ori + sher	0.5189	0.4480	0.3144	0.4413	0.3574	0.2091
ori + scal	0.5183	0.4168	0.2846	0.4346	0.3419	0.2151

5.1 Ablation Study

In this section, we perform ablation study using DEC [16] and IDEC [5] with and without considering augmentation. In Table 3, we provide the ablation study of DEC which is unsupervised clustering technique that jointly optimizes the cluster centers and the parameters of the CAE. KL-divergence between the auxiliary and target distribution optimizes the objective function. From Table 3, we infer that providing CAE with the augmented data followed by DEC considering original data increases the accuracy by **5.61%**. While providing augmented data to DEC, with CAE being trained with original data hinders the performance. Hence, only CAE is trained with original and augmented data ensuring the objective is met.

In Table 4 we provide the ablation study of the Improved Deep Embedded Clustering (IDEC). IDEC is an improvement over IDEC, which not only jointly optimize the cluster centers and parameters of the CAE, but also preserve the local structure information. They use KL-divergence between the auxiliary and target distribution as their objective function along with the MSE loss of the CAE. From Table 4 it can be observed that providing CAE with augmented data with MSE+CCL loss, then providing the trained CAE to IDEC, where IDEC is trained on original data improves the performance by depicting the increase in accuracy by **7.43%**. While training the IDEC with augmented data with CAE trained on original data only hinders the performance.

From the experiments we observe, it is better to train the CAE with MSE+CCL as the integrity loss, with augmented data. The CAE trained in such an environment is incorporated for initial feature representation to the IDEC by providing original data, to perform better than other methods.

Table 3 Comparing results of proposed methodology with DEC [16]. CAE-WAug and CAE-Aug refers to CAE trained without and with augmentation, respectively. DECWAug and DEC-Aug refers to DEC trained without and with augmentation respectively. We show how combination of augmentation applied to CAE and DEC may affect the clustering performance

Loss →	MSE			MSE + CCL		
Method ↓, Metric →	ACC	NMI	ARI	ACC	NMI	ARI
DEC [16] CAE – WAug + DEC – WAug	0.4113	0.3874	0.2271	0.3625	0.3625	0.8196
CAE – WAug + DEC Aug	0.3096	0.3013	0.1530	0.2927	0.2210	0.1201
CAE – Aug + DEC WAug	**0.4674**	**0.4876**	**0.3076**	**0.4492**	**0.4665**	**0.2716**

Table 4 Comparing results of the proposed methodology with IDEC [5]. CAE-WAug and CAE-Aug refers to CAE trained without and with augmentation respectively. IDECWAug and DEC-Aug refer to IDEC trained without and with augmentation, respectively. We show how the combination of augmentation applied to CAE and IDEC may affect the clustering performance

Loss →	MSE			MSE + CCL		
Method ↓ , Metric →	ACC	NMI	ARI	ACC	NMI	ARI
IDEC [5] CAE – WAug + IDEC – WAug	0.5098	0.4903	0.3020	0.3625	0.4340	0.2511
CAE – WAug + IDEC Aug	0.3611	0.3300	0.1703	0.3514	0.3335	0.1856
CAE – Aug + IDEC WAug	**0.4983**	**0.4942**	**0.3130**	**0.5841**	**0.4340**	**0.3662**

6 Conclusions

In this paper, we have defined data augmentation as an auxiliary plug-in for deep embedded clustering that densifies data helping in accurate clustering performance. We have demonstrated how data augmentation helps to increase the data density yielding superior clustering performance when the data is considerably less in amount. Extensive experimentation is done on setting up the right objective function. Our main objective is to cluster data with very less inter-class variance and very high intra-class variance. We have demonstrated our experiments on crowdsourced heritage dataset. We also show, how certain augmentation techniques uphold the clustering objectives (such as random shear and random scale), while some of them hinders the same (random rotation). We demonstrate our results on Indian Digital Heritage (IDH) dataset to show our methodology shows better performance to state-of-the-art clustering algorithms. Deploying clustering algorithms for critical applications warrants circumspection and is still a work in progress and we believe our work is a step in this direction.

Acknowledgements This project is partly carried out under the Department of Science and Technology (DST) through ICPS program—Indian Heritage in Digital Space for the project “Crowd-Sourcing” (DST/ ICPS/ IHDS/ 2018 (General)) and “Digital Poompuhar" (DST/ ICPS/ Digital Poompuhar/ 2017 (General)).

References

1. Bloice M, Stocker C, Holzinger A (Aug 2017) Augmentor: an image augmentation library for machine learning. J Open Sour Softw 2 . https://doi.org/10.21105/joss.00432
2. Caron M, Bojanowski P, Joulin A, Douze M (2019) Deep clustering for unsupervised learning of visual features
3. Chen J, Zhang L, Liang YC (May 2019) Exploiting Gaussian mixture model clustering for full-duplex transceiver design. IEEE Trans Commun 1. https://doi.org/10.1109/TCOMM.2019.2915225
4. Cui H, Zhang J, Cui C, Chen Q (Jan 2016) Solving large-scale assignment problems by Kuhn-Munkres algorithm. https://doi.org/10.2991/ameii-16.2016.160
5. Guo X, Gao L, Liu X, Yin J (2017) Improved deep embedded clustering with local structure preservation. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17, pp 1753–1759. https://doi.org/10.24963/ijcai.2017/243
6. Guo X, Zhu E, Liu X, Yin J (2018) Deep embedded clustering with data augmentation. In: Zhu J, Takeuchi I (eds) Proceedings of The 10th Asian conference on machine learning. Proceedings of machine learning research, 14–16 Nov 2018, vol 95, pp 550–565. PMLR. http://proceedings.mlr.press/v95/guo18b.html
7. Gupta SC (1964) Delta function. IEEE Trans. on Educ. 7(1):16–22. https://doi.org/10.1109/TE.1964.4321835,
8. Hamad D, Biela P (2008) Introduction to spectral clustering. In: 2008 3rd international conference on information and communication technologies: from theory to applications, pp 1–6. https://doi.org/10.1109/ICTTA.2008.4529994
9. Han K, Vedaldi A, Zisserman A (2019) Learning to discover novel visual categories via deep transfer clustering
10. Inoue H (2018) Data augmentation by pairing samples for images classification. https://openreview.net/forum?id=SJn0sLgRb
11. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892. https://doi.org/10.1109/TPAMI.2002.1017616
12. Mikołajczyk A, Grochowski, M (2018) Data augmentation for improving deep learning in image classification problem. In: 2018 international interdisciplinary PhD workshop (IIPhDW), pp 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338
13. Mrabah N, Khan NM, Ksantini R, Lachiri Z (2020) Deep clustering with a dynamic autoencoder: from reconstruction towards centroids construction
14. Ren Y, Wang N, Li M, Xu Z (2018) Deep density-based image clustering. CoRR **abs/1812.04287**
15. Shorten C, Khoshgoftaar T (2019) A survey on image data augmentation for deep learning. J Big Data 6:1–48
16. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis
17. Yang J, Parikh D, Batra D (2016) Joint unsupervised learning of deep representations and image clusters
18. Zhan X, Xie J, Liu Z, Ong YS, Loy CC (Jun 2020) Online deep clustering for unsupervised representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
19. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2017) Random erasing data augmentation