DOI 10.29042/2018-3561-3566

Helix Vol. 8(4): 3561- 3566

Semantic Image Clustering with Global Average Pooled Deep Convolutional Autoencoder

[1]Morarjee Kolla, [2] Dr. T. VenuGopal

[1]Research Scholar, Department of CSE, JNTUA, Anantapuramu, Andhra Pradesh, India.

[2]Professor, Department of CSE, JNTUH College of Engineering Jagityal, Nachupally, Telangana, India.

Email: **[1]morarjeek@gmail.com, [2] t_vgopal@rediffmail.com*

Received: 5th March 2018, Accepted: 9th April 2018, Published: 30th June 2018

Abstract

Deep Clustering learns feature representations in embedded space suitable for clustering. In Deep Convolutional Embedded Clustering (DCEC) algorithm, the last convolution layer feature map of encoder is used to build the embedded space. This considers spatial information retains in the last convolution layer of encoder, which unable to identify the discriminative parts of the image. To address this issue, we propose a solution using Global Average Pooling (GAP) of the last convolution layer feature maps in the encoder. This will encourage the network to identify all discriminative regions and an extent of an object to formulate semantic image clusters (SIC). Our experimental results prove the efficiency of proposed Global Average Pooled Deep Convolutional Embedded Clustering (GAPDCEC) for simultaneous feature learning and clustering.

Keywords: *Deep Clustering, Deep Convolutional Embedded Clustering, Global Average Pooling, Global Average Pooled Deep Convolutional Embedded Clustering, semantic image clusters*

1.Introduction

Clustering is an Unsupervised data analysis and visualization methodology for grouping similar images in Content based image retrieval (CBIR). Most of the clustering methodologies deal with different distance functions to measure the similarity between the raw pixels on data space. These distance driven approaches on handcrafted images are still in an infancy stage to extract meaningful clusters. Mapping raw pixel representations with human understandable representations is a quite challenging task. Recently data-driven approaches are more successful in building cluster friendly feature space. Clustering is a complex process in large data space. To avoid this, recent approaches [1,2,3] are using shallow embedded space.

Semantic Image Clustering (SIC) is the unsupervised machine learning approach using knowledge representation concepts to group unlabelled images meaning fully. It leans the efficient discriminative features to reduce semantic gap in retrieval systems. Most popular methodology in this area uses relevance feedback(RF) [4]. Some other approaches are using knowledge representation mechanisms like Ontology [5] in their contributions. RF deals with low level representation based traditional approach with human judgement. Ontology deals with concepts, constraints and domain knowledge. However, these approaches are still facing difficulties with complex structure of unlabelled images.

Recent success in deep learning approaches are using a shallow embedded space through representation learning [1] for feature learning and clustering individually. Some other approaches [3] are jointly learning features and clustering in embedded space with high performance.

Some of the interesting questions in deep learning are still unclear. For example, which framework CNN features are more suitable for clustering? How is the memory space to store these deep features? What type of autoencoders are more suitable for effective clustering? Which methodology will be useful in creating cluster friendly embedded space? Which loss functions are effective in reducing both clustering and reconstruction loss of deep autoencoders?

Our main idea is to reduce the space for valid deep features suitable for clustering. This embedded latent space captures the structure of data and used to group similar data points.

2.Related Work

Research on SIC uses Relevance Feedback, Ontology and deep learning. Some of the interesting contributions are summarized below.

SIC is developed based on RF approaches [4] are suffers with exact interpretation of context meaning. To overcome this difficulty, they are heavily depending on several iterations of human feedback. Different kinds of ontologies like domain ontology, constraint ontology, evaluation ontology [6, 7, 8, 9] was used in existing research contributions. Some other contributions are using both relevance feedback and ontology as a hybrid approach [10]. However, these approaches are not meeting the expectations of user in producing relevant results.

Nowadays deep representations are using heavily to perform clustering, make use of different types of auto-encoder representations, structure designs and clustering performance evaluations [1,3,11,12,13,14]. Stacked autoencoders(SAE) [1,11,12,15] are widely used in deep clustering. Due to layer-wise pretraining and involved complicated

procedures related to these methodologies, SAE consumes more resources for execution. Clustering loss is used to tune the parameters and cluster centres simultaneously in Deep Embedded Clustering(DEC) [1]. DEC feature space may be corrupted due to non-preservation of data properties. This problem is overcome by DCEC [3] with the help of local structure preservation. DCEC used Convolution autoencoder to learn feature and local structure preservation. DEPICT [13] prevents allocation of clusters to outlier samples through regularization and effectively maps a data into nonlinear embedded space. JULE [2], based on a recurrent framework and iteratively clustered using agglomerative clustering algorithm.

Conventional clustering methods are high computation cost for large-scale datasets. K-means is more suitable for large-scale clustering. But reducing the storage of the large amount of data computation cost is a challenging task. Our proposed approach will overcome these difficulties.

- Our main idea to introduce GAP to reduce overfitting, identify all discriminative regions. The contributions are Deep autoencoder with GAP provides all discriminative nonlinear embedded space.
- End to End joint learning approach, which simultaneously learns feature representation and clustering.
- GAP will be used to regularize model parameters and it overcomes the overfitting problem by reducing parameters.
- Existing methods are used to identify one discriminative region. But our method act as a detector to identify different patterns in the image.
- CNN followed by GAP will be used to identify the object regions and localizations in the image.

3. Global Average Pooled Deep Convolutional Embedded Clustering (GAPDCEC)

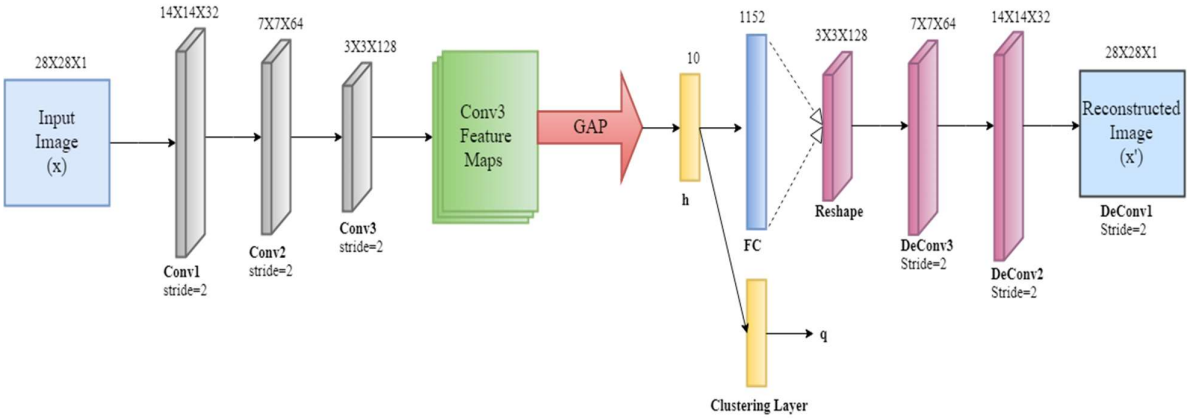

Fig. 1. The representative model of proposed GAPDCEC for MNIST

We modified the DCEC by introducing GAP to the last convolutional layer as shown in Fig. 1. In DCEC one discriminative region will be identified. But in GAPDCEC all discriminative regions along with their localizations will be identified. Consider a dataset x of the MNIST image as an input digit image (0-9) of size 28X28X1. This input image is convoluted with stride of size 2, then convert the image after convolution1 to the size of 14X14X32 feature maps. Same convolution operations are repeated up to convolution3. Then resulting the feature maps of size 3X3X128. We performed GAP on these feature maps instead of they flatten in DCEC. Then fully connected embedded layer with only 10-dimensional feature space. We keep the decoder as it is like in DCEC to learn upsampling.

3.1 Deep Convolutional Autoencoder

Convolutional Autoencoder [3] has two layers, corresponding to encoder $f_W(.)$ and decoder $g_U(.)$ is generally defined as

$$f_W(x) = \sigma(x * \mathrm{W}) \equiv h$$
$$g_U(h) = \sigma(h * \mathrm{U}) \quad (1)$$

where $f_W(x)$ and $g_U(h)$ are encoder and decoder respectively. x, h are vectors. σ is activation function. * is Convolution operator.

3.2 Global Average Pooling

In DCEC [3], the last convolution layer output of the encoder is flattened and fed by densely connected embedded layer with 10 units. This structure unable to capture discriminative regions of image. To avoid this in our Proposed GAPDCEC, we use the Global Average Pooling(GAP). Taking an inspiration from

Network-in -Network [16] and Googlenet [17], GAP followed by fully connected will improve localization ability with mapping resolution.
Global Average pooling (F^k) is defined [18] as

$$F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k \quad (2)$$

Where A_{ij}^k is Activation at location i,j of feature map A^k and Z is the number of pixels in the feature map.
We modified the encoder function in our case with a convolution followed by GAP is defined as

$$f_W(x) = \sigma[(x*W) + F^k] \equiv h \quad (3)$$

3.3 Clustering Loss and Reconstruction Loss

We calculated Clustering loss and reconstruction loss same as DCEC [3]. We pretrain the convolutional autoencoder before performing the clustering. Then remove the decoder and finetune the encoder using below objective function. Embedded points(Z_i) are valid feature representations for input data samples. Custer centres(μ_j), autoencoder weights (W, U) can be initialized by using K-means on Z_i [3].
Clustering loss is computed as

$$L_c = KL(P||Q) = \sum_i \sum_j p_{ij}\ log\ \frac{p_{ij}}{q_{ij}} \quad (4)$$

where p_{ij} is the target distribution, q_{ij} is the student's t-distribution.
Re-construction loss is defined as

$$L_r = \sum_{i=1}^{n} || x_i - x_i' ||_2^2 \quad (5)$$

where x_i is an input image and x_i' is re-constructed image.
Our objective loss function is defined as

$$L = L_r + \gamma L_c \quad (6)$$

Where $\gamma > 0$ is a coefficient that controls degree of distortion.

3.4 Optimization

We optimize (6) using mini batch stochastic gradient decent (SGD) with momentum and backpropagated. Our optimization function updates convolutional autoencoder weights, cluster centres and target distribution P same as DCEC [3].
Update the cluster centres by

$$\mu_j = \mu_j - \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\partial L_c}{\partial \mu_j} \quad (7)$$

Where λ is Learning Rate and m is Mini batch

Update the autoencoder weights by

$$W = W - \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\partial L_r}{\partial W} + \gamma \frac{\partial L_c}{\partial W}$$
$$U = U - \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\partial L_r}{\partial U} \quad (8)$$

We update the target distribution same as DCEC by considering stopping threshold δ [3].

The entire GAPDCEC algorithm is summarized below

Algorithm: GAPDCEC algorithm
Input:
Input data: X; Number of dataset samples: n;
Number of clusters: K; Learning Rate λ; Mini batch m; Stopping threshold: δ;
Output:
Autoencoder weights: W, U; Cluster centres: μ_j;
Embedded points: Z_i; Clustering Loss: L_c;
Reconstruction Loss: L_r ; Accuracy: Accuracy;
Normalized Mutual Information: NMI;
Algorithm:
Initialize μ_j, W, U \\according to section 3.3.
Compute all embedded points $Z_i = \{ f_W(x_i) \}_{i=1}^n$
Compute L_c, L_r, L using (4), (5), (6)
while not Converged **do**
Update μ_j using (7)
Update W,U using (8)
Update $\{Z_i\}_{i=1}^n$
Update Target distribution P
\\according to section 3.3
Ret urn μ_j, W, U

4. Experiments

We evaluate the proposed method (GAPDCEC) on three image datasets and compare the performance with other algorithms. We establish qualitative and quantitate experimental results that prove the efficiency of GAPDCEC compared to other competitive algorithms. Experiments are conducted on MNIST-Full [19], MNIST-Test [19], USPS datasets.

4.1 Experimental Setup

Experiments are conducted in intel i7 processor laptop with 8GB RAM and 4GB NVIDIA GPU memory. We developed our methodology using Python, TensorFlow and Keras.

4.1.1 Comparison Methods

We showed the efficacy of our system by comparing our algorithm with five state-of-the-art algorithms K-means, DEC, IDEC, DCN and DCEC. Existing GitHub codes are used for all comparative methods.

4.1.2 Parameter Setting

DEC, IDEC algorithms use dimensions for encoder as d-500-500-2000-10 and for decoder 10-2000-500-500-d for all data sets. All internal layers other than input, output and embedded layers are activated by ReLu. DCEC and GAPDCEC algorithms use convolution filters of 32,64,128 and kernels of size 5X5, 5X5, 3X3 for con1, con2, cov3 respectively with stride 2. we fixed a learning rate of 0.01, a momentum of 0.9, a convergence threshold of 0.1% and an update interval of 140.

  ISSN 2319 – 5592 (Online)

4.1.3 Evaluation Metric

The Clustering methods are tested with the aid of Accuracy(ACC) and Normalized Mutual Information(NMI) [1].

4.2 Results

Table 1. Performance Comparison of clustering algorithms on three databases

Datasets	MNIST-Full		MNIST-Test		USPS	
Criterion	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**
K-means	54.24	48.52	54.63	50.18	66.82	62.66
DEC	84.08	81.28	69.94	67.69	69.28	70.18
IDEC	84.21	83.81	71.45	69.40	72.10	73.23
DCN	83.24	81.34	69.86	67.34	69.14	70.12
DCEC	88.97	88.49	85.29	83.61	79.00	82.57
GAPDCEC	**89.54**	**89.12**	**87.17**	**85.03**	**80.36**	**83.56**

Fig. 2. Datasets VS Accuracy

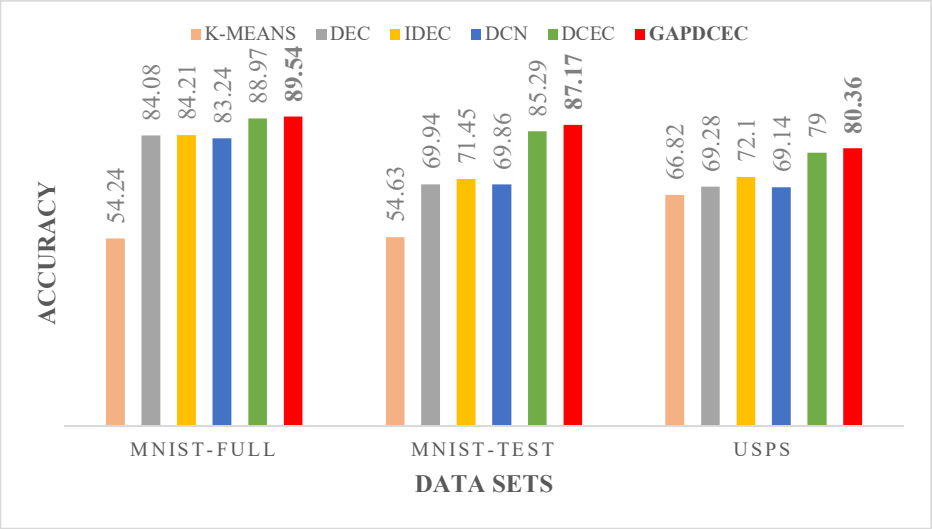

Fig. 2. Datasets VS Accuracy

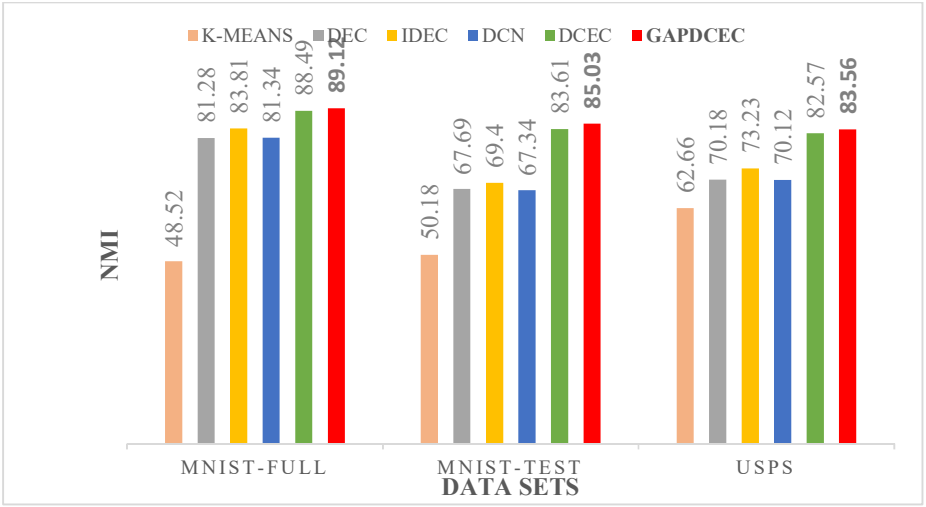

Fig. 3. Datasets VS NMI

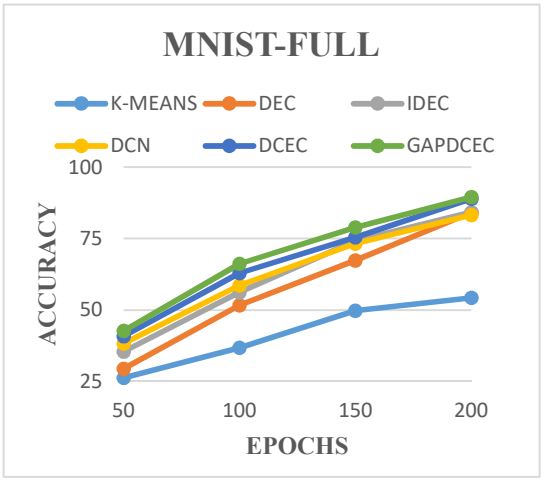

Fig. 4. Epochs VS Accuracy for MNIST-Full

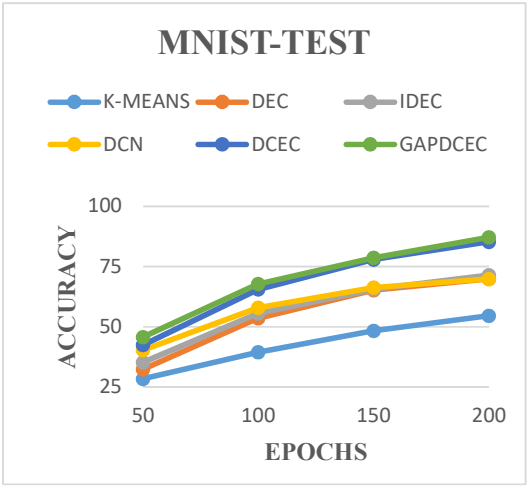

Fig. 5. Epochs VS Accuracy for MNIST-Test

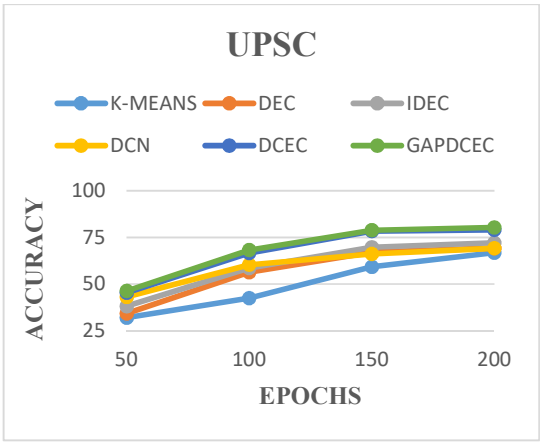

Fig. 6. Epochs VS Accuracy for UPSC

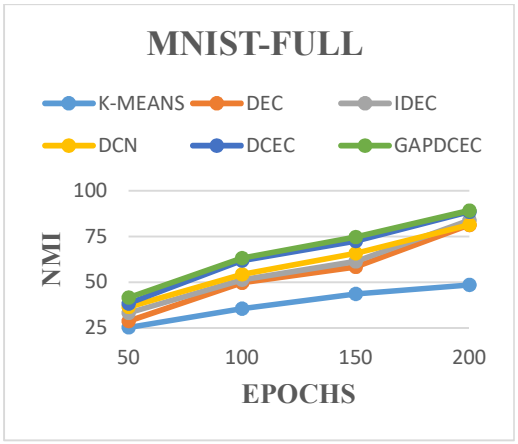

Fig. 7. Epochs VS NMI for MNIST-Full

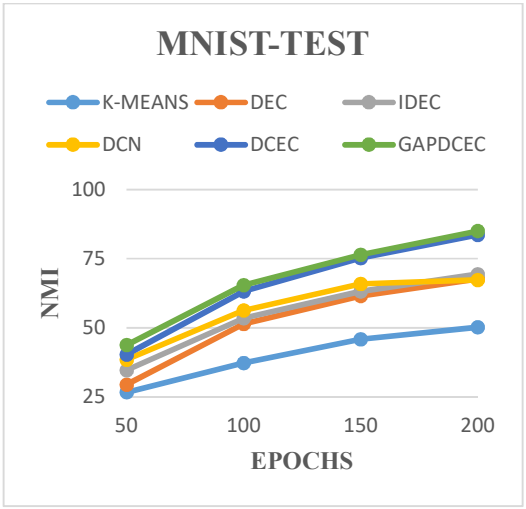

Fig. 8. Epochs VS NMI for MNIST-Test

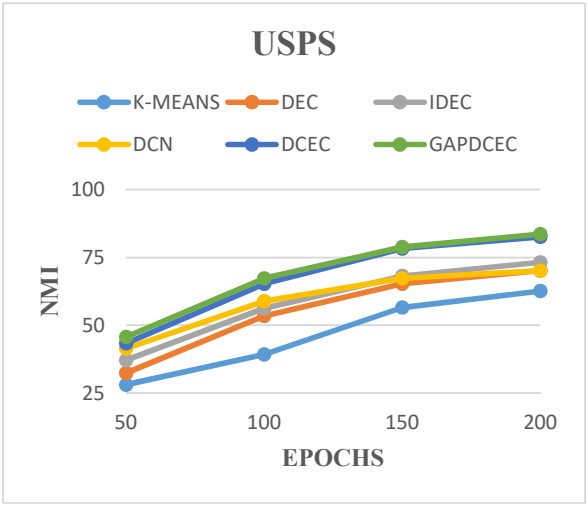

Fig. 9. Epochs VS NMI for UPSC

We evaluate the performance of our algorithm GAPDCEC with competitive algorithms of K-means, DEC, IDEC, DCN and DCEC as shown in the Table 1. Both accuracy and NMI have improved in GAPDCEC compared to other algorithms. Fig. 4, 5,6 show accuracy graphs with respect to different epochs. Fig. 7, 8,9 show the NMI graphs with respect to different epochs.

5. Conclusion

This paper proposes Semantic Image Clustering with GAPDCEC algorithm, which identifies all discriminative regions of image. This framework avoids an overfitting problem and concentrates on user query-based cluster target regions effectively. It learns discriminative features with localization, which can improve the cluster accuracy. The experiments conducted on five different comparative methods on three benchmarked data sets demonstrated the efficacy of GAPDCEC in terms of accuracy, NMI of clusters. In future, experiments will be conducted on complex and real time datasets to explore novel deep learning frameworks.

References

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

[2] Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5147{5156 (2016)

[3] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin:Deep Clustering with Convolutional Autoencoders. ICONIP 2017: Neural Information Processing pp 373-382.

[4] Ryota Hinami, Yusuke Matsui, Shin'ichi Satoh, "Region-Based Image Retrieval Revisited", DOI: 10.1145/3123266.3123312,ACM, 2017.

[5] Morarjee Kolla, T. Venu Gopal, Region based Semantic Image Retrieval using Ontology, pp.421-428, Vol. 5, LNNS Springer, 2017.

[6] Shamsfard, M., & Barforoush, A. (2003). The State of the Art in Ontology Learning. *The Knowledge Engineering Review*, Cambridge Univ. Press, 18(4), 293-316.

[7] Gómez-Pérez, A., & Manzano-Macho, D. (2003). A survey of ontology learning methods and techniques. *Deliverable 1.5, IST Project IST-20005-29243- OntoWeb*.

[8] Omelayenko, B. (2001). Learning of ontologies for the Web: the analysis of existent approaches. Paper presented at Proceedings of the international workshop on Web dynamics, London.

[9] M. Uma Devi and G. Meera Gandhi: Wordnet and Ontology Based Query Expansion for Semantic Information Retrieval in Sports Domain, Journal of Computer Science 2015, 11 (2): 361.371, DOI: 10.3844/jcssp.2015.361.371.

[10] Kolla Morarjee, T.Venu Gopal,Semantic Image Clustering using Region based on Positive and Negative Exaples, pp.261-264, ICICC, Feb 2015,ISBN:978-93-82163-59-6.

[11] Huang, P., Huang, Y., Wang, W., and Wang, L. (2014). Deep embedding network for clustering. In International Conference on Pattern Recognition (ICPR), pages 1532–1537. IEEE.

[12] Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: International Conference on Machine Learning (ICML). pp. 3861{3870 (2017)

[13] Dizaji, K. G., Herandi, A., and Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. arXiv preprint arXiv:1704.06327.

[14] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M- C- Lee, H. Salimbeni, K- Arulkumaran and M. Shanahan, Deep unsupervised clustering with gaussian mixture variational autoencoders, arXiv preprint arXiv:1611.02648, 2016.

[15] Peng, X., Xiao, S., Feng, J., Yau, W.Y., Yi, Z.: Deep subspace clustering with sparsity prior. In: International Joint Conference on Arti_cial Intelligence (IJCAI) (2016)\

[16] M. Lin, Q. Chen, and S. Yan. Network in network. International Conference on Learning Representations, 2014.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.

[18] Ramprasaath R. Selvaraju, Michael Cogswell et. al.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In IEEE International Conference on Computer Vision (ICCV), 2017.

[19] LeCun, Y., Bottou, L., Bengio, Y., Ha_ner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278{2324 (1998).

