Improving Deep Image Clustering With Spatial Transformer Layers

Thiago V.M. Souza  
Centro de Informática  
Universidade Federal de Pernambuco  
Recife, Brazil  
tvms@cin.ufpe.br

Cleber Zanchettin  
Centro de Informática  
Universidade Federal de Pernambuco  
Recife, Brazil  
cz@cin.ufpe.br

Abstract—Image clustering is an important but challenging task in machine learning. As in most image processing areas, the latest improvements came from models based on the deep learning approach. However, classical deep learning methods have problems to deal with spatial image transformations like scale and rotation. In this paper, we propose the use of visual attention techniques to reduce this problem in image clustering methods. We evaluate the combination of a deep image clustering model called Deep Adaptive Clustering (DAC) with the Visual Spatial Transformer Networks (STN). The proposed model is evaluated in the datasets MNIST and FashionMNIST and outperformed the baseline model in experiments.

I. INTRODUCTION

The clustering task consists of dividing a set of data into subgroups where elements belonging to the same group are similar to each other and different from the elements of the other groups. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields [2][3][4]. In some cases, clustering is even important to supervised learning. In many real applications of large-scale image classification, the labeled data is not available or is not enough to train supervised models, since the tedious manual labeling process requires a lot of time and labor. A widely used strategy is to applying clustering to the unlabeled training data to group in similar instances and then use minimal human effort to label annotation based in the group elements [5][6].

Image clustering is an important but challenging task due to the image intra-class variability. For a long time, classic techniques such as K-means were the best option to image clustering [7][8]. In recent years, deep neural networks have proved to be very effective in several image processing areas and deep clustering approaches reached the state-of-the-art in manifold image benchmarks using methods such as Deep Clustering Network (DCN) [9]. Joint Unsupervised Learning (JULE) [10], Deep Embed Cluster (DEC) [11] and Deep Adaptive Clustering (DAC) [1].

The deep neural networks are extremely powerful. However, they have some problems with spatial image transformations like scale and rotation. The majority of Convolutional Neural Networks (CNN) typically employ max-pooling layers using small pooling regions (e.g., 2 x 2 or 3 x 3 pixels). The max-pooling transformation provides a spatial invariance of up to only a small region and the intermediate feature maps in the CNN is not invariant to intra-class differences of the input data.

Some advanced techniques have been proposed to deal with this problem, such as Visual Attention Solutions and Spatial Transformer modules [12]. This modules can be inserted into the network architecture as a network layer and provides the ability to learn invariance to scale, rotation and the more general image deformations.

In this paper, we investigate the use of visual attention techniques in deep clustering models to making the networks more invariant to inter-class differences of the input data. To evaluate this approach, we added Spatial Transformers layers into the Deep Adaptive Clustering (DAC) model. We evaluate our approach performing experiments with the MNIST [13] and FashionMNIST [14] datasets.

The next section reviews the related work, specifically on deep clustering, invariant transformation clustering, and visual attention. In Section 3 we detail the proposed approach. In Section 4 we present the experiments and Section 5 the results. Section 6 presents the final considerations.

II. RELATED WORKS

Deep image clustering is a recent research area, but with exciting published works [15]. The approaches use the most diverse architectures varying the structure of the deep networks, the clustering algorithms and the combination of both parts.

Approaches such as the Deep Clustering Network (DCN) [9] use a pretrained autoencoder combined with the k-means algorithm. Methods such as Joint Unsupervised Learning (JULE) [10] combines deep convolutional networks with hierarchical clustering. Deep Embed Cluster (DEC) [11], also uses a pretrained autoencoder, then removes the decoder part and uses the encoder as a feature extractor to feed the clustering method. After that, the network is fine-tuned using the cluster assignment hardening loss. Meanwhile, the clusters are iteratively tuned by minimizing the KL-divergence between the distribution of soft labels and the auxiliary target distribution. Other interesting models are based on Generative Adversarial Networks (GAN) [16] and Variational Auto-
Encoders (VAE) \cite{17} like \cite{18} and \cite{19} are able of generating new images besides performing clustering.

The use of invariant features to deal with spatial transformations has been used for a long time to improve the results in the task of image clustering \cite{20} \cite{21} \cite{22}. The more classic methods use features such as SIFT\cite{23}, SURF\cite{24}, and ORB\cite{25} that are invariant to uniform scaling, orientation, illumination changes, and partially invariant to affine distortion.

Other Algorithms such as \cite{26} can learn models of different types of objects from unlabeled images including background clutter and spatial transformations. The method uses a statistical approach that can jointly normalize out transformations that occur in training data while learning a density model of the normalized data. The model extends the mixture of Gaussians, the factor analyzer and the mixture of factor analyzers to include image transformation as a latent variable. The model is trained with the expectation maximization (EM) algorithm.

In this paper, we investigate the use of visual attention techniques to improve the feature extraction in the convolutional layers. Visual attention is a new area in computer vision that emerged based on the human ability to look at a scenario and be able to focus attention on a specific area of the image. Attention is applied to decrease the search area for the object of interest in the image, providing to the network the ability to focus on regions of interest of the input image data.

There are many interesting studies in this area, considering supervised problems, one example is the Recurrent Attention Model (RAM) \cite{27}. The model is a recurrent neural network formed by a glimpse network, a core network, an action network and a location network. The image and the location coordinate is sent to the glimpse sensor to generate a new representation like retina. The location coordinate and the retina representation are merged in the glimpse network to generate a new representation, the glimpse vector. In the next part, the core network, or in this case, a Long-short Term memory (LSTM) \cite{28} compute the result, based on the last state and the glimpse vector. In the end, the action network and localization network generate the next location to look and the action/classification label. The method uses reinforcement learning to train the network. RAM has not the ability to dealing with multiple objects. DRAM\cite{29} handle this problem and extends the idea using two stacked LSTMs. One LSTM is to classify and the other one to take handle with the localization problem. In DRAM the model generates labels sequence for the multiple objects. The EDRAM\cite{30} is another improvement that combines DRAM with Spatial Transformer Layers (ST) and turns the network fully-differentiable, this approach has, in general, the same structure as presented in DRAM but uses modified ST module as an attention mechanism to locate the interest image region. This version of ST module employed in EDRAM can use information from previous states of the recurrent network to improve results.

III. PROPOSED APPROACH

A. Deep Adaptive Clustering - DAC

DAC is a single-stage convolutional-network-based method to cluster images. The motivation of the method is the underlying assumption that the relationship between pairwise images is binary and its optimizing objective is the binary pairwise-classification problem.

The label features extracted by a convolutional neural network represents the images and the cosine distance between the label features measures and the pairwise similarities. Furthermore, DAC introduces a constraint to make the learned label features tending to one-hot vectors. Moreover, since the ground-truth similarities are unknown, it adopts an adaptive learning algorithm, an iterative method for tuning the model. In each iteration, pairwise images with the estimated similarities are selected based through an upper and lower threshold; these thresholds are adjusted through the $\lambda$ parameter learned by the network, then the network is trained by the selected labeled samples. DAC stop when all training instances are used, and the objective cannot be improved more.

The DAC optimization function is defined as in the following equation:

$$\min_w E(w) = \sum_{i,j} v_{ij} L (r_{ij}, f(x_i; w) \cdot f(x_j; w))$$ (1)
In this equation \(x_i\) and \(x_j\) are the unlabeled input images, \(w\) are the actual parameters from the network, \(f_u\) is a mapping function that maps input images to label features and the operator represents the dot product between two label features, \(r\) is the unknown binary variable (which refer to the output label), \(r_{ij} = 1\) indicates that \(x_i, x_j\) belong to the same cluster and \(r_{ij} = 0\) otherwise. \(v\) is an indicator coefficient where \(v_{ij} = 1\) indicates that the sample is selected for training, and \(v_{ij} = 0\) otherwise, \(L\) is the function loss defined below:

\[
L (r_{ij}, g (x_i, x_j; w)) = -r_{ij} \log (g (x_i, x_j; w)) - (1 - r_{ij}) \log (1 - g (x_i, x_j; w))
\]  

Finally, we cluster the images according to the most significant response of label features. The DAC reached the state-of-the-art in several public databases. Figure 1 presents all steps of this solution.

B. Spatial Transformer - ST

The spatial transformer is a visual attention mechanism consisting of differentiable modules, which can be trained with the backpropagation algorithm and learn to perform spatial transformations conditioned to the input data map.

![Spatial Transformer Network](image)

Fig. 2. Spatial Transformer Network. Source: [12]

These modules can be inserted as layers of a convolutional network, receiving the input data map \(U \in \mathbb{R}^{H \times W \times C}\) with \(W, H, C\) are the width, height, and channels, respectively. That can be both, the input image or the map of features extracted by the inner layers of a convolutional network.

The \(\theta\) transformation map is predicted by the localization network, which can be any convolutional network with a regression layer at the output. The calculation of \(\theta\) can be followed in the equation below:

\[
\theta = f_{loc}(U)
\]

\(U \in \mathbb{R}^{H \times W \times C}\)  

(3)

The size of \(\theta\) may vary according to the desired transformation type \(\tau_{\theta}\), as we can see in the equation below.

\[
\begin{pmatrix}
  x_i \\
  y_i
\end{pmatrix}
= \tau_{\theta}
\begin{pmatrix}
  x_i^o \\
  y_i^o \\
  1
\end{pmatrix}
\]

(4)

\[
\begin{pmatrix}
  \theta_{11} & \theta_{12} & \theta_{13} \\
  \theta_{21} & \theta_{22} & \theta_{23}
\end{pmatrix}
\]

The transformation parameters are then used to generate a sampling grid produced by the grid generator. The model uses the grid and the features map produced by the sampler to generate the output map.

Through these mechanisms, the convolutional networks can become more robust and invariant to the transformations or variability inherent to the input image data set with a low computational cost.

C. Proposed pipeline

To evaluate our hypothesis, we used as a baseline clustering method the modification of the Deep Adaptive Clustering framework, named DAC*, presented in the original paper [1]. In the DAC* the upper and lower thresholds are set by the parameter \(\lambda\) which in this version is not learned by the network but added linearly at each time. At each iteration all examples are also selected for training. We chose this network because it is the state-of-the-art in several benchmarks and treating the clustering task as a binary classification problem, which makes it suitable for the use of Spatial Transformer Networks.

The convolutional network that composes the architecture present in the original paper is an AllConvNet [31]. However, in several experiments, we had difficulty in training the model using the ST layers. In this case the ST layers performed strange transformations in the images, distorting the object and making the input image noisily after some time of training and impairing the results. This behavior led us to believe that the problem could be due to the vanishing gradient. Another option is to find an appropriate learning rate that attended the training of the convolutional network and spacer layers at the same time.

To minimize this problem, we replace the standard network for a smaller model based on the VGG network [32]. The new model has similar results to the original approach presented in DAC. In Table I are presented the descriptions of the convolutional network. We verify different combinations of ST layers in the input image and also after the convolutional filters. We used 3 spatial layers, and Figure 3 shows the complete network architecture with ST layers. We inserted one ST layer near to each block of convolutional layers, applying the corrections to the original input data, like recommended in [12], and we choose correct transformations in the extracted features each time that they suffer a significant spatial transformation as after downsampled with max-pooling. With this structure we immediately correct the transformation in each resolution of the data, as we can see in the Figure 3. The model doesn’t use an ST layer after the first convolutional layers block, to reduce the computational cost, because the corrections are made in this same resolution before over the input data. The structure of the localization network in the spatial layers is the same proposed in [33] and is detailed in Table I. Usually, the localization networks structures use the Relu activation between the layers and leave the last layer without activation [12]. In an unconventional way, [33] uses tanh activation functions after all the convoluted and dense
The proposed convolutional architecture has three spatial transformer layers. The first one is inserted after the input layer and performs transformations in the initial image. The other spatial layers are applied in the feature maps after the second and third block of convolutional layers.

**TABLE I**

| Model | Spatial-ConvNet |
|-------|-----------------|
| Input | 28x28 monochrome image |
| 3 x 3 conv. | 64 BN ReLU |
| 2 x 2 Maxpooling | BN |
| 3 x 3 conv. | 128 BN ReLU |
| 2 x 2 Maxpooling | BN |
| 3 x 3 conv. | 256 BN ReLU |
| 2 x 2 Maxpooling | BN |
| 3096 dense | BN ReLU |
| 10 dense | BN ReLU SoftMax |

*In some experiments the ST layers were deactivated.*

**TABLE II**

| Localization Network |
|-----------------------|
| Input NxNxM monochrome image |
| 2 x 2 Maxpooling |
| 5 x 5 conv. | 20 Tanh |
| 2 x 2 Maxpooling |
| 5 x 5 conv. | 20 Tanh |
| 50 dense | Tanh |
| 6 dense | Tanh |

*At the input NxNxM corresponds the output dimensions of the anterior layers to ST module.*

Fig. 3. The proposed convolutional architecture has three spatial transformer layers. The first one is inserted after the input layer and performs transformations in the initial image. The other spatial layers are applied in the feature maps after the second and third block of convolutional layers.

Fig. 4. MNIST database, composed of manuscript digits images belonging to 10 different classes.

Fig. 5. Fashion MNIST database, composed of clothing images belonging to 10 different classes.

layers. In the initial experiments, it showed better results with the ST layers than the conventional structures.

**IV. EXPERIMENTS**

**A. Datasets**

The experiments were conducted with two well-used databases to evaluate computer vision problems: the MNIST handwritten digits and the Fashion MNIST based on clothing images.

The MNIST database is very used for evaluate problems of machine learning and computer vision. It is formed by 70,000 monochrom image sof handwritten digits divided in 60,000 images for training and 10,000 for test the model, all images have a size of 28x28 pixels with elements belonging to 10 classes.

The second dataset used is the Fashion MNIST, with 70,000 monochrome images of clothing pieces, composed of 60,000 images for training and 10,000 to test the model. The images
have a size of 28x28 pixels with elements belonging to 10 classes.

B. Evaluation Metrics

The metrics used to evaluate the methods are the Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and clustering Accuracy (ACC). These measures return results between a range [0,1], values close to 1 represent more precise results in clustering.

The ACC is calculated as follows, for the $i_{th}$ image, let us denote $q_i$ as the result from the clustering algorithm and $p_i$ as the ground truth label. The $ACC$ is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta (p_i, \text{map}(q_i))}{n}$$

(5)

$n$ is the total number of images in the dataset, $\delta$ is a function where $\delta(x, y) = 1$ if $x = y$ ($\delta(x, y) = 0$ otherwise), and $\text{map}(q_i)$ is a matching function that makes the optimal match between clustering labels and the ground truth labels. We obtained this matching using the Kuhn-Munkres algorithm [34].

The NMI computes the ratio of information that we can know about the classes $U$ given the clusters $V$ to the averaged information that is contained in $U$ and $V$.

$$\text{NMI} = \frac{\text{MI}(U, V)}{\text{mean}(H(U), H(V))}$$

(6)

The entropy $H$ is the amount of uncertainty for a partition, $P$ is the probability that an object picked at random from $U$ falls into class $U_{i_{th}}$. $P'$ likewise for $V$. $P(i, j)$ is the probability that an object picked at random falls into both classes $U$ and $V$. The mutual information is defined in equation below:

$$\text{MI}(U, V) = \sum_{i=1}^{\|U\|} \sum_{j=1}^{\|V\|} P(i, j) \log \left( \frac{P(i, j)}{P(i)P'(j)} \right)$$

(7)

The normalized mutual information is defined as:

$$\text{NMI} = \frac{\sum_{i=1}^{\|U\|} \sum_{j=1}^{\|V\|} \frac{|U_{i} \cap V_{j}|}{|U_{i}|} \log \left( \frac{|U_{i} \cap V_{j}|}{|U_{i}|} \right)}{\text{mean}(\sum_{i=1}^{\|U\|} P(i) \log(P(i)), -\sum_{i=1}^{\|V\|} P'(i) \log(P'(i)))}$$

(8)

Adjusted Rand index is a function that measures the similarity of the two assignments, ignoring permutations and with normalization. If $C$ is a ground truth class assignment and $K$ the clustering, let us define $a$ as the number of pairs of elements that are in the same set in $C$ and in the same set in $K$, and $b$ the number of pairs of elements that are in different sets in $C$ and different sets in $K$.

The raw (unadjusted) Rand index is then given by:

$$RI = \frac{a + b}{C^2}$$

(9)

However, the $RI$ score does not guarantee that random label assignments will get a value close to zero (especially if the number of clusters is in the same order of magnitude as the number of samples).

To work around this effect we can discount the expected $RI$ of the random labelings by defining the adjusted Rand index as follows:

$$\text{ARI} = \frac{RI - E[RI]}{\max(RI) - E[RI]}$$

(10)

C. Experimental Settings

We compare the proposed approach with the DEC [11] the DAC and its DAC* version. These models actually present the best literature results in the two evaluated databases. In order to verify how the use of the ST layers may contribute to the model final result, we compared our approach in four experiments: 1) with all the ST layers activated; 2) with the last layer off; 3) with the first layer activated; and 4) with no ST layers. The idea is to quantify the contribution of the ST layers in the model’s accuracy.

The original DAC [1] paper does not perform experiments in the Fashion MNIST database. For comparison purposes, we run the DAC* version in this database using the same parameters suggested in the original paper.

In the experiments with the two base images, we use the same data augmentation parameters presented in the original DAC paper. We modified the initial lower and upper selection thresholds respectively for a range between [0.9 0.99] in the MNIST dataset experiment and between [0.8 0.99] for the Fashion MNIST dataset. We used Adam optimizer in our model with a learning rate of 0.0001, as suggested in [33]. We run each experiment 10 times and calculate the mean of the results, which is used for comparison with the other methods.

![Fig. 6. Comparison between some original images and their respective output](image.png)
### Table III

Clustering performance of different methods on image datasets based on Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI)

| Model                        | MNIST          | Fashion MNIST  |
|------------------------------|----------------|---------------|
|                              | ACC | NMI | ARI | ACC | NMI | ARI |
| DEC (U)                      | 0.9430 | 0.7716 | 0.7114 | 0.6184 | 0.6289 | 0.4835 |
| DEC (H)                      | 0.9660 | 0.9246 | 0.9406 | 0.7885 | 0.6899 | 0.4857 |
| Proposed without ST Layer    | 0.9775 | 0.9351 | 0.9486 | -    | -    | -    |
| Proposed using 1 ST Layer    | 0.9568 ± 0.0388 | 0.9326 ± 0.0299 | 0.9230 ± 0.0517 | 0.6120 ± 0.0254 | 0.5910 ± 0.0244 | 0.4009 ± 0.0257 |
| Proposed using 2 ST Layers   | 0.9800 ± 0.0072 | 0.9551 ± 0.0102 | 0.9566 ± 0.0148 | 0.6501 ± 0.0346 | 0.6585 ± 0.0340 | 0.5209 ± 0.0323 |
| Proposed using 3 ST Layers   | 0.9612 ± 0.0338 | 0.9365 ± 0.0222 | 0.9276 ± 0.0402 | 0.6644 ± 0.0442 | 0.6683 ± 0.0251 | 0.5419 ± 0.0440 |

### V. Results

Table III presents the experiment results. The proposed model without the ST layers showed lower results than the ones obtained by the DAC model. This is an expected result because compared with other methods it uses a more simple model to extract features. However, we can observe that our approach using ST layers overcome the results obtained by the other methods in all metrics. Using only one ST layer after the input of the proposed model, we obtain a superior result in almost all the metrics in the tests with the two databases, compared to the best results previously obtained by the DEC and DAC, being below only the obtained NMI by DEC on the Fashion MNIST base, by a small difference of 0.0005. Adding one ST layer before the input and another applied to the features extracted after the second block of convolutional layers, we were able to overcome the DEC and DAC best previous results in both sets of data.

We extract the output of the first transforming layer to observe some evidence that proves its actions to improve the results. We can observe the outputs in Figure 6.

We observe that the network initially applies a powerful zoom in the image and over time reduces this zoom to fit the object in a region where it can better frame all its details in the frame, normalizing the objects, correcting distortions and rotation during the network training to get a better result. Based on the learning from data augmentation images, it rotates objects to a standard angle where it is possible to zoom in and fill a larger area of the full image without losing great details of the object. The output result images from ST layers also present a blur aspect and miss some details, but this loss is compensated by the transformation corrections previously set.

Comparing the results of the experiments using models with a different number of ST layers it is remarkable that even with only one layer the model achieves a considerable improvement, compared to its original version without ST layers. It is also notable that with the addition of more ST layers to the features, the model can obtain better results. However, the idea of using more layers to improve results cannot be applied in every context. Using three ST layers in the experiments with the Fashion MNIST base we achieve the better result on this dataset. However, in the experiments with the MNIST, the use of a third one ST Layer compromised the results, reducing your final result in comparison to the network with two ST layers running in this same dataset.

Figure 7 presents the comparison of the models performance curves with and without spatial layers obtained during the training. It is interesting that even during the first epochs, where the spatial layers have not sufficiently trained, the results are better than the model without ST layers. It is possible to see that even before converging the networks also benefit from the intermediate representations obtained by the ST layers during the training. The curves remained stable in the MNIST dataset following the same pattern of growth without significant declines in performance over time. Analyzing the results of the Fashion MNIST database we see that the networks with one and three ST layers fall out of the performance in some spots between the epochs two and six and after stabilizes and return to increase your performance. Among the layer combinations, the network growth with two ST layers remained more stable in comparison the others during all training epochs, despite being overcome in the final result by the network with 3 ST layers.

Finally, we see that by making convolutional networks more invariant to image samples variation by visual attention techniques allows simpler models to obtain superior results.

### VI. Conclusion

In this work we propose a new approach to Deep Adaptive Clustering solution, replacing the original convolutional features extraction network with a new simpler model combined with Spatial Transformer layers. We evaluated our approach by conducting experiments on two public databases and compared it with other promissor methods to the problem.

We also conducted experiments by varying the amount of ST layers in the convolutional model, to evaluate if, with the addition of new ST layers, using the spatial transformation correction over the internal extracted features, the model results grow proportionally. The experiments showed that our approach was able to outperform the other methods in the two evaluated databases, achieving the state-of-art results in both datasets.
Finally, we show that with the use of visual attention techniques, such as ST Layers, the deep image clustering method can obtain performance improvement. The use of ST layers has shown promise results in improving the performance of the DAC model, as the area of visual attention continues to advance and several new approaches were proposed that extend the capacity of standard ST layers, it is natural that the suggestion for future work is to use these new methods to improve deep image clustering models.

A promising method [33] that defines a layer capable of learning and correcting diffeomorphic transformations and combining them with the affine or homomorphic transformations of the conventional ST layers. The work already surpasses the conventional ST layers in some public databases in the image classification task and could also bring improvement in the results of deep image clustering.

ACKNOWLEDGMENT

This work was supported by CNPq (Brazilian research agencies). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan XP GPU used for this research.

REFERENCES

[1] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, “Deep adaptive image clustering,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct 2017, pp. 5880–5888.

[2] D. Zhang, Y. Rao, J. Zhao, J. Zhao, A. Hu, and B. Cai, “Feature based segmentation and clustering on forest fire video,” in 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 2007, pp. 1788–1792.

[3] D. Saravanan, “Cure clustering technique suitable for video data retrieval,” in 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Dec 2016, pp. 1–4.

[4] Y. Cheng, T. Zhang, and S. Chen, “Fast person-specific image retrieval using a simple and efficient clustering method,” in 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 2009, pp. 1973–1977.

[5] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang, “Correlative multi-label video annotation,” in Proceedings of the 15th ACM International Conference on Multimedia, New York, USA, Sep. 2007, p. 1726.

[6] S. Sclaroff, M. Cascia, S. Sethi, and L. Taycher, “Unifying textual and visual cues for content-based image retrieval on the world wide web,” Comput. Vis. Image Underst., vol. 75, p. 8698, Jul. 1999.

[7] J. Wang, J. Wang, Q. Ke, G. Zeng, and S. Li, “Fast approximate k-means via cluster closures,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 3037–3044.

[8] J. Wang, J. Wang, J. Song, X. Xu, H. T. Shen, and S. Li, “Optimized cartesian k-means,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 1, pp. 180–192, Jan 2015.

[9] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” 2016. [Online]. Available: http://arxiv.org/abs/1610.04794

[10] J. Yang, D. Parikh, and D. Batra, “Joint unsupervised learning of deep representations and image clusters,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 5147–5156.

[11] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in International Conference on Machine Learning, vol. 48.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, “Spatial transformer networks,” in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 2017–2025.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[14] H. Xiao, K. Rasul, and R. Vollgraf, (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[15] 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, vol. 6, pp. 39501–39514, 2018.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu,
D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680.

[17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013.

[18] J. T. Springenberg, “Unsupervised and semi-supervised learning with categorical generative adversarial networks,” 2015. [Online]. Available: https://arxiv.org/pdf/1511.06390.pdf

[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, ser. IJCAI’17, 2017, pp. 1965–1972.

[20] W. Zhang, X. Wu, W. Zhu, and L. Yu, “Unsupervised image clustering with sift-based soft-matching affinity propagation,” IEEE Signal Processing Letters, vol. 24, no. 4, pp. 461–464, April 2017.

[21] Y. Lee, H. Ahn, and S. Rhee, “ Efficient image retrieval using advanced clustering surf,” in 2012 15th International Conference on Network-Based Information Systems, Sep. 2012, pp. 749–753.

[22] Y. Zhang and Z. Miao, “Object recognition based on orb and self-adaptive kernel clustering algorithm,” in 2014 12th International Conference on Signal Processing (ICSP), Oct 2014, pp. 1397–1402.

[23] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, Jun. 2008.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in Proceedings of the 2011 International Conference on Computer Vision, ser. ICCV ’11, 2011, pp. 2564–2571.

[26] B. J. Frey and N. Jojic, “Transformation-invariant clustering using the em algorithm,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 1, pp. 1–17, Jan 2003.

[27] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, ser. NIPS’14, 2014, pp. 2204–2212.

[28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[29] J. Ba, V. Mnih, and K. Kavukcuoglu, “Multiple object recognition with visual attention,” 2014. [Online]. Available: http://arxiv.org/abs/1412.7755

[30] A. Ablavatski, S. Lu, and J. Cai, “Enriched deep recurrent visual attention model for multiple object recognition,” 2017. [Online]. Available: http://arxiv.org/abs/1706.03581

[31] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” 2014. [Online]. Available: https://arxiv.org/pdf/1412.6806.pdf

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014. [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf

[33] N. S. Detlefsen, O. Freifeld, and S. Hauberg, “Deep diffeomorphic transformer networks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 4403–4412.

[34] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Research Logistics (NRL), vol. 52, no. 1, pp. 7–21, 2005.

[35] Z. Research, “fashion-mnist,” https://github.com/zalandoresearch/fashion-mnist