UNSUPERVISED CONVOLUTIONAL NEURAL NETWORKS FOR LARGE-SCALE IMAGE CLUSTERING

Chih-Chung Hsu[1] *and Chia-Wen Lin*[2]

[1] Institute of Communication Engineering, National Tsing Hua University
E-mail: m121754@gmail.com
[2] Dept. of Electrical Engineering, National Tsing Hua University
E-mail: cwlin@ee.nthu.edu.tw

ABSTRACT

The paper proposes an unsupervised convolutional neural network (UCNN) to solve clustering and representation learning jointly in an iterative manner. The key idea behind the proposed method is that learning better feature representations of images leads to more accurate image clustering results, whereas better image clustering can benefit the feature learning with the proposed UCNN. In the proposed method, given an input image set, we first randomly pick k samples and extract their features as the initial centroids of image clusters using the proposed UCNN with an initial representation model pre-trained from the ImageNet dataset. Mini-batch k-means is then performed to assign cluster labels to individual input samples for a mini-batch of images randomly sampled from the input image set until all images are processed. Subsequently, UCNN simultaneously updates the parameters of UCNN and the centroids of image clusters iteratively based on stochastic gradient descent. Experimental results demonstrate the proposed method outperforms start-of-the-art clustering schemes in terms of accuracy and memory complexity on large-scale image sets containing millions of images.

Index Terms— Unsupervised learning, image clustering, deep learning, convolutional neural network

1. INTRODUCTION

Image clustering is a fundamental problem for many image processing and computer vision applications. Nowadays, a huge number of images have been uploaded to clouds for sharing or storage. How to efficiently organize such large scale image data is an emerging and challenging issue. In general, clustering methods can be roughly categorized into hierarchical clustering and centroid-based clustering. The most popular algorithms for hierarchical clustering are agglomerative clustering, which is computationally very expensive for large image data [3][4]. In contrast, centroid-based clustering (e.g., k-means and spectral clustering) [5]–[10] randomly picks k samples from input data as initial cluster centroids. Then, each unlabeled sample finds its closest cluster centroid and is assigned with the corresponding cluster label. As a result, the centroids of clusters are then updated according to the clustering result. The clustering and centroid updating are iterated until the clustering result converges [6][7]. Such centroid-based clustering is more suitable for large-scale data clustering than hierarchical clustering due to less memory usage and computational cost. The effectiveness of centroid-based clustering, nevertheless, highly relies on feature representational power.

Although deep learning has been successfully adopted for various multimedia and vision applications in supervised manners, unsupervised deep learning still remains a challenging problem. The first deep learning-based image clustering work adopts AutoEncoder to learn visual representation followed by concatenating conventional k-means to obtain the final clusters [11]. However, it has been shown in [12] that, compared to CNN-based architectures, AutoEncoder usually cannot learn representative features well from high-dimensional data such as images. The CNN with Connection Matrix (CNN-CM) method in [13] proposed a connection matrix that allows feeding in additional side information to assist learning discriminative representations for clustering. A full-set k-means is then performed to group all images into their corresponding clusters based on the learned features. The complexity of the full-set k-means will grow drastically when the size of image set becomes large, making large-scale clustering impractical. The CNN with Re-running Clustering (CNN-RC) method in [14] proposed to learn feature representations and cluster images jointly: hierarchical image clustering is performed in the forward pass, while representations are learned in the backward pass. In the hierarchical clustering, image samples are first regarded as initial centroids, and then reliable label information is extracted from an undirected affinity matrix established from the input image set. The network parameters are iteratively updated towards obtaining better feature representations by minimizing a predefined loss metric. Nevertheless, constructing an affinity matrix consumes high computation and memory complexity when the training set

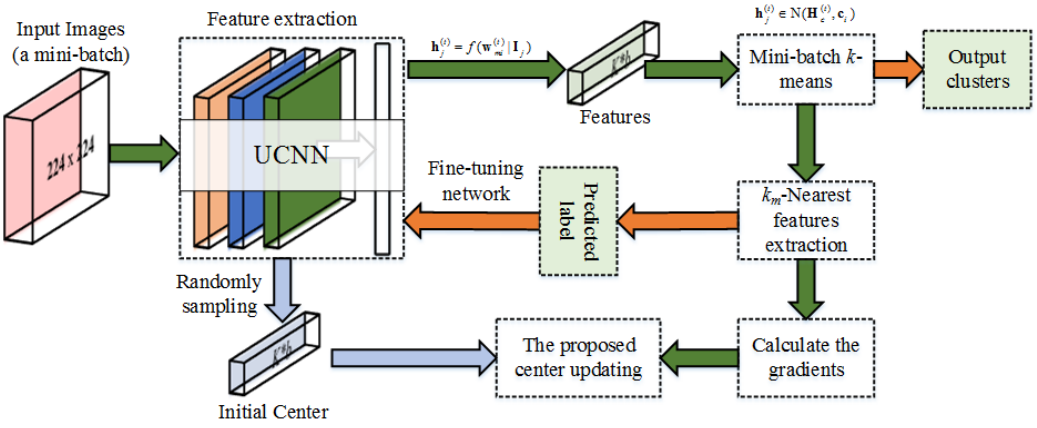

Fig. 1. Block diagram of the proposed UCNN for joint image clustering and representation learning.

becomes large. The memory cost can hardly be reduced since it is not a sparse matrix.

Although CNNs have been shown to achieve good performances in supervised learning-based image/video applications such as visual object localization, tracking, categorization, the existing CNNs cannot well address large-scale unsupervised image clustering. To address the problems with large-scale image clustering: (1) representative feature learning from unlabeled input images and 2) high computation and memory complexity, we propose a convolutional network to achieve joint clustering and representation learning. To reduce computation and memory cost, we nicely incorporate mini-batch *k*-means into the CNN-based clustering framework. The main contribution of this paper is three-fold: i) we are among the first to propose a framework that integrates mini-batch *k*-means with state-of-the-art CNNs to efficiently address the large-scale image clustering problem; ii) we propose a novel iterative centroid updating method that can avoid the feature mismatch problem caused by the gradient drift with mini-batch *k*-means; and iii) the proposed framework can be easily integrated into existing CNN-based networks.

2. JOINT CLUSTERING AND REPRESENTATION LEARNING BASED ON MINI-BATCH K-MEANS

In this paper, we propose an unsupervised CNN (UCNN) modified from the network structure in [15] to capture the compact image features. The proposed UCNN is composed of five convolutional layers *Conv1–Conv5* adopted from the first five convolutional layers of AlexNet [16], followed by three adaptation layers (*Conv6–8*) with channel numbers 6144, 2048, and *k*, respectively. Finally, we concatenate a fully connected layer (*FC9*) and *Softmax* layers to extract the image features. The adaptation layers consist of three convolutional layers, *Conv8*—*Conv8*, all with 3×3 kernels followed by a global max-pooling that finds the maximum value for each channel of *Conv8* so that the size of the output of global max-pooling is $1 \times k$.

The proposed scheme for iterative image clustering and representation learning is illustrated in Fig. 1. We first initialize the parameters of the UCNN with a pre-trained model for speeding up the convergence of iterations. We then randomly pick k images $\mathbf{I}_c$ from the input image set $\mathbb{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{N_x}\}$ containing N_x images and extract their features $\mathbf{H}_c$ using the UCNN as the k initial cluster centroids $\mathbf{C}$. After the initialization, we sample the input image set into mini-batches, and for the b-th mini-batch, perform mini-batch *k*-means [17] to assign cluster labels to features $\mathbf{h}_i^{(b)} = f\left(\mathbf{W}_{\text{FC9}}|\mathbf{I}_i^{(b)}\right) \in \mathbf{H}(\mathbb{I}^{(b)})$ extracted from individual frames of the mini-batch. Based on the assigned labels to the feature set of the *b*-th mini-batch, we can update the parameters of the UCNN using SGD. Then, features $\mathbf{h}_i^{(b)}$ in the *b*-th mini-batch are used to update their corresponding centroids using SGD. Since $\mathbf{W}_{\text{FC9}}$ will be updated after each iteration, the extracted feature $\mathbf{h}_i^{(b)}$ will also be updated as well, resulting in a possible mismatch between the features extracted in successive iterations. In this case, the centroid updating based on SGD may become unstable and unpredictable since the feature mismatch will lead to gradient drift error in SGD. To overcome this problem, we derive the gradient drift error between two successive iterations and compensate for the drift error by tracking backward the features in two successive iterations to ensure their consistency. Finally, the proposed method updates the extracted feature $\mathbf{h}$, centroids $\mathbf{C}$, and parameters $\mathbf{W}_{\text{FC9}}$ iteratively without such drifting error problem. We then show how to iteratively representation learning and centroid updating without gradient drift problem.

2.1. Representation Learning

In this work, we extract local salient features from the output of layer *Conv8* [15], and then feed the features into *FC9* to generate the features for clustering. To learn the parameters of *FC9* and *Softmax* of the proposed UCNN, we adopt a standard SGD process [18], where parameter sets $\mathbf{W}_{\text{FC9}} = \{w_{mi}\}$ and $\mathbf{W}_{\text{SMax}} = \{w_{ij}\}$ represent the parameter sets of *FC9* and *Softmax*, respectively. At each iteration, we use SGD to update the weights as follows:

$$w_{mi}^{(t+1)} = w_{mi}^{(t)} - \eta \sum_{j=1}^{k} \left[\left(y_j - t_j \right) \cdot \max\left(y_j, 0 \right) \cdot w_{ij} \right] \cdot \max(h_i, 0) \cdot x_m = w_{mi}^{(t)} - \eta \Delta w_{mi}^{(t)}. \quad (1)$$

As a result, the full gradient for updating the weights of *FC9* can be calculated by (1).

2.2. Cluster Centroid Updating

Assume that the size of a mini-batch is N_m, we randomly sample N_m images from the input image set $\mathbb{I}$ to form a mini-batch. Initially, we randomly pick k features $\mathbf{H}_c^{(0)} = \{\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, \ldots, \mathbf{h}_k^{(0)}\}$ from $\mathbb{I}$ as initial centroids $\mathbf{C}$, where $\mathbf{h}_j^{(0)}$ denotes the feature of the j-th image at first iteration (i.e., $t = 0$). Mini-batch k-means is then performed to assign individual samples of each mini-batch to their corresponding clusters. Based on the mini-batch clustering result, the centroids of those clusters that are assigned to the mini-batch's samples are updated based on SGD [18]. At iteration t, the i-th centroid $\mathbf{c}_i^{(t)}$ that is assigned to a new sample is updated by the weighted average of the feature of the $(t-1)$-th centroid and the feature of the newly assigned sample $\boldsymbol{h}_{\text{new}}^{(t)}$ as follows:

$$\boldsymbol{c}_i^{(t)} = (1 - \gamma_i)\boldsymbol{c}_i^{(t-1)} + \gamma_i \boldsymbol{h}_{\text{new}}^{(t)}, \quad (2)$$

where $\boldsymbol{h}_{\text{new}}^{(t)} \in NN\left(\boldsymbol{H}_c^{(t)}, \boldsymbol{c}_i\right)$ represents the extracted feature of the sample in mini-batch $\mathbf{H}_c$ that is newly assigned to its nearest neighbor centroid $\boldsymbol{c}_i$. We use per-centroid learning rates γ_i for the i-th centroid [18] as determined by

$$\boldsymbol{\gamma}_i = 1/\text{count}(\boldsymbol{c}_i), \quad (3)$$

where count($\mathbf{c}_i$) represents the number of the samples assigned to $\mathbf{c}_i$.

2.3. Compensation of Centroid Drifting

Note, at the t-th iteration, the features of the j-th image $\mathbf{h}_j^{(t)} = f(\mathbf{w}_{\text{FC9}}^{(t)}|\mathbf{I}_j)$ is extracted based on the filter coefficients $w_{mi}^{(t)}$ of *FC9*. However, $w_{mi}^{(t)}$ is updated along time during representation learning, thereby making $\mathbf{h}_j^{(t)}$ vary along time as well. The time-varying nature of $\mathbf{h}_j^{(t)}$ leads to the inconsistency between the features of the same image extracted at two successive iterations. For example, $\mathbf{h}_j^{(t)} = f(\mathbf{W}_{\text{FC9}}^{(t)}|\mathbf{I}_j)$ extracted at iteration t is different from $\mathbf{h}_{\text{j}}^{(t-1)} = f(\mathbf{W}_{\text{FC9}}^{(t-1)}|\mathbf{I}_{\text{j}})$ at iteration $t-1$, as $\mathbf{W}_{\text{FC9}}^{(t)}$ and $\mathbf{W}_{\text{FC9}}^{(t-1)}$ are different due to parameter updating. This makes centroid updating in (2) unreliable since $\boldsymbol{h}_j^{(t)}$ is time varying, which will degrade the performance of image clustering. To overcome this problem, we propose an approach to ensure feature consistency between successive iterations. At iteration t, we have $\mathbf{c}_i^{(t)} = (1 - \gamma_i)\mathbf{c}_i^{(t-1)} + \gamma_i \mathbf{h}_j^{(t)}$ and $w_{mi}^{(t)} = w_{mi}^{(t-1)} - \eta \Delta w_{mi}^{(t-1)}, \forall m, i$. At iteration t, the feature extracted at the $(t-1)$-th iteration can be backward tracked by

$$\mathbf{h}_j^{(t-1)} = f\left(\left(\mathbf{w}_{\text{FC9}}^{(t)} + \eta \Delta \mathbf{w}_{\text{FC9}}^{(t-1)} \right) | \mathbf{I}_j \right). \quad (4)$$

To maintain the consistency between the features used, we replace the features in (2) with the backward tracked features in (4), and rewrite the centroid updating formula as follows:

$$\mathbf{c}_i^{(t)} = (1 - \gamma_i)\mathbf{c}_i^{(t-1)} + \gamma_i f\left(\left(\mathbf{w}_{\text{FC9}}^{(t)} + \eta \Delta \mathbf{w}_{\text{FC9}}^{(t-1)} \right) | \boldsymbol{I}_j \right) \quad (5)$$

As a result, the cluster centroids can be properly updated. After iterating for several epochs with the proposed framework, the cluster labels of images will converge to their final values more reliably. Furthermore, the proposed mini-batch-based scheme achieves large-scale image clustering on a single personal computer with reasonable computational and memory complexity as will be shown in the experiment section.

3. EXPERIMENTAL RESULTS

3.1. Experiment Setup

1) *Comparison Schemes*: To evaluate the performance of the proposed method, we test our method against three state-of-the-art deep learning-based image clustering schemes including the AutoEncoder-based Deep Embedding Clustering (DEC) scheme proposed in [11] and the CNN with Connection Matrix (CNN-CM) method proposed in [13] and the CNN with Re-running Clustering (CNN-RC) [14]. Note, as explained above, these three deep-learning-based schemes cannot deal with large-scale image sets consisting of millions of images with commercial GPU support like Titan X. Therefore, besides the three methods, we also implemented three baseline schemes for performance evaluation: 1) Baseline-I: the proposed method without feature mismatch compensation, that is, use (2) instead of (5) to update cluster centroids; 2) Baseline-II: the pre-trained model without fine-tuning followed by mini-batch k-means clustering; 3) Baseline-III: the pre-trained model without fine-tuning following by full-set k-means clustering.

2) *Datasets for Pre-training and Testing*: We selected two large-scale image datasets, ILSVRC12 in ImageNet [16] and Places2 [19], for clustering performance evaluation. ILSVRC12 consists of 1.2 million training images and 50,000 validation images collected from 1,000 object categories, and Places2 consists of 1.6 million training images and 18,250 images validation images collected from 356 scene categories.

Since, for fast convergence, the parameters of UCNN was pre-trained based on the ILSVRC12 training set, we did not evaluate the performances of the clustering methods on the training set of ILSVRC12 for fairness. Instead, we conducted the performance evaluation on the Places2 training (denoted "Places-Train") and validation (denoted "Places-Val") sets, and also on the ILSVRC12 validation set (denoted "ILSVRC-Val"). For the Places2 training and validation sets, the channel number to *Conv8* and the number of neurons of *Softmax* in the proposed UCNN were both set to 365, whereas

Table I. NMI Performance Comparison of the Proposed Scheme and State-of-The-Art Schemes for Three Image Sets.

Evaluated methods	ILSVRC-Val	Places-Val	Places-Train
DEC [11]	0.155	0.113	N.A.
CNN-CM [13]	0.137	0.198	N.A.
CNN-RC [14]	0.295	0.213	N.A.
Baseline-I	0.181	0.153	0.047
Baseline-II	0.231	0.177	0.045
Baseline-III	0.293	0.201	N.A.
Proposed	**0.375**	**0.307**	**0.187**

Table II. Comparison of Run-time and Memory Costs of the Proposed Scheme and State-of-The-Art Schemes for Three Image Sets.

Evaluated methods	ILSVRC-Val	Places-Val	Places-Train
DEC [11][1]	0.9 hr/16 GB	0.75 hr/14 GB	N.A.
CNN-CM [13]	3 hr/ 7 GB	1.8 hr/5 GB	N.A.
CNN-RC [14]	5.1 hr/10 GB	4.6 hr/7 GB	N.A.
Baseline-I	1.1 hr/8 GB	0.5 hr/8 GB	40 hr/8 GB
Baseline-II	0.9 hr/22 GB	0.45 hr/19 GB	**36 hr/8 GB**
Baseline-III	4.28 hr/22 GB	3.68 hr/19 GB	N.A.
Proposed	**1.2 hr/8 GB**	**0.5 hr/8 GB**	43 hr/8 GB

for the ILSVRC12 validation set the number of channels to *Conv8* and number of neurons of *Softmax* were both set to 1000. Similar to [13], all test images were cropped to 256x256 center-surrounding sub-images.

3) *Computation Platform*: We implemented the proposed method on top of TensorFlow [20] on an Intel Core i7-4770 PC with 32 GB RAM which is equipped with an NVIDIA Titan X GPU with 12GB GPU RAM.

3.2. Performance Evaluation

To evaluate the objective clustering performances of the proposed method and the compared methods, we adopt the Normalized Mutual Information (NMI) [21]. The higher the NMI is, the more reliable the clustering result becomes. Table I compares the NMI performances of the proposed method, DEC [11], CNN-CM [13], CNN-RC [14] and three CNN-based baseline methods for three image sets. The result shows the proposed method achieves significantly higher NMI on all image sets compared to the other schemes. Compare with Baseline-I, we can observe that the feature mismatch in mini-batch-based centroid updating leads to significant drifting error which causes NMI performance degradation by 0.14–0.19. Compared with the direct combination of a pre-trained model with mini-batch k-means (Baseline-II) and full-set *k*-means (Baseline-III), the proposed joint optimization of clustering and parameter learning leads to performance improvement in NMI by 0.13–0.14 and 0.08–0.10, respectively. Besides, DEC, CNN-CM, and CNN-RC, and Baseline-III all cannot handle large-scale image clustering due to their high complexity as will be explained later.

Table II compares the memory and run-time costs for three image sets, where we set the number of epochs for parameter updating to 10. The run-time is proportional to the size of image set and the number of clustering iterations. The comparison shows that, for mini-batch size $N_m = 50$, our method consumed about 8GB GPU memory and 43 hours to obtain the clustering result of the Places-Train image set on Titan X, whereas the three existing deep-learning-based schemes [11][13][14] failed in this clustering task. Since DEC [11] is based on AutoEncoder to learn the feature representations from a training set, it has been shown that the representation learning performance of an AutoEncoder-based network is generally unsatisfactory for high-dimensional data (e.g., images) in terms of computation cost and clustering accuracy [8]. Note, CNN-CM [13] and Baseline-III both perform full-set *k*-means clustering which requires to extract the features of all images and compare the distances between features, leading to huge computation/memory costs and thereby making the large-scale clustering tasks impractical on a single general-purpose PC equipped with a GPU graphic card. Similarly, CNN-RC [14] relies on estimating an $N_x \times N_x$ affinity matrix, making the clustering process unsolvable when the size of dataset N_x is large. Instead of using computation/memory demanding operations like full-set *k*-means and affinity matrix construction, the proposed mini-batch-based approach with feature drift compensation can efficiently and reliably address the problem of large-scale joint representation learning and clustering.

4. CONCLUSION

In this paper, we proposed an unsupervised convolution neural network (UCNN) architecture, which can extract salient features in an unsupervised manner to benefit image clustering. On top of UCNN, we also proposed a mini-batch-based iterative representation learning and clustering centroid updating approach to efficiently address the problem of large-scale image clustering for up to millions of images at reasonable memory and computation costs. While the mini-batch iterative updating strategy offers good scalability to the proposed UCNN, we have also proposed a feature drift compensation scheme to avoid the performance degradation due to feature drifting in the iterative process. Our experimental results demonstrate the superior performance and scalability of the proposed scheme on several public image datasets.

5. REFERENCES

[1] T. Liu, C. Rosenberg and H. A. Rowley, "Clustering billions of images with large scale nearest neighbor search", in *Pro. IEEE Workshop Appl. Comput. Vis.*, pp. 28–33, 2007.

[2] Y. Gong, M. Pawlowski, F. Yang, L. Brandy, L. Boundev, and R. Fergus, "Web scale photo hash clustering on a single machine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, USA, June 2015.

[3] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Tran. Pattern Anal. Mach. Intell.*, vol. 23, pp. 1053–1074, 2001.

[4] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighborhood," *Pattern Recognit.* vol. 10, pp. 105–112, 1978.

[5] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symp. Mathematical Statistics Probability*, vol. 1, pp. 281–297, Oakland, CA, USA., 1967.

[6] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, 2004.

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Conf. Neural Inf. Process. Syst.,* vol. 2, pp. 849–856, Vancouver, British Columbia, Canada, Dec. 2001.

[8] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. Conf. Neural Inf. Process. Syst.*, pp. 494–502, Lake Tahoe, Nevada, Dec. 2013.

[9] D. Han and J. Kim, "Unsupervised simultaneous orthogonal basis clustering feature selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, USA, June 2015, pp. 5016–5023.

[10] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Proc. European Conf. Comput. Vis.,* pp. 459–472, Florence, Italy, Oct. 2012.

[11] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Learning Rep*., San Juan, Puerto Rico, May 2016.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet classification with deep convolutional neural networks," in *Proc. Conf. Neural Information Process, Syst.*, Lake Tahoe, Nevada, Dec. 2012.

[13] A. Dundar, J. Jin, and E. Culurciello, "Convolutional clustering for unsupervised learning," in *Proc. Int. Conf. Learning Rep.*, San Juan, Puerto Rico, May 2016.

[14] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, USA, June 2016.

[15] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? – Weakly-supervised learning with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, USA, June 2015.

[16] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. *ImageNet Large Scale Visual Recognition Competition* 2012. [Online] available: http://www.image-net.org/challenges/LSVRC/2012/.

[17] D. Sculley, "Web-scale k-means clustering," in Proc. *ACM Int. Conf. World Wide Web*, pp. 1177–1178, April 2010, New York, USA.

[18] Y. Avrithis, Y. Kalantidis, E. Anagnostopoulos, and I. Z. Emiris, "Web-scale image clustering revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," arXiv, 2015.

[20] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," [Online] available: http://tensorflow.org.

[21] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," *In Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 267–273. ACM, 2003.