# Understanding Center Loss Based Network for Image Retrieval with Few Training Data

Pallabi Ghosh(✉) and Larry S. Davis

University of Maryland, College Park, USA
pallabig@cs.umd.edu, lsd@umiacs.umd.edu

**Abstract.** Performance of convolutional neural network based image retrieval depends on the characteristics and statistics of the data being used for training. We show that for training datasets with a large number of classes but small number of images per class, the combination of cross-entropy loss and center loss works better than either of the losses alone. While cross-entropy loss tries to minimize misclassification of data, center loss minimizes the embedding space distance of each point in a class to its center, bringing together data-points belonging to the same class.

**Keywords:** Center loss · Image retrieval · Small training dataset

## 1 Introduction

A common approach to identifying features in CBIR is to train a multi-class deep model with a large fully supervised training set, and then use features from various layers of the network as a basis for coding database images (which need not be drawn from the classes used to train the network). Early attempts at retrieval were based on cross-entropy loss. Triplet loss has been used to train networks for image retrieval [4]. However optimizing triplet loss is challenging because the level of relative similarity or dissimilarity in each training triplet determines how fast the network learns.

In this paper we study the use of center loss [14,16] for image retrieval. Center loss reduces the distance of each data point to its class center. It is not as difficult to train as triplet loss and performance is not based on the selection process of the training data points (triplets). Combining it with a softmax loss, prevents embeddings from collapsing.

Experiments will show that for training datasets with few images per class but with a large number of classes, the improvement using center loss for retrieval is significant.

## 2 Related Works

Some of the classical papers in image retrieval include [3,7,8,13]. Most of the recent work is based on training CNN models [2,5,12]. Both [11] and [17] review these techniques.


L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11132, pp. 717–722, 2019.
https://doi.org/10.1007/978-3-030-11018-5_63

**Algorithm 1.** Resnet18 pre-trained on Imagenet is the base network. L is the output size of the pre-final (512 for Resnet18). The training set contain K classes and B is the batch size. The center for each class is calculated by averaging the pre-final layer output of the network after passing each image in the class through the network.

**Input** Dataset with K classes and $N_k$ images in $k_{th}$ class. Center matrix $C_{K\times L}$ containing the centers of each class.

**Output** The recomputed center matrix $C'$

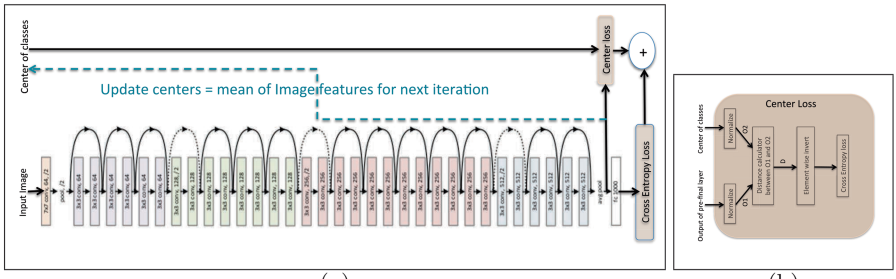
**procedure** CENTER LOSS + CROSS-ENTROPY LOSS BASED NETWORK TRAINING
  **while** not converge **do**
    Mini-batch of images is passed through the network
    $f^1 \leftarrow$ pre-final layer output
    $f^2 \leftarrow$ final FC layer output
    $L_s \leftarrow$ cross-entropy function applied to $f^2$
    $\bar{C} \leftarrow$ normalized $C_{K\times L}$
    $\bar{f}^1 \leftarrow$ normalized $f^1_{B\times L}$
    $D_{B\times L} \leftarrow \text{trace}(\bar{f}^1 \bar{f}^{1^T}) * \left[1 \cdots 1\right]_{1\times K} + \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}_{B\times 1} * \text{trace}(\bar{C}\bar{C}^T)^T - 2\bar{f}^1\bar{C}^T$
    $\frac{1}{D} \leftarrow$ element wise inverse of D after adding 1e-4 to prevent division by 0
    $L_c \leftarrow$ cross-entropy function applied to $\frac{1}{D}$
    Backpropagate loss $L = L_s + L_c$ and update weights of network
    Recompute centers $C'$ using updated network. All images in batches are passed through the network to get a new $f^1$.
    $C'_k \leftarrow \frac{1}{N_k}\sum_{y_i=k} f^1_{y_i}(x_i)$ where x are images, y are corresponding labels and k is a particular class in K.
    For the next epoch $C = C'$.

[2] achieved huge performance improvements by training the network on datasets related to the query. [9] showed that using intermediate layers captures local patterns of objects which performs better than using the final layer output for image retrieval. Similarly [15] uses the regional maximum activations of convolutions, R-MAC, for the same purpose. R-MAC uses a CNN to obtain a local descriptor of the image, which is then max pooled from different regions in a rigid grid, normalized, whitened and sum-aggregated to give a compact output vector. [4] also uses a similar process but with region proposals instead of the rigid grid to define the aggregation regions.

Center Loss was first used for face recognition by [16]. They update centers per mini-batch based on the gradient of center loss, and combines center loss with softmax loss for stability. [14] used a similar idea for few shot learning where they apply softmax over center distances. Instead of updating centers, they recalculate the centers per mini-batch based on the image classes in the support set in the mini-batch using episodic learning.

## 3 Our Algorithm

Our technique combines center loss with cross-entropy loss on a Resnet18 [6] based network as shown in Fig. 1. Suppose there are K classes and that the $k^{th}$ class has $N_k$ images. Let $f^1_{y_i}(x_i)$ be the pre-final layer output by passing the $i_{th}$ image $(x_i)$ with label $y_i$ through the network. Similarly let $f^2_{y_i}(x_i)$ be the final FC layer output and let B be the number of images per batch.




(a) (b)

**Fig. 1.** (a) Our Algorithm (Resnet18 image from [1]) (b) The center loss computation block

First the training images are passed through a network pre-trained on Imagenet, giving us $f^1_{y_i}(x_i)$ feature descriptor. Then the center $c_k$ of the $k^{th}$ class is computed as follows:

$$c_k = \frac{1}{N_k} \sum_{y_i=k} f^1_{y_i}(x_i) \tag{1}$$

Also the distance $d_{ik}$ of the feature descriptor for each image to each class center $c_k$ is calculated as follows:

$$d_{ik} = ||f^1_{y_i}(x_i) - c_k||^2_2 \tag{2}$$

Let this matrix be D with each element $d_{ik}$. Each $d_{ik}$ is inverted to get $\frac{1}{D}$ so that it can be equated to a normal cross-entropy loss model where the input to the loss layer is a scores array. Let each row in $\frac{1}{D}$ be represented as $\frac{1}{d_i}$ and the labels corresponding to each row be $y_i$. Finally $\frac{1}{D}$ values are passed into a cross-entropy loss function which yields the center loss, $L_c$. This is combined with a normal cross-entropy loss applied on the final Fully-Connected layer with number of classes as output size, $L_s$. The total loss $L$ can be expressed as:

$$L = L_s + L_c = -\sum_{i=1}^{B} log \frac{e^{W^T_{y_i} f^2_{y_i}(x_i) + b_{y_i}}}{\sum_{j=1}^{K} e^{W^T_j f^2_{y_i}(x_i) + b_j}} - \sum_{i=1}^{B} log \frac{e^{W^T_{y_i} \frac{1}{d_i} + b_{y_i}}}{\sum_{j=1}^{K} e^{W^T_j \frac{1}{d_i} + b_j}} \tag{3}$$

This is similar to the loss in [16] except that we replace the squared Euclidean center loss with cross entropy function being applied on this distance as in [14]. The difference with [14] is that we use inverse instead of negative distance function. The use of cross-entropy function on the squared Euclidean distance helps to remove the instability of the center loss. At the end of each epoch we use Eq. 1 to recompute the centers globally for the entire dataset. [16] uses an update formula to update the centers whereas [14] recomputes them, but both of them recalculate only at the mini-batch level, and not globally.

## 4 Dataset

Google Landmark [10] has 14951 classes with about 1 million images in the original train dataset. We split this into training set consisting of the first 8951 classes and the query set containing the remaining 6000 classes, so training and query partitions do not have any classes in common. Since each query class should have at least 2 images - one as the query and the other to be included in the retrieval/index set - the classes containing only one image are not used. We take a maximum of 10 images per class. So finally there are 8951 training classes with 72244 images, 5943 query classes with 1 query image per class and an index set consisting of 42709 images from these 5943 classes.

## 5 Results

We use Resnet models in Pytorch pre-trained on Imagenet as initialization. The final layer size is modified to suit the number of classes in our training set and it is initialized using Xavier uniform initialization. The output size of the pre-final layer is model dependent (512 for Resnet18), which would be the size of the feature descriptor for the image. For all networks, we used Adam optimization for training with a weight decay of 2e–4. The initial learning rate was set at 0.001 and a stepwise scheduler with drop rate of 0.92 per epoch was used. We ran the experiments with a batch size of 224.

Mean average precision or mAP score was used as the evaluation criterion. For Google Landmark dataset, given a query image all other images from the same class are correct retrieval results and images from other classes are incorrect retrieval results.

From Table 1 when the training datasets have few ($<= 10$) images per class, center loss leads to improvement. To understand the performance of center loss based network, we conducted a t-sne analysis for all the 3 models in Table 1 as seen in Fig. 2.

One main point of difference with previous works is that we are training on a very different data distribution with huge number of classes and few images per class. Unfortunately we do not have any previous results that have been trained on a similar data distribution as the partial Google Landmarks for comparison purposes.

**Table 1.** Comparative study of mAP scores for different losses using different models. We see that the model fine tuned using both cross-entropy loss and center loss performs better than just using cross-entropy loss

| Model | Pre-trained on Imagenet | Fine-tuned using cross-entropy loss | Fine-tuned using cross-entropy and center loss |
|---|---|---|---|
| Resnet18 | 31.43% | 49.54% | 56.42% |
| Resnet101 | 35.17% | 45.165% | 61.22% |

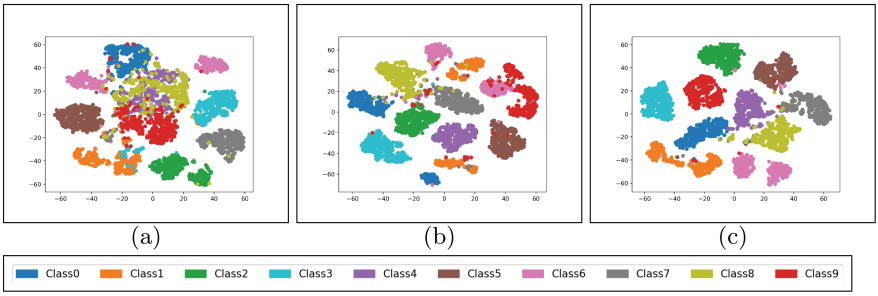



**Fig. 2.** We plot the t-SNE scatter plots for 10 random classes with 500 images from each class. The first figure (a) is the scatter plot for model pre-trained on Imagenet, the second (b) for model fine-tuned with cross-entropy loss only and the third figure (c) is for model fine-tuned with cross-entropy loss and center loss. As we can see in the figure, center+ cross-entropy loss performs better clustering than just cross-entropy loss and they both perform better than the model just pre-trained on Imagenet. Specifically between (b) and (c) - in (b) classes 0, 1 and 6 are split into 2 groups with other classes in between. This is not observed in (c)

## 6 Conclusion

We explored the effect of center loss training on image retrieval applications. A combination of center loss and cross-entropy loss performs better than just using cross-entropy loss or center loss separately. Also using cross-entropy on center distance to compute center loss instead of just the squared Euclidean distance stabilizes the center loss network. Any of the earlier techniques including VLAD encoding of intermediate layers, R-MAC etc can be used on top of this network for better results. Center loss based network is most useful when the training dataset has a large number of classes with few images per class. In the future, we plan to apply the model to other applications such as clustering and few-shot learning.

**Acknowledgement.** This work was supported by the DARPA MediFor program under cooperative agreement FA87501620191, "Physical and Semantic Integrity Measures for Media Forensics". The authors acknowledge the Maryland Advanced Research Computing Center (MARCC) for providing computing resources.

## References


1. https://www.kaggle.com/pytorch/resnet18
2. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 584–599. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_38
3. Chen, Y., Wang, J.Z., Krovetz, R.: Clue: cluster-based retrieval of images by unsupervised learning. IEEE Trans. Image Process. **14**(8), 1187–1201 (2005)
4. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: learning global representations for image search. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VI. LNCS, vol. 9910, pp. 241–257. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_15
5. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. Int. J. Comput. Vis. **124**(2), 237–254 (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Liu, Y., Zhang, D., Lu, G.: Region-based image retrieval with high-level semantics using decision tree learning. Pattern Recognit. **41**(8), 2554–2570 (2008)
8. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Trans. Pattern Anal. Mach. Intell. **18**(8), 837–842 (1996)
9. Ng, J.Y.H., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. arXiv preprint arXiv:1504.05133 (2015)
10. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3456–3465 (2017)
11. Rafiee, G., Dlay, S.S., Woo, W.L.: A review of content-based image retrieval. In: 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP), pp. 775–779. IEEE (2010)
12. Salvador, A., Giró-i Nieto, X., Marqués, F., Satoh, S.: Faster r-cnn features for instance search. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on. pp. 394–401. IEEE (2016)
13. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **19**(5), 530–535 (1997)
14. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4080–4090 (2017)
15. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. arXiv preprint arXiv:1511.05879 (2015)
16. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VII. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31
17. Zheng, L., Yang, Y., Tian, Q.: Sift meets CNN: a decade survey of instance retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **40**(5), 1224 (2017)