Distance-Based Class Activation Map for Metric Learning

Yeqing Shen[1], Huimin Ma[2](✉), Xiaowen Zhang[1], Tianyu Hu[2],
and Yuhan Dong[1,3]

[1] Tsinghua University, Beijing, China
{shenyq18,zhangxw18}@mails.tsinghua.edu.cn
[2] University of Science and Technology Beijing, Beijing, China
{mhmpub,tianyu}@ustb.edu.cn
[3] Tsinghua Shenzhen International Graduate School, Shenzhen, China
dongyuhan@sz.tsinghua.edu.cn

Abstract. The interpretability of deep neural networks can serve as reliable guidance for algorithm improvement. By visualizing class-relevant features in the form of heatmap, the Class Activation Map (CAM) and derivative versions have been widely exploited to study the interpretability of softmax-based neural networks. However, CAM cannot be adopted directly for metric learning, because there is no fully-connected layer in metric-learning-based methods. To solve this problem, we propose a **Distance-based Class Activation Map (Dist-CAM)** in this paper, which can be applied to metric learning directly. Comprehensive experiments are conducted with several convolutional neural networks trained on the ILSVRC 2012 and the result shows that Dist-CAM can achieve better performance than the original CAM in weakly-supervised localization tasks, which means the heatmap generated by Dist-CAM can effectively visualize class-relevant features. Finally, the applications of Dist-CAM on specific tasks, i.e., few-shot learning, image retrieval and re-identification, based on metric learning are presented.

Keywords: Neural network interpretability · Class activation map · Metric learning

1 Introduction

The past few years have witnessed a major development in deep learning, and impressive achievements have been made in several tasks, e.g., recognition, detection and reinforcement learning. However, current deep neural networks are facing the problem of interpretability. Specifically, deep neural networks contain a large number of learnable parameters, which must be trained on a grand scale of data using gradient descent strategy, and thus the prediction results of the neural network are less possible to be appropriately interpreted based on the parameters

H. Ma et al. (Eds.): PRCV 2021, LNCS 13022, pp. 336–347, 2021.
https://doi.org/10.1007/978-3-030-88013-2_28

of the network. Many studies [2,12,21] have been trying to solve this problem of network interpretability, and one representative approach is to visualize the output of the neural network, of which Class Activation Map (CAM) [20] has made significant progress. To be more specific, CAM is applied to the classification network based on Convolutional Neural Network (CNN). This classification network extracts the features of images through CNN, and the features will then be combined through fully connected layer to obtain the prediction results. To visualize the classification network, CAM combines the feature map of CNN with the weights of fully connected layers, based on which the heatmap of the image to be recognized can be generated. However, this approach relies heavily on the fully connected layers of the classification network, thereby making it difficult to be applied to metric learning which lacks fully connected layer.

Metric learning aims at learning a representation function which maps objects into a CNN network. The object's similarity should be reflected in the distance of the CNN network, i.e., the distance between similar objects is as reduced as possible while dissimilar objects are far from each other. This approach has been extensively applied in image retrieval [1,4,15,18], re-ID [5,9,10,17] and few-shot learning [3,6–8,14], etc. Therefore, it is of great value to interpret the neural networks in metric learning. Facing this problem of interpretability, we propose a Dist-CAM to achieve better interpretability of neural networks in metric learning. In particular, the current study focuses on solving the problem of CAM in its limited applicability to broader network architectures, and tries to extend CAM to metric learning based on the idea of class activation.

As shown in Fig. 1, in a recognition network with fully connected layer, the feature maps of different channels combine with the weights of the fully connected layer. The class with the highest probability in the output is considered as the final prediction result. On the other hand, different regions of the heatmap generated by CAM have distinct responses. Particularly, a region with a higher response indicates that the features of this area have more contributions to the output. Therefore, CAM actually establishes a relationship between the output of the classification network and different regions of the original image in which a stronger response implies a greater relevance.

By contrast, in a metric learning network without fully connected layer in Fig. 2, the features of the sample are firstly extracted by CNN, and then the distance between the features of the test sample and those of the training sample is calculated in the metric module. The class of the training sample which has the smallest distance from the test sample is considered as the prediction result. Following this idea of class prediction in metric learning, the Dist-CAM proposed by the current study establishes a relationship between the prediction results of metric learning and different locations of the original image. To be more specific, the main idea is to calculate the distance between the feature maps of the training sample and those of the test sample in different channels, in order to evaluate their relevance. The obtained relevance score is used as the weights corresponding to the different channels of the feature maps of the test sample, in which a channel with a higher relevance score has a greater weight.

The relevance between features of the test sample in a particular location and the class of the training sample is reflected in the responses of different locations in the generated heatmap, and a higher response suggests a greater relevance between the feature and the training set.

The main contribution of the current paper is that we propose the Dist-CAM which can be employed in the interpretation of the neural networks in metric learning. To evaluate the quality of heatmaps generated by Dist-CAM, comprehensive experiments have been conducted on weakly-supervised localization tasks in ImageNet [13]. The results show that the regions with higher responses in the heatmaps generated by Dist-CAM are indeed the class-relevant features, which clearly indicates that Dist-CAM can be used in metric learning to effectively interpret neural networks. Additionally, this paper also compares the performance of CAM and Dist-CAM in weakly-supervised localization tasks, and the results show that Dist-CAM with the same backbone network as CAM can achieve more accurate localization. To the best of our knowledge, this is the first method that can be used to generate class activation map for metric learning. In order to evaluate the applicability of Dist-CAM, the visualization results in several metric learning tasks are also presented in Sect. 5.

2 Distance-Based Class Activation Map

Details on CAM in softmax-based methods and Dist-CAM in metric learning as well as their comparison will be introduced in this section.

Class Activation Map

Softmax-based learning refers to neural networks with fully-connected(FC) layers, which calculates probabilities of different classes, and predicts the final result with the softmax layer. The operation of softmax layer is shown in Eq. 1, in which $\hat{y}$ is the prediction of softmax and v_i is the value of vector generated by FC layers.

$$\hat{y} = \frac{e^{v_i}}{\sum_j e^{v_i}} \tag{1}$$

The architecture of CAM [20] for the softmax-based classification networks is shown in Fig. 1(a), in which FC means FC layer. Test feature map $Q \in \mathbf{R}^{W \times H \times K}$ is encoded by a convolutional neural network (CNN) backbone with test sample as input. The width and height of feature map are denoted as W and H, and the number of its channels is denoted as K. Furthermore, the feature map of the k-th channel are denoted as $Q_k \in \mathbf{R}^{W \times H}$. Then Global Average Pooling (GAP) calculates the average of feature map in each channel, which is denoted as $\mathcal{GAP}(\cdot)$. Finally, as shown in Eq. 2, the output of the GAP is combined with weight of FC layers to obtain the confidence of each class. For a specific class c, w_k^c is the FC layer weight of the k-th channel and y^c is the class confidence.

$$y^c = \sum_k w_k^c \mathcal{GAP}(Q_k) \tag{2}$$

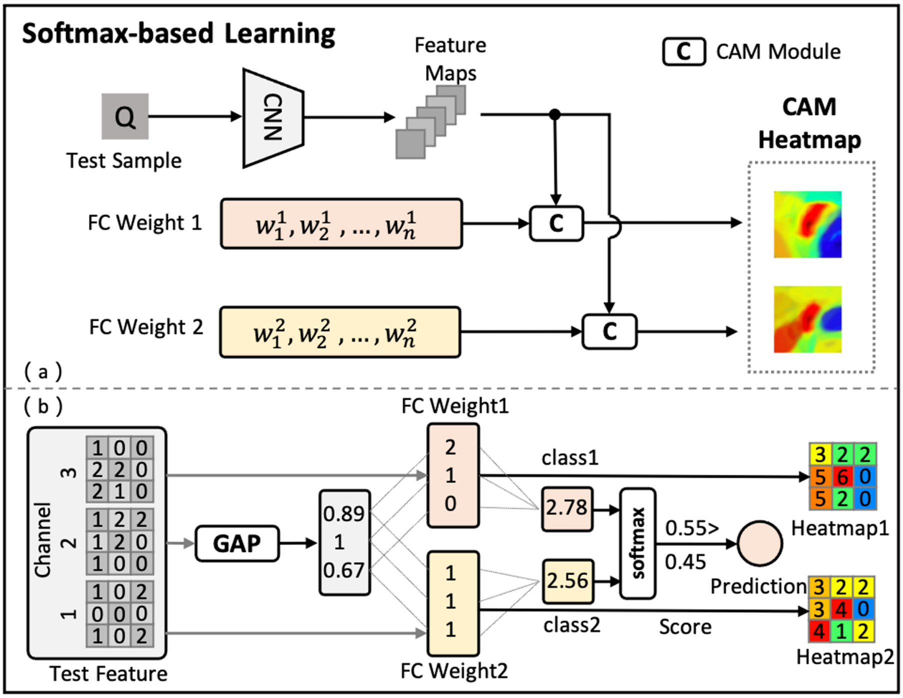

Fig. 1. Class activation map. In softmax-based learning, CAM combines the feature maps with FC weight to generate heatmaps. (a) The framework of softmax-base learning with CAM module. (b) A simplified illustration of CAM generation process.

As Eq. 3 shows, heatmap of class c is obtained by combining test feature maps with FC layers weight of class c in the CAM module. Different regions of the heatmap have distinct response. A larger response indicates a greater correlation between the region and the class c. High response region can be regarded as activation, so the heatmap of class c is the class c activation map.

$$M_{cam}^{c} = \sum_{k} w_k^c Q_k \tag{3}$$

As Fig. 1(b) shows, the final prediction is class 1, because the score of class 1 is 0.55 and the score of class 2 is 0.45. Heatmap 1 and heatmap 2 are generated by combining test feature maps with FC weight 1 and FC weight 2. A higher response in heatmap c indicates a greater correlation between the region and the class c.

However, due to the dependence on the weight of the FC layer, CAM can only be applied in the softmax-based methods, and cannot be used in metric learning. To solve this problem, the distance-based Class Activation Map is designed to analyze the models in metric learning.

Distance-Based Class Activation Map
As shown in Fig. 2 , the feature of unlabeled test sample Q is predicted in metric module according to the distances to other training sample features $\mathcal{S} = \{S_0, S_1, \cdots, S_N\}$. In addition, for class c, we denote the i-th feature of training sample of the k-th channel as $s^c_{i,k}$.

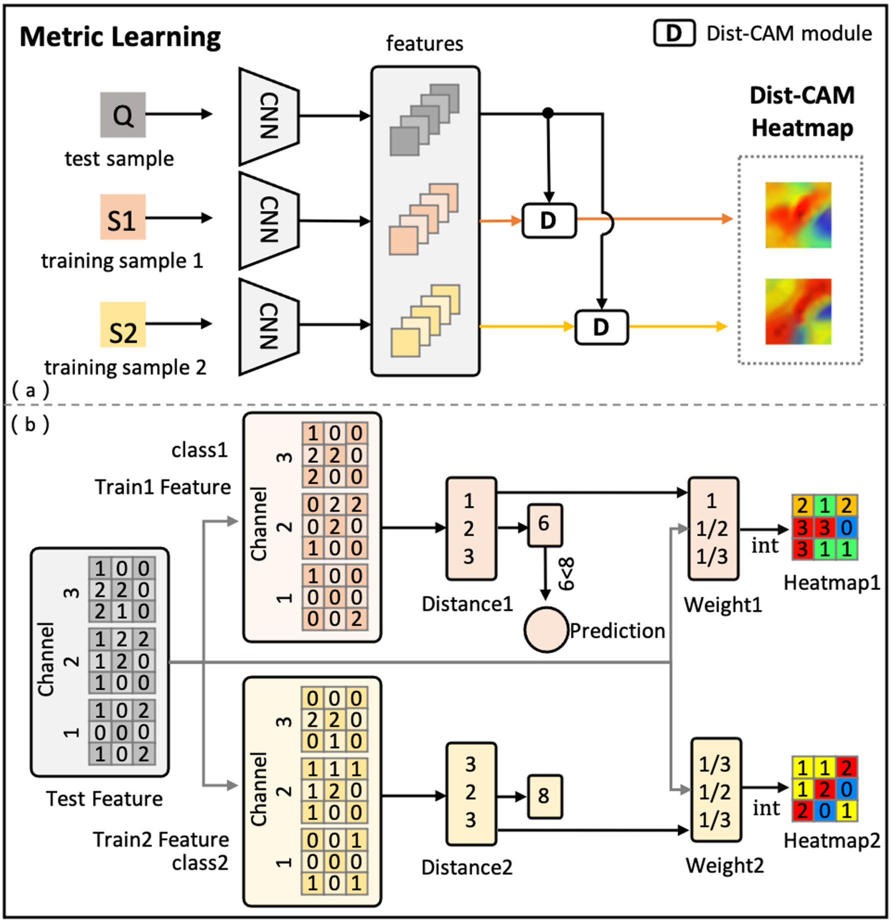

Fig. 2. Distance-based class activation map. In metric learning, Dist-CAM combines the feature maps with weight to generate heatmaps. (a) The framework of metric learning with Dist-CAM module. (b)A simplified illustration of Dist-CAM generation process.

As Eq. 4 shows, the distance D^c is calculated between the feature of training sample S^c and that of test sample Q via the GAP layer, where the operation $\mathcal{L}_1(\cdot)$ refers to element-wise 1-norm. The k-th dimension in the vector D^c represents the difference between S^c and Q in the k-th channel. It is clear that the difference is positively correlated with the distance value. Therefore, the smaller value of D^c_k is, the closer Q_k is to S^c_k. If the feature is closer to the training sample c in the channel k, it is supposed that the k-th channel plays a more important role in the prediction.

$$D^c = \mathcal{L}_1(\mathcal{GAP}(S^c) - \mathcal{GAP}(Q)), \quad D^c \in \mathbf{R}^K \tag{4}$$

Similar to the combination of the GAP outputs in Eq. 2, the output is obtained by the summation of D^c in metric learning. However, the largest response of class confidence y^c by the combination in Eq. 2 determines the output class c, while the smallest summation of D^c determines the output class c. Therefore, we transform D^c to W^c as shown in Eq. 5, where $norm(\cdot)$ is the operation of element-wise normalization. Thus the coefficient W reflects the distances from the different channels of Q to training sample S^c. The bigger the weight W^c_k is, the more relevant the k-th channel is with the class c.

$$W^c = norm(\frac{1}{D^c}), \quad W^c \in \mathbf{R}^K \tag{5}$$

$$M^c_{DistCAM} = \sum_k W^c_k Q^k \tag{6}$$

$M^c_{DistCAM}$ is calculated by combining Q^k with weight vector W^c_k as shown in Eq. 6. The different responses in Dist-CAM heatmap reflect distinct level of class activation. High response region can be regarded as activation, so the heatmap of class c is the class c activation map. As shown in Fig. 2(b), the prediction is class 1, because the distance in feature space between test sample and training sample 1 is 6, and the distance in feature space between test sample and training sample 2 is 8. In the Dist-CAM module, weight of class W^c is reciprocal of distance vector D^c. Heatmap 1 and heatmap 2 are generated by combining test feature maps with weight 1 and weight 2. A higher response in Dist-CAM heatmap c indicates a greater correlation between the region and the class c.

In Dist-CAM heatmap, a red pixel represents a large activation value, and a blue pixel represents a small activation value. The region with a large activation value suggests that feature of this region has a large contribution to the final prediction. As shown in Fig. 3, the response regions of Dist-CAM heatmaps generated with different categories of prototypes are also different. The input image contains a cat and a dog. The Dist-CAM modules measure the feature with the prototypes of cat, dog, bird, and table respectively. Among visualization results, the Dist-CAM heatmaps of cat class and dog class respond significantly at the locations of the two objects, while the other two activation regions are chaotic and scattered, which means Dist-cam effectively activates the class-relevant regions.

$$P^c_k = \frac{1}{N} \sum_i S^c_{(i,k)}, \quad P^c_k \in \mathbf{R}^{W \times H} \tag{7}$$

In some metric learning applications, the prediction of test sample is determined by the nearest class prototype, instead of the nearest training sample.

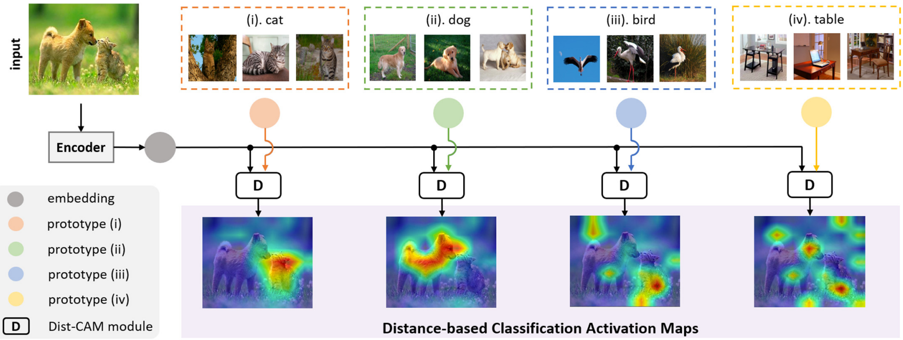

Fig. 3. Framework of Distance-based class activation Map. Different colors represent the corresponding images and prototypes of different categories (i)–(iv). The input image is encoded by a feature encoder, and the features are fed to Dist-CAM modules to class-wise calculate the distance-based Class Activation Maps with different prototypes. The Dist-CAM heatmaps of cat class and dog class respond significantly at the locations of the two objects, while the other two activation regions are chaotic and scattered.

The prototype of class c is denoted as P^c, which is obtained by calculating the mean value of features of all training samples in class c. The way to calculate prototype P^c is shown in Eq. 7, in which N is the number of training samples. In these applications, the prototype P^c replaces the training sample S^c to calculate the distance D^c and generate the Dist-CAM heatmap $M^c_{DistCAM}$.

3 Experiments

Weakly-supervised localization task is chosen for the evaluating the performance of Dist-CAM for the following reasons. On the one hand, neural networks in classification task and those in weakly-supervised localization task share the same kind of label, i.e., both are trained with class label. On the other, both weakly-supervised localization task and class activation task focus on localizing class-relevant regions. Therefore, weakly-supervised localization task is appropriate to be used in the experiments.

Experiments are conducted to quantitatively evaluate the weakly-supervised localization ability of Dist-CAM on the ILSVRC 2012 benchmark [13]. We adopt two main CNN backbones: ResNet and DenseNet. For each backbone, we have trained the model on ILSVRC 2012 training set to obtain softmax-based models with FC layers and distance-based metric learning models, respectively.

In order to ensure the fairness of the experiment, N in Eq. 7 is set to the number of training samples of each class when calculating the prototype in metric learning. Besides, we drop the last GAP layer and the FC layer of models to obtain a multi-channel feature map with spatial position information for metric learning experiments. For example, the input image with the size of 224 × 224 × 3 is encoded by a modified ResNet backbone into a feature map with the size of

Table 1. Localization error (top-1) of the Dist-CAM on ILSVRC2012 *val* sets.

Backbone	Method	Top-1 error
ResNet-18	CAM	59.5
ResNet-18	**Dist-CAM**	**56.7**
ResNet-50	CAM	56.2
ResNet-50	**Dist-CAM**	**54.7**
ResNet-101	CAM	56.6
ResNet-101	**Dist-CAM**	**54.9**
DenseNet-161	CAM	62.4
DenseNet-161	**Dist-CAM**	**57.6**
DenseNet-169	CAM	62.3
DenseNet-169	**Dist-CAM**	**56.6**
DenseNet-201	CAM	62.5
DenseNet-201	**Dist-CAM**	**56.4**

$2048 \times 7 \times 7$. When predicting the class, the feature map is transformed into a 2048-dimensional feature of test sample by the GAP layer. Then the network calculates the distances between the 2048-dimensional feature and each prototype and selects the nearest one as the prediction. When calculating the Dist-CAM, the network measures the channel-wise distances of the 2048-dimensional feature and the nearest prototype and obtains the channel weighting coefficients by reciprocal and normalization. Finally, a weighted summation of the original $2048 \times 7 \times 7$ feature map with the coefficients is performed to obtain the Dist-CAM with the size of 7×7.

To compare with the original CAM, we evaluate the localization ability of Dist-CAM with the same error metric (top-1) on the ILSVRC 2012 [13] *val* set. The intersection of union (IoU) threshold is set to generate bounding boxes on the positions with strong responses. Compared with groundtruth, when the IoU between them is lower than 0.5, the prediction is considered as a wrong localization.

As shown in Table 1, Dist-CAM has achieved a lower top-1 error than the original CAM, which indicates the effectiveness of our method. Without the FC layer, Dist-CAM has achieved accurate localization of discriminative features, which is effective to interpret the model in metric learning. Meanwhile, the Dist-CAM can also be regarded as a promotion of CAM, which can be used in the softmax-based methods by dropping the last FC layer.

4 Metric-Based Applications

Dist-CAM is applied to three main tasks of metric learning, including few-shot learning, image retrieval and re-identification. Experiments are conducted to show the visualization results to explore network interpretability. Note that in few-shot learning, support sample refers to training sample mentioned in Sect. 2 and query sample refers to test sample. In image retrieval and re-identification, training sample mentioned in Sect. 2 is defined as gallery sample and test sample is defined as probe sample.

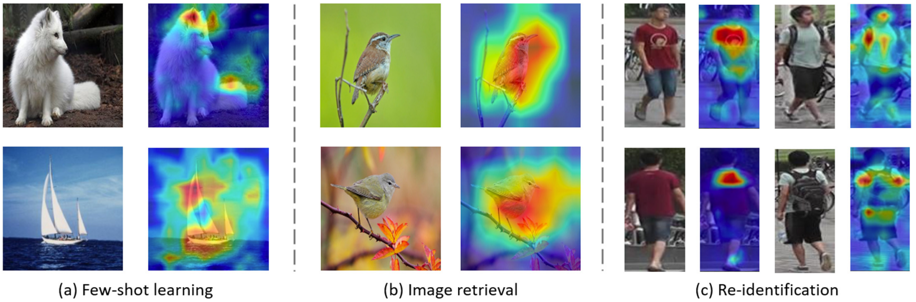

Fig. 4. Dist-CAM visualization for metric-based applications. (a) Dist-CAM heatmaps for few-shot learning on miniImageNet; (b) Dist-CAM heatmaps for image retrieval on CUB2011; Dist-CAM heatmaps for pedestrian re-identification on Market-1501.

4.1 Few-Shot Learning

Few-shot learning is one of the most significant applications for metric learning [3,6–8,14]. Studies on few-shot learning aim to overcome the lack of support samples to obtain a useful recognition model. Inspired by the idea of meta-learning, most of the current approaches adopt a number of tasks similar to the target one to train a meta-learner. Specifically, the majority of current mainstream methods are based on metric learning frameworks. To predict the class of unlabeled query sample, the metric learning frameworks measure the distances among the unlabeled query sample and the class prototypes, which are calculated from the labeled support samples.

As mentioned in Sect. 1, the lack of FC layers limits the original CAM methods to analyze the model performance. To explore the network interpretability of meta-learning, Dist-CAM is used to localize the discriminative features and analyze the advantages of metric-based few-shot learning.

We adopt ResNet-12 as a backbone and train the model [7] by meta-learning strategy. The performance has achieved a state-of-the-art performance with 77.14 ± 0.42 in the-5-way-1-shot task. Figure 4(a) shows Dist-CAM for few-shot learning on mini-Imagenet [11] dataset. According to the setting of the one-shot task, only one support sample is used to calculate the prototype. It can be seen that the activation location of the fox in the above image is concentrated on the ears and tail, and the activation location of the sailboat in the image below is concentrated on the sail and hull. These are indeed the class-defining features related to these classes. Results show that Dist-CAM is a useful tool to analyze the metric-based few-shot learning, which can also be used to improve the training strategies by analyzing failure cases.

4.2 Image Retrieval

Metric learning is widely used in image retrieval to search images by images. For softmax-based methods, it is exhausting to retrain the model when a novel class sample appears. By contrast, there is no need to retrain the model in metric learning when adding a novel class. Therefore, image retrieval is an important application of metric learning, trying to retrieve the same object or class as the probe sample from the existing gallery samples [1,4,15,18]. Different from the classification algorithm with fixed output classes, the metric learning algorithm increases the inter-class interval and reduces the intra-class variance by designing the loss function during the training process. Images are encoded by backbone and retrieved according to distance calculated by metric module. In the application stage of image retrieval, even if new categories are added, the trained backbone can be used for prediction in metric learning, which is difficult for classification algorithms with a fixed number of classes.

Dist-CAM is used to analyze an image retrieval model trained on the CUB2011 [16], a dataset containing 200 categories of birds, of which ResNet-50 is used as the backbone. According to the settings of the image retrieval task, only one gallery sample that is the closest to the probe sample is used to calculate the Dist-CAM heatmaps. It can be seen from Fig. 4(b) that Dist-CAM heatmaps have a good weakly supervised localization capability. The activation location is concentrated on the region of the birds. Therefore, it clearly shows that Dist-CAM can be used to analyze the results and failure cases of the image retrieval model.

4.3 Re-identification

As a sub-problem of image retrieval, re-identification aims to find the expected target in the image library [5,9,10,17]. During the test phase, the probe images to be retrieved are used to calculate feature distances with gallery images. Then top-k images are retrieved according to feature distances. Plenty of re-identification algorithms are based on metric learning to avoid retraining the model when adding a novel class as mentioned in previous section.

We adopt the ResNet-50 as a backbone and train the model with center loss and reranking strategy [10]. The rank-1 accuracy achieves 94.5 and the average precision achieves 85.9 on the Market-1501 [19]. According to the settings of the re-identification, only one gallery sample that is the closest to the probe sample is used to calculate the prototype. Figure 4(c) shows Dist-CAM of the probe samples in the Market-1501 dataset for pedestrian re-identification. It can be seen that the discriminative features are concentrated on the human clothes. The Dist-CAM heatmaps in Fig. 4(c) reflect that clothes are an important feature for re-identification in these probe samples.

5 Conclusion

In this paper, we propose **Distance-based Class Activation Map** for deep metric learning to explore the interpretability of metric-based networks. On weakly supervised object localization tasks on ILSVRC 2012 [13], comprehensive experiments are conducted and the result shows that Dist-CAM is better than the original CAM with ResNet or DenseNet as backbones. Besides, we demonstrate the visualization of Dist-CAM in specific applications of metric learning, including few-shot learning, image retrieval and re-identification. In the future, it is worthwhile to adopt the Dist-CAM to guide the innovation and improvement of metric learning methods.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No.U20B2062), the fellowship of China Postdoctoral Science Foundation (No.2021M690354), the Beijing Municipal Science & Technology Project (No.Z191100007419001).

References

1. Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S.: Deep metric learning to rank. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1861–1870 (2019)
2. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 839–847 (2018)
3. Chu, W., Wang, Y.F.: Learning semantics-guided visual attention for few-shot image classification. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 2979–2983 (2018). https://doi.org/10.1109/ICIP.2018.8451350
4. Ge, W., Huang, W., Dong, D., Scott, M.R.: Deep metric learning with hierarchical triplet loss. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 272–288 (2018)
5. Hao, Y., Wang, N., Li, J., Gao, X.: Hsme: Hypersphere manifold embedding for visible thermal person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8385–8392 (2019)
6. Li, X., et al.: Learning to self-train for semi-supervised few-shot classification. In: 33rd Conference on Neural Information Processing Systems. vol. 32, pp. 10276–10286 (2019)
7. Liu, J., Song, L., Qin, Y.: Prototype rectification for few-shot learning. In: ECCV, vol. 1. pp. 741–756 (2019)
8. Liu, L., Zhou, T., Long, G., Jiang, J., Yao, L., Zhang, C.: Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 3015–3022 (2019)
9. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: deep hypersphere embedding for face recognition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6738–6746 (2017)
10. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 0–0 (2019)
11. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: International Conference on Learning Representations (2018)

12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. **128**(2), 336–359 (2020)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations 2015 (ICLR 2015) (2015)
14. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. Adv. Neural Inf. Process. Syst. **30**, 4077–4087 (2017)
15. Song, H.O., Jegelka, S., Rathod, V., Murphy, K.: Deep metric learning via facility location. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2206–2214 (2017)
16. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
17. Wang, H., Zhu, X., Xiang, T., Gong, S.: Towards unsupervised open-set person re-identification. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 769–773 (2016). https://doi.org/10.1109/ICIP.2016.7532461
18. Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep metric learning with angular loss. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2612–2620 (2017)
19. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: a benchmark. In: IEEE International Conference on Computer Vision (2015)
20. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society (2016)
21. Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., Jiao, J.: Weakly supervised instance segmentation using class peak response. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3791–3800 (2018)