electronics

Article

Glocal Retriever: Glocal Image Retrieval Using the Combination of Global and Local Descriptors

Zeu Kim †, Youngin Kim † and Young-Joo Suh *

Graduate School of Artificial Intelligence, Pohang University of Science and Technology, Pohang 37673, Republic of Korea
* Correspondence: yjsuh@postech.ac.kr; Tel.: +82-54-79-2243
† These authors contributed equally to this work.

Abstract: Development of deep learning has led to progress in computer vision, including metric learning tasks such as image retrieval, through convolutional neural networks. In image retrieval, the metric distance (i.e., the similarity) between the images needs to be computed and then compared to return similar images. Global descriptors are good at extracting holistic features of an image, such as the overall shape of the main object and the silhouette. On the other hand, the local features extract the detailed features which the model uses to help classify similar images together. This paper proposes a descriptor mixer which takes advantage of both local and global descriptors (group of features combined into one) as well as different types of global descriptors for an effect of a lighter version of an ensemble of models (i.e., fewer parameters and smaller model size than those of actual ensemble of networks). As a result, the model's performance improved about 1.36% (recall @ 32) when the combination of the descriptors were used. We empirically found out that the combination of GeM and MAC achieved the highest performance.

Keywords: image retrieval; deep metric learning; computer vision

Citation: Kim, Z.; Kim, Y.; Suh, Y.-J. Glocal Retriever: Glocal Image Retrieval Using the Combination of Global and Local Descriptors. *Electronics* **2023**, *12*, 442. https://doi.org/10.3390/electronics12020442

Academic Editor: Hung-Yu Chien

Received: 7 November 2022
Revised: 26 December 2022
Accepted: 8 January 2023
Published: 14 January 2023

1. Introduction

Introduction of deep learning has immensely enhanced the capacity for capturing non-linear data and to be robust against changes in objects [1]. Deep learning in general allows researchers to automatically extract discriminative features through neural networks, making it easier to create a similarity function that could recognize objects in images the model has not seen before, while the introduction of convolutional neural networks [2–5] has significantly improved performances in computer vision tasks. The performance of metric learning did not improve as much when classification techniques were implemented directly to metric learning. It is mainly because while metric learning requires the network to identify features to help match similar images with the query image, which can be a similar task to image classification, the essence of metric learning is different from image classification [1]. In most cases, image classification classifies images by classes or categories while image retrieval does not necessarily classify images by classes but rather clusters and categorizes images by their similarities, which may not be as obvious as features exploited in classifications, compared to the query image. For example, in the case of image retrieval, the model needs to find other images which possibly match the object detected in the query image and not just simply identify what the object is.

To counter the issue, many models with CNN backbone networks have been developed by researchers. Having a backbone network enhances the performances of the models with some "pre-knowledge" on the exact, if not similar, datasets used to train and test the performance. Some of the networks exploit the use of different global descriptors (a group of features combined into one vector) while other models found ways to explore both local and global features through descriptors. However, there are not many models which have

different global descriptors and use local and global features in one model. In this paper, we present a descriptor mixer that utilizes both local and global descriptors as well as different types of descriptors for an effect similar to implementing an ensemble of models with a smaller size and fewer parameters. The paper achieves the goals by proposing a model which consists of various parts: a global module, a local module, an auxiliary module.

The contributions can be summarized as the following:

- Using both types of descriptors allow the features to complement each other while using only one type of descriptor has some drawbacks due to the limited information the features provide.
- The combination of descriptors does not add large numbers of parameters, allowing the model to stay relatively light.
- The use of an auxiliary module provides the model a complementary criterion to judge what images to retrieve.
- The paper also tests the good stage to concatenate the descriptors together to make it easier for the model to extract significant features.

2. Related Works

In the past, metric learning was performed in a classic (non-deep) learning way such as calculating the Mahalanobis Distance and the Large Margin Nearest Neighbor [6]. While those methods were able to help compute the distance and the similarities, classic metric learning did not have a high capacity to capture non-linear data and be robust against changes in objects [1]. The images are very likely to be in a different poses, illuminations, and expressions from the trained images the model experienced.

After deep metric learning was introduced, several models and methods were implemented. Some of the most popular backbone networks used for feature extraction are AlexNet [3], VGG [4], and ResNet [5]. However, it was not enough just to directly implement those classic convolutional neural networks to metric learning. Therefore, fine tuning methods were introduced. One example is fine tuning using Siamese networks [7], a pair of neural networks which share parameters. Similarly, triple networks were also implemented by having the network to obtain triplet results after forward propagating a mini batch [8]. Furthermore, there are researchers that use vision transformers as backbone networks. Ouyang et al. use the Swin transformer to build a Siamese network and then fine-tunes the model [9]. Patel et al. also utilizes a group of aggregated global features to provide global information exchange between the windows in a local transformer [10].

Deep feature enhancement was also implemented to embed features to improve the discrimination of deep features. Attention mechanisms underscore the most influential part of the feature map, bolstering the network to avoid observing irrelevant parts of the image. Li et al. [11] and Noh et al. [12] used either fully connected layers or convolutional layers to highlight important features. Wang et al. proposes the 'visual-test joint-embedding learning' which uses position attention to learn the relationship between the visual image and the textual sentence [13].

Ensemble is a technique that aims to improve performance of a model through a combination of several trained networks [14]. Ensemble models in deep metric learning [15,16] proposed by researchers include attention based ensemble, proposed by Kim et al., and random bagging method implemented by Xuan et al. In this paper, an ensemble of global descriptors are used to exploit the diversities the descriptors provide without heavily increasing the model size.

Some use methods other than descriptors to compute the metric distance and retrieve images. One method is hashing embedding, which allows the the features to be more compact and result in producing more efficient models and algorithms. Zhai et al. use hashing to transfer knowledge to a student network from a teacher network for speedier image retrieval [17]. Jin et al. use a similar approach using hashing [18] to find targeted people in person re-identification systems from non-overalpping cameras.

The concepts of local and global features were widely used in previous works in metric learning. In metric deep learning, different papers [19–21] have proposed how to extract local features in many ways. One example is DELF [21], proposed by Noh et al. It implements an attentive local feature descriptor using a single scale feature map. In case of global features, hardwired methods, such as SIFT [22], were used before deep learning was introduced. Currently, methods such as RMAC [23], SPoC [24], and GeM [25] are used with deep neural networks to extract features. There are also joint local and global feature extractors such as DELG [26] were previously proposed. Ko et al. recently proposed the group generalized mean pooling (GGeM) to retrieve images using a vision transformer.

There are also models which use both global and local descriptors. DOLG [27] combines a global descriptor and a local descriptor to take advantage of both holistic and delicate features of an image. DALG, presented by Y.Song et al, uses a cross-attention module to hierarchically (instead of heuristically) fuse the features [28]. Furthermore, Alappat et al. present a model that uses an Inception V3 backbone network and extracts the MS-RMAC feature matrix to retrieve images [29]. Global–local attention module (GLAM), proposed by C.Song, combines both local and global attention as well as spatial and channel attention and then computes a new feature tensor [30].

3. Methods

The overall model is illustrated in Figure 1. The model consists of a backbone network, a global module, a local module, and then an orthogonal fusion module [27] to combine the global and the local descriptors. Then, it is followed by a fully connected layer which reduces the dimension of the descriptor. Finally, all the outputs are then concatenated and fed into a L2 normalization layer to calculate the metric loss. A descriptor in this paper refers to a group of features combined into one vector [31]. Therefore, a local descriptor is a group of features extracted from distinctive parts of the image and a global descriptor is similarly a group of features that contain information on the holistic image.

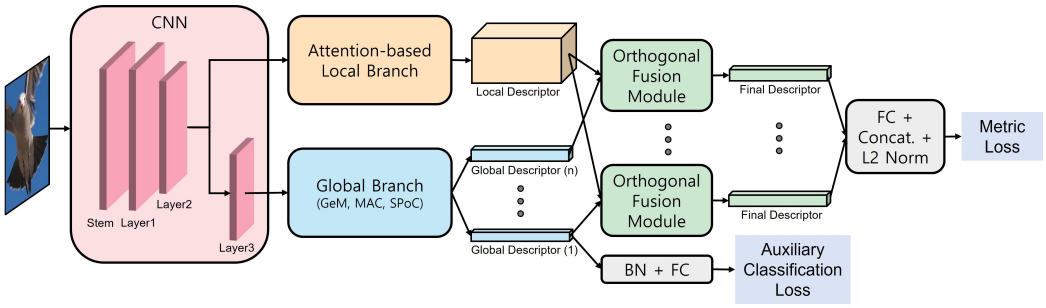

Figure 1. The full architecture of the model.

Our model is inspired by CGD [32] and DOLG [27]. CGD introduces a series of different global descriptors while DOLG proposes a way to effectively fuse local and global descriptors into a single stage solution. To upgrade the model even further, we combined and merged both local and global descriptors as well as several kinds of global descriptors.

3.1. Backbone Network

This model uses a pretrained ResNet-50 [5] as our backbone network. ResNet, introduced by He et al. in 2016, is a commonly used CNN renowned for its robustness against vanishing gradients and relatively low training errors when training deeper networks. ResNet is known for its deep layers and batch normalization. When ResNet was presented, it had a significantly large number of layers compared to other CNNs such as VGG [4], thanks to the structure of residual blocks; batch normalization [33]; and He initialization [34].

3.2. Global Module

The global descriptor extracts the overall shape and the silhouette of the image and enhances the model to learn the holistic features of an image. In image retrieval, it is not as important to classify the dataset into given classes but instead it is more important to retrieve similar images regardless of classes. In this model, the global branch consists of a total of three global descriptors: SPoC [24], MAC [35], and GeM [25]. Those global descriptors, which can be used separately or in a combination of two or three, output embedding vectors. A combination of several global descriptors allows the model to have an effect similar to having an ensemble model with fewer parameters and smaller resources compared to actually creating one with several neural networks [32]. Having various descriptors provides the model with large diversity [14], enlarging the model's point of view.

3.3. Local Module

While the global module provides comprehensive information on the input image to the model, a generalized group of features is not enough to search and retrieve similar images from the dataset. To complement the model, a local module is used to extract the local descriptor. The local module looks for similar parts in the images and then helps the model compare and match precise parts of the image with the input image. For example, the local module would look for parts which help the model in classification tasks. Although it is not used in this paper, class activation maps (CAM) [36] and gradient-weighted CAM (Grad-CAM) [37] visualize where the model is looking at to know what the object in the image is. The local module consists of atrous convolution layers [38], which help the network to obtain a larger field of view with the same computational cost as regular convolutional layers, and an attention map [39] to enhance the network to focus on the important features. The multi-atrous convolutional layers allow the model to take into consideration the scale variations among the images, enhancing the neural network to observe the data in different scales and perspectives. After processing the data with the attention map, both outputs from the attention map and the convolutional layer are then put together in what we call a fusion module.

3.4. Fusion Module

This model uses two types of fusion modules. One is the orthogonal fusion module (OFM) [27] which combines the local and the global descriptor. In this model, because there are a maximum of three global descriptors, each of the global descriptors is fused with the local descriptor and produces a total of three different outputs if all the global descriptors were implemented. To merge the final descriptors into one vector, we implemented another type of fusion module which consists of a fully connected layer to reduce the dimension of the final descriptors. Then the module concatenates the output and creates an L2 normalized single descriptor to compute the metric loss. For results related to concatenation, please refer to Table 3.

3.5. Auxiliary Module

This model also has an auxiliary module [32]. The auxiliary module is a part of the global module which computes the auxiliary classification loss. The auxiliary loss complements the model further by not only providing information on the class of the model (the images from the same class are more likely to be similar), but also preventing the model overfitting to the dataset by utilizing cross entropy loss, temperature scaling [40], and label smoothing [41]. Temperature scaling and label smoothing help generalize the model by restraining the model from trusting the classes too much. Cross entropy loss is commonly used in image classification. Temperature scaling modifies the model if it is overconfident and enhances the model to adapt better to more examples. Label smoothing was also used to calibrate the model and for domain generalization through considering the fact that labels on the training data might be wrong and in overall cause errors when

the model is too confident in real life applications. Those two calibration methods enhance the model to avoid overfitting.

3.5.1. Loss Function

Our loss function consists of the following: metric loss, auxiliary loss, and the hyper-parameter lambda. The loss function is:

$$Loss = L_M + \lambda L_A \tag{1}$$

L_M denotes the metric loss computed by the main module which consists of both local and global descriptors. L_A is the auxiliary loss computed by the auxiliary module. λ modifies the weight the auxiliary loss has on the total loss. For results related to the auxiliary module and the loss function, please refer to Table 2.

3.5.2. Hyperparameters

The auxiliary module is enabled for the first two epochs of training. The results (Table 2) show that it is better to include the module in recalls recalls at 1, 2, and 4 when λ equals 3, and 8 when λ equals 3. Tables 1 and 3 display results we obtained when λ equals 1. When λ is 1, there is not much benefit to include an auxiliary module. The tables manifest that we were able to achieve the best performance when λ is three (see Table 2).

Table 1. This table shows the performance of the model depending on the layer it received the feature map from. The row 'Layer 4' indicates the global module received the feature map from Layer 4 and it is the same for the row 'Layer 3'. The bold values are the best performances in each recall @ K.

Backbone Layer for GD	**Recall @ K (%)**			
	1	**2**	**4**	**8**
Layer 4	63.03	74.29	82.48	89.18
Layer 3	**71.67**	**81.14**	**88.62**	**93.40**

Table 2. The table shows whether the existence of the auxiliary module is beneficial to the overall performance. The row '2 Epochs' indicates that the auxiliary module was implemented for only first two epochs. The bold values are the best performances in each recall @ K.

λ	**Auxiliary Classification Loss**	**Recall @ K (%)**			
		1	**2**	**4**	**8**
0	**Not used**	71.67	81.14	88.58	*93.40*
1	**All epochs**	68.67	78.92	87.51	93.06
	2 epochs	71.37	80.82	88.54	**93.42**
3	**All epochs**	69.36	79.03	87.36	92.57
	2 epochs	**71.83**	**81.68**	**88.62**	*93.40*
5	**All epochs**	69.87	79.01	87.12	91.78
	2 epochs	71.75	81.64	88.40	92.16

Table 3. This table shows the stage in which the concatenation starts. The row 'Before OFM' shows the performance of the model when the concatenation begins before the descriptors are processed by the OFM. In contrast, the row 'After OFM' indicates the performance after the descriptors are processed by the OFM. The bold values are the best performances in each recall @ K.

Stage for Concatenating	Recall @ K (%)			
	1	**2**	**4**	**8**
Before OFM	69.36	79.03	87.36	92.57
After OFM	**71.67**	**81.14**	**88.62**	**93.40**

3.5.3. Implementation

The experiment was implemented using Pytorch on an NVIDIA RTX 3090 and trained with the CUB-200-2011 Dataset [42]. We used proxy-anchor loss [43] to compute the metric loss and CE loss with label smoothing and temperature scaling to compute the auxiliary classification loss. Some of the results were obtained when we were writing and submitting a paper to a conference [31].

4. Results

4.1. Backbone Network and Layers

The feature maps from Layer 2 are fed to the local module while the feature maps from Layer 3 are fed to the global module. Using Layers 3 and 4 resulted in lower performance (see Table 1). We assume it is because Layer 4 has a lower resolution than Layer 3, which could have made it more difficult for the model to extract accurate features.

Due to limited resources, we were not able to use a larger model as our backbone network. Although it would depend on the size of the dataset and the network, we predict that a larger backbone network might ameliorate the overall performance.

4.2. Use of λ

Table 2 indicates that it is better to include the module in recalls at 1, 2, and 8 when λ equals 3. When λ is 1, there is not much benefit to include an auxiliary module. It can be inferred that the information implied in the auxiliary loss (i.e., the overall classification loss and the confidence inhibitor) is somewhat crucial to the model's decision making process. However, if the auxiliary loss overshadows the main metric loss, the model's decision making process can be exacerbated, especially when the model has to retrieve more than two images.

4.3. Concatenation Stage

There are two ways to concatenate the descriptors. One way is to concatenate the global descriptors and then are processed by the OFM once. The other way is to process the descriptors through the OFM to combine the local and global descriptors and produce several final descriptors. Those final descriptors are then concatenated. In other words, concatenation occurs after the OFM stage. The results (see Table 3) demonstrate that the latter acquires better performance.

4.4. Effect of Descriptors

Table 4 displays the performance difference between models without certain types of descriptors. While the differences are not big, the results show that the recall values are the highest when both types of descriptors are used. Table 5 manifests the performances of the model depending on what kind of global descriptors were used. At the top of the table we included the performances of CGD, which is a model that did not implement a local descriptor. As displayed in Table 5, our model, which employed both local and global descriptors, achieves a better performance.

Table 4. Ablation study—the effect of descriptors. The bold values are the best performances in each recall @ K.

Ablation Study—Descriptors	Recall @ K (%)			
	1	**2**	**4**	**8**
Backbone only	69.72	79.66	86.90	92.03
Only LD	70.91	80.32	87.01	92.18
Only GD	70.53	80.96	87.91	92.82
Using both	**71.67**	**81.14**	**88.62**	**93.40**

As a result, the highest performing model consisted of MAC and GeM functions on the global branch with an auxiliary module with the parameter $\lambda = 3$. We empirically found out that λ of 3 achieves better performance. The overall results were computed and then organized after applying the best empirical hyperparameters we found.

Table 5. Full results with the best hyperparameters regarding the combination of global descriptors. The bold values are the best performances in each recall @ K.

Model		Recall @ K (%)					
		R@1	**2**	**4**	**8**	**16**	**32**
Baseline		69.72	79.66	86.90	92.03	95.58	97.42
CGD [32]	**G+M**	67.60	78.10	86.30	91.90	-	-
Transformer [44]	**GeM**	**78.60**	80.70	82.40	83.40	-	-
Our Method	**S**	62.20	74.43	84.45	90.85	94.99	97.60
	M	70.49	80.13	87.83	92.52	95.97	97.99
	G	70.63	81.04	88.05	93.11	96.20	98.06
	G+S	70.31	80.57	87.85	**93.48**	96.29	98.03
	M+S	70.48	80.37	88.42	93.06	96.27	98.00
	G+M+S	71.66	**81.20**	88.44	93.25	96.03	97.81
	G+M	71.83	81.14	**88.62**	93.40	**96.39**	**98.30**

5. Discussion

5.1. Combinations of Global Descriptors

Combining different descriptors produces a similar effect of having an ensemble of descriptors. In general, it is encouraged to choose functions (whether they are models, descriptors, vectors, etc.) with a large variance to create an ensemble. For this model, the ensemble can be depicted as a collection of experiences and methods of feature extraction. Different functions have factors which can affect the performance (such as the dataset used to train and test the model) and efficiency depending on the situation. The variance between those functions allow the model to adapt to those said situations.

When we trained and tested the model, the combination of GeM and MAC empirically worked the best. When the model only used one descriptor at a time, the performances, even though they had relatively small difference, were lower. The results show that using more than one descriptor is very likely to result in higher performance. However, the combination of all three did not achieve the highest results, which may seem to contradict the explanation above. This phenomenon can be interpreted as either the combination of too many descriptors or the effects of those three specific choices of global descriptor functions did not synergize as well as we expected to. In the former case, too many descriptors used at once can be overall become confusing. This case may occur due to conflicting results by the each individual descriptor. In the latter case, the dataset and the choices of those descriptors may not be the most ideal choices to use together.

As mentioned above, using a combination of global descriptors has a similar effect of using an ensemble of whole models with one global descriptor each. This as a result creates a lighter model overall which can provide some advantages. First, having a lighter model can contribute to a more eco-friendly development of algorithms and models. Strubell et al. states the carbon emission caused by training a deep neural network is about 7 times compared to that caused by a human life in a year and training a transformer emits about five times the amount of CO_2 a car produces in its lifetime [45]. To reduce the carbon footprint caused by developing deep learning models, Wu et al. introduces possible ways to reduce the carbon footprints and presents the empirical results [46].

5.2. The Significance of the Auxiliary Module

Initially, the results that did not use the auxiliary module were better. However, as more experiments were conducted, we realized that it is important to control the hyper-parameter λ. As displayed in Table 2, when the model utilized the auxiliary module for two epochs with the value of λ increased into 3 or 5, R@1 increased from 71.67 (without the auxiliary module) to 71.83 and 71.75%. It can be explained that calculating R@1 is similar to image classification and therefore increasing the impact of the auxiliary loss is likely to augment the recall values at small numbers. On the other hand, the auxiliary loss does not positively influence recall values at large numbers like R@8 because as the number K (from R@K) increases, the classification loss (used to calculate the auxiliary loss) is not as useful. As a result, when K = 1, 2 or 4, the model performance was generally better when λ = 3. In the case of R@8, the model worked best when GeM and SPoC were used. However, when λ = 5 (i.e., when the auxiliary loss was more significant than it should be), the performance degraded instead. The auxiliary loss accelerates the convergence rate but does not help increase the overall performance when used throughout the whole session. Thus, we tried to utilize the module only for the first two epochs for pragmatic purposes. Figure 2 displays that the implementation of two epochs bolstered the model's learning and resulted in faster convergence.

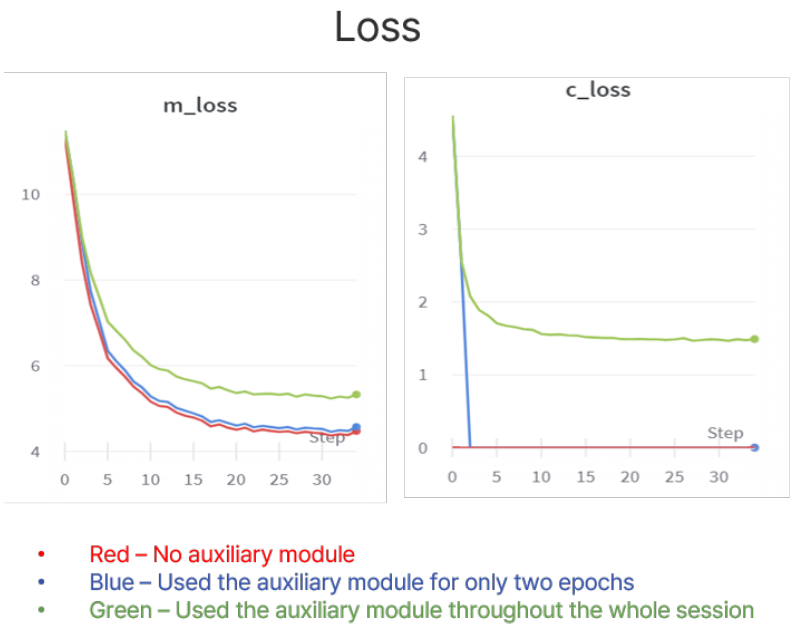

Figure 2. Graphs that show losses depending on the partial use of the module. Because the auxiliary module is disabled after two epochs, the classification loss is zero after the second epoch.

5.3. Miscellaneous Points to Cover

- The model performed better when the descriptors were fused after the descriptors were made into projections. It can be inferred that the process of concatenating the

descriptors and then computing projections resulted in lower performance because the descriptors are all mixed.

- One could argue that using several small patches of local features can allow for detailed comparison between images. However, doing so will take up vast amount of memory and computational time. If there are copious memory, power, and time, the performance definitely would be promising. In this case, the resources were limited by the time this thesis was written. Therefore, implementing several types of modules (not just the local ones) and partly exploiting local descriptors from the local module is the most realistic choice to make.
- The feature maps from Layer 2 are fed to the local module while the feature maps from Layer 3 are fed to the global module. Initially, the target model went a layer deeper and used Layers 3 and 4. Using Layers 3 and 4, which is how CGD [32] was implemented, resulted in lower performance (see Table 1). We assume it is because Layer 4 has a lower resolution than Layer 3, which could have made it more difficult for the model to extract accurate features. However, if the input had a larger image size and if we could have provided images with higher resolution so that the feature maps had better resolutions in deeper layers, it might have resulted in higher performances.

6. Conclusions

Ultimately, we propose a descriptor mixer model that incorporates both local and global descriptors as well as different types and combinations of descriptors. The combination of global descriptors have an effect similar to implementing an ensemble of models with a smaller size and fewer parameters for image retrieval. The overall model retrieves the images using the auxiliary module using the hyperparameter λ, the concatenation of descriptors, and the combination of several types of global descriptors. Empirically, the results show that when two global descriptors are used with a local descriptor with λ of 3, the performance of the model was generally the best. Modifying factors such as controlling the hyperparameter λ and changing the backbone model may further improve the performance depending on the dataset and other hyperparameters such as the optimizer and the loss function. This paper will hopefully provide some input on the effects of a descriptor mixer in image retrieval.

This paper does have some areas that need to be improved. One would be the lack of diversity in the dataset in the results section. While we have only used CUB-200-2011 [42] for our evaluation, it could have been better if we have obtained results using other datasets such as MS COCO [47] and CARS196 [48]. However, due to limited resources, we were not able to do so. Moreover, this paper explores the combinations of three global descriptors and uses one local branch that extracts one type of local descriptor. The descriptors we have used in this paper may not be the most ideal combinations and utilizing different kinds of descriptors outside the ones we have used could yield better performance.

Author Contributions: Conceptualization, Z.K. and Y.K.; methodology, Z.K. and Y.K.; software, Z.K. and Y.K.; validation, Z.K., Y.K. and Y.-J.S.; formal analysis, Z.K. and Y.K.; investigation, Z.K. and Y.K.; data curation, Z.K. and Y.K.; writing—original draft preparation, Z.K. and Y.K.; writing—review and editing, Z.K., Y.K. and Y.-J.S.; visualization, Y.K.; project administration, Y.-J.S.; funding acquisition, Y.-J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH), No. 2021-0-01096; Development of merging techniques of face animations and Korean/English voice audios for further development of conversational avatar programs); the Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korean government (MOTIE) (No. 20214810100010); and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2022R1A6A1A03052954).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: This model was trained using the Caltech-UCSD Birds-200-2011 (CUB-200-2011) Dataset by Wah et al. from Caltech Vision Lab [42]. The details of the dataset can be read from this link: https://www.vision.caltech.edu/datasets/cub_200_2011/, accessed on 7 January 2022.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CE	Cross entropy

References

1. Chen, W.; Liu, Y.; Wang, W.; Bakker, E.M.; Georgiou, T.; Fieguth, P.W.; Liu, L.; Lew, M.S. Deep Image Retrieval: A Survey. *arXiv* **2021**, arXiv:2101.11282.
2. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1097–1105.
4. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representation (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
5. 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 (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
6. Weinberger, K.Q.; Saul, L.K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
7. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, 20–26 June 2005; IEEE Computer Society: Washington, DC, USA, 2005; Volume 1, pp. 539–546. [CrossRef]
8. Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wang, J.; Philbin, J.; Chen, B.; Wu, Y. Learning Fine-Grained Image Similarity with Deep Ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 1386–1393.
9. Ouyang, X.; Zhou, T.; Vidal, R.; Dhua, A. SwinTransFuse: Fusing Swin and Multiscale Transformers for Fine-Grained Image Recognition and Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, 19–20 June 2022.
10. Patel, K.; Bur, A.M.; Li, F.; Wang, G. Aggregating Global Features into Local Vision Transformer. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR 2022), Montreal, QC, Canada, 21–25 August 2022; IEEE Computer Society: Los Alamitos, CA, USA, 2022; pp. 1141–1147. [CrossRef]
11. Li, G.; Yu, Y. Visual Saliency Detection Based on Multiscale Deep CNN Features. *IEEE Trans. Image Process.* **2016**, *25*, 5012–5024. [CrossRef] [PubMed]
12. Noh, H.; Araujo, A.; Sim, J.; Han, B. Image Retrieval with Deep Local Features and Attention-based Keypoints. *arXiv* **2017**, arXiv:1612.06321.
13. Wang, Y.; Yang, H.; Bai, X.; Qian, X.; Ma, L.; Lu, J.; Li, B.; Fan, X. PFAN++: Bi-Directional Image-Text Retrieval With Position Focused Attention Network. *IEEE Trans. Multimed.* **2021**, *23*, 3362–3376. [CrossRef]
14. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *arXiv* **2021**, arXiv:2104.02395. [CrossRef]
15. Kim, W.; Goyal, B.; Chawla, K.; Lee, J.; Kwon, K. Attention-Based Ensemble for Deep Metric Learning. In *Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11205, pp. 760–777.
16. Xuan, H.; Souvenir, R.; Pless, R. Deep Randomized Ensembles for Metric Learning. In *Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11220, pp. 751–762.
17. Zhai, H.; Lai, S.; Jin, H.; Qian, X.; Mei, T. Deep Transfer Hashing for Image Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 742–753. [CrossRef]
18. Jin, H.; Lai, S.; Zhao, G.; Qian, X. Hashing person re-ID with self-distilling smooth relaxation. *Neurocomputing* **2021**, *455*, 111–124. [CrossRef]
19. Revaud, J.; Weinzaepfel, P.; de Souza, C.R.; Humenberger, M. R2D2: Repeatable and Reliable Detector and Descriptor. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.

20. Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019), Long Beach, CA, USA, 16–20 June 2019.
21. Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; Han, B. Large-Scale Image Retrieval with Attentive Deep Local Features. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017. [CrossRef]
22. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
23. Li, Y.; Xu, Y.; Wang, J.; Miao, Z.; Zhang, Y. MS-RMAC: Multiscale Regional Maximum Activation of Convolutions for Image Retrieval. *IEEE Signal Process. Lett.* **2017**, *24*, 609–613. [CrossRef]
24. Babenko, A.; Lempitsky, V.S. Aggregating Local Deep Features for Image Retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015; pp. 1269–1277.
25. Radenovic, F.; Tolias, G.; Chum, O. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1655–1668. [CrossRef] [PubMed]
26. Cao, B.; Araujo, A.; Sim, J. Unifying Deep Local and Global Features for Image Search. In *Proceedings of the European Conference on Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12365, pp. 726–743.
27. Yang, M.; He, D.; Fan, M.; Shi, B.; Xue, X.; Li, F.; Ding, E.; Huang, J. DOLG: Single-Stage Image Retrieval With Deep Orthogonal Fusion of Local and Global Features. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021), Montreal, QC, Canada, 10–17 October 2021; pp. 11772–11781.
28. Song, Y.; Zhu, R.; Yang, M.; He, D. DALG: Deep Attentive Local and Global Modeling for Image Retrieval. *arXiv* **2022**, arXiv:2207.00287.
29. Alappat, A.L.; Nakhate, P.; Suman, S.; Chandurkar, A.; Pimpalkhute, V.; Jain, T. CBIR using Pre-Trained Neural Networks. *arXiv* **2021**, arXiv:2110.14455.
30. Song, C.; Han, H.; Avrithis, Y. All the attention you need: Global-local, spatial-channel attention for image retrieval. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV 2022), Waikoloa, HI, USA, 3–8 January 2022; IEEE Computer Society: Los Alamitos, CA, USA, 2022; pp. 439–448. [CrossRef]
31. Kim, Z.; Kim, Y.; Suh, Y.-J. Image Retrieval Using the Combination of Global and Local Descriptors. In Proceedings of the Symposium of the Korean Institute of Communications and Information Sciences (KICS 2022), Pyeongchang, Republic of Korea, 9–11 February 2022; pp. 85–86.
32. Jun, H.; Ko, B.; Kim, Y.; Kim, I.; Kim, J. Combination of Multiple Global Descriptors for Image Retrieval. *arXiv* **2019**, arXiv:1903.10663.
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015; pp. 1026–1034. [CrossRef]
35. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. In Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016.
36. Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Wachinton, DC, USA, 2016; pp. 2921–2929. [CrossRef]
37. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]
38. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122. [CrossRef]
39. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the 9th International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
40. Zhang, X.; Yu, F.X.; Karaman, S.; Zhang, W.; Chang, S.F. Heated-Up Softmax Embedding. *arXiv* **2018**, arXiv:1809.04157.
41. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
42. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*; Technical Report CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.
43. Kim, S.; Kim, D.; Cho, M.; Kwak, S. Proxy Anchor Loss for Deep Metric Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, 13–19 June 2020; pp. 3235–3244.
44. Ko, B.; Kim, H.G.; Heo, B.; Yun, S.; Chun, S.; Gu, G.; Kim, W. Group Generalized Mean Pooling for Vision Transformer. *arXiv* **2022**, arXiv:2212.04114.
45. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. *arXiv* **2019**, arXiv:1906.02243.

46. Wu, C.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Chang, G.; Behram, F.A.; Huang, J.; Bai, C.; et al. Sustainable AI: Environmental Implications, Challenges and Opportunities. *arXiv* **2021**, arXiv:2111.00364.
47. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
48. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, 1–8 December 2013.