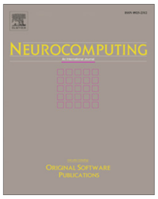
Neurocomputing 439 (2021) 374–382

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



The feature generator of hard negative samples for fine-grained image recognition

Taehung Kim, Kibeom Hong, Hyeran Byun *

Department of Computer Science, Yonsei University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 22 June 2020
Revised 27 August 2020
Accepted 13 October 2020
Available online 22 October 2020
Communicated by Z. Wang

Keywords:
Fine-grained image recognition
Metric learning
Hard negative sample
Feature generation
Deep neural networks

ABSTRACT

The key to solving the fine-grained image recognition is exploring more discriminative features for capturing tiny hints. In particular, the triplet objective function fits well with the fine-grained image recognition task because they capture the semantic similarity between images. However, triplet loss needs many pairs of tuples with hard negative samples, and it takes too much cost. To alleviate this problem, we propose a new framework that generates features of the hard negative samples. The proposed framework consists of three stages: learning part-wise features, enriching refined hard negative samples, and fine-grained image recognition. Our proposed method has achieved state-of-the-art performance in CUB-200-2011, Stanford Cars, FGVC-Aircraft, and DeepFashion datasets. Also, our extensive experiments demonstrate that each stage has a good effect on the final goal.

1. Introduction

Fine-grained image recognition is a more challenging task than coarse-level image recognition because the inter-class variance of the target dataset is very small. Many studies have tried to find a subtle difference between data to overcome this problem. Some works [1,2] proposed methods to localize important areas of images from the parts classifiers and use them for learning. Other works [3–5] used emphasized images from the attention mechanism for learning. Also, Chen et al. [6] improved performance by learning the network with the original image and shuffled image which has remained discriminative features even if the image shuffled like a puzzle. Cui et al. [7] leveraged humans in the loop during deep metric learning and obtained better tuples for training, and Tan et al. [8] and Yu et al. [9] developed a click prediction model beyond the employed user click frequency data, for improved image recognition. Wang et al. [10] proposed a feature embedding method using semantic similarity between images from patch clustering. Zhang et al. [11] and Li et al. [12] proposed a feature refining method and group-based learning method for extracting more representative features. Recently, Zhang et al. [13] has shown good recognition accuracy using joint learning for classification loss and triplet loss. These researches [5,14,15] have focused on both object and part simultaneously, not focusing only on the object or the part. There are various fusion types between object and part, such as feature-level [5,15] and classifier-level [14].

A triplet tuple consists of a reference sample, a positive sample, and a negative sample. Based on the tuple, triplet loss trains the model by clustering the data by adjusting the distance between the positive sample and negative sample based on the reference sample in the embedding space. Canévet et al. [16] defined a sample that is very similar to the reference image as a hard negative sample, which is classified as false positive among negative images. If we want to get more discriminative features through triplet loss, then choosing a hard negative sample is more important. Afterwards, there are hard negative samples mining methods [10,17] for fine-grained image recognition tasks.

In this paper, we propose a pipeline framework that generates features of hard negative samples (GHNS), instead of the conventional mining method from the given dataset. As shown in Fig. 1, the whole framework is composed of three stages: learning part-wise features, enriching refined hard negative samples, and final prediction. In the first stage, we train each part network by joint learning with classification, triplet, and the adversarial objective function to obtain discriminative features of three parts. Next, we generate several sets of candidates that are made of parts images by various combinations from the "parts shuffle module." and obtain the features of candidate sets by utilizing the trained part network in the first stage. In the next stage, a filtering network (FilterNet) filters out improper features of the generated candidate to

* Corresponding author.
E-mail address: hrbyun@yonsei.ac.kr (H. Byun).

https://doi.org/10.1016/j.neucom.2020.10.032
0925-2312/© 2020 Elsevier B.V. All rights reserved.

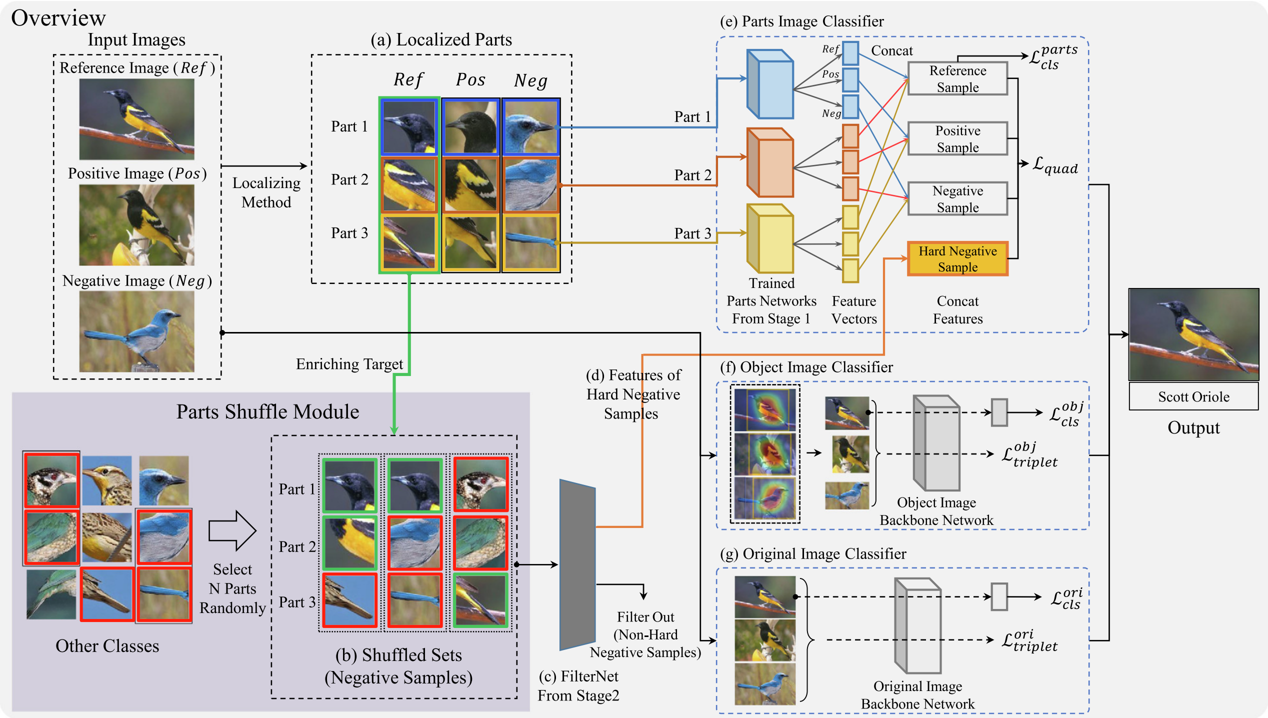

Overview
Input Images
Reference Image (Ref)
Positive Image (Pos)
Negative Image (Neg)
Localizing Method
(a) Localized Parts
Ref Pos Neg
Part 1
Part 2
Part 3
Enriching Target
Parts Shuffle Module
Other Classes
Select N Parts Randomly
(b) Shuffled Sets (Negative Samples)
(c) FilterNet From Stage2
Filter Out (Non-Hard Negative Samples)
(d) Features of Hard Negative Samples
(e) Parts Image Classifier
Ref
Pos
Neg
Concat
Reference Sample
Positive Sample
Negative Sample
Hard Negative Sample
$\mathcal{L}_{cls}^{parts}$
$\mathcal{L}_{quad}$
Trained Parts Networks From Stage 1
Feature Vectors
Concat Features
(f) Object Image Classifier
$\mathcal{L}_{cls}^{obj}$
$\mathcal{L}_{triplet}^{obj}$
Object Image Backbone Network
(g) Original Image Classifier
$\mathcal{L}_{cls}^{ori}$
$\mathcal{L}_{triplet}^{ori}$
Original Image Backbone Network
Scott Oriole
Output

Fig. 1. The overview of our proposed framework with stage 1, stage 2, and stage 3. First, we leverage parts localized data (a) from previous work to train parts networks. We make various combinations (b) of the target sample from step (a) and filter out improper samples using filterNet (c). We obtain hard negative samples (d) from the step (b) and (c). We add hard negative samples to train in parts image classifier (e) for better performance. Finally, we predict final score by three classifiers (e), (f), and (g).

obtain *harder* negative sample features. In the final prediction stage, the networks consist of part image classifier, object-level classifier, original image classifier. We obtain the final result from the part classifier with additional features of hard negative samples generated in the previous step. The proposed method can generate hard negative samples without limit of the number beyond the hard negative samples mining method in the given dataset.

We report state-of-the-art performance on four standard benchmark datasets (CUB-200-2011, FGVC-Aircraft, Stanford Cars, and DeepFashion), where our framework consistently outperforms the existing method. We also show the importance and necessity of each objective function in parts network, performance change for the number of hard negative samples for each class, and the importance of each classifier in the final prediction.

In Section 2, we briefly review related works on fine-grained image recognition and deep metric learning. In Section 3, we introduce our proposed method, "Generator of Hard Negative Samples." Experiments and result analyses are presented in Section 4, and finally, we conclude this paper in Section 5.

2. Related works

2.1. Fine-grained image recognition

There have been many studies to improve the performance of fine-grained image recognition. The method proposed by Lin et al. [18] is one of the most popular methods to train the relationship between the two feature vectors using the outer product. Previous methods have had huge computational costs, due to performing outer product operations, and subsequent studies have proposed novel methods to solve this problem. Lin et al. [19] and Zheng et al. [20] proposed additional matrix square-root normalization and a DBT block which uniformly divides the input channels according to semantic meaning, to reduce the computational cost of bilinear CNN. Tan et al. [21] also proposed a method that removes background noise using an aggregated slack mask. In addition, Li et al. [22] alleviated the computational burden of an improved CNN, by using sandwiching Newton-Schulz iteration.

CUB-200-2011 [23], the most representative dataset in fine-grained image recognition task, has additional information such as part annotation and label. Previous researches [24–27] directly used parts annotation to improve object recognition performance through classifiers for each part. However, labeling of image annotations is labor-intensive information because experts have to work on them manually. Also, the CUB-200-2011 dataset is the only annotated dataset. Thus many methods [28,29,25,30] proposed various ways to learn to find discriminative parts instead of using annotated data. They proposed the method to localize the discriminative parts using external segmentation information and detection module for finding each part. Some approaches [2,31,32] used the weakly-supervised method. Peng et al. [14] proposed novel spatial constraints between parts and an object for finding the region of the discriminative part. Zheng et al. [5] produced part attention from feature channels and trains the networks using attention information. Recently, Chen et al. [33] used additional puzzle images with adversarial loss because they preserved essential local information even if the image was shuffled. Our proposed method differs from previous works. We trained the models with joint optimizing by the triplet loss, classification loss, and adversarial loss, instead of using only classification loss. From the joint optimizing, we obtain well-trained backbone networks and generate discriminative parts features based on the backbone networks.

2.2. Deep metric learning

Chopra et al. [34], Hoffer et al. [35], and Chen et al. [36] proposed representative methods of deep metric learning and proposed pairwise, triplet, and quadruplet objective function, respectively. Some works have tried to use deep metric learning for fine-grained image recognition tasks. Zhang et al. [13] proposed quadruplet loss by using the coarse-level label in the data set and trained with joint optimizing by the triplet loss and classification loss, respectively. He has achieved better performance in a fine-grained image recognition task because triplet loss captures the semantic similarity between images. Also, Cui et al. [7] obtained hard negative samples by bootstrapping more training data from the web and showed good performance by adding the samples to the given training data. Sun et al. [3] proposed a novel multi-attention multi-class constraint with an excitation module [37] using the triplet loss in a given training data. Almost existing studies using triplets have found the hard negative samples in an only given set of data or from the web. In contrast, our proposed framework generates more discriminative features of hard negative samples without limit. Therefore, we improve the performance by generating more hard negative training samples from the proposed framework. Our proposed method also differs from the Zhang's method [13] that redefines the sample (p_i^-). We have proposed a quadruplet with a new composition, comprised of generated hard negative samples (n_i^+).

3. Proposed method

We propose a pipeline framework to train models of each part, generate features of hard negative samples with the parts network, and finally predict with fine-grained image recognition. Each stage is as follows.

3.1. Stage 1: Feature learning for each part

As shown in Fig. 2, parts networks consist of three networks. In typical coarse-grained image recognition, most previous studies widely used softmax. However, because it is challenging to capture subtle differences in the inter-class variance of the fine-grained image recognition task with the conventional softmax alone, we need additional similarity constraints (i.e., triplet loss, adversarial loss). Because triplet loss captures the semantic similarity among images, enforces a margin, and leads to tighter clustering of samples around the class, many studies used triplet loss. We also applied the triplet loss to the fine-grained image recognition task by using these characteristics. We use discriminative cropped images from the existing weakly-supervised learning method [38] to train the model as inputs. The triplet consists of three samples: reference sample r_i, the same class as r_i, positive sample p_i, the difference class as r_i, negative sample n_i. We expect that feature distance from r_i and n_i is larger than r_i and p_i by a certain margin $m > 0$ in embedding space. Triplet-driven networks generate a feature vectors $f(\cdot) \in \mathbb{R}^D$, where D is the embedding feature dimension. This constraint can be formulated as in the following:

$$\|f(r_i) - f(p_i)\|_2^2 + m \leqslant \|f(r_i) - f(n_i)\|_2^2. \tag{1}$$

The loss function for each part can be defined as:

$$L_{triplet}^{part} = \sum_{i=1} \frac{1}{2} max\left\{0, \|f(r_i) - f(p_i)\|_2^2 - \|f(r_i) - f(n_i)\|_2^2 + m\right\}. \tag{2}$$

We assume that all parts are of the same importance. And in stage 1, the triplet loss consists of the sum of the 3 parts triplet loss as follows:

$$L_{triplet} = L_{triplet}^{part1} + L_{triplet}^{part2} + L_{triplet}^{part3}. \tag{3}$$

Also, the classification loss consists of the sum of the 3 parts as follows:

$$L_{cls} = L_{cls}^{part1} + L_{cls}^{part2} + L_{cls}^{part3}. \tag{4}$$

Target dataset images vary widely because of various lighting and angles, even though they are of the same class. This large intra-class variance causes performance degradation in the fine-grained image recognition task. We want to solve this problem through adversarial loss. If we make networks learn to distinguish

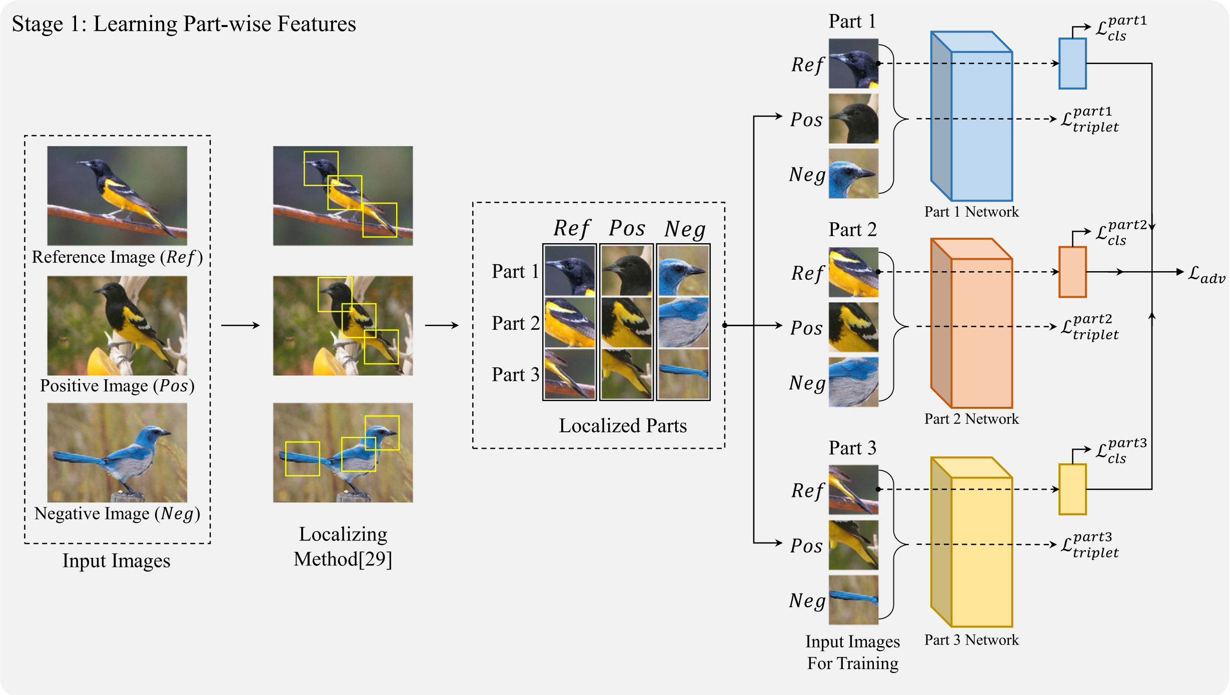


Fig. 2. Stage 1 includes three parts networks (i.e., head, body, tail) optimized by classification loss, triplet loss, and adversarial loss. Each loss requires different sets of image parts. Localized parts come from the previous work [38]. Each set consists of three images: reference sample, positive sample, and .negative sample.

between input images composed of the same class and input images included any other class, it will be helpful to solve the intra-class variance problem. For that reason, we incorporate the adversarial loss. Our research is the first attempt to jointly optimize three loss functions by incorporating the adversary loss.

Adversarial loss plays the role of distinguishing between input images composed of the same class and input images included any other class. When we training, we label each input image set as a one-hot vector $d \in \{0, 1\}^2$ in case unshuffled and shuffled, respectively. We train the networks to distinguish between three input images from the same image and another using the adversarial loss. We define the composition of the image set shuffled set I and unshuffled set $s(I)$, like $I = \left\{C^i_{part1}, C^i_{part2}, C^i_{part3}\right\}$ and $s(I) = \left\{C^i_{part1}, C^j_{part2}, C^k_{part3}\right\}$, respectively. We set which parts will be selected randomly in the shuffled set.

$$Set^{Shf}_{Img} = \left\{C^i_{part_1}, C^j_{part_2}, C^k_{part_3}\right\} \qquad s.t \begin{cases} i = j = k & Shf = 0. \\ otherwise & Shf = 1. \end{cases} \tag{5}$$

$$L_{adv} = -\sum d \cdot \log\left[D\left(Set^{Shf=0}_{Img}\right)\right] + (1 - d) \cdot \log\left[D\left(Set^{Shf=1}_{Img}\right)\right]. \tag{6}$$

For example, if all input images (i.e., head, wing, tail) composed of the same i^{th} image, the discriminator judges that the inputs are not shuffled $(d = 0)$. On the other hand, if some input images (i.e., head, wing) composed of the j^{th} image and other (i.e., tail) from the k^{th} image, the discriminator judges that some inputs are shuffled $(d = 1)$. D is a simple linear mapping network and discriminator for recognition which indicates whether there has been shuffled or not. In this stage, we finally define the following objective function:

$$L_{stage1} = \alpha L_{cls} + \beta L_{triplet} + \gamma L_{adv}, \tag{7}$$

where α, β, γ are weight parameters for each loss.

3.2. Stage 2: Enriching refined hard negative samples

Many studies using deep metric learning have tried in various ways to find hard negative samples. However, early studies did not improve much performance, and because they find hard negative samples within a limited given data, there are also fewer hard negative samples. We solve the problem through the enriching refined hard negative sample stage rather than the conventional mining method. Also, the proposed method can generate features of hard negative samples without restrictions on the volume of given data sets. This stage consists of two components: a "part shuffle module" and a "filtering module." The first component is to make various candidates for the hard negative sample. The second component is a filterNet to filter out the improper candidates among the various candidate generated earlier for additional training data in stage 3.

1) Parts shuffle module: In this module, we make many hard negative sample candidates. As shown in Fig. 3, we leverage three cropped parts from the previous research [5] for each reference image in "class A." We denote the three cropped images as a set. If we replace some parts with the image parts of the other classes, the candidate cannot be considered corresponding to the pure "class A." However, the candidate can be considered as hard negative samples. For example, we obtain a set similar to March Wren's ('class A') by replacing the wing part from a set of images corresponding to the March Wren's class with the wing of Canyon Wren's.

We generate many candidates in the following steps. First, we decide how many of the three parts to replace for making candidates. The hard negative sample should have only subtle differences to the extent that it is misclassified as false positive. Therefore, we replace only one or two parts in a set consisting of three parts. Second, we decide which parts to replace and which classes of images other than the reference image to use. We define the combination of the set per reference image as follows:

$$\begin{aligned} Sets = \Big\{&(C^i_P, C^j_Q, C^k_R) \mid \forall i \in P, \forall j \in Q, \forall k \in R \\ &s.t\ P = Q \neq R, P \neq Q = R, P = R \neq Q, P \neq R\ and \\ &\quad Q \neq R, P \neq Q\ and\ Q \neq R, P \neq Q\ and\ P \neq R\Big\}. \end{aligned}$$

where C is cropped image. P, Q, R indicate class, i, j, k are the image numbers in the class.

2) Filtering module: In the parts shuffle module, we replace some parts with other parts to generate candidates having subtle differences. However, candidates generated from "part shuffle modules" do not correspond to the definition of hard negative samples. It is just a random shuffled set of negative samples. Therefore, we design an additional filtering module to obtain more significant hard negative samples that correspond to the definition. The filtering module is a network that recognizes the image set when we feed-forward the image set as input. If the shuffle module generates many candidates from a reference image and the filtering network classifies some candidates as "class A," the same class of reference images, we can regard them as hard negative samples. Moreover, if we select a small number of parts to replace in the shuffle module and replace them with images much similar to the reference image, the generated candidates are less filtered by filterNet.

FilterNet is based on the trained parts networks in stage 1. We freeze feature extractor except for the last FC layers in stage 1. And we call the FC layers as filterNet, which are trained by the unshuffled image sets with the labels. As Fig. 3 shows, we feed-forward the set of candidates to filterNet, leaving only the features of candidates classified as "class A" with a score of τ or higher. We can use those features for additional training data in stage 3. The more well-trained parts network from stage 1, get better candidates and features of hard negative samples in stage 2. And better hard negative samples, we obtain more improve performance in the final prediction stage.

3.3. Stage 3: Final prediction

Stage 3 is the final step to recognize images. For better recognition performance, we focus on various types of images (parts image, object image, original image). Each type has different impacts and strengths for aiding recognition. As shown in step (e), (f), and (g) of Fig. 1, the final prediction consists of three networks: a parts image classifier, an object image classifier, and the original image classifier. We will describe each classifier.

Parts image classifier: Zhang et al. [13] proposed a quadruplet (r_i, p^+_i, p^-_i, n_i), where positive sample (p^-_i) has the same coarse class as r_i in the only given dataset. However, our proposed method is different from Zhang's method. We design a framework to *generate* hard negative samples features, which play a more important role than the positive samples in deep metric learning. To supply many features to the parts image classifier, we leverage a set of candidates filtered by the filtering module in stage 2. Because the combinations of the "image parts" generate features, we can only supply the additional features to the "parts image classifier." To use the features of the hard negative samples, we propose a quadruplet with a new composition, comprised of hard negative samples (n^+_i) in stage 3. Each quadruplet, (r_i, p_i, n^+_i, n_i), consists of four features. Reference, positive, negative features are obtained by feed-forwarding with the parts networks in stage 1. We select a feature of the hard negative sample corresponding to the refer-

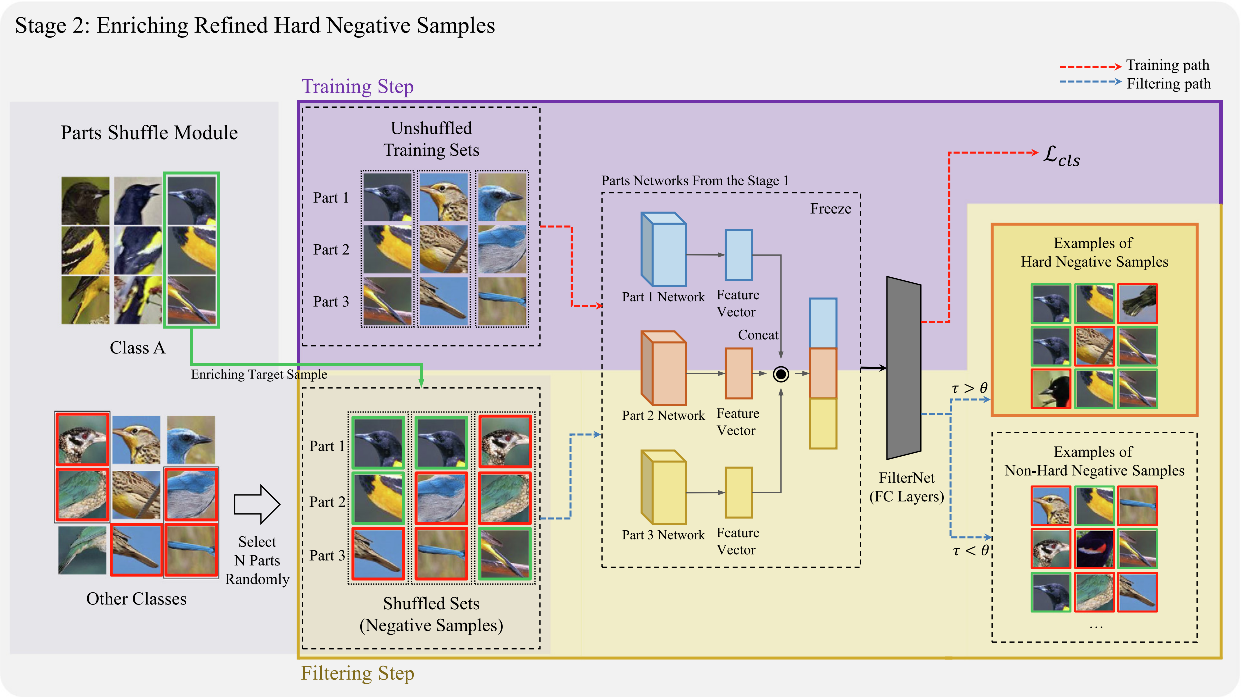


Fig. 3. Stage 2 composes two parts, "Parts shuffle module" and "FilterNet." Parts shuffle module makes set of image parts candidates. FilterNet plays a role in filtering features of improper hard negative samples. FilterNet is based on parts networks from stage 1.

ence image in stage 2. Given a quadruplet, the relation among the four features can be defined in two inequalities.

$$\|f(r_i)-f(p_i)\|_2^2+m_1 \leqslant \left\|f(r_i)-f(n_i^+)\right\|_2^2+m_2 \leqslant \|f(r_i)-f(n_i)\|_2^2, \qquad (8)$$

where m_1 and m_2 are the values of the margin in two terms. It is satisfying $m_1 > m_2 > 0, f_t(n^+)$ is a feature of the hard negative sample from stage 2. Compared to a single triplet, the quadruplet makes a model much richer relationships among the features. We propose to decompose Eq. (8) into two triplets, (r_i, p_i, n_i^+) and (r_i, n_i^+, n_i). Similar to Eq. 2, we can define quadruplet loss as follows:

$$L_{quad} = \sum_{i=1} \frac{1}{2} max\left\{0, \|f(r_i)-f(p_i)\|_2^2 - \left\|f(r_i)-f(n_i^+)\right\|_2^2 + m_1 - m_2\right\} + \sum_{i=1} \frac{1}{2} max\left\{0, \left\|f(r_i)-f(n_i^+)\right\|_2^2 - \|f(r_i)-f(n_i)\|_2^2 + m_2\right\}. \qquad (9)$$

The parts image classifier jointly learns the quadruplet and classification loss as follows:

$$L_{parts} = \lambda_p L_{cls}^{parts} + (1-\lambda_p) L_{quad}^{parts}. \qquad (10)$$

Original image classifier: The original image classification network only uses the original image without cropping or any other pre-processing. The network is trained by optimizing the classification and the triplet loss jointly.

Object image classifier: We use a CAM [38] to localize the object areas in images, to avoid using the bounding box information. We use localized images to train network classification and triple loss. Finally, we merge each prediction result of the original, object, and parts image classification using the following equation:

$$final\ score = a \cdot original\,score + b \cdot object\,score + c \cdot parts\,score. \qquad (11)$$

We select a, b and c values by experiments.

4. Experiments

4.1. Datasets

CUB-200-2011 [23] CUB-200-2011 is the most widely used dataset in fine-grained recognition. The dataset is about twice as large as the CUB-200 image volume. The number of images in the dataset is 11,788 of 200 different subcategories. The dataset is divided into 5,994 images for training and 5,794 images for testing. The annotation of the dataset includes the bounding box information of the object, and 312 attribute information of the characteristics of a bird (i.e., the color of the wing, the length of the peak).

Stanford Cars [39] Stanford Cars contains 16,185 images of the car. It is divided into 8,144 for training and 8,041 for testing. Each subcategory is consists of 24–84 images for training and 24–83 images for testing. It has one bounding box and label.

FGVC-Aircraft [40] FGVC-Aircraft is an aircraft dataset of 102 subcategories. The dataset consists of 10,200 images and is equally divided into training, test, and validation. Each subset contains 33–34 images. All images are annotated with the model, variant, family, and manufacturer information.

DeepFashion [41] DeepFashion dataset is used for image retrieval, detection, and recognition. We experimented with the first set of the DeepFashion dataset, the "Category and attribute Prediction Benchmark" data set. This dataset consists of 289,222 images and 50 subcategories. The entire images are annotated by a bounding box and clothing type information.

4.2. Implementation

All experiments in this paper were trained with and tested using a PC with 192GM RAM, Cascade Lake 24C processors @ 2.5 GHz, and 4× T4 NVIDIA GPUs. For a fair comparison, we design the framework based on ResNet-50 and VGG-16. We set the SGD optimizer, and the initial learning rate is set to 0.05, which decays by 0.09 for every 40 epochs. In stage 1, we set $\alpha = 1$, $\beta = 0.8$, $\gamma = 1$ for all experiments reported in this paper, except DeepFashion (1, 1, 0.5). The τ value is set to 0.95. As we mentioned, when recognizing the image, we have to focus both object and part simultaneously. We traverse the weight parameters (a, b, c) from 0 to 1 by 0.05 step. We select the parameters that show the highest recognition performance. For the highest results in stage 3, we set (a, b, c) as (0.3, 0.3, 0.4), (0.5, 0.2, 0.3), (0.4, 0.4, 0.2), (0.35, 0.3, 0.35) for CUB-200-2011, Stanford Cars, FGVC-Aircraft, DeepFashion, respectively. Since the classification loss has more information than a quadruplet, we set each weight as follows: λ_p, λ_{ori}, λ_{obj} = 0.75, 0.8, 0.8. We fine-tuned a model pre-trained using ImageNet [42] in stage 1. In the training phase, we trained backbone x3 and x5 networks in stage1 and stage 3, respectively. We trained the stage models using 300, 200, and 200 epochs, respectively. In the testing phase, the inference time was 384 ms for the ResNet-50 backbone network, and 440 ms for the VGG-16 backbone network.

4.3. Comparison with State-of-the-Art

The "Backbone" column denotes which CNN model was used as the backbone network. The result in fine-grained image recognition with CUB-200-2011, Stanford Cars, FGVC-Aircraft, and DeepFashion dataset is described in Tables 1–4, respectively. The columns in each table are the method, the "Backbone," and its accuracy. For a fair comparison, we compared with studies using ResNet-50 and VGG-16, as in other works. In stage 3, we set the final prediction setting based on the OPAM [14] experiment setting that fusion of three classifiers. Also, all the results are fairly obtained without external information such as a bounding box or annotation. Compared to MGE-CNN [48], which consists of several experts and a gating network, our GHNS framework outperforms it by 0.56% in the same ResNet-50 network. We confirmed that the proposed method was 0.96% higher than the second-highest result, ISQRT-COV [22]. The results show state-of-the-art performance not only CUB-200-211 dataset but also Stanford Cars, FGVC-Aircraft, DeepFashion datasets. Each dataset result shows 1.08%, 1.4%, and 6.22% higher than the second-highest result, respectively, in

Table 1
Comparison of our approach (GHNS) to recent results on CUB-200-2011.

Method	Backbone	Acc(%)
Bilinear-CNN [18]	VGGNet	84.1
RA-CNN [15]	VGG-19	85.3
Improved B-CNN [19]	VGG-16	85.8
OPAM [14]	VGG-16	85.83
Kernel-Pooling [43]	VGG-16	86.2
Refined-CNN [11]	VGG-16	86.4
MA-CNN [5]	VGG-19	86.5
DFL-CNN(2-scale) [44]	VGG-16	86.7
DCL [6]	VGG-16	86.9
iSQRT-COV [22]	VGG-16	87.2
GSFL-Net ([43] based) [12]	VGG-16	87.60
GHNS(Ours)	VGG-16	88.42
Kernel-Pooling [43]	ResNet-50	84.7
MAMC [17]	ResNet-101	86.5
HBPASM [21]	ResNet-34	86.8
DFL-CNN(1-scale) [44]	ResNet-50	87.4
DBTNet-50 [20]	ResNet-50	87.5
Cross-X [45]	ResNet-50	87.7
DCL [6]	ResNet-50	87.8
TASN [46]	ResNet-50	87.9
iSQRT-COV [22]	ResNet-50	88.1
S3N [47]	ResNet-50	88.5
MGE-CNN [48]	ResNet-50	88.5
MGE-CNN [48]	ResNet-101	89.4
GHNS(Ours)	ResNet-50	89.06

Table 2
Comparison of our approach (GHNS) to recent results on Stanford Cars.

Method	Backbone	Acc (%)
Bilinear-CNN [18]	VGGNet	91.3
Improved B-CNN [19]	VGG-16	92.0
OPAM [14]	VGG-16	92.19
Kernel-Pooling [43]	VGG-16	92.4
Refined-CNN [11]	VGG-16	92.4
RA-CNN [15]	VGG-19	92.5
iSQRT-COV [22]	VGG-16	92.5
MA-CNN [5]	VGG-19	92.8
TASN [46]	VGG-19	93.2
DFL-CNN(2-scale) [44]	VGG-16	93.8
GSFL-Net ([43] based) [12]	VGG-16	93.92
DCL [6]	VGG-16	94.1
GHNS(Ours)	VGG-16	94.54
iSQRT-COV [22]	ResNet-50	92.8
MAMC [17]	ResNet-101	93.0
DFL-CNN(1-scale) [44]	ResNet-50	93.1
MGE-CNN [48]	ResNet-101	93.6
TASN [46]	ResNet-50	93.8
HBPASM [21]	ResNet-34	93.8
MaxEnt [49]	ResNet-50	93.85
MGE-CNN [48]	ResNet-50	93.9
DBTNet-50 [20]	ResNet-50	94.1
DCL [6]	ResNet-50	94.5
Cross-X [45]	ResNet-50	94.6
GHNS(Ours)	ResNet-50	95.68

Table 3
Comparison of our approach (GHNS) to recent results on FGVC-Aircraft.

Method	Backbone	Acc(%)
Bilinear-CNN [18]	VGGNet	84.1
Kernel-Pooling [43]	VGG-16	86.9
Refined-CNN [11]	VGG-16	87.7
RA-CNN [15]	VGG-19	88.2
HIHCA [50]	VGG-16	88.3
Improved B-CNN [19]	VGG-16	88.5
GSFL-Net ([18] based) [12]	VGG-16	89.26
MA-CNN [5]	VGG-19	89.9
iSQRT-COV [22]	VGG-16	90.0
DFL-CNN(1-scale) [44]	VGG-16	91.1
DCL [6]	VGG-16	91.2
DFL-CNN(2-scale) [44]	VGG-16	92.0
GHNS(Ours)	VGG-16	93.0
Kernel-Pooling [43]	ResNet-50	85.7
iSQRT-COV [22]	ResNet-50	90.0
DBTNet-50 [20]	ResNet-50	91.2
HBPASM [21]	ResNet-34	91.3
DFL-CNN(1-scale) [44]	ResNet-50	91.7
Cross-X [45]	SENet-50	92.7
S3N [47]	ResNet-50	92.8
DCL [6]	ResNet-50	93.0
GHNS(Ours)	ResNet-50	94.40

Table 4
Comparison of our approach (GHNS) to recent results on Deepfashion.

Method	Backbone	Top3(%)	Top5(%)
WTBI [51]	Custom	43.73	66.26
DARN [52]	Custom	59.48	79.58
FashionNet [41]	Custom	82.58	90.17
FAFS [53]	VGG-16	86.72	92.51
AFGN [54]	VGG-16	90.99	95.78
FAN [55]	VGG-16	91.16	96.12
GHNS(Ours)	VGG-16	91.33	96.72
LWAD [56]	ResNet-50	86.30	92.8
GHNS(Ours)	ResNet-50	92.52	96.88

ResNet-50. VGG-16 results are 0.44%, 1.0%, and 0.17% higher than the second-highest result, respectively.

4.4. Ablation study

The effect of objective functions We studied how each stage affected the overall performance. We designed a variety of experiments and performed evaluations on the CUB-200-2011 dataset. We obtain networks that distinguish subtle differences because the networks use three parts as input and judge whether the images are mixed. Also,we train each part network through triplet loss that makes the reference sample a closer positive sample than the negative sample in embedding space. We confirm that triplet loss performs better using only classification loss. The results are shown in Table 5. In stage 1, the triplet loss and adversarial loss respectively show performance improvements of 1.31% and 1.74% over the classification loss. The result shows that the adversarial loss is more important than triplet loss in stage 1. We can confirm that each classification loss, triplet loss, and adversarial loss affects the overall performance of stage 1.

The effect of the number of additional hard negative samples In stage 2, we set the number of features to 10–250. As shown in Fig. 4, the result shows that increasing the number of features boots the performance significantly all datasets. The best result is achieved with 150 additional hard negative samples in the CUB-200-2011 and DeepFashion datasets. On the other hand, for the FGVC-Aircraft and Stanford Cars dataset, the proposed method achieves the best result when we set the number of features to 200. Datasets with greater numbers of initial training samples (such as DeepFashion) require more features of hard negative samples to boost performance. The performance improvements become less prominent when we use more than 250 additional hard negative samples.

The importance of each classifier In stage 3, we design an ablation study about the importance of each classifier. We experiment with the original image classifier, the object image classifier, and the part image classifier. As shown in Table 6, the result of each classifier is 83.52%, 84.65%, and 84.82%, respectively. The part image classifier has the optimal result among the classifiers. Focusing on the discriminative parts and additional features boosts the performance of the part image classifier. Therefore, part image classifier has the most important role in the final prediction. As shown in Table 6, we combine two classifiers to achieve more accurate results than one classifier. There are three combinations: “original image classifier + object image classifier,” “object image classifier + part image classifier,” and “original image classifier + part image classifier.” It makes the better result at least 3.88%. We observe that the “original image classifier + part image classifier” has a better result than the “object image classifier + part image classifier.” The result shows that all classifiers are important, and we use them all by the fusion of classifiers. Totally, we experiment with three classifiers: the original images classifier, the object images classifier, and the parts images classifier, which show the best performance when we train them together.

Table 5
Ablation performance on each loss function on CUB-200-2011 in stage 1.

Method	Acc(%)
Classification	86.24
Classification + Triplet	87.55
Classification + Adversarial	87.98
Classification + Triplet + Adversarial	89.06

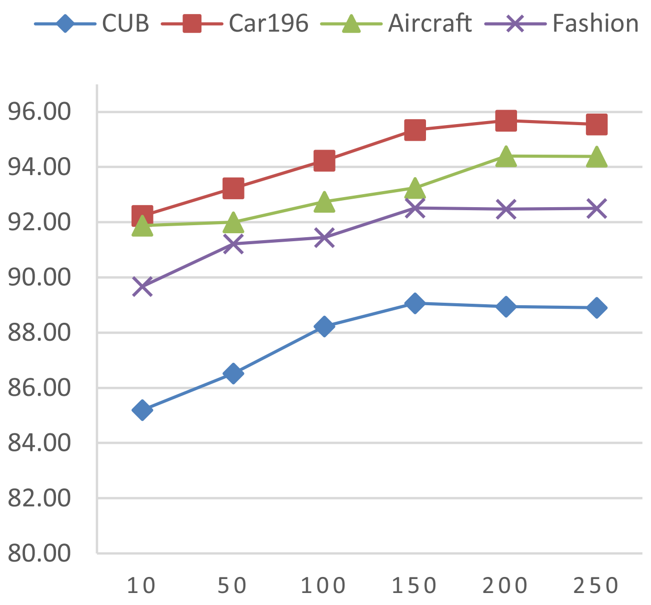


Fig. 4. Ablation analysis of the number of additional hard negative samples on the four datasets.

Table 6
Performance of components on CUB-200-2011 in stage 3.

Method	Acc(%)
Original	83.52
Object-level	84.65
Object-level(w/BB)	85.21
Part-level	84.82
Part-level (OPAM [14])	80.65
Original + Object-level	88.70
Original + Part-level	88.81
Object + Part-level	88.75
Original + Object-level(w/BB)+Part-level	89.38
Original + Object-level + Part-level	89.06

4.5. Discussions

We judge that our framework has a double-edged sword effect. As we mentioned, we exploit an existing method [5]. The results are influenced by the number of parts and the parts localizing quality from the existing method. The number of parts affects the number of parts networks in stage 1, causing the difference in the computing powers during training. If we leverage better part localizing methods, our framework can show a better performance.

In stage 3, it is possible to enhance the performance by improving the original classifier and the object classifier. As shown in Fig. 5, some failure cases (i.e., occlusion) exist in the object localization method [38]. Localizing failures cause decreasing performance in the object classifier and final result. As shown in Table 6, we have conducted additional experiments on the CUB-200-2011 dataset, using bounding box (BB) information instead of CAM [38]. Since there are no localizing failures, the results of “object-level” and “original + object-level (with bounding box) + part-level” improved by 0.56% and 0.32%, respectively. The results give the upper bounds of the performance. If we select better object localizing methods than CAM [38] in our framework, we could improve the performance.

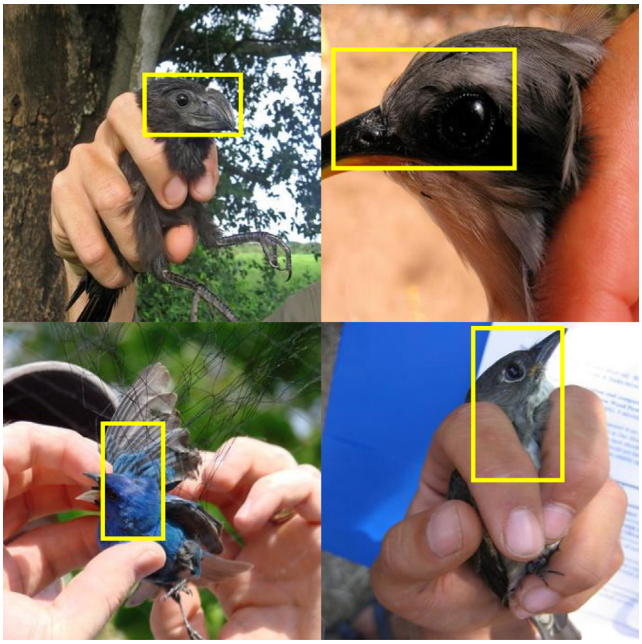


Fig. 5. There are some failure cases of object localizing in CUB-200-2011.

5. Conclusion

In this paper, we propose a novel GHNS framework for fine-grained image recognition. The first stage of the proposed framework is training the model for the well-feature embedding of each part using three loss functions. In the second stage, the parts generation module makes candidates of hard negative samples and obtains features of hard negative samples by the filtering module. In the final stage, we add the generated features to the quadruplet. We show state-of-the-art performance using an original image, an object image, and parts image classifiers together. Extensive experiments in the CUB-200-2011, Stanford Cars, FGVC-Aircraft, and DeepFashion dataset show outperform the other methods. Also, we verify the necessity of each stage and loss from the ablation study.

CRediT authorship contribution statement

Taehung Kim: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Kibeom Hong:** Validation, Data curation, Resources, Investigation. **Hyeran Byun:** Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C2003760) and Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01361, Artificial Intelligence Graduate School Program (YONSEI UNIVERSITY)).

References

[1] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A.M. Elgammal, D.N. Metaxas, Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1143–1152.

[2] X. Zhang, H. Xiong, W. Zhou, W. Lin, Q. Tian, Picking deep filter responses for fine-grained image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1134–1142..

[3] M. Sun, Y. Yuan, F. Zhou, E. Ding, Multi-attention multi-class constraint for fine-grained image recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[4] P. Sermanet, A. Frome, E. Real, Attention for fine-grained categorization, CoRR abs/1412.7054 (2014)..

[5] H. Zheng, J. Fu, T. Mei, J. Luo, Learning multi-attention convolutional neural network for fine-grained image recognition, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5219–5227.

[6] Y. Chen, Y. Bai, W. Zhang, T. Mei, Destruction and construction learning for fine-grained image recognition, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5152–5161..

[7] Y. Cui, F. Zhou, Y. Lin, S.J. Belongie, Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1153–1162.

[8] M. Tan, J. Yu, H. Zhang, Y. Rui, D. Tao, Image recognition by predicted user click feature with multidomain multitask transfer deep network, IEEE Transactions on Image Processing 28 (2019) 6047–6062, https://doi.org/10.1109/TIP.2019.2921861.

[9] J. Yu, M. Tan, H. Zhang, D. Tao, Y. Rui, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1, 1.

[10] Y. Wang, J. Choi, V.I. Morariu, L.S. Davis, Mining discriminative triplets of patches for fine-grained classification, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 1163–1172. URL: https://doi.org/10.1109/CVPR.2016.131. doi: 10.1109/CVPR.2016.131..

[11] W. Zhang, J. Yan, W. Shi, T. Feng, D. Deng, Refining deep convolutional features for improving fine-grained image recognition, EURASIP Journal of Image and Video Processing 2017 (2017) 27. URL: https://doi.org/10.1186/s13640-017-0176-3. doi: 10.1186/s13640-017-0176-3..

[12] X. Li, V. Monga, Group based deep shared feature learning for fine-grained image classification, in: 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9–12, 2019, BMVA Press, 2019, p. 143. URL: https://bmvc2019.org/wp-content/uploads/papers/0885-paper.pdf..

[13] X. Zhang, F. Zhou, Y. Lin, S. Zhang, Embedding label structures for fine-grained feature representation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1114–1123.

[14] Y. Peng, X. He, J. Zhao, Object-part attention model for fine-grained image classification, IEEE Transactions on Image Processing 27 (2018) 1487–1500.

[15] J. Fu, H. Zheng, T. Mei, Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4476–4484.

[16] O. Canévet, Object detection with active sample harvesting, 2017..

[17] M. Sun, Y. Yuan, F. Zhou, E. Ding, Multi-attention multi-class constraint for fine-grained image recognition, in: ECCV, 2018..

[18] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear cnn models for fine-grained visual recognition, in: 2015 IEEE International Conference on Computer Vision (ICCV) , 2015, pp. 1449–1457.

[19] T. Lin, S. Maji, Improved bilinear pooling with cnns, in: British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017, BMVA Press, 2017. URL:https://www.dropbox.com/s/fc6qtzvno7ln684/0395.pdf?dl=1..

[20] H. Zheng, J. Fu, Z. Zha, J. Luo, Learning deep bilinear transformation for fine-grained image representation, in: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada, 2019, pp. 4279–4288. URL: http://papers.nips.cc/paper/8680-learning-deep-bilinear-transformation-for-fine-grained-image-representation..

[21] M. Tan, G. Wang, J. Zhou, Z. Peng, M. Zheng, Fine-grained classification via hierarchical bilinear pooling with aggregated slack mask, IEEE Access 7 (2019) 117944–117953. URL: https://doi.org/10.1109/ACCESS.2019.2936118. doi: 10.1109/ACCESS.2019.2936118..

[22] P. Li, J. Xie, Q. Wang, Z. Gao, Towards faster training of global covariance pooling networks by iterative matrix square root normalization, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, IEEE Computer Society, 2018, pp. 947–955. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Li_Towards_Faster_Training_CVPR_2018_paper.html. doi: 10.1109/CVPR.2018.00105..

[23] C. Wah, S. Branson, P. Welinder, P. Perona, S.J. Belongie, The caltech-ucsd birds-200-2011 dataset, 2011..

[24] R. Farrell, O. Oza, N. Zhang, V.I. Morariu, T. Darrell, L. S. Davis, Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance, in: D.N. Metaxas, L. Quan, A. Sanfeliu, L.V. Gool (Eds.), IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain,

November 6–13, 2011, IEEE Computer Society, 2011, pp. 161–168. URL: https://doi.org/10.1109/ICCV.2011.6126238. doi: 10.1109/ICCV.2011.6126238..

[25] S. Branson, G.V. Horn, S.J. Belongie, P. Perona, Bird species categorization using pose normalized deep convolutional nets, CoRR abs/1406.2952 (2014). URL: http://arxiv.org/abs/1406.2952. arXiv:1406.2952..

[26] N. Zhang, J. Donahue, R.B. Girshick, T. Darrell, Part-based r-cnns for fine-grained category detection, in: D.J. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision - ECCV 2014–13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I, volume 8689 of Lecture Notes in Computer Science, Springer, 2014, pp. 834–849. URL: https://doi.org/10.1007/978-3-319-10590-1_54. doi: 10.1007/978-3-319-10590-1_54..

[27] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L.D. Bourdev, PANDA: pose aligned networks for deep attribute modeling, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014, IEEE Computer Society, 2014, pp. 1637–1644. URL: https://doi.org/10.1109/CVPR.2014.212. doi: 10.1109/CVPR.2014.212..

[28] T. Berg, P.N. Belhumeur, Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 955–962.

[29] L. Xie, Q. Tian, R. Hong, S. Yan, B. Zhang, Hierarchical part matching for fine-grained visual categorization, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 1641–1648.

[30] N. Zhang, J. Donahue, R.B. Girshick, T. Darrell, Part-based r-cnns for fine-grained category detection, in: ECCV, 2014..

[31] M. Simon, E. Rodner, Neural activation constellations: unsupervised part model discovery with convolutional networks, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1143–1151.

[32] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.A. Nguyen, M.N. Do, Weakly supervised fine-grained categorization with part-based image representation, IEEE Transactions on Image Processing 25 (2016) 1713–1725.

[33] Y. Chen, Y. Bai, W. Zhang, T. Mei, Destruction and construction learning for fine-grained image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, Computer Vision Foundation/ IEEE, 2019, pp. 5157–5166. URL:http://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Destruction_and_Construction_Learning_for_Fine-Grained_Image_Recognition_CVPR_2019_paper.html. doi: 10.1109/CVPR.2019.00530..

[34] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, 2005, pp. 539–546..

[35] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: SIMBAD, 2014..

[36] W. Chen, X. Chen, J. Zhang, K. Huang, Beyond triplet loss: A deep quadruplet network for person re-identification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1320–1329.

[37] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, IEEE Computer Society, 2018, pp. 7132–7141. URL:http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html. doi: 10.1109/CVPR.2018.00745..

[38] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2921–2929.

[39] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: 2013 IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.

[40] S. Maji, E. Rahtu, J. Kannala, M.B. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, ArXiv abs/1306.5151 (2013)..

[41] Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1096–1104.

[42] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: P.L. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3–6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks..

[43] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, S.J. Belongie, Kernel pooling for convolutional neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3049–3058.

[44] Y. Wang, V.I. Morariu, L.S. Davis, Learning a discriminative filter bank within a cnn for fine-grained recognition, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2016) 4148–4157.

[45] W. Luo, X. Yang, X. Mo, Y. Lu, L.S. Davis, J. Li, J. Yang, S.-N. Lim, Cross-x learning for fine-grained visual categorization, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8241–8250.

[46] H. Zheng, J. Fu, Z.-J. Zha, J. Luo, Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5007–5016.

[47] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, J. Jiao, Selective sparse sampling for fine-grained image recognition, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6598–6607.

[48] L. Zhang, S. Huang, W. Liu, D. Tao, Learning a mixture of granularity-specific experts for fine-grained categorization, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8330–8339.

[49] A. Dubey, O. Gupta, R. Raskar, N. Naik, Maximum-entropy fine-grained classification, in: NeurIPS, 2018..

[50] S. Cai, W. Zuo, L. Zhang, Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 511–520.

[51] H. Chen, A.C. Gallagher, B. Girod, Describing clothing by semantic attributes, in: ECCV, 2012..

[52] J. Huang, R.S. Feris, Q. Chen, S. Yan, Cross-domain image retrieval with a dual attribute-aware ranking network, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1062–1070.

[53] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, R.S. Feris, Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1131–1140.

[54] W. Wang, Y. Xu, J. Shen, S.-C. Zhu, Attentive fashion grammar network for fashion landmark detection and clothing category classification, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4271–4280.

[55] J. Liu, H. Lu, Deep fashion analysis with feature map upsampling and landmark-driven attention, in: ECCV Workshops, 2018..

[56] C. Corbière, H. Ben-younes, A. Ramé, C. Ollion, Leveraging weakly annotated data for fashion image retrieval and label prediction, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 2268–2274.

Taehung Kim received the B.S. degree from Korea Aerospace University, Seoul, Korea, in 2012. He is currently a Ph.D. student at Yonsei University. His research focuses on fine-grained image recognition, object detection and deep neural network.

Kibeom Hong is currently a Ph.D. student in Computer Science at Yonsei University, Seoul, Korea. He received the B.S. degree in Computer Science from Yonsei University, Seoul, Korea. His research interests include generative adversarial networks, generative models, and neural style transfer.

Hyeran Byun received the B.S. and M.S. degrees in mathematics from Yonsei University, Seoul, Korea, and the Ph.D. degree in computer science from Purdue University, West Lafayette, IN, USA. She is currently a Professor of Computer Science at Yonsei University. Her research interests include computer vision, artificial intelligence, and pattern recognition.