Dynamic Position-aware Network for Fine-grained Image Recognition

Shijie Wang[1, 2], **Haojie Li**[1, 2*], **Zhihui Wang**[1, 2], **Wanli Ouyang**[3]
[1]International School of Information Science & Engineering, Dalian University of Technology, China
[2]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China
[3] The University of Sydney, SenseTime Computer Vision Research Group, Australia
1027@mail.dlut.edu.cn, {hjli, zhwang}@dlut.edu.cn, wanli.ouyang@sydney.edu.au

Abstract

Most weakly supervised fine-grained image recognition (WFGIR) approaches predominantly focus on learning the discriminative details which contain the visual variances and position clues. The position clues can be ***indirectly*** learnt by utilizing context information of discriminative visual content. However, this will cause the selected discriminative regions containing some non-discriminative information introduced by the position clues. These analysis motivates us to ***directly*** introduce position clues into visual content to only focus on the visual variances, achieving more precise discriminative region localization. Though important, position modelling usually requires significant pixel/region annotations and therefore is labor-intensive. To address this issue, we propose an end-to-end Dynamic Position-aware Network (DP-Net) to directly incorporate the position clues into visual content and dynamically align them without extra annotations, which eliminates the effect of position information for discriminative variances among subcategories. In particular, the DP-Net consists of: 1) Position Encoding Module, which learns a set of position-aware parts by directly adding the learnable position information into the horizontal/vertical visual content of images; 2) Position-vision Aligning Module, which dynamically aligns both visual content and learnable position information via performing graph convolution on position-aware parts; 3) Position-vision Reorganization Module, which projects the aligned position clues and visual content into the Euclidean space to construct a position-aware feature maps. Finally, the position-aware feature maps are used which is implicitly applied the aligned visual content and position clues for more accurate discriminative regions localization. Extensive experiments verify that DP-Net yields the best performance under the same settings with most competitive approaches, on CUB Bird, Stanford-Cars, and FGVC Aircraft datasets.

Introduction

Weakly Supervised Fine-grained Image Recognition (WFGIR) aims at identifying sub-category of a given image label, e.g., different species of birds, and models of cars and aircrafts (Guo, Ouyang, and Xu 2020). It is a much more challenging problem than general image recognition due to the inherently subtle intra-class object variations among subcategories. As a result, the key to fine-grained image recognition lies in picking out the discriminative regions in an image to address the aforementioned challenge of WFGIR.

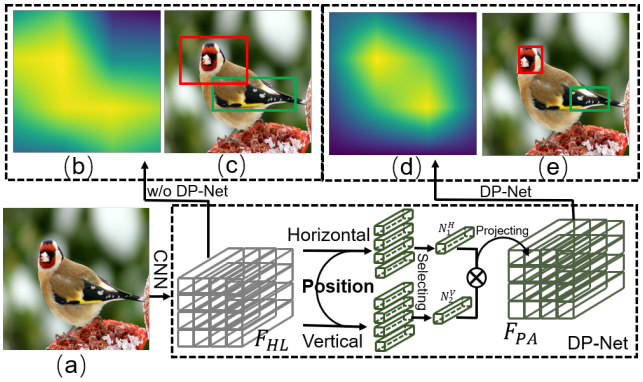

Figure 1: The motivation of the Dynamic Position-aware Network (DP-Net). F_{HL} denotes the high-level semantic feature maps and F_{PA} denotes the position-aware feature maps. (a) is the original image, (b)(d) are the discriminative response maps to guide network to sample the discriminative regions and (c) (e) are localization results without and with DP-Net learning, respectively. We can see that after introducing position clues, (d) is more compact and sparse than (b) and the resulted regions in (e) are more accurate and discriminative than those in (c).

Existing fine-grained image recognition methods can be roughly divided into three categories. The first category, *multi-stage schemes* is that the object and local parts/patches are localized either by explicitly detecting semantic parts (Zhang et al. 2016; Wei, Xie, and Wu 2016) or via implicit saliency localization (He and Peng 2017; He, Peng, and Zhao 2017; Peng, He, and Zhao 2018). The shortcoming of multi-stage schemes is that they are difficult to guarantee the selected regions are discriminative enough. Therefore, the second category, *end-to-end setting* focuses on automatically localizing the most discriminative patches via the attention mechanism in a weakly supervised manner (Fu, Zheng, and Mei 2017; Ding et al. 2019; Zheng et al. 2019b).

*Corresponding author.

Instead of picking out discriminative regions independently, the third category, *group learning* (Wang et al. 2019, 2020b) tends to select discriminative region group automatically by the correlation-guided discriminative learning.

All previous works tend to pick out discriminative details, which contain the discriminative visual content and position clues. The position clues can be indirectly learnt by utilizing context information of discriminative visual content, which leads into the position clues containing some non-discriminative information, thus degrading the recognition performance. For example, a bird can be identified using the spot on the wings in Fig. 1, but previous works by indirectly introducing position clues need to locate the entire wing to make sure that the spot is on the wings, making the selected regions contain much noisy and therefore increase the difficulty of recognition. These analysis motivate us to directly incorporate the position clues into visual content to make the network only pay attention to the discriminative visual content, which is more powerful to grab finer details.

Though important, position modelling usually requires significant pixel/region annotations and therefore is labor-intensive. Inspired by the natural language processing (Vaswani et al. 2017), we propose an end-to-end Dynamic Position-aware Network (DP-Net) to directly incorporate the position clues into visual content and dynamically align them without extra annotations, which eliminates the effect of position information for discriminative variances among subcategories. The DP-Net consists of Position Encoding Module (PEM), Position-vision Aligning Module (PAM) and Position-vision Reorganization Module (PRM). PEM learns a set of position-aware parts by adding the learnable position information into the horizontal and vertical visual content of images. PRM dynamically aligns both visual content and learnable position information via performing graph convolution on position-aware parts. PRM projects the aligned horizontal and vertical position-aware parts into the Euclidean space to construct a position-aware feature maps. Finally, the position-aware feature maps are used which is applied the visual content and corresponding position clues for more accurate discriminative region localization.

Main contributions of this paper can be summarized:

- To the best of our knowledge, we are the first to directly introduce the position clues into visual content in WFGIR.
- We propose an end-to-end Dynamic Position-aware Network (DP-Net) to directly incorporate the position clues into visual content and dynamically align them without extra annotations, eliminating the effect of position information for discriminative variances among subcategories.
- We evaluate the proposed method on three datasets (CUB-Bird, Stanford Cars, and FGVC Aircraft), and the results demonstrate that our PR-Net achieves state-of-the-art.

Related Work

Position encoding: Position encoding was originally proposed for natural language processing (Vaswani et al. 2017), where the model makes use of the order of the sequence and must inject some information about the relative or absolute position of the tokens in the sequence. Recently, position encoding is mainly applied to vision tasks including text recognition (Yue et al. 2020) and semantic segmentation (Choi, Kim, and Choo 2020). RobustScanner (Yue et al. 2020) enables the encoder to output characters encoding their own sequence positions for scene text recognition. HANet (Choi, Kim, and Choo 2020) introduces the absolute position prior for improving semantic segmentation for urban-scene images. Different from their works, we incorporate a learnable instead of absolute position encoding into visual content and dynamically align them.

Discriminative region localization: Recent WFGIR works mainly focus on designing end-to-end learning frameworks (Ding et al. 2019; Yang et al. 2018; Wang et al. 2020a,c; Ji et al. 2020). S3Ns (Ding et al. 2019) produces sparse attention to localize object and discriminative parts by collecting local maximums of class response maps. TASN (Zheng et al. 2019b) learns subtle feature representations from hundreds of part proposals and uses an attention-based sampler to highlight attention regions. DCL (Chen et al. 2019) automatically detects the discriminative regions by region confusion mechanism. More recent works (Wang et al. 2019; Zheng et al. 2017; Wang et al. 2020b) try to find discriminative region groups to improve discriminative ability for WFGIR. CGP (Wang et al. 2020b) establishes correlation between regions by graph propagation to discover the more discriminative region groups for WFGIR.

Most approaches predominantly tends to focus on learning the discriminative visual patterns which contain the discriminative visual content and position clues. However, the position clues can be indirectly learnt by utilizing context information of discriminative visual content, which leads into the selected tegions containing some non-discriminative information introducing by the position clues. Based on these consideration, we propose DP-Net to directly incorporate the position clues into visual content and dynamically align them for grabbing finer details. To our best knowledge, this is the first work to directly introduce the position clues into visual content for more accurate discriminative regions localization.

Proposed Method

We present our proposed Dynamic Position-aware Network in Fig.2. In order to better introduce position clues into different visual content, we propose a dual pathway to learn the aligned horizontal and vertical position-aware parts from horizontal and vertical direction of images, respectively. Finally, we project the aligned horizontal and vertical position-aware parts into the Euclidean space to construct a position-aware feature maps for discriminative region localization.

Position Encoding Module

Our proposed Position Encoding Module (PEM) consists of two sub-modules to learn position-aware horizontal and vertical parts: 1) part generator (PG) which divides the images into the horizontal and vertical parts from the horizontal and vertical direction; 2) learnable position encoding (LPE) which is adopted to add the learnable position information into the horizontal and vertical parts.

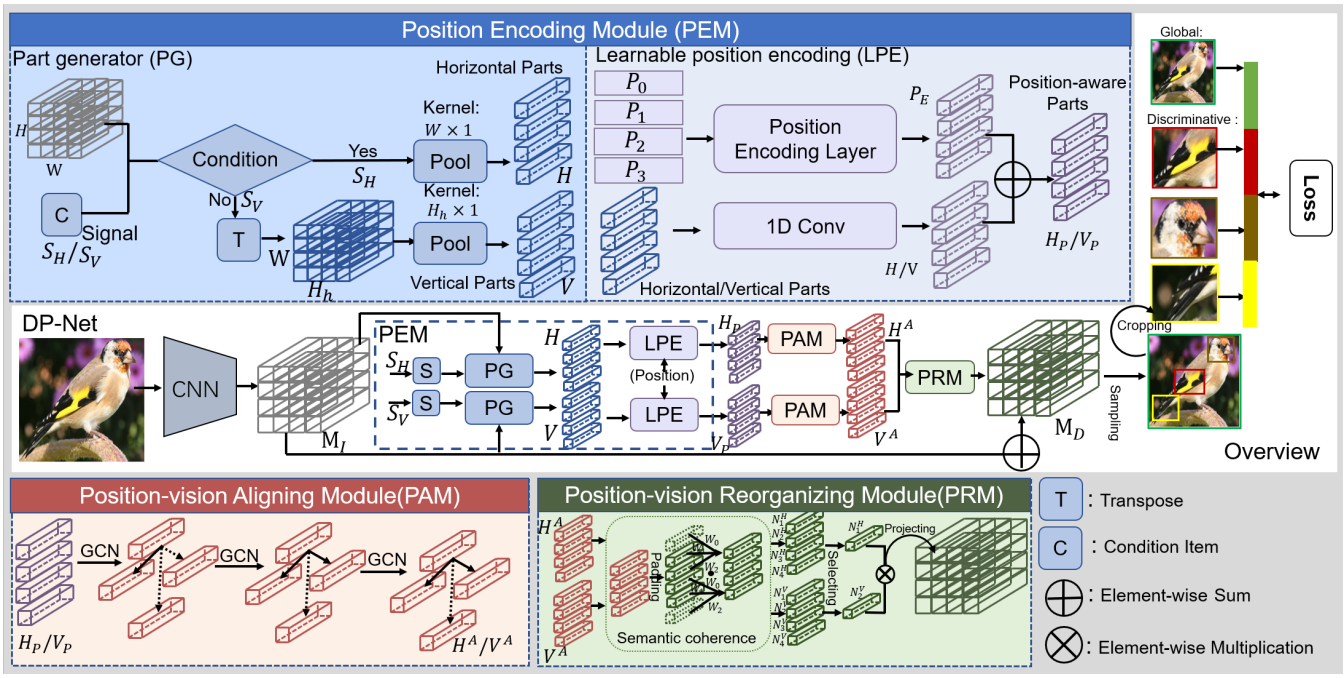

Figure 2: An overview of our DP-Net to directly introduce the position clues into visual content and dynamically align them for recognition. We begin by adding the learnable position information into horizontal and vertical parts by the Position Encoding Module (PEM) and aligning the position clues and visual content of position-aware parts through Position-vision Aligning Module (PAM). With these, we project the aligned position clues and visual content into the Euclidean space by reorganizing the aligned horizontal and vertical parts based on their feature similarity in Position-vision Reorganizing Module (PRM). At the sampling phase, the discriminative patches are located by collecting local maximums from M_D. Next, we crop and resize the patches to 224×224 from the original image and extract the corresponding features through CNN. Finally, the features of all branches are aggregated to produce the final recognition vectors.

Part generator. Given an image X, we feed X into the CNN backbone and extract the high-level feature maps from the top convolutional layer. The high-level feature maps are indicated as $M_I \in \mathbb{R}^{C \times H_h \times W}$, where C, H_h and W denote the channel, height and width of feature maps. Then, let M_I be an input tensor with specific signal S into the module PG to obtain a set of specific parts:

$$H = PG(M_I, S = S_H), V = PG(M_I, S = S_V), \quad (1)$$

where $H \in \mathbb{R}^{H_h \times C}$ and $V \in \mathbb{R}^{W \times C}$ denote the horizontal parts and vertical parts obtained by selecting corresponding horizontal pooling operation with different signals ($S = S_H$ and $S = S_V$), respectively. concretely, the input of PIE can be divided into two cases to obtain different parts: 1) $input = (M_I, S = S_H)$, which processes M_I by horizontal pooling operation with pooling kernel $W \times 1$ to obtain the horizontal parts H:

$$H_i = \frac{1}{W} \sum_{0 \le j < W} M_I^{ij}. \quad (2)$$

2) $input = (M_I, S = S_V)$, which reshapes M_I into $\tilde{M}_I \in \mathbb{R}^{C \times W \times H_h}$ followed by a horizontal pooling operation with pooling kernel $H_h \times 1$ to generate vertical parts V:

$$V_j = \frac{1}{H_h} \sum_{0 \le i < H_h} \tilde{M}_I^{ji}. \quad (3)$$

According to the horizontal pooling operation, it is easy to gather sufficient local details and destroy the global semantic information for WFGIR. It should be clarified that the horizontal and vertical parts cannot always guarantee the completeness of all the divided parts which are smaller than the size of certain semantic regions. However, it should not be a bad news for model training, since we also adopt random cropping which is a standard data augmentation strategy and leads to the result that divided parts are different compared with those of previous iterations. Small semantic parts, which are split at this iteration due to the horizontal pooling, will not be always split in other iterations. Hence, it brings an additional advantage of forcing our model to find more different granularity parts.

Learnable position encoding. For injecting visual content-related positional encodings, we propose to extend the basic sine and cosine functions of different frequencies proposed in Transformer by introducing a learnable mixture of position encoding. Concretely, the learnable positional encoding layer is defined as:

$$P_E = PE(pos, 2i) = w_{pos}^{2i} \cdot sin(pos/10000^{2i/C}), \quad (4)$$

$$P_E = PE(pos, 2i+1) = w_{pos}^{2i+1} \cdot cos(pos/10000^{2i/C}), \quad (5)$$

where $P_E \in \mathbb{R}^{N\times C}$ indicates the position information, and N and C are the number of position encoding and the dimension of parts, respectively. pos denotes the index of position encoding ranging from 0 to $N-1$, and w_{pos}^{2i} and w_{pos}^{2i+1} denote the learnable weight coefficients of the pos-th row the $2i$-th and the $(2i+1)$-th columns in weight matrix parameters $W \in \mathbb{R}^{N\times C}$, respectively. For horizontal parts H with horizontal position information $P_E^H \in \mathbb{R}^{H_h\times C}$ and vertical parts V with vertical position information $P_E^V \in \mathbb{R}^{W\times C}$, they can be incorporated by element-wise sum $\oplus$ as follows:

$$H_P = H \oplus P_E^H, V_P = V \oplus P_E^V. \tag{6}$$

Note that absolute position information is randomly jittered by up to two positions to generalize over different part location from various images to prevent an inordinately tight position-object coupling. Besides, we regard the horizontal and vertical parts of the image as the smallest unit of position encoding instead of each pixel of the feature maps. Concretely, the number of horizontal and vertical parts is selected manually and usually satisfies $(H_h + W) < (H_h \times W)$. There are two obvious disadvantages of all pixels with position encoding. First, the position information are lying in a low dimensional manifold, so the position encodings are over-complete. Second, the computation overhead is heavy and the memory cost is also large.

Position-vision Aligning Module

Each image can be represented into the horizontal and vertical position-aware parts by PEM. Although the parts contains the corresponding learnable position encoding, they are not aligned between the visual content and position information. We propose the Position-vision Aligning Module (PAM) to learn the correlation between the vision content and position information to align them. Here, let's take the horizontal position-aware parts as example to represent a detailed formulation for one graph propagation process.

The horizontal position-aware parts can be regarded as the nodes G of graph, and the adjacent matrix of nodes can be calculated, which indicates correlation intensity between nodes. Inspired by the work of multiple kernel learning (Dereli, Oguz, and Gönen 2019; Zhu et al. 2017), we propose to extend the basic graph neural network by introducing a learnable mixture of adjacent matrix for capturing more complex relations between nodes. Concretely, each element of the adjacent matrix can be calculated as below:

$$R_{ij} = c_{ij} \cdot < h(G_i), h(G_j) >, \tag{7}$$

where R_{ij} denotes the correlation coefficient between each two nodes (G_i, G_j), and h is a fully connected function where all nodes are mapped in a same space, followed by the similarity measure operation $<,>$. c_{ij} is a learnable correlation weight coefficient in weighted matrix $C \in \mathbb{R}^{|G|\times|G|}$, and c_{ij} can be learned to adjust correlation coefficient R_{ij} through back propagation. Note that the naive multiplication between correlation coefficients R and nodes G will completely change the scale of the feature vectors. Therefore, we perform normalization on each row of adjacent matrix to ensure that the sum of all the edges connected to one node equals to 1. The normalization of the adjacent matrix $A \in \mathbb{R}^{|G|\times|G|}$ is realized by a symmetric normalization, which corresponds to taking the average of neighboring node features. This formulation arrives at the new propagation rule shown as follows:

$$A = Q^{-\frac{1}{2}}(R + I)Q^{-\frac{1}{2}}, \tag{8}$$

where Q is the diagonal node degree matrix of $R + I$, and $R + I$ is the adjacent matrix of the graph with added self-connections for considering its own representation of each node and I is the identity matrix, which could directionally attend over other nodes' and its own features to learn the correlation between position information and visual content and align them.

After we obtain the adjacent matrix, we both take feature representations $G \in \mathbb{R}^{|G|\times C}$ and the corresponding adjacent matrix $A \in \mathbb{R}^{|G|\times|G|}$ as inputs, and update the node features as $G^{'} \in \mathbb{R}^{|G|\times C'}$. Formally, one layer process of GCN can be represented as:

$$G^{'} = f(G, A) = l(A \cdot G \cdot W), \tag{9}$$

where $W \in \mathbb{R}^{C\times C'}$ is the learned weight parameters, and l is a non-linear function (we use Rectified Linear Unit in the experiments). Since the visual content and learnable position information of horizontal and vertical parts are in a same graph space, they are mutually correlated and can align each other after multiple graph propagations, which obtains the aligned horizontal and vertical parts H^A V^A.

Position-vision Reorganization Module

Learning aligned horizontal and vertical parts only forces the parts in alignment between visual content and position information rather than to consider their spatial context. The network has difficulty in selecting discriminative patches/regions without spatial information. To deal with this limitation, we propose a Position-vision Reorganization Module (PRM) to resume the spatial information by reorganizing the aligned horizontal and vertical parts.

Before resuming the spatial information, we propose a semantic reassembly layer to stay semantic coherency of the aligned horizontal and vertical parts H^A V^A. Let's take aligned horizontal parts H^A as an example to represent the semantic coherence layer:

$$N_i^H = \sum_{j=-1}^{1} W_j \otimes H_{i+j}^A; \tag{10}$$

where the reassembly horizontal part $N_i^H \in \mathbb{R}^C$ can be obtained through performing element-wise multiplication $\otimes$ between the neighboring part H_{i+j}^A and the corresponding learnable weight coefficients $W_j \in \mathbb{R}^C$. With the reassembly process, each part with neighboring parts contributes to the reassembly parts N^H and N^V differently for guaranteeing the neighboring parts more correlated, based on the content of features instead of distance of locations.

And then, we utilize the feature similarity between N^H and N^V to more easily resume the original Euclidean space,

Feature Map	Stride	Scale	Scale Step	Aspect ratio
M_D	32	32	$2^{\frac{1}{3}}, 2^{\frac{2}{3}}$	$\frac{2}{3}, 1, \frac{3}{2}$
M_D^1	64	64	$2^{\frac{1}{3}}, 2^{\frac{2}{3}}$	$\frac{2}{3}, 1, \frac{3}{2}$
M_D^2	128	128	$1, 2^{\frac{1}{3}}, 2^{\frac{2}{3}}$	$\frac{2}{3}, 1, \frac{3}{2}$

Table 1: The stride, patch scale size, scale step and aspect ratios of the three different layers. M_D^1 and M_D^2 are feature maps after down-sampling M_D.

since the i-th row and j-th column feature vectors in the original Euclidean space can find a unique part group $< N_i^H, N_j^V >$:

$$V_{ij} = N_i^H \otimes N_j^V; \tag{11}$$

where $V_{ij} \in \mathbb{R}^C$ represents the i-th row and j-th column projecting feature vectors of the position-aware feature maps M_P, which can be obtained through element-wise multiplication $\otimes$ between the horizontal reassembly node N_i^H and vertical reassembly node N_j^V.

Finally, we integrate the original discriminative features M_I and the position-aware features M_P into a new feature maps M_D, which leads to more stable performance:

$$M_D = M_I + M_P. \tag{12}$$

Discriminative Information Sampling

We use M_D with three different scales to generate default patches, inspired by Feature Pyramid Network (Lin et al. 2017). Table 1 displays the design details, containing the scale size, scale step and aspect ratio of default patches.

Let's take M_D as an example. We feed the features M_D into a score layer. Concretely, we add a $1 \times 1 \times N$ convolution layer to learn discriminative response maps $R \in \mathbb{R}^{N \times H_h \times W}$, which indicates the impact of discriminative regions on the final classification, as follows:

$$R = W_D * M_D, \tag{13}$$

where $W_D \in \mathbb{R}^{C \times 1 \times 1 \times N}$ represents the convolution kernels, N is the number of the default patches at a given location in the feature maps. Meanwhile, we assign the discriminative response value to each default patch p_{ijk}:

$$p_{ijk} = [t_x, t_y, t_w, t_h, R_{ijk}], \tag{14}$$

where R_{ijk} denotes the value of the i^{th} row, the j^{th} column and the k^{th} channel, and (t_x, t_y, t_w, t_h) denotes each patch's coordinates. Finally, the network picks the top-M patches with a response value, where M is a hyper-parameter.

Loss Function

The full multi-task loss $\mathcal{L}$ can be represented as the following:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{gud} + \mathcal{L}_{rela} + \mathcal{L}_{rank}, \tag{15}$$

where $\mathcal{L}_{cls}$ represents the fine-grained classification loss. $\mathcal{L}_{gud}$, $\mathcal{L}_{rela}$ and $\mathcal{L}_{rank}$ represent the guided loss, correlation loss and rank loss, respectively.

We denote the selected discriminative patches as $P = \{P_1, P_2, ..., P_N\}$ and the corresponding discriminative response values as $R = \{R_1, R_2, ..., R_N\}$. Then the guided loss and the correlation loss as well as the rank loss are defined as follows:

$$\mathcal{L}_{gud}(X, P) = \sum_i^N (max\{0, log\mathcal{C}(X) - log\mathcal{C}(P_i)\}), \tag{16}$$

$$\mathcal{L}_{rela}(P_c, P) = \sum_i^N (max\{0, log\mathcal{C}(P_i) - log\mathcal{C}(P_c)\}), \tag{17}$$

$$\mathcal{L}_{rank}(R, P) = \sum_{log\mathcal{C}(P_i) < log\mathcal{C}(P_j)} (max\{0, (R_i - R_j)\}), \tag{18}$$

where X is the original image and the function $\mathcal{C}$ is the confidence function which reflects the probability of classification into the correct category, P_c is the concatenation of all selected patch features.

The guided loss is designed to guide the network to select the more discriminative regions. The correlation loss can guarantee that the prediction probability of combined features is greater than that of single patch features. The rank loss strives for consistency of the discriminative scores and the final classification probability values of the selected patches, encouraging them in the same order.

Method	Accuracy
BL (Li et al. 2017)	84.5%
BL + Sample	87.1%
BL + Sample + PEM + PRM	88.4%
BL + Sample + PEM + PRM + PAM	89.3%

Table 2: The ablative recognition results of different variants of our method. We test the models on CUB.

Experiments

Datasets

We comprehensively evaluate our algorithm on Caltech-UCSD Birds (Branson et al. 2014) (CUB-200-2011), Stanford Cars (Krause et al. 2013) (Cars) and FGVC Aircraft (Airs) (Maji et al. 2013) datasets, which are widely used benchmark for fine-grained image recognition. The CUB-200-2011 dataset contains 11,788 images spanning 200 subspecies. The ratio of train data and test data is roughly 1:1. The Cars dataset has 16,185 images from 196 classes officially split into 8,144 training and 8,041 test images. The Airs dataset contains 10,000 images over 100 classes, and the train and test sets split ratio is around 2 : 1.

Implementation Details

In all our experiments, all images are resized to 448×448, and we crop and resize the patches to 224×224 from the original image. We use fully-convolutional network ResNet-50 as feature extractor and apply Batch Normalization as regularizer. We also use Momentum SGD with initial learning rate 0.001 and multiplied by 0.1 after 60 epochs. We use weight decay $1e^{-4}$. To reduce patch redundancy, we adopt the non-maximum suppression (NMS) on default patches based on their discriminative scores, and the NMS threshold

Method	Box Annotation	Part Annotation	CUB Acc.	Cars Acc.	Airs Acc.
PN-DCN (Branson et al. 2014)	BBox	Parts	85.4%	-	-
M-CNN (Wei, Xie, and Wu 2016)	n/a	Parts	84.2%	-	-
PG (Krause et al. 2015)	BBox	n/a	82.8%	92.8%	-
SCDA (Wei et al. 2017)	n/a	n/a	80.1%	85.1%	79.5%
AutoBD (Yao et al. 2018)	n/a	n/a	81.6%	88.9%	-
OPAM (Peng, He, and Zhao 2018)	n/a	n/a	85.8%	92.2%	-
lB-CNN (Kong and Fowlkes 2017)	n/a	n/a	84.2%	90.9%	87.3%
Kernel-Activation (Cai, Zuo, and Zhang 2017)	n/a	n/a	85.3%	91.7%	88.3%
Kernel-Pooling (Cui et al. 2017)	n/a	n/a	86.2%	92.4%	85.7%
DBT-Net (Zheng et al. 2019a)	n/a	n/a	88.1%	94.5%	91.6%
PA-CNN (Zheng et al. 2020)	n/a	n/a	87.8%	93.3%	91.0%
DCL (Chen et al. 2019)	n/a	n/a	87.8%	94.2%	93.0%
TASN (Zheng et al. 2019b)	n/a	n/a	87.9%	93.8%	-
CDL (Wang et al. 2019)	n/a	n/a	88.4%	94.5%	-
LIO (Zhou et al. 2020)	n/a	n/a	88.0%	94.5%	92.7%
ACNet (Ji et al. 2020)	n/a	n/a	88.1%	94.6%	92.4%
CGP (Wang et al. 2020b)	n/a	n/a	88.3%	94.0%	93.2%
S3Ns (Ding et al. 2019)	n/a	n/a	88.5%	94.7%	92.8%
Our DP-Net	n/a	n/a	**89.3%**	**94.8%**	**93.9%**

Table 3: Comparison of different methods on CUB-200-2011(CUB), Cars 196 (Cars) and Aircraft (Airs).

is set to 0.25. Note that the architecture in principle contains multiple CNN modules and for clarity, these CNN modules share the same parameters.

Ablation Experiments

For understanding the influence of different components in our proposed method, we conduct some ablation studies. As shown in Table 2, we design different settings on CUB-200-2011 dataset by using ResNet-50 as the backbone network. First, the features are extracted from the original image through ResNet-50 (He et al. 2016) without any object or partial annotation for fine-grained recognition, and we set it as the baseline (BL) of our model. When we introduce the score mechanism (Sample) to only preserve the highly discriminative patches and reduce the number of patches to single-digit, the top-1 recognition accuracy on CUB-200-2011 dataset improves 2.6%. Finally, we take account into the position clues into visual content through DP-Net for more accurate discriminative region localization, and achieve the state-of-the-art result of 89.3%. The PEM and PRM can effectively make the network focus on the local details and introduce the learnable position information into parts, thus and still improves the accuracy. For locating discriminative regions, the PAM can learn and align both the visual content and position information of parts to eliminates the effect of location for discriminative variances among subcategories, thus outperforming by 0.9%. Ablation experiments have verified that the proposed DP-Net focuses on the finer details to precisely find the discriminative details, thus effectively improves the recognition accuracy.

Performance Comparison

Due to the proposed model only utilizing image-level annotations, our comparisons focus on the weakly supervised methods. In Tab.3, the performance of different methods on CUB-200-2011 dataset, Stanford Cars-196 dataset and FGVC-Aircraft dataset is reported, respectively. In the table from top to bottom, the methods are separated into six groups, which are (1) supervised multi-stage methods, (2) weakly supervised multi-stage frameworks, (3) weakly supervised end-to-end feature encoding, (4) end-to-end localization-classification networks, and (5) DP-Net.

Earlier multi-stage methods rely on the object and even part annotations to achieve comparable results. However, using the object or part annotations limits the performance due to the fact that human annotations only give the coordinates of important parts rather than the accurate discriminative region location. Weakly supervised multi-stage frameworks gradually exceed the strong supervised methods though picking out discriminative regions. The end-to-end feature encoding methods have good performance via encoding the CNN feature vectors into high-order information, while they result in high computational cost. Although the localization-classification sub-networks works well on various datasets, they neglect that the indirect position clues contain some non-discriminative information obtained by utilizing context information of visual content, which leads to the effect of location for discriminative variances among subcategories. Our end-to-end DP-Net approach achieves new state-of-the-art without any extra annotations and enjoys consistent performance on various datasets.

As shown in Table 3, our approach outperforms these strong supervised methods in the first group, which indicates that the proposed method can find the discriminative patches without any fine-grained annotations. Compared with recent weakly supervised end-to-end methods, they find discriminative patches from high-level feature maps directly. We run DP-Net to directly learn the position-aware feature maps that the highlighted regions are finely related to visual content and position clues for discriminative region localization and

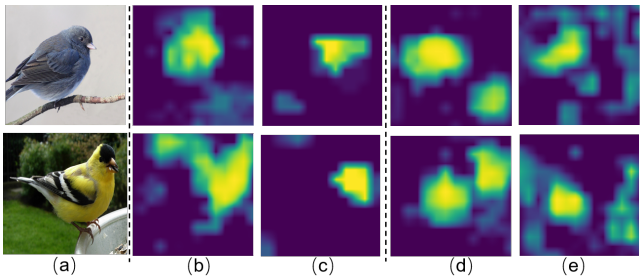

Figure 3: Visualization of intermediate results in DP-Net. (a) is the original images, (b)(d)indicate the original feature maps M_I and (c)(d) denote the reconstructing feature maps of the special channel, respectively. (b)(c) are the same channel feature. (d)(e) are also the same channel feature.

Depth	1	2	3	4	5
Accuracy	88.2%	88.7%	89.3%	89.1%	88.6%

Table 4: The recognition accuracy on CUB-200-2011 of model trained with different depth of graph in PAM.

achieves the new state-of-the-arts.

Visualization Analysis

Insights about the influence of our proposed approach can be obtained by visualizing the effects of feature maps M_I and M_P, i.e. the feature maps without and with DP-Net respectively. Compared with the feature maps M_I without DP-Net, the feature map response can be shrank to focus on the accurate discriminative regions without attending on background noise in Fig. 3, which indicates our method can directly learn and use the position clues instead of introducing position information by utilizing context information of discriminative visual content. To intuitively display the effect of position clues, We draw the discriminative regions and display the discriminative response map predicted by our model without and with DP-Net in Fig. 4, respectively. It can be seen that the discriminative response maps without DP-Net focus on the wide area which results in the problem of hard localization, as shown in Fig. 4(b). However, Our DP-Net could pay attention to a small region or region group in discriminative response maps, where the discriminative patches can be located more easily and accurately. For more intuitive presentation, we display the localization results in original images, as shown in Fig. 4(c)(e).

Discussions

The deeper, the better? We show the recognition results with different depth of graph, as shown in Tab. 4. It is obvious that the performance of DP-Net drops when the depth of graph increases to 4. The possible reason of the performance drop is that after using more graph layers, the propagation between nodes will be overwhelmed.

The importance of position clues: We show the recognition results with different branch of our model, as shown in Tab.5. The original feature branch, position-aware feature branch and the combination of both original and position-aware feature branches denote that the sampling process is on feature maps M_I, M_P and M_D, respectively. To make fair comparison, we pick out 4 discriminative regions on corresponding feature maps. Compared to the original feature branch without the DP-Net, it can be seen that the original features M_I can be optimized through the position clue learning to achieve more accurate discriminative localization. Moreover, it is clear that the position-aware feature branch can exceed the original feature branch, outperforming by 0.3%. Finally, the original feature branch and the position-aware feature branch can promote each other, thus achieving the state-of-the-art.

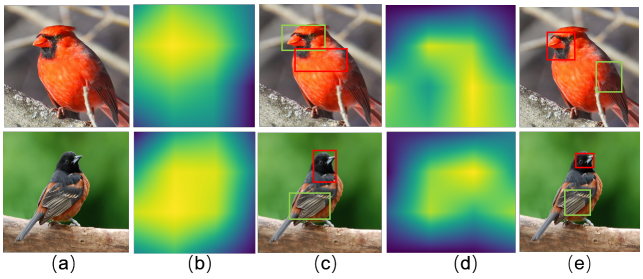

Figure 4: Visualization of discriminative response maps and localization results with and without DP-Net.(a) is the original images. (b)(c) are the discriminative response maps through sampling stage with and without the DP-Net, respectively. (d)(e) are the localization results with and without DP-Net, respectively.

Branch	Accuracy
Original Feature Branch w/o DP-Net	87.1%
Original Feature Branch	88.2%
Position-aware Feature Branch	88.5%
Original + Position-aware Feature Branch	89.3%

Table 5: The recognition accuracy on CUB-200-2011 of model trained with the different branch.

Conclusion

In this paper, we first directly introduce the position clues into the visual content and align them in WFGIR. We argue that learning the indirect position information aggravates the difficulty of recognition for existing methods. We propose an end-to-end Dynamic Position-aware Network (DP-Net) to directly incorporate the position clue into visual content by dynamically aligning visual content and learnable position encoding without extra annotations. Extensive experiments show that the recognition accuracy can be improved significantly by localizing patches on the position-aware feature maps. The last but the most important, our algorithm is end-to-end trainable and achieves state-of-the-art in CUB-Bird, FGVC Aircraft and Stanford Cars datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No.61932020, 61976038, U1908210 and 61772108.

References

Branson, S.; Horn, G. V.; Belongie, S. J.; and Perona, P. 2014. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. *CoRR* abs/1406.2952.

Cai, S.; Zuo, W.; and Zhang, L. 2017. Higher-Order Integration of Hierarchical Convolutional Activations for Fine-Grained Visual Categorization. In *ICCV 2017, Venice, Italy, October 22-29, 2017*, 511–520. doi:10.1109/ICCV.2017.63.

Chen, Y.; Bai, Y.; Zhang, W.; and Mei, T. 2019. 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*, 5157–5166.

Choi, S.; Kim, J. T.; and Choo, J. 2020. Cars Can't Fly Up in the Sky: Improving Urban-Scene Segmentation via Height-Driven Attention Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 9370–9380. IEEE. doi:10.1109/CVPR42600.2020.00939.

Cui, Y.; Zhou, F.; Wang, J.; Liu, X.; Lin, Y.; and Belongie, S. J. 2017. Kernel Pooling for Convolutional Neural Networks. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3049–3058. doi:10.1109/CVPR.2017.325.

Dereli, O.; Oguz, C.; and Gönen, M. 2019. A Multitask Multiple Kernel Learning Algorithm for Survival Analysis with Application to Cancer Biology. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 1576–1585. PMLR.

Ding, Y.; Zhou, Y.; Zhu, Y.; Ye, Q.; and Jiao, J. 2019. Selective Sparse Sampling for Fine-Grained Image Recognition. In *The IEEE International Conference on Computer Vision (ICCV)*.

Fu, J.; Zheng, H.; and Mei, T. 2017. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 4476–4484. doi:10.1109/CVPR.2017.476.

Guo, J.; Ouyang, W.; and Xu, D. 2020. Multi-Dimensional Pruning: A Unified Framework for Model Compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 1505–1514. IEEE. doi:10.1109/CVPR42600.2020.00158. URL https://doi.org/10.1109/CVPR42600.2020.00158.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778. doi:10.1109/CVPR.2016.90.

He, X.; and Peng, Y. 2017. Weakly Supervised Learning of Part Selection Model with Spatial Constraints for Fine-Grained Image Classification. In *AAAI February 4-9, 2017, San Francisco, California, USA.*, 4075–4081.

He, X.; Peng, Y.; and Zhao, J. 2017. Fine-grained Discriminative Localization via Saliency-guided Faster R-CNN. In *ACM MM 2017, Mountain View, CA, USA, October 23-27, 2017*, 627–635. doi:10.1145/3123266.3123319.

Ji, R.; Wen, L.; Zhang, L.; Du, D.; Wu, Y.; Zhao, C.; Liu, X.; and Huang, F. 2020. Attention Convolutional Binary Neural Tree for Fine-Grained Visual Categorization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 10465–10474. IEEE. doi:10.1109/CVPR42600.2020.01048.

Kong, S.; and Fowlkes, C. C. 2017. Low-Rank Bilinear Pooling for Fine-Grained Classification. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 7025–7034. doi:10.1109/CVPR.2017.743.

Krause, J.; Jin, H.; Yang, J.; and Li, F. 2015. Fine-grained recognition without part annotations. In *CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 5546–5555. doi:10.1109/CVPR.2015.7299194.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, 554–561. doi:10.1109/ICCVW.2013.77.

Li, Z.; Yang, Y.; Liu, X.; Zhou, F.; Wen, S.; and Xu, W. 2017. Dynamic Computational Time for Visual Attention. In *ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, 1199–1209. doi:10.1109/ICCVW.2017.145.

Lin, T.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature Pyramid Networks for Object Detection. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 936–944. doi:10.1109/CVPR.2017.106.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M. B.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. *CoRR* abs/1306.5151.

Peng, Y.; He, X.; and Zhao, J. 2018. Object-Part Attention Model for Fine-Grained Image Classification. *TIP* 27(3): 1487–1500. doi:10.1109/TIP.2017.2774041.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 5998–6008.

Wang, S.; Wang, Z.; Li, H.; and Ouyang, W. 2020a. Category-specific Semantic Coherency Learning for Fine-grained Image Recognition. In Chen, C. W.; Cucchiara, R.; Hua, X.; Qi, G.; Ricci, E.; Zhang, Z.; and Zimmermann, R., eds., *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, 174–183. ACM. doi:10.1145/3394171.3413871.

Wang, Z.; Wang, S.; Li, H.; Dou, Z.; and Li, J. 2020b. Graph-Propagation Based Correlation Learning for Weakly Supervised Fine-Grained Image Classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 12289–12296. AAAI Press.

Wang, Z.; Wang, S.; Yang, S.; Li, H.; Li, J.; and Li, Z. 2020c. Weakly Supervised Fine-Grained Image Classification via Guassian Mixture Model Oriented Discriminative Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 9746–9755. IEEE. doi:10.1109/CVPR42600.2020.00977.

Wang, Z.; Wang, S.; Zhang, P.; Li, H.; Zhong, W.; and Li, J. 2019. Weakly Supervised Fine-grained Image Classification via Correlation-guided Discriminative Learning. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, 1851–1860. doi:10.1145/3343031.3350976.

Wei, X.; Luo, J.; Wu, J.; and Zhou, Z. 2017. Selective Convolutional Descriptor Aggregation for Fine-Grained Image Retrieval. *TIP* 26(6): 2868–2881. doi:10.1109/TIP.2017.2688133.

Wei, X.; Xie, C.; and Wu, J. 2016. Mask-CNN: Localizing Parts and Selecting Descriptors for Fine-Grained Image Recognition. *CoRR* abs/1605.06878. URL http://arxiv.org/abs/1605.06878.

Yang, Z.; Luo, T.; Wang, D.; Hu, Z.; Gao, J.; and Wang, L. 2018. Learning to Navigate for Fine-Grained Classification. In *ECCV, Germany, September 8-14, 2018, Proceedings, Part XIV*, 438–454. doi:10.1007/978-3-030-01264-9_26.

Yao, H.; Zhang, S.; Yan, C.; Zhang, Y.; Li, J.; and Tian, Q. 2018. AutoBD: Automated Bi-Level Description for Scalable Fine-Grained Visual Categorization. *TIP* 27(1): 10–23. doi:10.1109/TIP.2017.2751960.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition abs/2007.07542.

Zhang, X.; Xiong, H.; Zhou, W.; Lin, W.; and Tian, Q. 2016. Picking Deep Filter Responses for Fine-Grained Image Recognition. In *CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 1134–1142. doi:10.1109/CVPR.2016.128.

Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition. In *ICCV 2017, Venice, Italy, October 22-29, 2017*, 5219–5227. doi:10.1109/ICCV.2017.557.

Zheng, H.; Fu, J.; Zha, Z.; and Luo, J. 2019a. Learning Deep Bilinear Transformation for Fine-grained Image Representation. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., 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*, 4279–4288.

Zheng, H.; Fu, J.; Zha, Z.; and Luo, J. 2019b. Looking for the Devil in the Details: Learning Trilinear Attention Sampling Network for Fine-Grained Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 5012–5021.

Zheng, H.; Fu, J.; Zha, Z.; Luo, J.; and Mei, T. 2020. Learning Rich Part Hierarchies With Progressive Attention Networks for Fine-Grained Image Recognition. *IEEE Trans. Image Processing* 29: 476–488. doi:10.1109/TIP.2019.2921876.

Zhou, M.; Bai, Y.; Zhang, W.; Zhao, T.; and Mei, T. 2020. Look-Into-Object: Self-Supervised Structure Modeling for Object Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 11771–11780. IEEE. doi:10.1109/CVPR42600.2020.01179.

Zhu, X.; Jing, X.; Wu, F.; Wu, D.; Cheng, L.; Li, S.; and Hu, R. 2017. Multi-Kernel Low-Rank Dictionary Pair Learning for Multiple Features Based Image Classification. In Singh, S. P.; and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2970–2976. AAAI Press.