Localizing Discriminative Regions for Fine-Grained Visual Recognition: One Could be Better than Many

Fen Fang, Yun Liu*, Qianli Xu

Institute for Infocomm Research, Singapore, 138632
** Corresponding author.*

Abstract

Fine-grained visual recognition (FGVR) involves distinguishing between highly similar subcategories. To capture subtle differences among closely related subcategories, prevalent methodologies first localize the discriminative region in the image and then classify it. However, previous detection-based and attention-based methods for discriminative region localization possess inherent limitations, thus limiting the performance. Deep reinforcement learning (DRL) is a good choice as it can autonomously determine optimal actions to achieve a given objective, *e.g.*, localizing the discriminative region. Nevertheless, existing DRL-based approaches learn to simultaneously localize an *uncertain* number of discriminative regions, which may pose challenges for the DRL agent. Additionally, the optimization of DRL relies on recognition feedback from an FGVR classifier, whereas existing approaches just employ standard networks as the classifier. To address these challenges, we propose a Reinforced Most-Discriminative Region Localization (RMDRL) module to adaptively localize *a single, most discriminative region* in the image. To provide precise feedback for training the RMDRL module, we propose a Discriminative Knowledge Self-Distillation (DKSD) module to cultivate a robust FGVR classifier. Our extensive experimentation across nine benchmarks validates the efficacy of our approach for FGVR. Our findings also support that one discriminative region localized by DRL could be better than multiple discriminative regions.

Keywords: Fine-grained, Visual Recognition, Discriminative Region, Deep Reinforcement Learning, Knowledge Distillation.

Figure 1: **Localize-to-classify paradigm for FGVR**. In adherence to the localize-to-classify paradigm, the approach begins by identifying the discriminative region in the input image, highlighted by the red box. This pinpointed discriminative region is then input into the classifier to produce the image category output, which, in this instance, is "Western Gull".

1. Introduction

Fine-grained visual recognition (FGVR) involves the classification of visual objects into more specific subordinate categories within the same basic category, such as distinct bird species [1], varied dog breeds [2], specific vehicle models [3], various aircraft types [4], and diverse food categories [5]. In contrast to traditional basic-level visual categorization [6], which aims to recognize basic-level categories, FGVR is more challenging because it requires capturing and distinguishing the subtle differences among closely related subcategories. This enables a range of challenging applications that may pose difficulties even for human annotators.

To effectively capture and discern subtle differences among closely related subcategories, state-of-the-art FGVR approaches [3, 2, 1, 7, 8, 9, 10] typically begin by localizing the *discriminative region* within the image before training a classifier to recognize its category, as depicted in Fig. 1. The discriminative region is defined as the area containing the distinctive visual characteristics required by the classifier for FGVR [8, 9, 10, 3, 2, 1, 11, 12, 13]. It can be either a single part of the image covering the object, such as the region of a bird [8, 9, 10], or multiple parts corresponding to different semantic elements of the object, such as the head, torso, and tail of a bird [3, 2, 1].

Modern FGVR approaches primarily concentrate on determining how to localize the discriminative region in the input image. Some methods [14, 15, 8, 9, 16, 17, 18] utilize object or object part annotations to train detectors for localizing the discriminative region. Nevertheless, it proves challenging for human annotators to precisely identify the discriminative region, considering that objects of different subcategories or even objects of the same subcategory but with varying poses and views may regard distinct

object parts as discriminative. Furthermore, bounding box annotations for objects or object parts are expensive to obtain. Consequently, certain methods [19, 20, 21, 22, 23, 24, 25, 26] adopt attention mechanisms to automatically localize the discriminative region, eliminating the need for additional annotations. However, these attention-based methods often fix the number of discriminative regions, contrary to the variability observed across different subcategories and even within the same subcategory with diverse object poses and views. Additionally, the performance of attention-based methods is limited by attention mechanisms that may focus on the wrong region. Last but not least, detection-based methods confine the pinpointed discriminative region within the object region, thereby disregarding valuable contextual information [27].

To tackle these problems, deep reinforcement learning (DRL) emerges as a promising solution due to its ability to autonomously determine optimal actions to achieve a given objective, such as localizing the discriminative region for FGVR [28, 29]. These works employ DRL to decide both *which* regions and *how many* regions are discriminative for FGVR. However, two key challenges are identified in existing DRL-based methods [28, 29]. First, there is a potential difficulty when DRL is tasked with simultaneously locating an *uncertain* number of discriminative regions, which may impact its performance. Second, the reward mechanism in DRL relies on recognition feedback from a classifier, yet [28, 29] employ a conventional neural network as the classifier, potentially limiting the model's overall performance.

Our efforts fall into the category of DRL-based FGVR. To address the first challenge, we draw upon previous studies demonstrating the effectiveness of using *a single discriminative region* [8, 9, 10], which appears to be a more attainable goal for DRL. Thus, this paper proposes a **Reinforced Most-Discriminative Region Localization (RMDRL)** module to adaptively localize a single, most discriminative region in the input image. The RMDRL module involves training an adapter to adjust the initialized region on the image, optimizing a policy network with parameters iteratively updated based on recognition feedback from a classifier. To tackle the second challenge and provide accurate feedback for DRL, we further propose a **Discriminative Knowledge Self-Distillation (DKSD)** module to learn a robust classifier. This module features a shared feature extractor followed by multiple branches, with each branch taking image features within a predefined region as input, representing individual FGVR classifiers. The outputs from all branches are aggregated through averaging. This aggregated knowl-

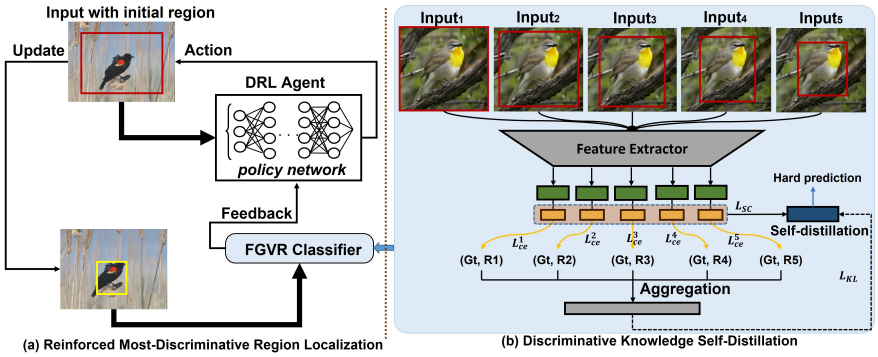

Figure 2: **Overview of our methods**. Our method comprises two integral modules: (a) reinforced most-discriminative region localization (b) discriminative knowledge self-distillation. In the first module, the DRL agent processes an image with an initial region as input, making decisions on the region update. The refined region is then introduced to an FGVR classifier, which takes the image as input and generates feedback for updating the parameters of the policy network for DRL agent learning. In the knowledge distillation module, each image is associated with multiple predefined regions(refer to Section 4 for the definition and generation of predefined regions),which are used for cropping to generate $Input_x$. Then, a shared feature extractor is employed to create individual FGVR classifiers, with each task using $Input_x$ as input for soft predictions, as indicated by the yellow arrows. Subsequently, knowledge aggregation and self-distillation are applied to distill the acquired knowledge from multiple classifiers into a single classifier for a more decisive, hard prediction. The single classifier trained in the second module severs as FGVR classifier in the first module.

edge is distilled into a single classifier using a self-distillation mechanism for enhanced efficiency. During the inference stage, the RMDRL module predicts an optimal discriminative region covering the most discriminative semantic parts in the image. This identified region is then fed into the distilled classifier to predict the image subcategory. Fig. 2 offers an overview of the proposed method. Extensive experiments have demonstrated that utilizing DRL to localize a single discriminative region is superior to locating multiple discriminative regions, making the proposed method achieve state-of-the-art performance for FGVR.

Our main contributions can be summarized as follows:

- Instead of locating an uncertain number of discriminative regions as in existing DRL-based methods, we introduce an RMDRL module that adaptively localizes *a single, most discriminative region*, as illustrated in Fig. 2(a).

- Recognizing that the reward mechanism in DRL hinges on recognition feedback from the classifier, we propose a DKSD module that distills multiple individual classifiers into a single robust classifier to provide accurate feedback for DRL, as depicted in Fig. 2(b).

- We conduct comprehensive experiments demonstrating that utilizing DRL to localize a single discriminative region surpasses the performance of locating multiple discriminative regions, establishing the state-of-the-art performance of the proposed method for FGVR.

2. Related Works

Based on the discriminative region localization strategy, FGVR approaches can be categorized into three groups: detection-based approaches, attention-based approaches, and DRL-based approaches. The following subsections introduce each of these categories in turn.

2.1. Discriminative region localization approaches

2.2. Detection-based FGVR Approaches

As discussed earlier in Sec. 1, the discriminative region can either constitute a single object region or encompass multiple object parts. Consequently, certain detection-based approaches aim to identify a single discriminative region covering the entire object, while others concentrate on detecting multiple discriminative regions corresponding to distinct object parts.

Object detection-based localization. These approaches employ the object's bounding box as the discriminative region. Conventional approaches, such as [1], [3], [2], [4], typically rely on using the ground-truth bounding box. However, the drawback of these approaches lies in their requirement for the ground truth box of objects, entailing unnecessary human labeling efforts. To address this, subnetworks are integrated to facilitate object localization. SPDA-CNN [30] introduces a top-down method, combining it with Faster R-CNN [31], to localize objects by inheriting prior geometric constraints. Additionally, segmentation models play a role in PS-CNN [18] and Mask-CNN [32], where they are employed to obtain object masks, aiding in object localization.

Object part detection-based localization. These approaches capture diverse discriminative parts or regions and encode them into a descriptor,

forming a representation of the complete image. Typically, prior works [17, 32, 33, 34, 35] employ bounding boxes to define these discriminative regions, acquired through weak supervisions or manual labeling. For example, in [17], a part-based R-CNN is introduced to identify both the entire object and its constituent parts. The learned geometric constraints subsequently enhance the accuracy of representation. Another approach, presented in [16], proposes a sequential search method for localizing discriminative parts. This method incorporates bounding box annotations and integrates heuristics into an LSTM network.

Detection-based approaches necessitate bounding box annotations during model training. However, these annotations are not only expensive but also suboptimal for FGVR, as the determination of discriminative regions depends on the classifier and is thus difficult for human annotators. Moreover, while bounding boxes cover objects or object parts, the contextual information in the background and environment, crucial for visual recognition, is ignored by detection-based methods. In our approach, the discriminative region is automatically determined by DRL with the reward of recognition feedback from the classifier, thus achieving better performance.

2.3. Attention-based FGVR Approaches

To eliminate the requirement for bounding box annotations, several recent works [24, 36, 37, 38, 39, 40, 41, 42, 43] leverage attention mechanisms to automatically localize discriminative regions. For instance, in [20], a hierarchical structure is employed to automatically identify the most discriminative parts through the construction of an attention proposal subnetwork. In [42], an end-to-end Structure-driven Relation Graph Network (SRGN) is proposed for fine-grained object recognition, utilizing object structure information without additional annotations to highlight discriminative details influenced by personalized differences. In [23, 44], multi-level attention is employed to simultaneously pinpoint multiple discriminative parts. In [45], a Progressive Multi-Granularity (PMG) training framework is proposed to effectively fuses features from different granularities. This strategy is similar to a 'multi-level attention mechanism' that simultaneously identifies multiple discriminative parts and then fuses these parts together for final object recognition. Meanwhile, [46] introduces a trilinear attention network to sample the most informative parts from hundreds of part proposals. [10] presents an attention-based convolutional binary neural tree that incorporates attention

mechanisms within a tree structure to facilitate hierarchical learning of discriminative characteristics. More recently, [47] used self-attention and graph neural networks for relation-aware feature transformation and refinement via context-aware attention, enhancing feature discriminability in an end-to-end learning process. [48] treats each head in multi-head self-attention as a weak learner, voting for discriminative region tokens based on attention maps and spatial relationships to improve FGVR performance.

While attention-based methods address certain limitations of detection-based approaches, they introduce a new challenge. Objects belonging to different subcategories, and even those within the same subcategory but with distinct poses and views, may perceive different object parts as discriminative regions. Thus, the number of discriminative regions can vary. However, attention-based methods usually fix the number of discriminative regions. Furthermore, attention mechanisms often exhibit high activation within the object region [49, 50] and contextual information is thus neglected, akin to detection-based approaches. Additionally, the performance of attention-based methods is restricted by attention mechanisms that may concentrate on the wrong region. In contrast, this paper employs DRL to automatically identify a single discriminative region that encapsulates crucial visual characteristics and contextual information in the image for FGVR.

2.4. DRL-based FGVR Approaches

DRL has been integrated with deep visual learning to enhance performance across various vision tasks such as object localization [51, 52]. In the field of fine-grained recognition, a notable DRL approach is presented in [53], leveraging part-attribute text descriptions to localize discriminative regions. The text attribute labeling required by this approach is more amenable than the accurate part location annotation required by traditional part-based FGVR methods. Instead of using part location annotations or text attributes, our approach employs DRL to automatically determine discriminative regions. Notably, a Navigator is proposed in [54] to direct the model to focus on the most informative regions, while the Teacher evaluates the regions proposed by the Navigator and provides feedback. Although they did not employ reinforcement learning, their mechanisms are quite similar.

Moreover, He *et al.* proposed a two-stage DRL architecture, M2DRL [28], to determine *which* regions and *how many* regions are discriminative and representative for FGVR. This architecture sequentially localizes the object and its parts ("which"), and adaptively determines the number of discriminative

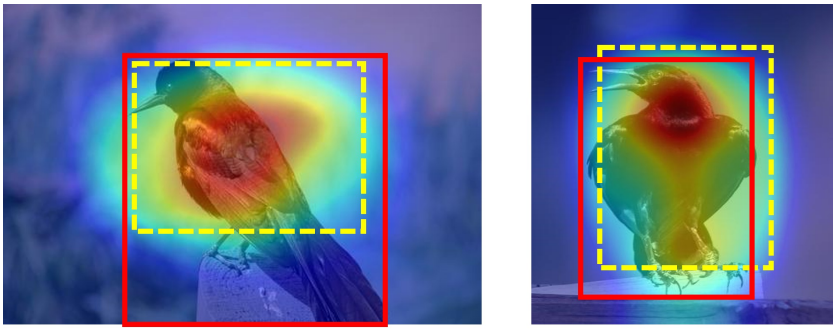

Figure 3: **Variation in discriminative regions for the same object due to different poses.** Samples are Boat Tailed Grackle birds from CUB-200 [1]. Dashed yellow box and red box indicate discriminative region obtained by our method and ground-truth box of object, respectively.

regions ("how many"). A semantic reward function is also introduced for this DRL scheme to joint consider the attention and category information. He *et al.* [29] further extended this work by incorporating multi-scale learning to localize regions in different scales as well as encode images at different scales. However, as discussed in Sec. 1, two challenges arise in this two-stage DRL architecture: 1) it is challenging for DRL to simultaneously localize an uncertain number of discriminative regions; 2) the classifier upon which the reward function of DRL relies is not robust enough. This paper aims at enhancing DRL-based FGVR by utilizing DRL to localize a single large discriminative region and employing self-distillation to construct a robust classifier. Experiments demonstrate the effectiveness of our novel techniques.

3. Methodology

Prior to introducing our method, we conduct an analysis of the variation in discriminative regions across different images. As depicted in Fig. 3, objects of the same subcategory may exhibit different poses, resulting in varying ratios of the discriminative region (depicted by the dashed yellow box) to the object region (depicted by the red box), exemplified by approximately 0.4 in Fig. 3(a) and 0.8 in Fig. 3(b). It is evident that objects belonging to different subcategories also exhibit diverse discriminative regions. Furthermore, since the discriminative region is determined by the classifier based on its contribution to classification performance, there are no explicit guidelines

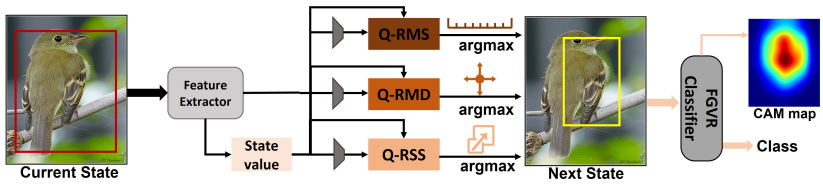

Figure 4: **Architecture of DDQN for multiple actions selection.** Upon receiving the current state, the agent predicts three actions. Upon executing these actions, the current state transitions to the next. To compute R_c, classification is performed on both the current and next states using the FGVR classifier. For calculating R_{cam}, we extract features from the last non-reduced convolutional layer of the classifier into the CAM map.

for human annotators to identify the discriminative region within a particular image, making it suboptimal to train object/part detectors for detecting discriminative regions [14, 15, 8, 9, 16, 17, 18]. In this paper, we employ DRL to adaptively localize the discriminative region, guided by the classifier through the reward function. Our work shows that localizing a single large discriminative region in our approach works better than localizing multiple discriminative regions in existing DRL-based approaches [28, 29].

3.1. Reinforced Most-Discriminative Region Localization

We continue by introducing the proposed RMDRL module. Unlike existing DRL-based approaches that primarily focus on exploring multi-scale and multi-granularity information to identify numerous discriminative regions [28, 29], our approach takes a different stance. Recognizing the challenges inherent in DRL when attempting to simultaneously localize an uncertain number of discriminative regions, we leverage a DRL methodology to autonomously and adaptively pinpoint the most discriminative region contributing most significantly to classification performance. To elaborate, we define a set of "actions" that transition from an initial region to the optimal discriminative region. The input image, accompanied by an initial region, serves as the "state" from which various actions for region adjustment are selected. The classifier plays the role of the "environment", generating evaluated results as "feedback" to calculate a "reward" for optimizing the learning process of the DRL agent. The region is denoted as a bounding box, defined by its center point along with its width and height.

In this work, we adopt a single-step DRL formulation to train the agent for adjusting the center point and width/height of the region, aiming to enhance the accuracy of the FGVR classifier. Single-step DRL algorithms

have demonstrated considerable potential as effective black-box optimizers [55, 56]. The primary distinction between standard DRL and single-step DRL lies in the states transition. Unlike standard DRL, single-step DRL involves only one state s_0, as there is a single attempt per learning episode. Single-step DRL aims to determine the optimal mapping $f_{\theta^*_{rl}}$ such that the optimal action $a^* = f_{\theta^*_{rl}}(s_0)$, where θ^*_{rl} denotes optimal parameters of the mapping function f. By consistently presenting s_0 to the agent, the learned mapping reflects the transformation from given state s_0 using the optimal action a^*. The essential components of DRL agent training are elaborated as follows.

State. As previously mentioned, within each episode, the state represents the input image along with an initial region. This region is characterized by a center point and a tuple denoting its height and width. For this study, as shown in Fig. 2(b), we define the initial region using the center point of the image and $\frac{4}{5}$ of the image's height and width.

Policy Network. It is also referred to as the "agent" in DRL. Using a standard neural network architecture, it undergoes training to predict the most effective policy. It takes current state to predict a probability distribution of actions to be taken from that state. It often contains a feature extractor responsible for generating visual representation from the input image, alongside multiple heads for state value and action prediction.

Actions. Since the region's location involves three aspects, we define three actions for its adjustment: Region-Moving Direction (RMD) for "where", Region-Moving Scalar (RMS) for "size", and Region-Scaling Scalar (RSS) for "shape". The action space for RMD is fixed at five, representing four directions (up, down, left, and right) and no movement. The action space for RMS is set to 8, with a step size of 4, allowing the center point to move from 4 to 32 pixels in a given direction. The action space for RSS is 9, with scaling ratios ranging from 0.4 to 1.2 and a step size of 0.1.

Environment and Reward. The environment in this scenario is an FGVR classifier, which can be independently trained or distilled from multiple branches. During the DRL agent learning process, we have designed a reward computation function $R(s, a)$ for the transition from the current state s to the next state $\hat{s}$. This function encourages the agent to take effective action a in adjusting the region. It achieves this by combining two aspects of rewards: (i) $R_c(s, a)$ represents the reward of class prediction from

the FGVR classifier with parameter of θ. This reward encourages the agent to take actions that enhance classification accuracy. (ii) $R_{cam}(s, a)$ indicates the reward of Intersection over Union (IoU) between the discriminative region and the "hot" part of the class activation map (CAM) [49], retrieved from the last non-reduced convolutional layer of the classifier. This reward prompts the agent to adjust the region to include, as much as possible, class-specific regions of the image (the hot region of the CAM map in Fig. 4). Hence, it accelerates the convergence of the DRL agent during training, leveraging the substantial overlap between the discriminative region and the class-specific area within the CAM map. In this work, we apply the typical method, GardCAM [57]. The reward functions are thus defined as

$$R(s, a) = \lambda R_{cam}(s, a) + R_c(s, a), \tag{1}$$

$$R_{cam}(s, a) = \text{IoU}(re_{\hat{s}}, re_{cam}) - \text{IoU}(re_s, re_{cam}), \tag{2}$$

$$R_c(s, a) = P(y|\hat{s}; \theta) - P(y|s; \theta). \tag{3}$$

Here, s is the current state, and $\hat{s}$ is the next state taking the action a from the current state s. re_s indicates the discriminative region at the state of s, while re_{cam} represents the hot part in the CAM map. y is the ground-truth class of the image. P represents the function used to compute the probability belonging to the class y when inputting the discriminative region at a state to the FGVR classifier with parameter θ. This region can be obtained by applying threshold algorithms (*e.g.*, OTSU [58]) and finding the largest connected areas. $\lambda \in (0, 1)$ is a coefficient to control the contribution of R_c and R_{cam} to the reward. Consequently, the class prediction result takes precedence in determining the reward, with the CAM region serving as a facilitator.

We utilize the Duelling Deep Q-Network (DDQN) [59] as an off-policy DRL algorithm to train the agent for selecting the three actions. In this method, the Q function, mapping a state-action pair to the expected cumulative reward, is decomposed into a state-value function and an action advantage function, as expressed in Eq. (4). The architecture of the DDQN designed for selecting multiple actions is depicted in Fig. 4. Given a state, following feature extraction, the network is divided into two streams: one for state-value estimation and the other for estimating state-dependent action advantages. The current Q-values (Q-RMD, Q-RMS, and Q-RSS) of the ac-

tions are calculated by combining the state-value and the action advantages:

$$\begin{aligned} Q(s,a;\theta_{rl}) =& V(s;\theta_{rl},\beta) + V^a(s,a;\theta_{rl},\alpha) \\ & - \frac{1}{N}\sum_{\hat{a}\in A} V^a(s,\hat{a};\theta_{rl},\alpha). \end{aligned} \tag{4}$$

where θ_{rl} is the parameter of the feature extractor in DRL, and α and β are parameters of the action advantage branch and state-value branch, respectively. V is the state-value function, which outputs a 1-d vector. V^a is the action advantage function, N is the size of the action space. A is the action set. We can estimate the optimal action a^* after optimizing the policy network with optimal parameters θ^*_{rl} by

$$a^* = \arg\max_{a\in A} Q(s,a;\theta^*_{rl}). \tag{5}$$

From Eq. (5), it is evident that the optimal action depends on the optimal parameters of the feature extractor. To optimize the parameters of the feature extractor, we use the mean square error (MSE) as the loss between the current Q-value and the target Q-value. The target Q-value on state s given action a is defined as

$$Q_{target}(s,a;\theta_{rl}) = R(s,a) + \gamma \max_{\hat{a}} Q(\hat{s},\hat{a};\theta_{rl}), \tag{6}$$

where $\hat{s}$ and $\hat{a}$ are the next state and next action. The joint loss on the three actions can be computed as:

$$L(s,a_i;\theta_{rl}) = \sum_{i=1}^{3}(Q(s,a_i;\theta_{rl}) - Q_{target}(s,a_i;\theta_{rl}))^2. \tag{7}$$

where a_i ($i \in \{1,2,3\}$) denotes three actions for Q-RMD, Q-RMS and Q-RSS computation. For instance, $Q(s,a_1;\theta_{rl})$ computes Q-RMD using Eq. (4). The three sub-losses carry equal weight since all actions (RMD, RMS, and RSS) equally contribute to maximizing the reward, thereby optimizing the policy network parameter θ_{rl}.

Eq. (1) and Eq. (3) illustrate how the performance of the FGVR classifier influences the reward computation. This, in turn, affects the optimization of the policy network regarding discriminative region adjustment, ultimately impacting the final image classification. Thus, boosting the final FGVR performance requires a robust classifier. Once the discriminative region within

an image is identified, this classifier should accurately classify it. To this end, we construct a robust classifier by introducing a DKSD module in the next section.

3.2. Discriminative Knowledge Self-Distillation

To ensure the robust training of the RMDRL module, a robust FGVR classifier is indispensable. To address this necessity, we introduce the DKSD module. Our strategy involves training multiple classifiers, and the averaged predictions from these classifiers serve as a robust prediction. Drawing inspiration from ensemble learning theory [60], we enhance the ensemble performance by intentionally increasing the variance among individual models. To achieve this, the classifiers process distinct predefined discriminative regions, each centered around the image's center but with varying height and width. Subsequently, we leverage the knowledge distillation technique [61, 62, 63, 64] to amalgamate knowledge from these individual classifiers, culminating in a robust single classifier capable of making decisive predictions. This process is depicted in Fig. 2(a). Concretely, multiple inputs featuring predefined discriminative regions are fed into a shared feature extractor (*e.g.*, ResNets [65]). The resulting features are then transmitted to multiple classifiers and implemented using a multi-branch architecture.

Individual classifiers. A fundamental challenge in this module is determining the number of individual classifiers, which corresponds to the number of branches. In Fig. 2, we illustrate five branches as an example. Theoretically, incorporating more branches in the module would enhance the performance of the distilled classifier by aggregating knowledge from a larger pool of individual classifiers and predefined discriminative regions. However, the increased number of branches also demands more computational resources. In practical terms, deciding on the number of branches involves striking a balance between accuracy and efficiency.

Empirically, when the incremental gain in learned knowledge, measured by the accuracy of the distilled classifier, becomes negligible due to an increase in the number of branches (*e.g.*, from n to $n+1$), the number of branches can be set at that level (*e.g.*, n). Regarding the range of predefined regions, assuming the number of branches is n, the design involves using the entire image as the first region and progressively shrinking it $n-1$ times by a certain height/width. Subsequently, the n branches can be trained simultaneously. As shown in Fig. 2(b), each branch takes an image and one

of the predefined regions within it as input. For an image x accompanied with different predefined regions, different branches may predict its category as different. Given image x and branch j with learnable parameter θ_j, the probability of class i is calculated using the formula:

$$P(i|x, \theta_j) = \exp(w_{ij} z_i) / \sum_{k=1}^{C} exp(w_{kj} z_k), \tag{8}$$

in which z_i represents the normalized logits for class i, w_{ij} is the importance score of branch j on z_i, and C is the total number of classes in the dataset. Specifically, the computation of w_{ij} depends on KD methods, for instance, in [63], the gate module uses features from the second block of its backbone network as input to generate the importance score of the corresponding branch. The common cross-entropy loss is employed to measure the difference between the predicted probability of class i by branch j and the ground-truth label y. Consequently, the average loss over the n branches is defined as

$$L_{MB} = \frac{1}{n} \sum_{j=1}^{n} L_{CE}^{j}(P(i|x; \theta_j), y), \tag{9}$$

where L_{CE} denotes the cross-entropy loss function.

Knowledge self-distillation. Through the optimization of the loss function in Eq. (9), individual classifiers are trained to predict the correct class with maximum likelihood. To facilitate effective knowledge transfer, we aggregate knowledge from multiple branches, thereby enhancing the distilled classifier's performance. The aggregated prediction is computed by first averaging the logits from individual branches (Eq. (10)) and then calculating the aggregated softmax probability of input image x for class i (Eq. (11)):

$$s_i = \frac{1}{n} \sum_{j=1}^{n} w_{ij} z_j, \tag{10}$$

$$P_{agg}(i|x) = \frac{exp(s_i)}{\sum_{k=1}^{C} exp(s_k)}. \tag{11}$$

While the accuracy of the aggregated prediction surpasses that of each single classifier, it comes at the cost of n times higher computational complexity. To expedite the inference process, we implement self-distillation from the

aggregated prediction $P_{agg}(\cdot|x)$ to another single classifier σ with parameter θ_{kd} (namely distilled classifier). The Kullback-Leibler (KL) divergence from the aggregated prediction to the distilled classifier is employed to quantify the difference between $P_{agg}(\cdot|x)$ and $\sigma(x)$ in their predictions:

$$L_{KL} = D_{KL}(P_{agg}(\cdot|x), \sigma(x)), \tag{12}$$

where $\sigma(x) = P(\cdot|x; \theta_{kd})$. The loss between the prediction of the distilled classifier and the ground truth is defined as cross-entropy loss:

$$L_{KD} = L_{CE}(\sigma(x), y). \tag{13}$$

Therefore, the overall loss function for distilling from the aggregated prediction to the distilled classifier is

$$L_{total} = L_{MB} + \tau L_{KL} + L_{KD}. \tag{14}$$

where τ is a hyper-parameter ($\tau \in (0, 1]$), ensuring that the soft loss (L_{KL}) and the hard loss (L_{KD}) contribute almost equally to the gradient magnitudes.

Following the incorporation of knowledge self-distillation, the distilled classifier σ amalgamates insights derived from various individual classifiers. During testing, if any of the trained individual classifiers can accurately categorize a given image, there is a strong likelihood that the distilled classifier will also accurately classify it [61, 62]. Consequently, the distilled classifier demonstrates exceptional performance. Ultimately, the distilled classifier serves the role of the FGVR classifier within the RMDRL module. Specifically, in Eq. (3), the parameter θ should be replaced with θ_{kd}.

3.3. Inference Process

During the inference process, an input image with an initial region is fed into the RMDRL module for discriminative region identification (refer to the definition of 'initial region' in the 'State' section of Sec. 3.1). The image, along with the identified discriminative region, is then input into the robust FGVR classifier, which has been trained using our DKSD module, for final object classification.

Table 1: Statistics of datasets and their training/test splits. "BBox" indicates whether this dataset provides object bounding box.

Dataset	Meta-class	#Train	#Test	#Class	BBox
Aircraft	Aircrafts	6667	3333	100	√
CUB-200	Birds	5994	5794	200	√
Stanford Cars	Cars	8144	8041	196	√
Stanford Dogs	Dogs	12000	8586	120	√
Oxford Flower	Flowers	2040	6194	102	×
BMW-10	Cars	258	254	10	√
Food-101	Food dishes	75750	25250	101	×
Oxford Pets	Dogs & cats	3680	3669	37	√
NaBirds	Birds	23929	24633	555	√

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate the performance of our method on nine FGVR benchmark datasets, including *Aircraft* [4], *CUB-200 Birds* [1], *Stanford Cars* [3], *Stanford Dogs* [2], *Oxford Flower* [66], *BMW-10* [67], *Food-101* [68], *Oxford Pets* [69], and *NaBirds* [70]. The dataset details are provided in Table 1. Notably, our method does not rely on ground-truth bounding box annotations of objects or object parts, making it applicable to datasets without such annotations, like *Oxford Flower* and *Food-101* datasets.

Implementation Details. We use PyTorch [80] to implement our method. ResNet-50 and ResNet-101 [65] serve as backbones for training FGVR classifiers, aligning with previous works [74, 24, 40, 72]. The policy network in the RMDRL module comprises five convolution layers, following the architecture in [55]. The three branches, Q-RMD, Q-RMS, and Q-RSS, corresponding to three action predictions, each contain one linear layer. In our predefined regions, we utilize the entire image as the first predefined region, ensuring that at least one predefined region contains the discriminative object region regardless of the object's position. Subsequently, the region is successively reduced by $\frac{1}{10}$ of the height/width for $n-1$ times (where n is the number of branches used) to generate other predefined regions.

In all our experiments, we resize input images to 224×224. Data augmentation includes random horizontal flipping, while random cropping is omitted to prevent potential loss of discriminative information in the image.

Table 2: Comparison of performance (Top-1 accuracy in %) against three methods: the recent state-of-the-art method, the leading attention-based method, and the DRL-based method.

Aircraft		**CUB-200 Birds**		**Stanford Cars**	
M2DRL [29]	-	M2DRL [29]	87.2	M2DRL [29]	93.3
SJFT* [71]	-	SJFT* [71]	-	SJFT* [71]	-
P2P-Net[72]	94.2	P2P-Net[72]	90.2	P2P-Net[72]	95.4
SMGVT [73]	-	SMGVT [73]	91.6	SMGVT[73]	-
SR_GNN [47]	95.4	SR_GNN [47]	91.9	SR_GNN[47]	96.1
CAP [74]	94.9	CAP [74]	91.0	CAP[74]	95.7
Ours_Res50	95.5	Ours_Res50	92.2	Ours_Res50	**96.8**
Ours_Res101	**95.9**	Ours_Res101	**92.6**	Ours_Res101	96.6
Stanford Dogs		**Oxford Flower**		**BMW-10**	
M2DRL [29]	-	M2DRL [29]	-	M2DRL [29]	-
SJFT* [71]	90.3	SJFT* [71]	97.0	SJFT* [71]	-
CPM* [75]	97.1	MC*loss [76]	97.7	G-PCANet [77]	57.9
SMGVT [73]	92.3	SMGVT [73]	-	SMGVT[73]	-
SR_GNN [47]	**97.3**	SR_GNN [47]	97.9	SR_GNN[47]	-
CAP [74]	96.1	CAP [74]	97.7	LE-PCANet* [77]	71.6
Ours_Res50	96.0	Ours_Res50	97.9	Ours_Res50	**74.2**
Ours_Res101	96.5	Ours_Res101	**98.3**	Ours_Res101	74.0
Food-101		**Oxford Pets**		**NaBirds**	
M2DRL[29]	-	M2DRL [29]	-	M2DRL [29]	-
SJFT* [71]	-	SJFT* [71]	-	SJFT* [71]	-
GPipe[78]	93.0	GPipe [78]	95.9	DSTL* [79]	87.9
SMGVT [73]	-	SMGVT [73]	-	SMGVT[73]	90.6
SR_GNN[47]	-	SR_GNN [47]	-	SR_GNN[47]	91.2
CAP [74]	**98.6**	CAP [74]	97.3	CAP [74]	91.0
Ours_Res50	97.8	Ours_Res50	97.0	Ours_Res50	**91.5**
Ours_Res101	98.2	Ours_Res101	**97.9**	Ours_Res101	91.1

* Methods marked with * use secondary dataset besides primary dataset. The best accuracy in each dataset is in bold.

For both stages (Fig. 2(a) and (b)) of training, the optimizer is Adam [81] with a learning rate of 0.001, a batch size of 64, and training epochs of 100 and 500 for the former and latter, respectively. During the agent training, we incorporate prioritized experience replay [82] to sample experiences from a

replay buffer. The replay memory has a capacity of 100,000, a discount factor (γ) of 0.99 for future rewards, and the target Q-value and current Q-value are initialized with random weights. The former is updated by the latter, loading its learned weights after every 40 episodes. To encourage exploration across action spaces, we employ the ϵ-greedy policy [83]. The RMDRL training involves 5-fold cross-validation, randomly splitting the dataset into five parts, designating one part for testing, and using the remaining four for training in each fold. The final accuracy is the average across the 5-folds. All experiments are conducted on an NVIDIA RTX A5000 GPU.

4.2. Performance Comparison

As previously discussed, to demonstrate the efficacy of our approach, we compare our method with two closely related groups of existing approaches: attention-based approaches (*e.g.*, CAP[74]) and DRL-based approaches (*e.g.*, M2DRL[29]). In addition, we add one more recent state-of-the-art method for each dataset, for instance, P2P-Net [72] on *Aircraft* [4], *CUB-200 Birds* [1], and *Stanford Cars* [3] datasets; and CPM* [75] on *Stanford Dogs* [2].

The comparison results are shown in Table 2. Our method demonstrates superior overall performance compared to state-of-the-art approaches, outperforming the most recent P2P-Net [72], the top-performing attention-based method CAP [74], and the DRL-based method M2DRL [29] across all datasets, except for *Stanford Dogs* and *Food-101*. The performance is achieved using various backbone architectures. This illustrates the efficacy of our proposed method in discerning subtle changes for fine-grained object recognition. Notably, among all the datasets, only BMW-10 is an ultra-fine-grained dataset, consisting exclusively of 10 classes of BMW cars. Hence, the performance on this dataset is lower than that of other datasets.

On the *Stanford Dogs* dataset, the leading method is CPM* [75], which utilizes additional datasets (secondary datasets such as ImageNet and COCO) for joint/transfer learning of objects, patches, and regions. In contrast, our method exclusively employs the primary dataset focused on fine-grained object recognition. Regarding the *Food-101* dataset, CAP [74] achieves slightly better performance with a 0.4% advantage over our method. This discrepancy can be attributed to the fact that discriminative features in most *Food-101* images are uniformly distributed and repeated across the entire images. Consequently, identifying the discriminative region on each image becomes more flexible and easier. This is illustrated by the samples in Fig. 5, where (a) from *CUB-200* [1] exhibits centralized discriminative regions at the throat

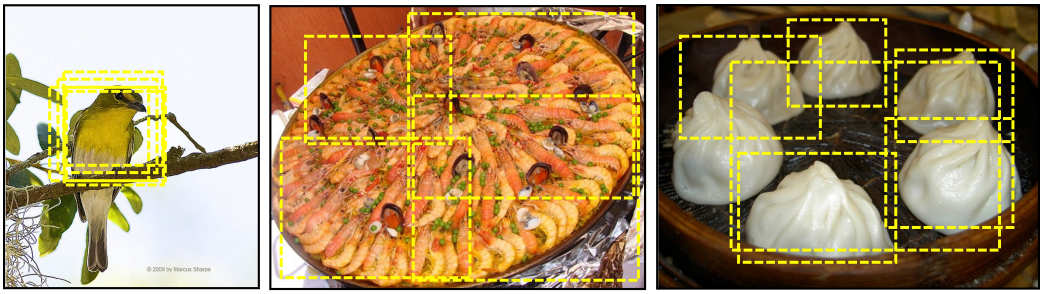

Figure 5: **Alternative discriminative regions** (in dotted yellow boxes) on images (a) from *CUB-200* [1], (b) and (c) from *Food-101* [68].

Table 3: Comparison of trainable parameters (1^{st} row), training time (2^{nd} row) and GPU memory usage (3^{rd} row) between CAP and our method with ResNet-50 [65] as the backbone.

Backbone	CAP	Ours
Res50	36.9M [74]	23.0M + 5.03M
Res50	114 sec [74]	296 sec+14 sec
Res50	12GB [74]	10/24GB

Table 4: Recognition accuracy of the final classifier by distilling knowledge from various number of weak classifier (branch).

Branch No.	Aircraft	CUB-200	Stanfords Cars	Stanfords Dogs	Oxford Flowers	BMW-10	Food-101	Oxford Pets	NaBirds
3	0.932	0.902	0.950	0.948	0.961	0.693	0.944	0.954	0.880
4	0.948	0.918	0.960	0.959	0.976	0.725	0.967	0.968	0.904
5	**0.959**	**0.926**	**0.966**	**0.965**	**0.983**	**0.740**	**0.982**	**0.979**	**0.915**
6	0.961	0.925	0.968	0.966	0.980	0.743	0.979	0.974	0.917

part of the bird with small offset variations in shape and size, making them less apparent. Conversely, (b) and (c) from *Food-101* [68] showcase alternative discriminative regions across the entire images, implying that identifying any region leads to correct image recognition. Hence, the identification of discriminative regions is not particularly challenging on the *Food-101* dataset, suggesting that our method does not offer a significant advantage over state-of-the-art methods in this context. This highlights a limitation of our work.

In the detailed dataset-wise analysis, our method consistently exhibits superior performance compared to the state-of-the-art approach CAP [74] and other top-performing methods: Our method achieves a 1% accuracy improvement over CAP on *Aircraft*. Our method outperforms CAP by 1.6%

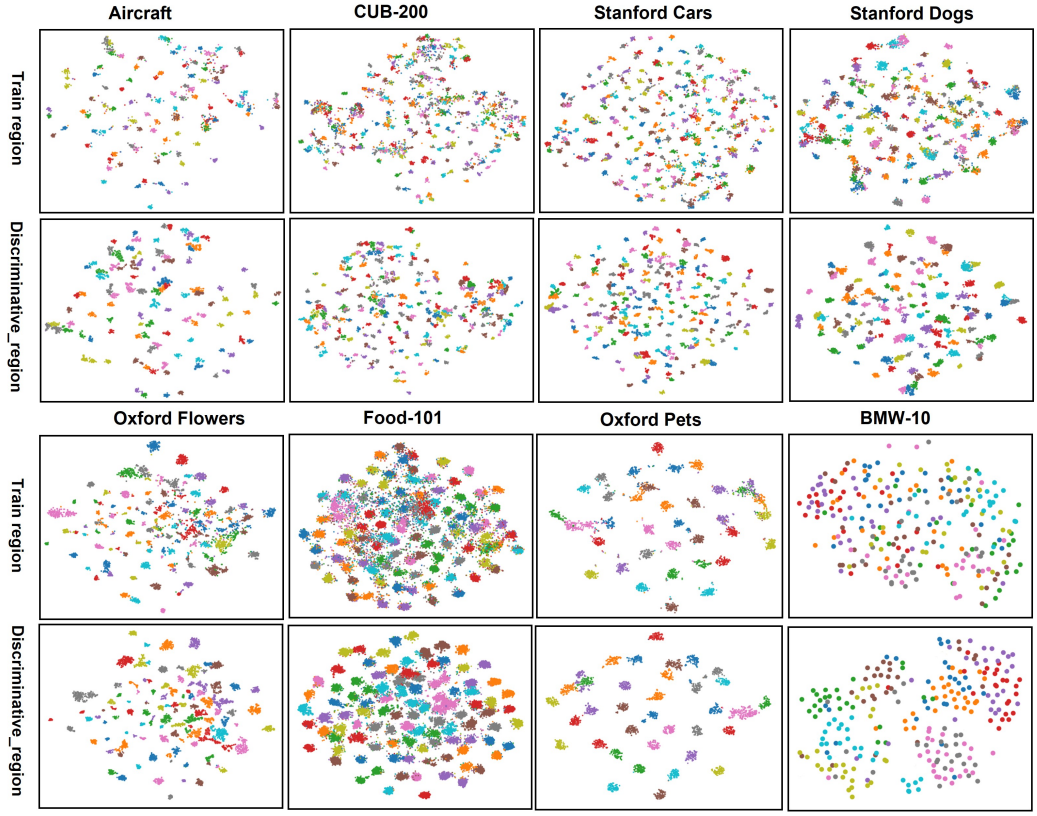

Figure 6: **Comparison of discriminability.** This is conducted by forwarding testing images, appending Train_region (upper row) and Discriminative_region (lower row), to an FGVR classifier. Visualization is accomplished using t-SNE to depict class separability and compactness in testing images across all datasets, excluding NaBirds. Notably, the lower row in each dataset exhibits more distinct and compact clusters, indicating clearer distinctions among clusters representing diverse subcategories.

on *CUB-200*, demonstrating its effectiveness on fine-grained bird recognition. With a 1.1% improvement on *Stanford Cars*, our method surpasses the accuracy attained by CAP. Our method achieves a 0.4% performance gain over the best-performing method (CAP) on *Stanford Dogs*, which exclusively uses primary data. Our method slightly outperforms the state-of-the-art method CAP in achieving accurate flower recognition on *Oxford Flower*. Our approach demonstrates a remarkable 2.6% accuracy boost over LE-PCANet* [77], the best-performing method in recognizing BMW models. While our method is slightly below CAP by 0.4% accuracy *Food-101*, it significantly

Table 5: Comparison among evaluation accuracy of five independently trained classifiers with different regions on testing images of all datasets. The results in columns from FGVR_1 to FGVR_5 are their evaluation accuracy by using corresponding Train_regions, results marked in blue indicates the best accuracy. The results in column Avg_CAM are the average accuracy of testing the five classifiers with CAM regions. The results in column Avg_RL are the average accuracy of evaluating the five classifiers with the RL agent tuned regions.

Dataset	FGVR_1	FGVR_2	FGVR_3	FGVR_4	FGVR_5	Avg_CAM	Avg_RL
Aircraft	0.893	0.905	0.887	0.898	0.901	0.911	**0.928**
CUB-200	0.858	0.862	0.866	0.860	0.856	0.864	**0.905**
Stanford Cars	0.916	0.924	0.919	0.908	0.921	0.915	**0.942**
Stanford Dogs	0.885	0.894	0.887	0.890	0.906	0.914	**0.948**
Oxford Flower	0.912	0.909	0/916	0.925	0.918	0.928	**0.944**
BMW-10	0.672	0.696	0.681	0.705	0.702	0.708	**0.732**
Food-101	0.921	0.934	0.928	0.919	0.911	0.937	**0.955**
Oxford Pets	0.919	0.928	0.930	0.927	0.922	0.925	**0.953**
NaBirds	0.852	0.859	0.868	0.855	0.849	0.870	**0.889**

outperforms the second-best method, GPipe [78], by 4.8%. Our method outshines CAP by 0.6% in accurate pet classification on *Oxford Pets*. Achieving a 0.5% higher accuracy than CAP on *NaBirds*, our method excels in recognizing fine-grained bird species.

Specifically, the representative DRL-based method, M2DRL[29] was exclusively implemented on two datasets: *CUB-200* and *Stanford Cars*. One can observe a clear performance margin favoring our method over M2DRL on these two datasets (5% on *CUB-200* and 3.3% on *Stanford Cars*). This aligns with our earlier discussion, highlighting that the segmentation of discriminative characteristics into multiple parts within M2DRL adds complexity to the localization process, potentially increasing the risk of overlooking crucial elements.

Notably, on *Oxford Flower*, *BMW-10* and *NaBirds*, our method outperforms approaches MC*loss [76], LE-PCANet* [77] and DSTL* [79] that rely on secondary datasets, showcasing the robustness and effectiveness of our proposed method.

The aforementioned observations across all datasets highlight the data efficiency of our method, as it requires no additional secondary datasets. Moreover, our model (28.03M parameters) is more parameter-efficient compared to CAP (36.9M parameters), as indicated in Table 3. Despite our method comprising two stages, it avoids the use of additional subnetworks, and the DRL agent training policy network [55] is notably shallow, resulting in fewer trainable parameters compared to CAP. Besides, we compared

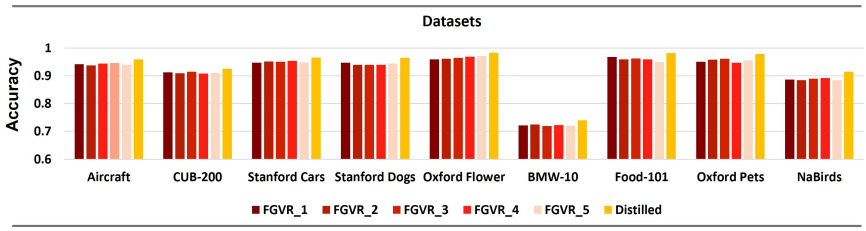

Figure 7: **Comparison between individual classifiers and the distilled classifier.** All classifiers receive images with bounding boxes fine-tuned by the DRL agent, which are specifically trained to align with the preferences of individual FGVR classifiers.

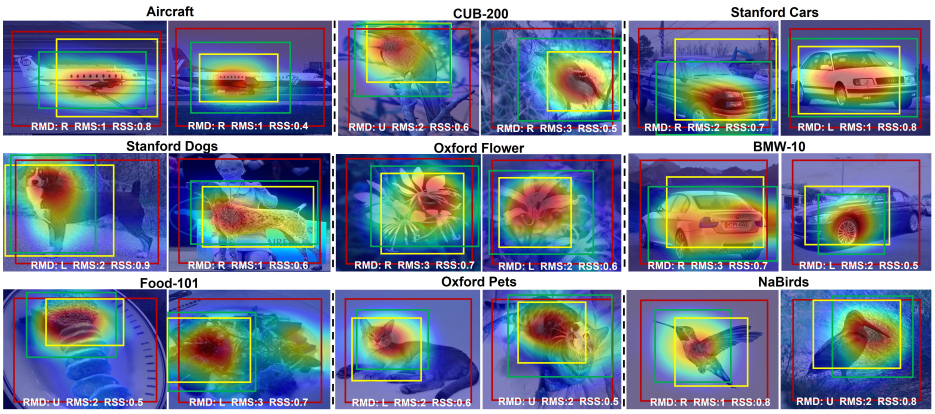

Figure 8: **Visualization of Discriminative region adaptation**. For each dataset, we present two samples overlaid with CAM regions (in green boxes), initial regions (in red boxes), and RMDRL-tuned discriminative regions (in yellow boxes). The symbols N, U, D, L, and R indicate that the region's moving direction is "none", "up", "down", "left", and "right", respectively. The associated values with RMS and RSS represent the region-moving scalar towards a direction and the region-scaling scalar.

training time and GPU memory usage between CAP and our method with 5 individual classifiers in DKSD module on Aircraft dataset in Table 3. Notably, our method requires training 5 classifiers simultaneously within the DKSD module, resulting in a significantly higher training time compared to CAP.

In essence, our approach relies on the capability of any individually trained classifier to accurately categorize an image when provided with a discriminative region. Only in such cases can the distilled classifier ensure correct classification. The RMDRL module plays a pivotal role in automatically adjusting an initial region towards the discriminative region. It is crucial

to note, however, that if the distilled classifier itself struggles to recognize an image even with the discriminative region, our method may fail to classify the image even if the DRL agent successfully identifies the discriminative region.

4.3. Ablation Study

We conducted three ablation studies to examine the impact of different factors on our method's performance: the influence of the number of individual classifiers, the effectiveness of RMDRL, and the efficacy of DKSD. The results for the impact of the number of individual classifiers are presented in Table 4. The outcomes for the other two ablation studies are discussed in Sec. 4.3.2 and Sec. 4.3.3, respectively. These studies analyze the contributions of four components: (1) Base component (without the use of RMDRL and DKSD), represented by the first five columns of Table 5. (2) Base component with RMDRL, illustrated by the first five bars on each dataset in Fig. 7. (3) Base component with both RMDRL and DKSD, represented by the last bar on each dataset in Fig. 7. The detailed results of these ablation studies are as follows.

4.3.1. Impact of the number of individual classifiers

The number of individual classifiers is a critical parameter, and we conducted experiments to analyze its impact (Table 4). The recognition accuracy of the distilled classifier changes with the number of individual classifiers, showing the following trends: (1) A significant improvement is observed across all datasets when the number of individual classifiers increases from 3 to 4. (2) The improvement rate slows down when the number of individual classifiers goes from 4 to 5. (3) Setting the number of individual classifiers to 6 results in insignificant accuracy improvements on *Aircraft* (0.2%), *Stanford Cars* (0.2%), *Stanford Dogs* (0.1%), *BMW-10* (0.3%), and *NaBirds* (0.2%). Additionally, there is a slight accuracy drop on *CUB-200* (-0.1%), *Oxford Flower* (-0.3%), *Food-101* (-0.5%), and *Oxford Pets* (-0.5%). According to the computation of predefined regions described in 'Implementation Details', the fifth region in an image contains only half of the original image's content, and the sixth region contains even less. Due to variations in data distribution across datasets like CUB-200 and Oxford Flowers, the center of the discriminative region may not align with the center of the image. Thus, the sixth region is less likely to provide additional useful information for the individual classifiers to aggregate. This, combined with training randomness, can lead

to a slight performance drop on some datasets when the number of classifiers exceeds five, explaining trend (3).

After careful consideration of accuracy and model complexity, we determine that the optimal number of individual classifiers for the DKSD module is 5. Subsequently, individual classifiers are named from FGVR_1 to FGVR_5 in the following studies.

4.3.2. Effectiveness of RMDRL

Table 5 provides a comprehensive accuracy comparison among independently trained individual FGVR classifiers using three types of regions as inputs. The columns FGVR_1 to FGVR_5 (base component) present results from classifiers trained with Train_regions corresponding to five areas employed for training five individual classifiers. The Avg_FGVR column shows the average values obtained from FGVR_1 to FGVR_5. The Avg_CAM column displays the average accuracy obtained by feeding CAM regions to the five classifiers. The Avg_RL column values represent the average results from classifiers with the discriminative region which is tuned by RMDRL module.

Across all datasets, the Avg_RL (Base component with RMDRL) consistently outperforms the best result (highlighted in blue) from columns FGVR_1 to FGVR_5 and the one in column Avg_CAM by a significant margin. This indicates that the RMDRL module effectively adjusts the initial region on a testing image, providing better discriminative characteristics tailored to a specific classifier and consequently improving recognition accuracy. Further more, comparing the Avg_RL value with the results of M2DRL on the CUB-200 and Stanford Cars datasets (in Table 2) reveals that the Avg_RL is 0.905 on CUB-200, compared to M2DRL's 0.872, and 0.942 on Stanford Cars, compared to M2DRL's 0.933. This comparison demonstrates that using a single region produces better results.

In examining individual classifiers (FGVR_1 to FGVR_5), the accuracy differences are not significant due to variations in object size, pose, and location across the dataset. The dilemma of enhancing accuracy on some objects at the cost of accuracy on others is observed, depending on the selected regions. This trade-off is mitigated by employing the RMDRL module in our method.

For qualitative analysis, we investigated the impact of the RMDRL's tuning on the initial region to enhance classifier discriminability. Testing images from all datasets (except *NaBirds*) were used for this purpose. An individual FGVR classifier (*e.g.*, FGVR_2) was chosen as an example, and discrim-

inability was compared when fed with Train_region and the RMDRL-tuned discriminative region. The discriminability comparison in Fig. 6 reveals that forwarding the discriminative region as input results in clusters that are farther apart and more compact, enhancing differentiation among various clusters denoting different subcategories. This improvement is observed across all datasets, underscoring the importance of the RMDRL's tuning for boosting recognition performance.

4.3.3. Effectiveness of DKSD

We conducted a comparison between individual classifiers (Base component with RMDRL) and the distilled classifier (Base component with RMDRL and DKSD) by utilizing the discriminative region tuned by RMDRL module. The results, as illustrated in Fig. 7, reveal that the best accuracy among individual classifiers varies across different datasets. For instance, FGVR_4, FGVR_3, and FGVR_4 performed the best on *Aircraft*, *CUB-200*, and *Stanford Cars*, respectively, while FGVR_1, FGVR_5, and FGVR_2 achieved the highest accuracy on *Stanford Dogs*, *Oxford Flower*, and *BMW-10*.

This variability is attributed to differences in instance size across datasets. Importantly, the DKSD module plays a crucial role in aggregating the learned knowledge from multiple individual classifiers into the distilled classifier. This distilled classifier consistently achieves the highest accuracy across all datasets, underscoring the effectiveness of the knowledge distillation module in enhancing the overall performance of our method.

4.4. Qualitative Analysis

In Fig. 8, we present qualitative results showcasing discriminative regions adapted from different datasets. Each sample is overlaid with a CAM region (green boundary), an initial region (red boundary), and the RMDRL-tuned discriminative region (yellow boundary). The results vividly illustrate the capability of the RMDRL (DRL adapter) to fine-tune the initial region towards the discriminative region, encompassing the majority of informative characteristics (pixels of hot color in the CAM region) and an appropriate portion of context information.

A comparison between two examples from the same dataset reveals variations in the discriminative region across images within the same subcategory. Furthermore, we observe that even when the Intersection over Union (IoU) between the CAM region and the RMDRL tuned discriminative region is

large, their center point, shape, and size may differ. This underscores that the CAM region serves as a facilitator for identifying the discriminative region but is not equivalent to it. Thus, the visualization results provide additional confirmation of the imperative role played by the RMDRL module in learning for effective region tuning.

The visualization results reveal that, although we identify only one discriminative region, the RMDRL module ensures that this tuned region captures the most discriminative features while minimizing the inclusion of background elements. This demonstrates the module's effectiveness in focusing on the critical parts of the image for accurate classification, thereby enhancing the overall performance despite working with a single region. This supports the idea that 'one could be better than many.

4.5. Discussion

In the DKSD module, we leverage self-distillation to enhance the classifier's performance. An alternative approach to knowledge aggregation is the use of an *ensemble* [84, 85]. Ensemble methods have indeed demonstrated success by improving classifier performance through the averaging of outputs from multiple independently trained neural networks [61, 62, 63]. In contrast, knowledge distillation involves training a single model to match the output of the ensemble by minimizing a loss, as depicted in Eq. (12). Since this loss generally cannot be minimized to 0, the accuracy of the model based on knowledge distillation may experience a slight reduction.

We opt for self-distillation in our framework for two primary reasons. Firstly, it enhances efficiency in the inference stage. Ensembles demand considerable computational resources during inference, as they involve running inference on multiple individually trained models. Secondly, self-distillation reduces the complexity in learning the adaptation of the discriminative region. Ensemble models generate multiple classifiers with potential variations in performance across images. Consequently, adapting the region adjustment process would need to account for the preferences of multiple models, introducing additional learning complexity. It is important to note that, as shown in Section 1, we do not claim to have designed a new KD module itself, but rather that we have effectively applied the existing KD approach in a novel way to enhance the overall system's performance.

5. Conclusion

In this work, we employ the localize-to-classify paradigm for FGVR: first localizing the discriminative region in the image and then classifying it. We observe that the neural network's ability to accurately recognize objects is highly dependent on the specific discriminative regions it utilizes, and these regions can vary greatly even within the same object category. Existing approaches [28, 29] tackle this issue by leveraging DRL to simultaneously determine which and how many regions are discriminative, which introduces inherent complexities for the DRL agent. To this end, we introduce a Reinforced Most-Discriminative Region Localization (RMDRL) module to infer a single, most discriminative region in the image, with guidance/feedback from a trained FGVR classifier. To guarantee that the FGVR classifier can accurately classify an image once its discriminative region is identified, we construct a robust classifier by introducing a Discriminative Knowledge Self-Distillation (DKSD) module. It aggregates knowledge about discriminative regions from multiple individual classifiers with different predefined coarse bounding boxes, resulting in a distilled, more robust classifier. Our experiments, including comparisons against recent state-of-the-art methods and ablation studies on nine datasets, demonstrate the effectiveness of our approach for FGVR. Notably, our findings support the contention that a single discriminative region localized by DRL can outperform multiple discriminative regions, as advocated in prior works such as [28, 29], in the context of FGVR.

Acknowledgement

This work was supported by the Agency for Science, Technology and Research (A*STAR) AME Programmatic Grant No. A18A2b0046, Singapore.

References

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD birds-200-2011 dataset, Tech. Report.

[2] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel datasets for fine-grained image categorization, in: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. (CVPRW), Citeseer, 2011, pp. 806–813.

[3] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: Int. Conf. Comput. Vis. Worksh. (ICCVW), 2013, pp. 554–561.

[4] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv preprint arXiv:1306.5151.

[5] S. Jiang, W. Min, L. Liu, Z. Luo, Multi-scale multi-view deep feature aggregation for food recognition, IEEE Trans. Image Process. (TIP) 29 (2019) 265–276.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2009, pp. 248–255.

[7] Z. Lin, J. Jia, W. Gao, F. Huang, Fine-grained visual categorization of butterfly specimens at sub-species level via a convolutional neural network with skip-connections, Neurocomputing 384 (2020) 295–313.

[8] F. Zhou, Y. Lin, Fine-grained image classification by exploring bipartite-graph labels, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 1124–1133.

[9] Y. Chen, Y. Bai, W. Zhang, T. Mei, Destruction and construction learning for fine-grained image recognition, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2019, pp. 5157–5166.

[10] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, F. Huang, Attention convolutional binary neural tree for fine-grained visual categorization, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2020, pp. 10468–10477.

[11] Z.-C. Zhang, Z.-D. Chen, Y. Wang, X. Luo, X.-S. Xu, A vision transformer for fine-grained classification by reducing noise and enhancing discriminative information, Pattern Recognition 145 (2024) 109979.

[12] W. Li, Z. Li, X. Yang, H. Ma, Causal-vit: Robust vision transformer by causal intervention, Engineering Applications of Artificial Intelligence 126 (2023) 107123.

[13] Q. Zhu, Z. Li, W. Kuang, H. Ma, A multichannel location-aware interaction network for visual classification, Applied Intelligence 53 (20) (2023) 23049–23066.

[14] B. Yao, A. Khosla, L. Fei-Fei, Combining randomization and discrimination for fine-grained image categorization, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2011, pp. 1577–1584.

[15] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, P. N. Belhumeur, Birdsnap: Large-scale fine-grained visual categorization of birds, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2014, pp. 2011–2018.

[16] M. Lam, B. Mahasseni, S. Todorovic, Fine-grained recognition as HSnet search for informative image parts, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2017, pp. 2520–2529.

[17] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based R-CNNs for fine-grained category detection, in: Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 834–849.

[18] S. Huang, Z. Xu, D. Tao, Y. Zhang, Part-stacked CNN for fine-grained visual categorization, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 1173–1182.

[19] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, Y. Lin, Fully convolutional attention networks for fine-grained recognition, arXiv preprint arXiv:1603.06765.

[20] J. Fu, H. Zheng, T. Mei, Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2017, pp. 4438–4446.

[21] P. Rodríguez, J. M. Gonfaus, G. Cucurull, F. XavierRoca, J. Gonzalez, Attend and rectify: A gated attention mechanism for fine-grained recovery, in: Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 349–364.

[22] B. Zhao, X. Wu, J. Feng, Q. Peng, S. Yan, Diversified visual attention networks for fine-grained object classification, IEEE Trans. Multimedia (TMM) 19 (6) (2017) 1245–1256.

[23] H. Zheng, J. Fu, T. Mei, J. Luo, Learning multi-attention convolutional neural network for fine-grained image recognition, in: Int. Conf. Comput. Vis. (ICCV), 2017, pp. 5209–5217.

[24] Y. Peng, X. He, J. Zhao, Object-part attention model for fine-grained image classification, IEEE Trans. Image Process. (TIP) 27 (3) (2017) 1487–1500.

[25] H. Li, M. Li, Q. Peng, S. Wang, H. Yu, Z. Wang, Correlation-guided semantic consistency network for visible-infrared person re-identification, IEEE Transactions on Circuits and Systems for Video Technology 34 (6) (2024) 4503–4515.

[26] T. Chen, L. Lin, R. Chen, Y. Wu, X. Luo, Knowledge-embedded representation learning for fine-grained image recognition, in: IJCAI, 2018, p. 627–634.

[27] F. Fang, L. Li, H. Zhu, J.-H. Lim, Combining Faster R-CNN and model-driven clustering for elongated object detection, IEEE Trans. Image Process. (TIP) 29 (2019) 2052–2065.

[28] X. He, Y. Peng, J. Zhao, StackDRL: Stacked deep reinforcement learning for fine-grained visual categorization., in: Int. Joint Conf. Artif. Intell. (IJCAI), 2018, pp. 741–747.

[29] X. He, Y. Peng, J. Zhao, Which and how many regions to gaze: Focus discriminative regions for fine-grained visual categorization, Int. J. Comput. Vis. (IJCV) 127 (2019) 1235–1255.

[30] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, D. Metaxas, SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 1143–1152.

[31] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Annu. Conf. Neur. Inform. Process. Syst. (NeurIPS), 2015, pp. 91–99.

[32] X.-S. Wei, C.-W. Xie, J. Wu, C. Shen, Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization, Pattern Recogn. 76 (2018) 704–714.

[33] X. He, Y. Peng, Weakly supervised learning of part selection model with spatial constraints for fine-grained image classification, in: AAAI Conf. Artif. Intell. (AAAI), 2017, pp. 4075–4081.

[34] Z. Lin, J. Jia, F. Huang, W. Gao, A coarse-to-fine capsule network for fine-grained image categorization, Neurocomputing 456 (2021) 200–219.

[35] C. Huang, Z. He, Task-driven progressive part localization for fine-grained object recognition, in: IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1–9.

[36] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, J. Jiao, Selective sparse sampling for fine-grained image recognition, in: Int. Conf. Comput. Vis. (ICCV), 2019, pp. 6599–6608.

[37] Z. Huang, Y. Li, Interpretable and accurate fine-grained recognition via region grouping, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2020, pp. 8662–8672.

[38] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, Z. Zhang, The application of two-level attention models in deep convolutional neural network for fine-grained image classification, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2015, pp. 842–850.

[39] J. Ji, Y. Guo, Z. Yang, T. Zhang, X. Lu, Multi-level dictionary learning for fine-grained images categorization with attention model, Neurocomputing 453 (2021) 403–412.

[40] Y. Zhao, K. Yan, F. Huang, J. Li, Graph-based high-order relation discovery for fine-grained recognition, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2021, pp. 15079–15088.

[41] H. Choi, K. Cho, Y. Bengio, Fine-grained attention mechanism for neural machine translation, Neurocomputing 284 (2018) 171–176.

[42] S. Wang, Z. Wang, H. Li, J. Chang, W. Ouyang, Q. Tian, Accurate fine-grained object recognition with structure-driven relation graph networks, Int. J. Comput. Vis. (IJCV) 132 (1) (2024) 137–160.

[43] S. Wang, J. Chang, Z. Wang, H. Li, W. Ouyang, Q. Tian, Content-aware rectified activation for zero-shot fine-grained image retrieval, IEEE Trans Pattern Anal Mach Intell (2024) 4366–4380.

[44] X. He, Y. Peng, J. Zhao, Fast fine-grained image classification via weakly supervised discriminative localization, IEEE Trans. Circ. Syst. Video Technol. (TCSVT) 29 (5) (2018) 1394–1407.

[45] R. Du, D. Chang, A. K. Bhunia, J. Xie, Z. Ma, Y.-Z. Song, J. Guo, Fine-grained visual classification via progressive multi-granularity training of jigsaw patches, in: ECCV, 2020, p. 153–168.

[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: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2019, pp. 5012–5021.

[47] A. Bera, Z. Wharton, Y. Liu, N. Bessis, A. Behera, SR-GNN: Spatial relation-aware graph neural network for fine-grained image categorization, IEEE Trans. Image Process. (TIP) 31 (2022) 6017–6031.

[48] Q. Xu, J. Wang, B. Jiang, B. Luo, Fine-grained visual classification via internal ensemble learning transformer, IEEE Trans. Multimedia (TMM) 25 (2023) 9015–9028.

[49] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 2921–2929.

[50] Y. Liu, Y.-H. Wu, P. Wen, Y. Shi, Y. Qiu, M.-M. Cheng, Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 44 (3) (2020) 1415–1428.

[51] J. C. Caicedo, S. Lazebnik, Active object localization with deep reinforcement learning, in: Int. Conf. Comput. Vis. (ICCV), 2015, pp. 2488–2496.

[52] S. Mathe, A. Pirinen, C. Sminchisescu, Reinforcement learning for visual object detection, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 2894–2902.

[53] X. Liu, J. Wang, S. Wen, E. Ding, Y. Lin, Localizing by describing: Attribute-guided attention localization for fine-grained recognition, in: AAAI Conf. Artif. Intell. (AAAI), 2017, pp. 4190–4196.

[54] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, L. Wang, Learning to navigate for fine-grained classification, in: ECCV, 2018, pp. 438–454.

[55] F. Fang, Q. Xu, Y. Cheng, Y. Sun, J.-H. Lim, Image understanding with reinforcement learning: Auto-tuning image attributes and model parameters for object detection and segmentation, IEEE Trans. Circ. Syst. Video Technol. (TCSVT) 32 (10) (2022) 6671–6685.

[56] H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, E. Hachem, Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows, Physical Review Fluids 6 (5) (2021) 053902.

[57] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Int. Conf. Comput. Vis. (ICCV), 2017, pp. 618–626.

[58] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. on Systems, Man, and Cybernetics 9 (1) (1979) 62–66.

[59] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, in: Int. Conf. Mach. Learn. (ICML), 2016, pp. 1995–2003.

[60] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J. T. Zhou, G. Zheng, Z. Zeng, Nonlinear regression via deep negative correlation learning, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 43 (3) (2019) 982–998.

[61] Z. Allen-Zhu, Y. Li, Towards understanding ensemble, knowledge distillation and self-distillation in deep learning, in: Int. Conf. Learn. Represent. (ICLR), 2023, pp. 1–12.

[62] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: Annu. Conf. Neur. Inform. Process. Syst. Worksh. (NeurIPS Workshop), 2014, pp. 1–9.

[63] X. Lan, X. Zhu, S. Gong, et al., Knowledge distillation by on-the-fly native ensemble, in: Annu. Conf. Neur. Inform. Process. Syst. (NeurIPS), 2018, pp. 7528–7538.

[64] X. Cheng, Z. Rao, Y. Chen, Q. Zhang, Explaining knowledge distillation by quantifying the knowledge, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2020, pp. 12925–12935.

[65] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2016, pp. 770–778.

[66] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Indian Conf. on Comp. Vision, Graphics and Image Processing, 2008, pp. 722–729.

[67] J. Krause, J. Deng, M. Stark, L. Fei-Fei, Collecting a large-scale dataset of fine-grained cars, in: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. (CVPRW), 2013, pp. 1–2.

[68] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–Mining discriminative components with random forests, in: Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 446–461.

[69] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2012, pp. 3498–3505.

[70] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, S. Belongie, Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2015, pp. 595–604.

[71] W. Ge, Y. Yu, Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2017, pp. 1086–1095.

[72] X. Yang, Y. Wang, K. Chen, Y. Xu, Y. Tian, Fine-grained object classification via self-supervised pose alignment, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2022, pp. 7399–7408.

[73] D. Demidov., M. Sharif., A. Abdurahimov., H. Cholakkal., F. Khan., Salient mask-guided vision transformer for fine-grained classification,

in: Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, INSTICC, SciTePress, 2023, pp. 27–38.

[74] A. Behera, Z. Wharton, P. R. Hewage, A. Bera, Context-aware attentional pooling (CAP) for fine-grained visual classification, in: AAAI Conf. Artif. Intell. (AAAI), 2021, pp. 929–937.

[75] W. Ge, X. Lin, Y. Yu, Weakly supervised complementary parts models for fine-grained image classification from the bottom up, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2019, pp. 3034–3043.

[76] D. Chang, Y. Ding, J. Xie, A. K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, Y.-Z. Song, The devil is in the channels: Mutual-channel loss for fine-grained image classification, IEEE Trans. Image Process. (TIP) 29 (2020) 4683–4695.

[77] Q. Wang, Y.-D. Ding, A novel fine-grained method for vehicle type recognition based on the locally enhanced PCANet neural network, Journal of Computer Science and Technology 33 (2018) 335–350.

[78] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, et al., GPipe: Efficient training of giant neural networks using pipeline parallelism, in: Annu. Conf. Neur. Inform. Process. Syst. (NeurIPS), 2019, pp. 103–112.

[79] Y. Cui, Y. Song, C. Sun, A. Howard, S. Belongie, Large scale fine-grained categorization and domain-specific transfer learning, in: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2018, pp. 4109–4118.

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, in: Annu. Conf. Neur. Inform. Process. Syst. (NeurIPS), 2019, pp. 8026–8037.

[81] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1–15.

[82] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, in: Int. Conf. Learn. Represent. (ICLR), 2016, pp. 1–21.

[83] R. Agarwal, The epsilon greedy algorithm - a performance review, International Journal of New Technology and Research 6 (9) (2020) 1–3.

[84] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (2010) 1–39.

[85] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137 (1-2) (2002) 239–263.