An Erudite Fine-Grained Visual Classification Model

Dongliang Chang[1], Yujun Tong[1], Ruoyi Du[1], Timothy Hospedales[2], Yi-Zhe Song[3], and Zhanyu Ma[1]*
[1] Beijing University of Posts and Telecommunications, China
[2] University of Edinburgh, UK [3]SketchX, CVSSP, University of Surrey, UK
[1]{changdongliang,tongyujun,duruoyi,mazhanyu}@bupt.edu.cn
[2]{t.hospedales}@ed.ac.uk [3]{y.song}@surrey.ac.uk

Abstract

Current fine-grained visual classification (FGVC) models are isolated. In practice, we first need to identify the coarse-grained label of an object, then select the corresponding FGVC model for recognition. This hinders the application of FGVC algorithms in real-life scenarios. In this paper, we propose an erudite FGVC model jointly trained by several different datasets[1]*, which can efficiently and accurately predict an object's fine-grained label across the combined label space. We found through a pilot study that positive and negative transfers co-occur when different datasets are mixed for training, i.e., the knowledge from other datasets is not always useful. Therefore, we first propose a feature disentanglement module and a feature re-fusion module to reduce negative transfer and boost positive transfer between different datasets. In detail, we reduce negative transfer by decoupling the deep features through many dataset-specific feature extractors. Subsequently, these are channel-wise re-fused to facilitate positive transfer. Finally, we propose a meta-learning based dataset-agnostic spatial attention layer to take full advantage of the multi-dataset training data, given that localisation is dataset-agnostic between different datasets. Experimental results across 11 different mixed-datasets built on four different FGVC datasets demonstrate the effectiveness of the proposed method. Furthermore, the proposed method can be easily combined with existing FGVC methods to obtain state-of-the-art results. Our code is available at* https://github.com/PRIS-CV/An-Erudite-FGVC-Model.

1. Introduction

In daily life, most people can quickly identify the coarse-grained label of an object (*e.g.,* car, bird, or aircraft). Then if we want to go further and identify its fine-grained labels (*e.g.,* "Ferrari FF Coupe" [20], "Sayornis" [36], "Boeing 727-200" [25]), we must learn and master the relevant knowledge [7]. However, it is impossible to master the knowledge and the classification topology of all objects in the world. A critical way to address this problem is to develop FGVC algorithms which can assist humans to recognise the fine-grained labels of different objects. Moreover, with the rapid development of deep learning, current FGVC algorithms have already abandoned the reliance on additional information [2, 5] (*e.g.,* attributes, bounding boxes) and have achieved recognition performance of over 90% on a wide range of fine-grained datasets [38], with the ability to be applied in practice.

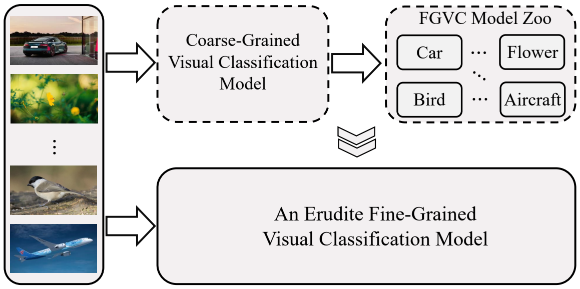


Figure 1. How to identify the fine-grained labels of an object? Current paradigms require two stages: coarse-grained visual classification and fine-grained visual classification. This paper transforms the two stages of recognition into an erudite fine-grained visual classification model, which can directly recognise the fine-grained labels of objects across different coarse-grained label spaces.

However, current FGVC algorithms are all based on a single source of training data, *e.g.,* a model trained on the CUB-200-2011 [36] dataset can only be used to recognise the species of a bird. If we want to identify a model for a car, we have to use another FGVC model. Specifically, as shown in Figure 1, if we want to recognise the fine-grained label of an object, we first need to know its coarse-grained label (*e.g.,* birds *vs.* cars) through a coarse-grained visual classification model, then select its corresponding fine-grained model from the FGVC model zoo and recognise

[*] indicates the corresponding author.

[1] In this paper, different datasets mean different fine-grained visual classification datasets.

its fine-grained label. This two-stage approach faces four challenges: Firstly, the inference time becomes longer (first coarse-grained image recognition, then fine-grained image recognition); Secondly, more storage space is required (different FGVC datasets require different fine-grained models to be stored); Thirdly, the accumulation of errors occurs (the accuracy of coarse-grained image recognition directly affects the accuracy of fine-grained recognition); Fourthly, the positive and negative transfers between different datasets is ignored. The above challenges greatly hinder the application of FGVC algorithms in practice.

A key solution to solve the challenges is to jointly train an *erudite* FGVC model with all training data from different datasets, as shown in Figure 1. However, our pilot study found that a vanilla erudite model fails to make accurate predictions because both positive and negative transfer occurs between different datasets. Specifically, after joint training, although each dataset's overall distribution of features almost always becomes better (*i.e.,* larger inter-class variance and intra-class similarity) than training alone, it becomes clear that only some categories get a better feature representation, and others get a worse feature representation. At the same time, the boundaries between different datasets sometimes become more blurred, confounding the model's predictions between them. Unfortunately, negative transfer dominates in practice, resulting in a significant drop in the test accuracy of the model on each dataset compared to training alone.

To make the erudite model more accurate, in this paper, we propose a feature disentanglement module and a feature re-fusion module to balance the positive and negative transfer between different datasets. In detail, we decouple the deep features through many dataset-specific feature extractors to obtain dataset-specific features, thus reducing the negative transfer. However, after decoupling the features, we need to know which dataset-specific classifier to use at the inference stage (but we cannot access the coarse-grained label of an object), and lost the positive transfer between datasets. Therefore, inspired by the mixture of experts (MoE) [26], we propose a gating-based feature re-fusion module to channel-wise re-fuse the dataset-specific features to facilitate positive transfer between different datasets. Finally, we obtain features with higher inter-class variance and intra-class similarity while maintaining positive transfer and suppressing negative transfer between datasets.

Meanwhile, an advantage of joint training with many different datasets is that we have more training data. Although the feature representations should be dataset-specific, salient feature localisation should be dataset-agnostic. Therefore, we can take full advantage of the increased training data to train the model to locate many different discriminative regions. Naturally, we can use a traditional spatial attention layer to locate regions that are useful for FGVC. However, directly applying a traditional spatial attention layer fails to work well due to domain-shift when training on different datasets [21, 28, 48]. To address this issue, we propose a meta-learning based spatial attention layer that drives the model to acquire a dataset-agnostic spatial attention that enhances the models' localisation ability to further increase performance.

We demonstrate our resulting framework on 11 different mixed-datasets built on four different FGVC datasets, and show that it can easily be combined with existing FGVC methods to obtain state-of-the-art results.

2. Related Work

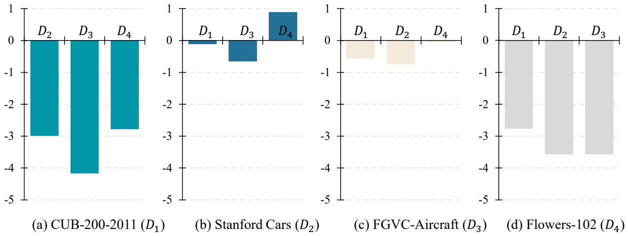
2.1. Fine-grained Visual Classification (FGVC)

Increasing inter-class variance and intra-class similarity is the central challenge of FGVC [42]. Current deep learning-based FGVC methods fall into two main categories. The first caterogy of approaches explicitly localise fine-grained visual regions and subsequently fuse the localised regions to perform fine-grained classification [42, 44, 50]. The second category is implicit approaches, which use higher-order feature representations [22, 47], end-to-end feature encoding [8, 13, 32, 33] or specially designed loss functions [6, 14, 23, 39] to drive the model to implicitly discover fine-grained visual regions that reflect subtle differences and explore the relationships between them. Recently, Chang *et al.* [7] extended the traditional FGVC to a multi-granularity visual classification task and achieved state-of-the-art results by feature disentanglement and reinforcement. Choudhury *et al.* [9] tried to loosen the need for annotation information further and explored using Wikipedia for fine-grained recognition without any annotations. Meanwhile, Yang *et al.* [41] incorporates geographical and temporal information into the deep model to improve performance.

Different from the above methods, this paper does not focus on improving performance on a particular FGVC dataset where the coarse grained category is always considered known. Instead we focus on how to train a model on a combined dataset of multiple datasets such that it can make predictions in the combined label-space, without assuming coarse-gradined category annotation during inference.

2.2. Joint Training of Multiple Datasets

Several studies have tried to combine several different datasets for joint training [16, 29, 37, 46, 49]. While this potentially benefits from more data for representation learning, the challenge is eliminating the potential conflicts and negative between datasets which can outweigh the benefits of joint training in practice. Kim *et al.* [19] attempted to train several object segmentation datasets jointly, proposed a gradient conflict loss for locating conflicting label



(a) CUB-200-2011 (D_1) (b) Stanford Cars (D_2) (c) FGVC-Aircraft (D_3) (d) Flowers-102 (D_4)

Figure 2. Performance differences (Δ (%)) between multi-dataset training *vs.* training each dataset alone. Subplots indicate target datasets and bars correspond to extra data used for training.

spaces, and mitigated this problem using class-independent binary cross-entropy loss. Zhao *et al.* [46] trying to train a single object detector predicting over the union of all the label spaces, and using the pseudo labelling approach to integrate the data from different datasets. Almazán *et al.* [1] further combined joint training of multiple datasets with self-supervised learning. A strong pre-training model was used to generate pseudo-labels for unlabelled images in each dataset, leading to improved zero-shot retrieval.

Following these works, we extend the vision of joint training of multiple datasets to the field of FGVC. In contrast to these methods, as different FGVC datasets belong to different coarse-grain labels with disjoint fine-grained labels, we cannot simply achieve positive transfer between datasets in the label space as in the previous approach. The key challenge of the joint training of multiple FGVC datasets is managing positive and negative transfer between dataset-specific features.

3. Pilot Study

In this section, we first define and analyse a baseline method (a vanilla erudite model) for joint training of multiple datasets. The results show that positive and negative transfers co-occur when different datasets are mixed for training.

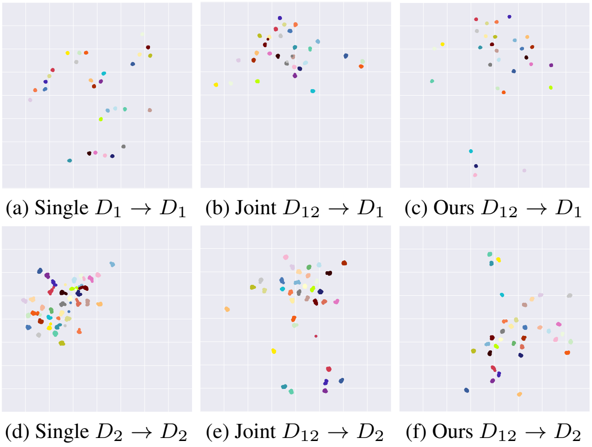
3.1. Notations and definitions

Suppose we have N different FGVC datasets: $\{D_n\}_{n=1}^{N}$, where $D_n = \{x^n, y^n\}$, x^n are the samples of the n^{th} dataset, and y^n are their corresponding ground truth. And $D = \{x, y\}$, where x is the union of all datasets samples ($x \in \{x^1 \cup \ldots \cup x^N\}$), and y is the joint label-space of all datasets ($y \in \{y^1 \cup \ldots \cup y^N\}$).

3.2. Baseline method

To enable the model can make predictions across different fine-grained label spaces, the most straightforward method is to train a feature extractor $\mathcal{F}(\cdot)$ and a classifier $\mathcal{G}(\cdot)$ with the mixed data from different datasets as

$$Loss(\cdot) = \mathcal{L}(\mathcal{G}(f), y), \quad f = \mathcal{F}(x), \tag{1}$$



(a) Single $D_1 \rightarrow D_1$ (b) Joint $D_{12} \rightarrow D_1$ (c) Ours $D_{12} \rightarrow D_1$

(d) Single $D_2 \rightarrow D_2$ (e) Joint $D_{12} \rightarrow D_2$ (f) Ours $D_{12} \rightarrow D_2$

Figure 3. Feature visualisation by t-SNE [34] after single dataset-learning, vanilla multi-dataset joint learning and our method. Subplots indicate training data and evaluation data for plotting: $D_1 \rightarrow D_1$ indicates single dataset training and evaluation on dataset D_1, while $D_{12} \rightarrow D_1$ indicates training on $D_1 + D_2$ and then evaluating t-SNE on D_1. In each sub-figure, colours indicate different classes.

where $\mathcal{L}(\cdot)$ is the cross entropy loss, $x \in D$, and the goal is to make predictions $\mathcal{G}(\mathcal{F}(\cdot))$ in the union label-space.

3.2.1 The baseline method is all we need?

Datasets In this section, we select all pairs the four datasets (CUB-200-2011 (D_1) [36], Stanford Cars (D_2) [20], FGVC-Aircraft (D_3) [25], and Flowers-102 (D_4) [27]) to form six different datasets (*e.g.*, $D_{12} = \{D_1, D_2\}$, please see Section 5.1 for details) to evaluate the performance of the baseline. Please refer to Section 5.2 for training details.

Evaluation The performance of the baseline method is evaluated by the difference in prediction accuracy between multiple datasets trained together and each dataset trained alone as

$$\Delta = acc(D_n) - \tilde{acc}(D_n), \tag{2}$$

where $acc(D_n)$ indicates the model's performance of D_n after D_n joint training with other datasets, and $\tilde{acc}(D_n)$ indicates the model's performance when trained alone.

Results From the Figure 2, we can see that: (i) a significant drop in the model's prediction accuracy regardless of which dataset the D_1 was trained jointly with, *e.g.,* a maximum drop of 4.17% (D_{13}), as shown in Figure 2a. A similar phenomenon can be observed in Figure 2d. (ii) the performance degradation is less when the D_2 are trained jointly with other datasets and is improved in the joint training with D_4, as shown in Figure 2b. A similar phenomenon can be observed in Figure 2c. Considering (i) and (ii) together, we see that both negative and positive transfer can occur when jointly training multiple datasets. However, negative transfer is usually more significant than the positive transfer.

CUB-200-2011 (D_1)						Stanford Cars (D_2)					
Metric	Single$^\flat$	Baseline$^\flat$	Ours$^\flat$	Baseline†	Ours†	**Metric**	Single$^\flat$	Baseline$^\flat$	Ours$^\flat$	Baseline†	Ours†
SC (↑)	0.127	0.150 ± 0.120	$\underline{0.163\pm0.021}$	–	–	**SC** (↑)	0.199	0.281 ± 0.064	$\underline{0.350\pm0.026}$	–	–
CHI (↑)	419.7	420.8 ± 42.02	$\underline{495.6\pm36.61}$	–	–	**CHI** (↑)	753.6	994.4 ± 121.9	$\underline{1409\pm47.12}$	–	–
DBI (↓)	3.909	3.757 ± 2.236	$\underline{3.642\pm0.236}$	–	–	**DBI** (↓)	4.175	2.548 ± 0.369	$\underline{2.243\pm0.215}$	–	–
Positive (↑)	–	86.33 ± 4.041	$\underline{120.0\pm2.0}$	–	–	**Positive** (↑)	–	111.67 ± 4.059	$\underline{123.7\pm15.04}$	–	–
Negative (↓)	–	113.7 ± 4.041	$\underline{80.0\pm2.0}$	–	–	**Negative** (↓)	–	84.33 ± 4.509	$\underline{72.33\pm15.04}$	–	–
MMD (↑)	–	–	–	5095 ± 1101	$\underline{5572\pm312.5}$	**MMD** (↑)	–	–	–	4781 ± 1105	$\underline{5406\pm953.4}$
FGVC Aircraft (D_3)						Flowers-102 (D_4)					
Metric	Single$^\flat$	Baseline$^\flat$	Ours$^\flat$	Baseline†	Ours†	**Metric**	Single$^\flat$	Baseline$^\flat$	Ours$^\flat$	Baseline†	Ours†
SC (↑)	0.271	0.299 ± 0.042	$\underline{0.359\pm0.016}$	–	–	**SC** (↑)	$\underline{0.642}$	0.533 ± 0.004	0.590 ± 0.005	–	–
CHI (↑)	515.0	565.9 ± 78.25	$\underline{651.3\pm52.44}$	–	–	**CHI** (↑)	2288	2033.6 ± 93.2	$\underline{2354\pm183.0}$	–	–
DBI (↓)	3.246	2.427 ± 0.357	$\underline{2.170\pm0.289}$	–	–	**DBI** (↓)	$\underline{0.837}$	1.160 ± 0.083	0.955 ± 0.068	–	–
Positive (↑)	–	67.33 ± 7.506	$\underline{70.33\pm1.528}$	–	–	**Positive** (↑)	–	54.0 ± 4.582	$\underline{63.0\pm1.732}$	–	–
Negative (↓)	–	32.67 ± 7.506	$\underline{29.67\pm1.528}$	–	–	**Negative** (↓)	–	48.0 ± 4.583	$\underline{39.0\pm1.732}$	–	–
MMD (↑)	–	–	–	5097 ± 478.5	$\underline{5692\pm588.0}$	**MMD** (↑)	–	–	–	4061 ± 597.4	$\underline{4874\pm462.4}$

Table 1. Evaluation of the feature distribution of test samples after joint training. ♭: denotes the distribution of samples within each individual dataset, and †: represents the distribution of samples between any two datasets. Underline indicates the best results.

In addition, we can see that: (i) the positive and negative transfer between any two datasets is asymmetric: take the D_1 as an example: when mixed with the D_3, its performance dropped by 4.17%, but the performance of the D_3 dropped by only 0.57%. (ii) Datasets of non-rigid objects (D_1 and D_4) are more vulnerable to negative transfer than rigid object datasets (D_2 and D_3).

3.3. Positive and negative transfer

In this section, we take D_{12} as an example to analyse why a positive or negative transfer occurred.

The central challenge of FGVC is obtaining a feature representation with large inter-class variance and intra-class similarity. Therefore, we first visualised the trained features through t-SNE [34] to analyse the intra-class and inter-class variance. Since showing all $\sim$ 200 classes simultaneously is hard, we use the hierarchical label structure of FGVC datasets in [7] to visualise 30 classes for D_1 and 35 classes for D_2. The selected classes all belong to the same parent, and are thus particularly difficult to distinguish.

From the Figure 3, we can observe: **Positive transfer:** (i) After joint training, the inter-class variance of D_2 is increased significantly (compare Figure 3d and Figure 3e). **Negative transfer:** after joint training, the inter-class similarity of the D_1 is increased significantly (compare Figure 3a and Figure 3b) and led to worse generalisation performance (78.31% (Single $D_1 \rightarrow D_1$) *vs.* 75.32% (Ours $D_{12} \rightarrow D_1$)). Please refer to Section 5.3 for a detailed analysis about **Ours**.

In addition, we have calculated intra-class similarity and inter-class variance quantitatively, as shown in Table 1. Specifically, we use three metrics: **SC**: Silhouette Coefficient [30], **CHI**: Calinski-Harabasz Index [4], and **DBI**: Davies-Bouldin Index [10], where the larger SC and CHI, and the smaller DBI indicate larger intra-class similarity and inter-class variance. By comparing **Single** (train each dataset alone) and **Baseline** (joint training), we can see that for most datasets, regardless of which dataset is mixed together results in a better feature representation. The positive transfer between datasets is fully validated. We have also analysed the positive and negative transfers at the category level through per-class classification accuracy improvement relative to the **Single**. Specifically, we use two metrics: **Positive** and **Negative**. **Positive** denotes the number of classes that increases (and unchanged) accuracy after joint training. **Negative** denotes the number of classes that reduces accuracy after joint training. As we can see from the Table 1, when training the dataset jointly, both positive and negative transfers occur, and the relative proportions of positive and negative transfers are different in different datasets. Overall, the negative transfer is more severe. This phenomenon is consistent with Figure 2 and Figure 3. Furthermore, we have measured the distance between any two datasets by **MMD**: Maximum mean discrepancy [3], please see Section 5.3 for a detailed analysis.

4. Methodology

Eliminating the negative transfer and boosting the positive transfer between different datasets is the key to obtain a more accurate and erudite FGVC model. In Section 4.1, we first discuss our proposed feature disentanglement module which decouple the features through dataset-specific feature extractors to alleviate negative transfer. In Section 4.2, we then discuss our proposed refusion module which re-fuse the dataset-specific features through a gating-based feature re-fusion module to enhance the positive transfer between datasets. Finally, we discuss our proposed meta-learning based dataset-agnostic spatial attention layer in Section 4.3. Figure 4 depicts our framework with proposed modules.

4.1. Feature Disentanglement Module

As analysed in Section 3, learning a single dataset-agnostic feature introduce a severe negative transfer problem. Therefore, we first decouple the mixed feature embeddings into different dataset-specific features, to eliminate the negative transfer between different datasets.

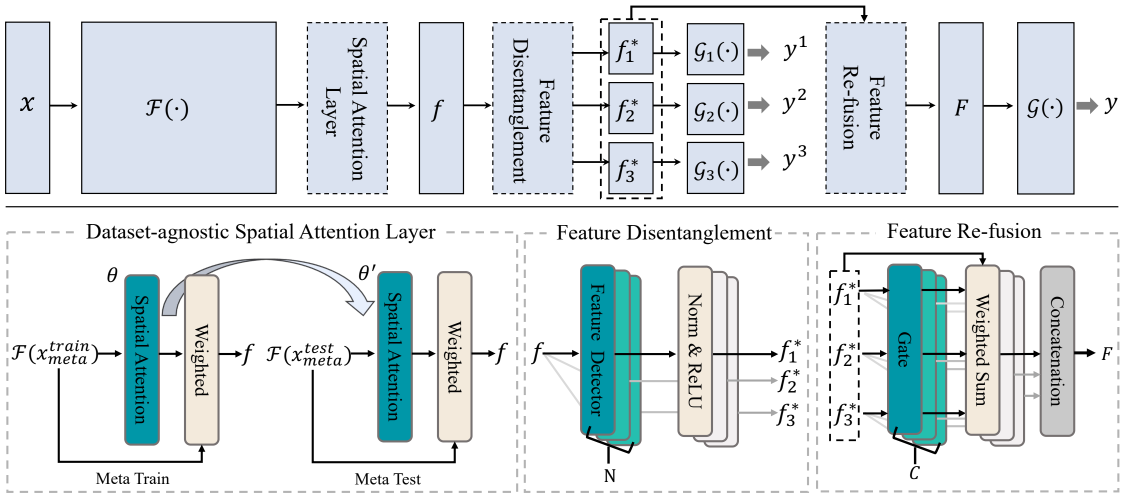


Figure 4. A schematic illustration of the proposed methods. The input x contains data belonging to multiple datasets. Here is a mix of 3 datasets as an example (*i.e.,* $N = 3$). The dataset-specific classifiers ($\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$) are only used in the training stage.

We first obtain dataset-specific features $\{f_n^*\}_{n=1}^N$ using multiple dataset-specific projection heads $\{m_n(\cdot)\}_{n=1}^N$ and constrain the task-specific features to be discriminative by using dataset-specific classifiers $\{\mathcal{G}_n(\cdot)\}_{n=1}^N$:

$$\begin{aligned} Loss_{aux}(\cdot) &= \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\mathcal{G}_n(f_n^*), y^n), \\ f_n^* = m_n(f_n),\ \ m_n(\cdot) &= \mathrm{ReLU}(\mathrm{BN}_n(\mathrm{Conv}_n(\cdot))), \end{aligned} \tag{3}$$

where f_n denotes the features belong to the n^{th} dataset, after disentangling from their common encoding $f_n = \mathcal{F}(x^n)$, x^n are the samples of the n^{th} dataset. $\mathrm{Conv}_n(\cdot)$ is used to extract the dataset-specific features of the n^{th} dataset and is a 1×1 convolution layer with both input and output channels the same as the channels of f_n, $\mathrm{BN}_n(\cdot)$ is the batch normalization layer [18], ReLU is the non-linear activation function, $\mathcal{L}(\cdot)$ is the cross entropy loss. $\mathcal{G}_n(\cdot)$ is used only during the training stage, and will be discarded in the inference stage.

4.2. Feature Re-fusion Module

The feature disentanglement alleviates the negative transfer effect due to the use of dataset-specific features, but we also lose the advantage of positive transfer at the same time. Also, we we cannot easily decide which dataset-specific feature during testing, as the coarse-grained label is unknown during the inference time.

To address the above issues, we propose a gating-based channel-wise feature re-fusion module that learns how to fuse the feature array for use with a common classifier. We define the re-fused features as $F = \{F_c\}_{c=1}^C$ defined by

$$F_c = \sum_{n=1}^{N}\mathrm{Softmax}(\mathrm{FC}(\mathrm{Cat}(\{f_{n,c}^*\}_{n=1}^N)) \cdot (\{f_{n,c}^*\}_{n=1}^N)), \tag{4}$$

where $f_{n,c}^*$ denotes the c^{th} channel of f_n^*, $c \in [1, C]$, C denotes the number of channels, N denotes the number of datasets, $\mathrm{Cat}(\cdot)$ denotes the concatenation of multiple features at channel level, $\mathrm{FC}(\cdot)$ denotes the fully connected layer with output dimension equals to N.

Finally, the fused features F are passed into the joint classifier $\mathcal{G}(\cdot)$ to predict a joint classification result. The loss function is formulated as

$$Loss_{main}(\cdot) = \mathcal{L}(\mathcal{G}(F), y), \tag{5}$$

where $\mathcal{L}(\cdot)$ is the cross entropy loss.

Therefore, the overall optimisation objective is

$$Loss_{all}(\cdot) = Loss_{main}(\cdot) + Loss_{aux}(\cdot). \tag{6}$$

In summary, the feature re-fusion module learns to predict the set of fusion weights with which to fuse the feature ensemble $\{f_n^*\}$. These fusion weights are trained so as to optimise performance of the joint label-space classifier using the fused features (Eq 5).

4.3. Dataset-agnostic Spatial Attention Layer

In this section, we aim to take full advantage of multi-dataset training data, given that localisation is dataset agnostic. We propose a meta-learning based dataset-agnostic spatial attention layer that can be used across different datasets for localisation.

Vanilla Spatial Attention Layer A key to the FGVC task is to localise different fine-grained regions for recognition [38]. Therefore, similar to self-attention, we first propose a multi-head spatial attention layer for the model to focus on different visual regions.

The channels in the feature represent different patterns [31]. To obtain different feature regions, referring to the set-up in the previous work [35], we first divide the deep features $\mathcal{F}(x)$ into 8 groups at the channel wise: $\mathcal{F}(x) = \{\text{group}_k\}_{k=1}^{8}$, where $\text{group}_k \in F^{\frac{C}{8}\times H\times W}$, C is the channel numbers of the deep features $\mathcal{F}(x)$, H and W are the height and width for each channel. Subsequently, the grouped features are fed into their corresponding spatial attention layers to obtain spatial attention respectively as

$$f = \text{Cat}(\{\text{Sigmoid}(\text{Conv}_k(\text{group}_k)) \cdot \text{group}_k\}_{k=1}^{8}), \tag{7}$$

where each $\text{Conv}_k(\cdot)$ is 1×1 convolution layer (the input channel is $\frac{C}{8}$ and the output channel is 1). $\text{Cat}(\cdot)$ denotes the concatenation of multiple features at channel level.

As mentioned earlier, both the training and test data contain data from multiple datasets. The vanilla spatial attention layer cannot work well due to potential conflicts between different datasets. Therefore, similar to [21], we use the model-agnostic meta-learning [15] to optimise the parameters of the spatial attention layer. Finally, we obtained a dataset-agnostic spatial attention layer so that the potential conflicts can be better handled.

Meta train In the training process, the training data for each mini-batch is a mixture of data from different datasets. We randomly select the data from one dataset as the meta-test set ($D_{meta}^{test} = \{x_{meta}^{test}, y_{meta}^{test}\}$) and the rest of the data as the meta-train set ($D_{meta}^{train} = \{x_{meta}^{train}, y_{meta}^{train}\}$). In addition, we use meta-learning only to optimise the parameters of the spatial attention layer. Thus, the loss function of the meta-train is

$$Loss_{meta}^{train} = Loss_{all}(\theta; D_{meta}^{train}), \tag{8}$$

where θ means the parameters of the spatial attention layer. Therefore, the gradient of θ with respect to $Loss_{meta}^{train}(\cdot)$ is ∇_θ, and the parameters of the spatial attention layer will be updated after optimisation to $\theta' = \theta - \alpha\nabla_\theta$, where α is the learning rate.

Meta test In each mini-batch the model is also virtually evaluated on the meta test set from a different dataset. The loss for the adapted parameters on the meta-test data is

$$Loss_{meta}^{test}(\cdot) = Loss_{all}(\theta'; D_{meta}^{test}), \tag{9}$$

where θ' means the updated parameters from meta-train. This means that for optimisation with $Loss_{meta}^{test}(\cdot)$, we will need the second derivative with respect to the θ.

Summary The meta-train and the meta-test are optimised simultaneously, so the final objective is

$$L_{meta}(\cdot) = Loss_{meta}^{train}(\cdot) + Loss_{meta}^{test}(\cdot). \tag{10}$$

4.4. Summary

Since equation 10 requires a second-order derivative and is optimised with different data and parameters with equation 6, for each mini-batch, we adopt a two-stage approach for the optimisation of the parameters of the model.

Stage I: we train the feature extractor, the feature disentanglement and re-fusion modules, and the classifier with

$$Loss(\cdot) = Loss_{all}(\Theta), \tag{11}$$

where Θ denotes the parameters of the whole model, excluding the parameters of the spatial attention layer.

Stage II: we train the dataset-agnostic spatial attention layer with

$$Loss(\cdot) = L_{meta}(\theta), \tag{12}$$

where θ means the parameters of the spatial attention layer.

5. Experiments Setting

5.1. Datasets

We consider four widely used FGVC datasets to evaluate the performance of the proposed method in a scenario where multiple datasets are trained jointly: D_1: **CUB-**200**-**2011 [36] (contains 200 classes, 5994 for training and 5794 for test), D_2: **Stanford Cars** [20] (contains 196 classes, 8144 for training and 8041 for test), D_3: **FGVC-Aircraft** [25] (contains 100 classes, 6667 for training and 3333 for test), and D_4: **Flowers-**102 [27] (contains 102 classes, 2040 for training and 6149 for test).

In these four datasets, D_2 and D_3 are relatively similar in terms of dataset gap (both datasets are vehicles). While D_1 and D_3 are similar in terms of object shape (e.g., birds and airplanes have heads and wings), thus D_1 and D_3 can be used to evaluate the model performance under mixed training of two datasets with similar object shapes. In particular, D_{12} denotes a mixture of D_1 and D_2, D_{123} denotes a mixture of D_1, D_2, and D_3, and D_{1234} denotes a mixture of D_1, D_2, D_3, and D_4. Therefore, we totally have 11 different datasets ($nCr(4,2) + nCr(4,3) + nCr(4,4)$, where nCr calculates the number of unique ways to select r from n). In summary, we hope that the different combinations of the four datasets described above will adequately simulate the situation in a real-life scenario.

5.2. Training Details

The number of the training samples varies significantly between different datasets (from 2040 to 8144). The mixing of different datasets for training will cause a long-tail distribution problem, which could affect the performance evaluation. Therefore, we use a down-sampling approach [40] to alleviate the long-tail distribution problem in the training process. Specifically, we sample the same amount of data from different datasets, and mix them as training data per mini-batch (*e.g.,* 16 images per dataset in a mini-batch). We adopt ResNet-50 [17] as our backbone feature extractor and initialise it with ImageNet [11] pre-trained weights. All our proposed modules are initialised randomly. The image input is resized to 224 × 224. We adopt Momentum SGD

Dataset	Single	Baseline	Baseline+	Baseline++	Ours	Ours$_{PCGrad}$	Ours$_{PMG}$
D_{12}	83.63	82.17	83.31	82.89	83.94	83.38	84.93
D_{13}	83.15	80.84	82.06	82.22	83.35	82.55	84.18
D_{14}	86.43	83.76	85.81	85.97	86.76	86.00	88.13
D_{23}	88.55	87.92	87.82	87.23	88.74	88.14	89.23
D_{24}	91.89	90.61	91.14	91.31	91.66	91.30	92.43
D_{34}	91.28	89.62	90.73	90.55	91.52	90.76	91.60
Avg.	87.50	85.82	86.81	86.69	87.66	87.02	88.41
D_{123}	85.10	84.13	84.65	84.04	84.83	84.58	85.61
D_{124}	87.32	84.98	86.86	86.43	87.48	87.07	87.73
D_{134}	86.99	84.80	86.00	85.52	86.88	86.45	87.70
D_{234}	90.61	89.56	89.65	89.58	90.46	89.93	90.85
Avg.	87.50	85.87	86.79	86.39	87.41	86.99	87.98
D_{1234}	87.49	85.54	86.54	86.06	87.45	86.86	88.08

Table 2. Comparisons with different baselines ($a\bar{c}c(\%)$). Underlining indicates the best results.

as our optimiser and cosine annealing [24] as our learning rate scheduler. The initial learning rate is 0.1 and gradually decay to 0 over 100 epochs. The backbone feature extractor has a $10\times$ smaller initial learning rate (*i.e.,* 0.01). We adopt common data augmentation techniques such as random horizontal flips and random crops. Note that we only have access to the coarse-grained labels (*e.g.,* car, bird) of the sample during the training phase. Unless mentioned otherwise, we do not have the coarse-grained labels of the test samples during the inference phase.

5.2.1 Evaluation Protocol

In this paper, we use Arithmetic Mean ($a\bar{c}c = \frac{1}{N}\sum_{n=1}^{N} acc_n$) to evaluate the model's performance, where acc_n is the test accuracy of the n^{th} test dataset, $n \in [1, N]$, and N is the total number of the test datasets.

5.2.2 Comparison methods

Single denotes that each dataset is trained separately. In addition, we trained a coarse-grained classification model to select the corresponding fine-grained model from the FGVC model zoo. **Baseline** denotes directly mixing data from different datasets to train the model (see Section 3.2 for details). **Baseline+** denotes our proposed method without the dataset-agnostic spatial attention layer. **Baseline++** denotes our proposed method without meta-learning. **Ours**: denotes our proposed method. To evaluate the effectiveness of our meta-learning strategy we also adopt another SOTA method, PCGrad [43], to optimize the spatial attention layer. This is denoted **Ours**$_{PCGrad}$. PCGrad handles the gradient conflicts between datasets by projecting a task's gradient onto the normal plane of the gradient of any other task that has a conflicting gradient. Finally, **Ours**$_{PMG}$ represents our proposed method on top of the state-of-the-art FGVC method PMG [12].

5.3. Results and Analysis

From Table 2, we can see that: (i) With our proposed feature disentanglement and re-fusion modules (**Baseline+**), the model performance is significantly improved over **Baseline**. This validates the effectiveness of our proposed modules. (ii) When mixing two datasets for training, **Baseline++** achieves comparable results to **Baseline+**. However, when the number of mixed datasets is more than three, the performance of **Baseline++** decreased significantly (*e.g.,* 86.0% to 85.52% on D_{134}). This suggests that the vanilla spatial attention layer cannot work well under the joint training of multiple datasets. (iii) With meta-learning strategy on the spatial attention layer, **Ours** obtain performance gains in all datasets as compared to **Baseline+** (*e.g.,* 86.0% to 86.88% on D_{134}). This validates the positive role of meta-learning to train a dataset-agnostic localisation module. (iv) Although performance gains were obtained after training the spatial attention layer using PCGrad (**Ours**$_{PCGrad}$), the gains is smaller than that of **Ours**. This phenomenon suggests that our meta-learning strategy handles the gradient conflicts problem caused by multiple data distributions better than PCGrad. (v) Even against the **Single** with more than $3\times$ the computational complexity of **Ours**, **Ours** still comparable with it, which validates the effectiveness of our proposed method. (vii) When the proposed method is combined with traditional FGVC algorithm (*i.e.,* PMG [12]), the model's performance is further improved, this validates the flexibility of the proposed method as a plugin.

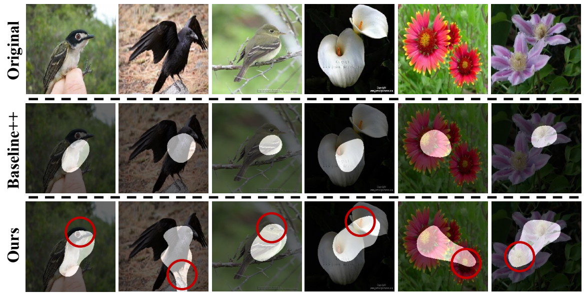


Figure 5. We highlight the supporting visual regions for attention layers of two compared models. The red circles denote the exclusive visual regions that **Ours** focus on.

Furthermore, from the Table 1 and Figure 3, we can see that: (i) Our proposed method has achieved the best results on the metric **SC**, **CHI**, and **DBI**, which indicate that the proposed method obtained better feature representations (*i.e.,* large inter-class variance and intra-class similarity) in most datasets. (ii) The proposed method can effectively enhance the positive transfer (compare Figure 3e and Figure 3f) and eliminate the negative transfer (compare Figure 3b and Figure 3c), which led to better generalisation performance. (iii) The proposed method has achieved the best results on the metric **Positive** and **Negative**, which provides ample evidence for the conclusion of (ii). (iv) The proposed method obtains the best results on the metric **MMD**, which indicates lower inter-dataset confusion.

	Two Datasets	Three Datasets	Four Datasets
Single	8.24/73.88	8.24/98.54	8.24/123.2
Baseline	4.12/24.77	4.12/24.82	4.12/24.87
Ours$_{train}$	4.33/30.54	4.38/31.96	4.42/33.84
Ours$_{test}$	4.22/29.23	4.28/30.59	4.33/32.48

Table 3. Computational complexity and number of parameters (**MACs** (G)/ **Params** (M)).

6. Future Analysis

6.1. Feature Visualisation

We further carry out the model visualisation with D_{12} to show that the spatial attention layer under **Baseline++** and **Ours** captures different regions that are useful for FGVC. From Figure 5, we can see that the visual regions **Ours** finds are more discriminative than the **Baseline++**, which demonstrates the meta-learning's positive role in learning a dataset-agnostic spatial attention layer.

6.2. Feature Re-fusion Module

We further analyse how the feature re-fusion module works (take D_{24} as an example). When the input data is D_2, the feature re-fusion module tends to select the channels that belong to D_4-specific features ($54.4_{\pm 21}\%$). When the input data is D_4, the feature re-fusion module tends to select features that belong to itself ($54.4_{\pm 21}\%$), rather than D_2-specific features. The above phenomenon is consistent with the findings in Section 3.2.1: the positive and negative transfer between any two datasets is asymmetric (*e.g.,* D_4 has a significant positive transfer to D_2, and D_2 a has a significant negative transfer to D_4).

6.3. Computational Complexity

In this section, we use the number of Multiply Accumulate Operations (**MACs**) to measure models' computational complexity and the number of parameters (**Params**) to measure the models' size. With the same accuracy, models with low **MACs** and **Params** are more efficient. From Table 2 and Table 3, we can see that: (i) Although the **MACs** and **Params** of our proposed method are much lighter than that of the **Single**, **Ours** still obtained comparable results to **Single**. (ii) Compared to **Baseline**, **Ours** gains over 2% performance on almost every dataset with a slight increase in **MACs** and **Params**. (i) and (ii) verify that the proposed method is efficient and accurate.

6.4. Why not use multi-task learning?

Multi-task learning (MTL) [45] aims to leverage commonality across several tasks to improve the performance on all tasks. Conventional multi-task learning shares an encoder and learns task-specific classifier/decoder heads. Without the knowledge of the coarse-grained class during inference, one cannot select task-specific heads. Thus our problem setting requires classification into the joint-label space of all tasks.

Hierarchical multi-task learning An alternative solution is to construct a hierarchical multi-task learning, where we first determine the coarse-grained category of the sample belongs and subsequently select its corresponding fine-grained classifier to recognise it. In other words, we have one feature extractor, one coarse-grained classifier, and N task-specific classifiers.

Results We evaluated the hierarchical multi-task learning with D_{12}. The performance of the coarse-grained classifier is 99.30%. However, the performance of the two fine-grained classifiers was very poor (*i.e.,* 26.5% on D_1, 28.3% on D_2). We attribute this to the coarse-grained classifier reducing the inter-class distance with each task [7]. For instance, both "Flamingo" and "Gray-backed albatross" belong to the category "Birds".

6.5. Limitations

This paper only considers the relationships between different FGVC datasets in the same domain (natural scenario). In cross-domain scenarios (*e.g.,* painting and sketching), it needs to be further explored whether positive and negative transfers still co-occur between datasets.

7. Conclusion

We introduce the problem of joint multi-dataset FGVC. This goes beyond traditional FGVC algorithms dedicated to solving particular fine-grained datasets, and requires more realistic inference in the joint label-space of all datasets, without assuming the coarse grained label is given during inference. To address this scenario, we train a single erudite FGVC model across multiple fine-grained recognition datasets. We analyse the challenges entailed in terms of negative transfer dominating over positive transfer across datasets, and propose a solution based on feature disentanglement and re-fusion modules to balance positive and negative transfer; as well as a dataset–agnostic spatial attention layer to enhance the generalisation of the model on localisation. The results across 11 different mix-datasets built on four different FGVC datasets verify the effectiveness of our method.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (NSFC) No. U19B2036, 62225601, in part by Beijing Natural Science Foundation Project No. Z200002, in part by scholarships from China Scholarship Council (CSC) under Grant CSC No. 202006470036, 202206470055, in part by BUPT Excellent Ph.D. Students Foundation No. CX2020105, CX2022152, in part by the Program for Youth Innovative Research Team of BUPT No. 2023QNTD02, and in part by the Supported by High-performance Computing Platform of BUPT.

References

[1] Jon Almazán, Byungsoo Ko, Geonmo Gu, Diane Larlus, and Yannis Kalantidis. Granularity-aware adaptation for image retrieval over multiple tasks. In *ECCV*, 2022. 3

[2] Thomas Berg and Peter Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013. 1

[3] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 2006. 4

[4] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 1974. 4

[5] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013. 1

[6] Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Transactions on Image Processing*, 2020. 2

[7] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "Flamingo" is my "Bird": Fine-grained, or not. In *CVPR*, 2021. 1, 2, 4, 8

[8] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, 2019. 2

[9] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. The curious layperson: Fine-grained image recognition without expert labels. In *BMVC*, 2021. 2

[10] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979. 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[12] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Yi-Zhe Song, Zhanyu Ma, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *ECCV*, 2020. 7

[13] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Pairwise confusion for fine-grained visual classification. In *ECCV*, 2018. 2

[14] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine grained classification. In *NeurIPS*, 2018. 2

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 6

[16] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Generalizable pedestrian detection: The elephant in the room. In *CVPR*, 2021. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[19] Dongwan Kim, Yi-Hsuan Tsai, Yumin Suh, Masoud Faraki, Sparsh Garg, Manmohan Chandraker, and Bohyung Han. Learning semantic segmentation from multiple datasets with label shifts. *arXiv preprint arXiv:2202.14030*, 2022. 2

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshop*, 2013. 1, 3, 6

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2, 6

[22] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015. 2

[23] Kangjun Liu, Ke Chen, and Kui Jia. Convolutional fine-grained classification with self-supervised target relation regularization. *IEEE Transactions on Image Processing*, 2022. 2

[24] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 7

[25] Subhransu Maji, Juho Kannala, Esa Rahtu, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 1, 3, 6

[26] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 2014. 2

[27] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 3, 6

[28] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2

[29] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[30] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987. 4

[31] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 5

[32] Guolei Sun, Hisham Cholakkal, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Fine-grained recognition: Accounting for subtle differences between similar classes. In *AAAI*, 2020. 2

[33] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In *ECCV*, 2018. 2

[34] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Kearning Research*, 2008. 3, 4

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 6

[36] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Caltech, Technical Report*, 2011. 1, 3, 6

[37] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, 2019. 2

[38] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 5

[39] Furong Xu, Meng Wang, Wei Zhang, Yuan Cheng, and Wei Chu. Discrimination-aware mechanism for fine-grained representation learning. In *CVPR*, 2021. 2

[40] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 2022. 6

[41] Lingfeng Yang, Xiang Li, Renjie Song, Borui Zhao, Juntian Tao, Shihao Zhou, Jiajun Liang, and Jian Yang. Dynamic mlp for fine-grained image classification by leveraging geographical and temporal information. In *CVPR*, 2022. 2

[42] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *ECCV*, 2018. 2

[43] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 7

[44] Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing*, 2016. 2

[45] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 8

[46] Xiangyun Zhao, Samuel Schulter, Gaurav Sharma, Yi-Hsuan Tsai, Manmohan Chandraker, and Ying Wu. Object detection with a unified label space from multiple datasets. In *ECCV*, 2020. 2, 3

[47] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Learning deep bilinear transformation for fine-grained image representation. In *NeurIPS*, 2019. 2

[48] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[49] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022. 2

[50] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *AAAI*, 2020. 2