Fine-Grained Few-Shot Classification with Part Matching

Samuel Black
Temple University
sam.black@temple.edu

Richard Souvenir
Temple University
souvenir@temple.edu

Abstract

In this paper, we describe a parts-based approach tailored for fine-grained, few-shot classification, particularly for scenes where the parts distribution is more significant than the broader visual characteristics. By focusing on part-level representations within scenes, our method provides robust classification with limited examples. Our approach, Simple Matching Parts Learner (SMPL), leverages off-the-shelf components in a straightforward manner to optimize few-shot classification using a meta-training phase. We demonstrate the performance of this approach on existing few-shot benchmarks. Additionally, we repurpose an existing fine-grained dataset with higher class diversity and variability than the standard benchmarks for the few-shot setting. SMPL not only achieves state-of-the-art few-shot classification performance, but at a much lower computational cost than compared to the other methods. Code at https://github.com/vidarlab/smpl-fsl.

1. Introduction

Parts-based analysis is well suited for fine-grained classification, where subtle distinctions between categories often hinge on small, discriminative parts rather than on broader scene-level features, which may vary widely even within the same category. Figure 1 presents four images from the same hotel. Although the images differ significantly in their overall appearance, they share common objects distributed across multiple images. The aggregate object similarity is a stronger cue than the scene similarity between images. This observation is particularly relevant in the fine-grained, few-shot setting, where large-scale visual differences between images of the same category, coupled with limited examples for training, make global scene-based features less reliable. Instead, parts-based features isolate components of the scene that remain consistent across varied contexts.

Part-based approaches have differed in the semantic level of the constituent features (i.e., patch, part, object) but generally enforce some constraint regarding the distribution of part features in each class. While some methods learn localized features without explicit part-level guidance (e.g., [44, 45]), most methods rely on part annotations for supervision (e.g., [15, 38, 51, 53]). However, despite the large amount of work in this area, part-based approaches have not been widely adapted for few-shot learning, where data is sparse, and the challenge lies in achieving reliable classification with minimal labeled examples.

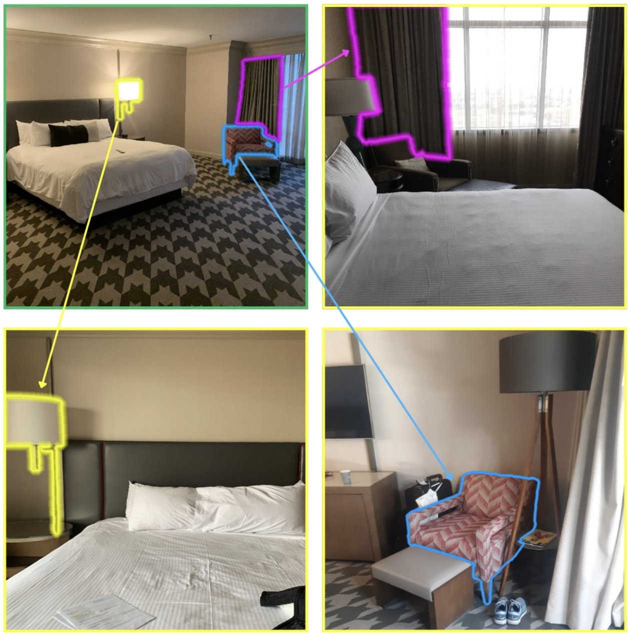

Figure 1. Four images from the same hotel vary widely in scene appearance yet share consistent parts (color-coded). SMPL learns to match these parts for fine-grained, few-shot classification where broader scene features are unreliable.

Our method, the Simple Matching Parts Learner (SMPL), builds on the parts-based paradigm in a straightforward manner. SMPL employs off-the-shelf components for part encoding and matching. Through a meta-training phase, SMPL learns to recognize distinctive parts across categories, equipping it with the capability to generalize to novel classes with just a few labeled examples. The simplicity is key; by leveraging established feature extractors and a streamlined matching process, our approach achieves competitive few-shot classification performance with a much lower computational overhead than more elaborate models. We demonstrate the benefits of this approach on existing benchmarks and a hotel room recognition dataset, which provides a more rigorous testing ground for few-shot methods, particularly in settings with subtle intra-class differences and diverse class representations. The contributions of this paper are as follows.

- We present a simple and efficient method for part-based, few-shot classification.
- Our method achieves state-of-art performance on multiple few-shot benchmarks with lower computational cost than competing approaches.
- We repurpose an existing hotel recognition dataset as a challenging few-shot benchmark, leveraging its large number of classes and high intra-class variability to better evaluate fine-grained few-shot classification performance compared to common benchmarks.

2. Related Work

The literature on few-shot learning (FSL) is vast; see [30] for a survey. Within the inductive, meta-learning paradigm, there are three dominant approaches. *Optimization-based* methods learn a novel set of weights for each task (e.g., [2, 10, 16, 17, 21, 22, 31, 32, 36, 50, 55]). *Metric-based* methods rely on a learned distance or matching function (e.g., [4, 6, 7, 13, 40, 41, 48, 52]). The third category, *model-based* methods, are trained to directly classify novel tasks, which is the paradigm that SMPL follows. In this section, we cover the related model-based FSL methods in addition to matching and benchmarking in FSL.

Model-Based FSL Model-based methods bypass gradient updates and a global feature space, prioritizing rapid adaptation. Some store support representations in external memory [3, 37], while others train networks to predict meta-learner parameters [25–27, 34]. MetaNAS [9] uses neural architecture search to optimize both weights and architecture, and SNAIL [24] structures the few-shot input as a sequence, using an RNN for classification. Similarly, SMPL employs a meta-learning phase to learn part-based matching for few-shot tasks.

FSL Matching Techniques Previous FSL methods have employed explicit matching techniques. Early approaches extract a global feature from each image and rely on nearest-neighbor search, sometimes pooling support image embeddings from the same class [7, 40, 48]. For fine-grained tasks, more recent methods sacrifice the computational efficiency of global features to compute matching scores from local features. RelationNet [41] applies convolutions across concatenated feature maps from the query and each support class to compute a relation score, while LRPABN [14] learns an alignment network applied to bilinearly pooled features. Local features have been matched in various ways, including as sums of distances across corresponding local features [4], Earth Mover's Distance to compare local feature distributions [52], and feature matching via embedded cross-attention modules [13]. These methods, however, do not distinguish between the images within a support class prior to matching. In contrast, methods such as MATA [6] RenNet, [19], and CPEA [11] perform local feature matching without intra-class pooling.

FSL Benchmarks Few-shot classification methods have been evaluated on general image recognition benchmarks repurposed for the few-shot paradigm, like CIFAR-FS [2], miniImageNet [48], and tieredImageNet [43]. Often, the breadth of the task is kept relatively narrow, with 5-way being the most common. However, these datasets often fail to capture the complexity and specificity of real-world applications. As such, fine-grained datasets such as CUB [49] (birds), Stanford-Cars [20], and VGG Flower [28] have become more common. Still, these fine-grained alternatives do not typically have a noticeable difference in global similarity between the query and support sets. We repurpose a large scale hotel room image dataset, Hotels-8k [18], for few-shot learning by curating the dataset such that query and support sets consist of images taken from different guests at different times in different rooms, introducing variations in room type, lighting conditions, and object composition. This results in a more natural and complex domain shift compared to previous benchmarks.

Summary Recent FSL methods have increasingly favored metric-based and optimization-based paradigms over model-based approaches. While SMPL incorporates elements of metric-based methods, it revisits the model-based strategy by leveraging recent advances in network architectures and semantic segmentation. Additionally, SMPL introduces novel strategies for learning part-to-part correspondences, demonstrating their effectiveness across a wide range of benchmarks, including a new object-centric dataset that presents fresh challenges in the few-shot setting.

3. Method

Simple Matching Parts Learner (SMPL), outlined in Figure 2, is our parts-based, few-shot classification method. In this section, first, we establish the foundation for the few-shot learning (FSL) setting and introduce the notation used in the paper. Next, we describe the two main aspects of SMPL: part encoding and part matching, and conclude with the algorithm for training and deployment.

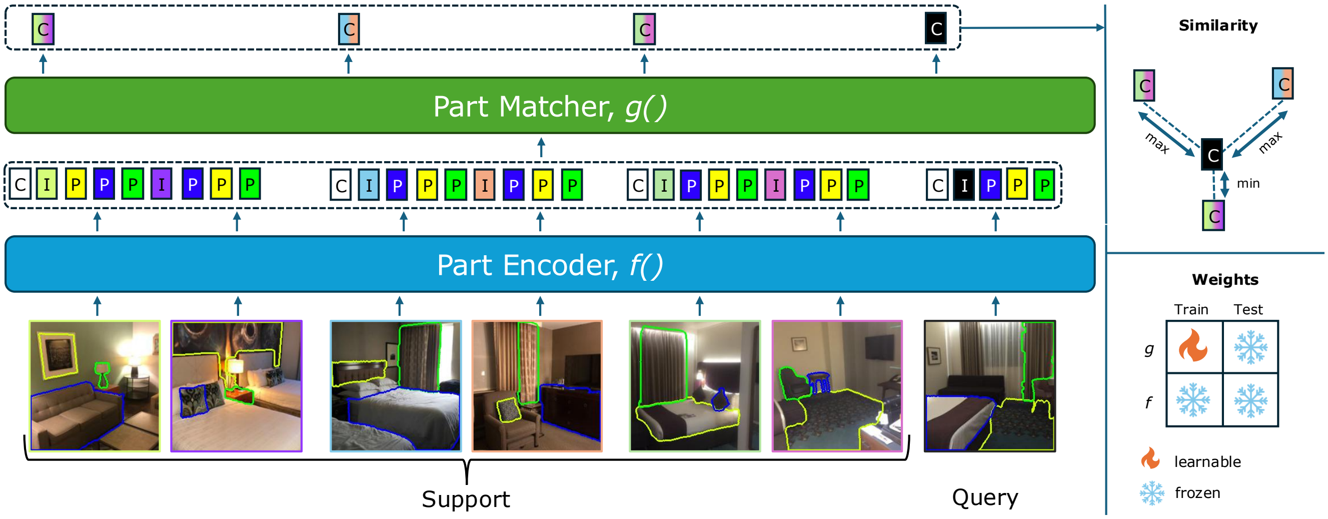

Figure 2. Illustration of SMPL for a 3-way, 2-shot task. Parts are extracted from images from the support and query sets. Part (P), image (I), and class (C) tokens are passed to a Transformer and the output class (C) tokens are compared for classification. In training, the parameters of the part matcher are optimized, and frozen during testing. Part and image encoder parameters are frozen for both training and testing.

3.1. Background

For FSL, we have a base dataset of images, $\mathcal{X}_{base}$, with known labels, $\mathcal{Y}_{base}$, for meta-learning. For evaluation, there is a set of images and labels, $\mathcal{X}_{novel}$ and $\mathcal{Y}_{novel}$, respectively, that are disjoint from their base counterparts. These sets can be subsampled to generate a few-shot task consisting of a support set $\mathcal{S} = \{(x_i, y_i)\}$ with NK images from N classes with K images each and query set, $\mathcal{Q} = \{(x_i, y_i)\}$, with separate images drawn from the same label space as $\mathcal{S}$. For our approach, we generate few-shot episodes during both training and testing. Our method relies on a model for part encoding and another for part matching. Let $\mathbf{f}$ be the output of the pre-trained feature encoder, f. The learnable part matching model, $\mathcal{Z} = g_\theta(\mathcal{T})$, is parameterized by θ and outputs a token sequence, $\mathcal{Z}$, given input sequence, $\mathcal{T}$.

3.2. Part Encoding

For part extraction, our approach leverages recent advances in image semantic segmentation to isolate and represent each part within an image, as illustrated in Figure 3. Following the approach described in [38], each part is represented by the mean pooled feature values extracted from a pre-trained model, providing a D-dimensional descriptor for each segment. This approach ensures that the feature representation remains compact for computational efficiency, while capturing essential discriminative characteristics. In Section 4.4.2, we investigate various approaches for part feature extraction and representation, comparing the performance. Additionally, to capture broader contextual information, we compute an image-level feature, also of dimensionality D, to complement the part-level representations. In Section 4.4, we demonstrate that few-shot performance is enhanced by incorporating both local (part-based) and global (image-level) perspectives.

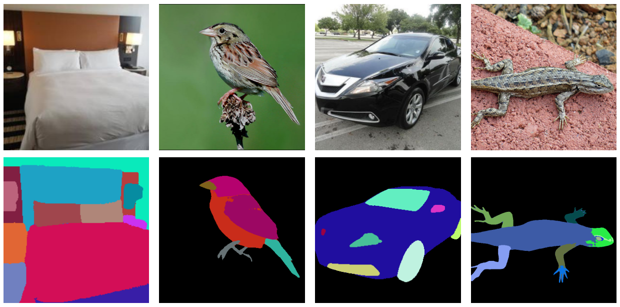

Figure 3. SMPL uses off-the-shelf segmentation methods to extract parts from images. For each image, a feature is generated for each part in addition to a image-level context feature.

3.3. Part Matching

The core of SMPL is the meta-learning phase which learns to match parts. The input is a sequence of tokens, $\mathcal{T} = \{\mathbf{t}_i^j\}$, for each image i and part j. We designate the query as $i = 0$ and the support images as $i = \{1, 2, \cdots, NK\}$ for an N-way, K-shot task. For a given image, the features include the local features and global context feature. Given a feature representation, $\mathbf{f}_i^j$, the corresponding token is:

$$\mathbf{t}_i^j = \mathbf{E}^\top \mathbf{f}_i^j + \mathbf{C}_i \qquad (1)$$

where $\mathbf{C}_i$ is a fixed sinusoidal embedding [47] that can take on one of $N + 1$ distinct values, distinguishing whether a token originates from the query or from one of the support classes. $\mathbf{E}$ is an optional learnable linear projection matrix,

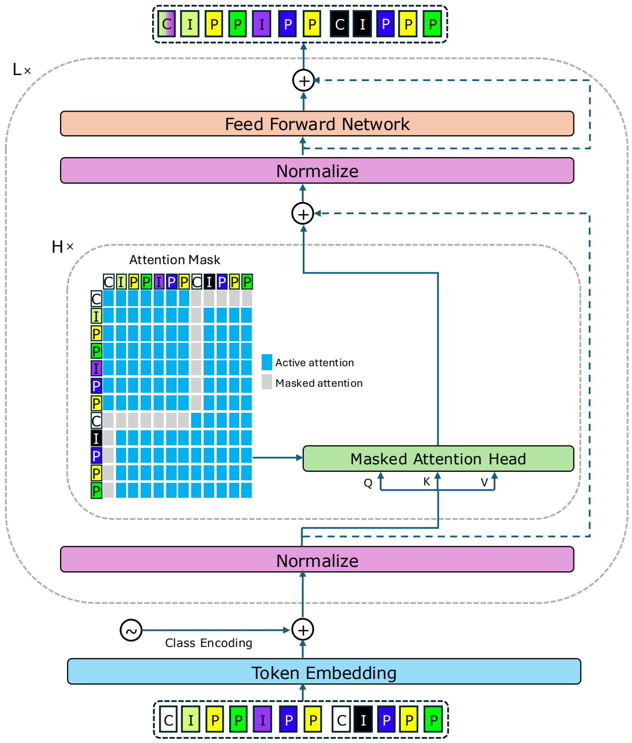

Figure 4. Part matching Transformer architecture diagram.

which can be used if the dimensionality of the feature from the pre-trained model differs from the input dimensionality of the part matching model, $g_\theta()$. For the query and each of the N support classes, we include a learnable class token to serve as the aggregate representation for matching.

Figure 4 shows the part matching Transformer of SMPL, which uses selective attention to match the set of part features extracted from the query set with the set of features extracted from the support set. The network consists of L encoder blocks, each containing a multi-head attention module with H heads followed by a 2-layer MLP with a ReLU activation. The attention modules use class-selective attention via masking, such that each class token attends exclusively to its respective part and context tokens. We follow the common approach to masked attention [8], where masked entries of the query-key matrix are set to a value representing negative infinity, effectively preventing attention between certain tokens. The part and context tokens across the query and support sets are fully connected.

From the output token sequence, $\mathcal{Z} = g_\theta(\mathcal{T})$, the class tokens are used to match. We denote the output class tokens as $\mathbf{z}_0$ for the query and $\mathbf{z}_1 ... \mathbf{z}_N$ for the support classes. The query class token is compared to each of the support class tokens with cosine similarity and normalized using softmax. Following previous work [7], we incorporate a tunable temperature parameter τ to scale the similarity scores, which leaves the following query-support matching score:

$$\hat{y}_i = \frac{\exp\left(\tau\langle \mathbf{z}_0, \mathbf{z}_i\rangle\right)}{\sum_{i=1}^{N} \exp\left(\tau\langle \mathbf{z}_0, \mathbf{z}_i\rangle\right)} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity. The part matching network can be optimized with cross-entropy loss.

3.4. SMPL Algorithm

Algorithm 1 outlines the meta-training process for SMPL, which uses multi-task training, such that the number of shots, K and number of ways, N, for each training task are randomly sampled between $[K_{min}, K_{max}]$ and $[N_{min}, N_{max}]$, respectively. The ensures that a single model can be deployed to handle tasks of varying and unknown sizes.

Algorithm 1 SMPL

```
1:  while not converged do
2:      ▷ Sample N-way, K-shot task                                   ◁
3:      $K \leftarrow$ RANDINT($K_{\min}$, $K_{\max}$)
4:      $N \leftarrow$ RANDINT($N_{\min}$, $N_{\max}$)
5:      $\mathcal{Q}, \mathcal{S} \sim (\mathcal{X}_{base}, \mathcal{Y}_{base})$
6:      $\mathcal{F} \leftarrow$ COMPUTE-FEATURES($\mathcal{Q}$,$\mathcal{S}$)
7:      for all $\mathbf{f} \in \mathcal{F}$ do
8:          ▷ Generate part matching input tokens                     ◁
9:          $\mathbf{t} \leftarrow$ TOKENIZE($\mathbf{f}$)                       ▷ Eq. 1
10:         $\mathcal{T} \leftarrow \mathcal{T} \cup \mathbf{t}$
11:     $\mathcal{T} \leftarrow$ ADD-CLASS-TOKENS($\mathcal{T}$, $N+1$)
12:     $\mathcal{Z} \leftarrow g_\theta(\mathcal{T})$            ▷ Compute output class tokens
13:     $\mathbf{y} \leftarrow$ MATCH-SCORE($\mathcal{Z}$)                       ▷ Eq. 2
14:     ▷ Compute loss and update weights                             ◁
15:     $\theta \leftarrow$ WEIGHT-UPDATE($\theta$, $\nabla\mathcal{L}$)
```

Testing follows the training process, with two modifications. The number of shots and ways, query image, and support images are provided as input rather than sampled (line 3-5). There is no optimization step (line 15); the highest similarity support class is returned as the best match.

4. Results

Datasets We evaluate SMPL on 5 fine-grained datasets. 4 are single-object datasets commonly used for few-shot evaluation: CUB [49], Stanford-Cars [20], Reptilia [46], and VGG-Flower [28]. We follow the most common protocol for training, validation, and testing splits and report accuracy on 1,000 1-shot/20-way and 5-shot/20-way tasks from the test set with 15 queries per class. We also evaluate on the hotel recognition dataset, Hotels-8k [18], which contains over 100,000 images from nearly 8,000 hotels. We repurpose the dataset for the few-shot setting by ensuring that query and support set images in the test set are submitted by different guests and use a 70/10/20 class split. Hotels-8k contains classes with only a small number of examples, so each task includes 3 queries per class.

Implementation We use Grounded-SAM [33] for part segmentation and a DINO [5] pre-trained ViT-Base model with 16×16 patches for feature extraction. The part matching transformer consists of 4 layers and 8 attention heads, with 768-d features. The model is trained with stochastic gradient descent with a 1-cycle learning rate scheduler [39]. The temperature parameter is $\tau = 20$. The training episode ranges are $[K_{min}, K_{max}] = [1, 5]$ and $[N_{min}, N_{max}] = [5, 20]$. Regularization includes random dropout of 20% of the input features and label smoothing [42].

Baselines We compare SMPL with the following few-shot methods: R2D2 [2], MetaOptNet [21], Linear Probing, full fine-tuning (FT), VPT [17], SSF [22], ProtoNet [40], RelationNet [41], MetaBaseline [7], MATA [6], and the FORT [50] variants, FORT-FT and FORT-SSF.

4.1. Few-Shot Classification

For each dataset, we trained separate baseline models for 1-shot and 5-shot tasks, while SMPL uses a single multi-task model for both. Table 1 shows that SMPL matches or surpasses all baselines across the four datasets and settings, demonstrating its robustness and adaptability. For Hotels-8k, CUB, and Reptilia, SMPL outperformed the next closest method by 13%, 8% and 14%, respectively, for 1-shot classification. For CUB, though all the methods performed well, the substantial improvement of SMPL over other methods suggests that it effectively captures subtle inter-class differences. The Reptilia dataset presented unique challenges due to the less distinctive visual characteristics of the reptile classes. However, SMPL still exceeded the performance of the baselines by wide margins. Based on the aggregate results, Hotels-8k was the most challenging dataset for few-shot classification, where methods with competitive performance on other datasets performed significantly worse than SMPL and the other top methods. This dataset exhibits a much higher level of query-support distribution shift in the form of object composition, lighting, and viewpoint compared to the other benchmarks.

Figure 5 illustrates challenging 1-shot classification cases where SMPL accurately classified the query. Each row presents a query image, the correct match from the support class, and a false positive selected by multiple baseline methods. These examples underscore the advantages of the part-centric approach of SMPL. In the first row, the small red pillow is a distinguishing feature, though the false positive exhibits high visual similarity in the bed, headboard, and red wall. In the second row, there is a distinct pattern on the seating for the correct match. In the third row, the beak and white underbelly match, even though the incorrect image displays a similar-looking bird in the same pose. In the fourth row, there are subtle details like the matching side-view mirrors, windshield, and side decals on the correct car. In each of these cases and others, SMPL successfully pairs the query with the correct support image, even in the presence of highly similar distractors.

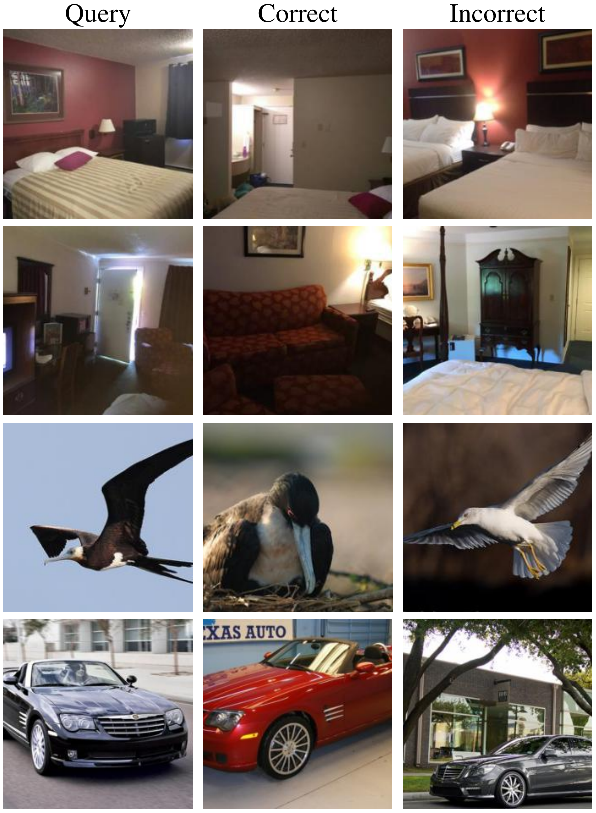

Figure 5. Each row shows a query, correct match, and false positive match selected by multiple baseline methods.

4.2. Performance Analysis

SMPL training is efficient as it does not require fine-tuning the backbone, significantly reducing computation by leveraging precomputed part features. As shown in Table 2, SMPL requires only a fraction of GPU memory and TFLOPs compared to other meta-learning approaches.

For inference, Figure 6 plots the average GPU memory consumption and inference time per episode on 20-way tasks with the Reptilia dataset. For each method, we plot for both the 1-shot and 5-shot measures. The optimization-based methods, VPT (purple) and FORT-SSF (yellow), took longer, due to the optimization steps involved for inference. Most metric-based methods, such as RelationNet (green), typically were faster, but required much higher memory consumption. MATA (blue), in particular, scaled poorly, as the memory usage increased linearly with the size of the task. SMPL (orange) not only achieves superior classification performance compared to the other methods but also competitive computational efficiency across both dimen-

	Hotels-8k		CUB		CARS		Reptilia		VGG-Flower	
Method	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
R2D2	.171	.445	.608	.881	.245	.629	.433	.671	.793	.972
MetaOptNet	.168	.420	.611	.887	.237	.589	.437	.693	.799	.974
Linear Probing	.126	.325	.438	.814	.176	.466	.358	.639	.602	.937
FT	.177	.457	.612	.907	.238	.624	.432	.684	.812	**.978**
VPT	.158	.375	.570	.856	.229	.563	.404	.655	.774	.964
SSF	.172	.452	.612	.915	.235	.638	.426	.687	.814	.977
FORT-FT	.172	.434	.622	.912	.245	.639	.435	.693	.810	**.978**
FORT-SSF	.152	.383	.653	.901	.249	.610	.432	.674	.801	.970
ProtoNet	.238	.362	.658	.883	.610	.853	.322	.532	.756	.964
RelationNet	.446	.580	.642	.780	.476	.773	.318	.513	.599	.787
MATA	.388	.729	.696	.868	.706	.864	.384	.583	.820	.963
MetaBaseline	.452	.676	.762	.889	**.711**	.867	.474	.667	.854	.966
SMPL	**.515**	**.737**	**.826**	**.934**	**.711**	**.876**	**.539**	**.717**	**.858**	**.978**

Table 1. Few shot classification accuracy on 20-way tasks. All methods use a DINO-pretrained ViT backbone.

Method	Mem (GB)	TFLOPs
RelationNet	39.9	51.3
MATA	66.1	27.3
MetaBaseline	36.3	41.6
SMPL	6.62	2.78

Table 2. Training efficiency on the Reptilia dataset.

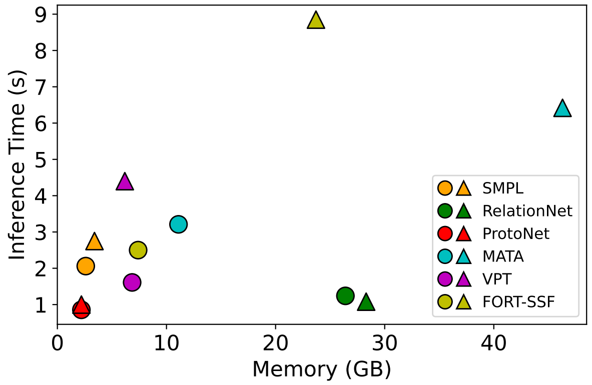

Figure 6. Computational efficiency (time and GPU memory) for 1-shot (circle) and 5-shot (triangle) inference on the Reptilia dataset.

sions. While the metric-based ProtoNet (red) was slightly more efficient, SMPL outpeformed it on 1-shot and 5-shot classification on this dataset by 67% and 35% respectively.

4.3. Part Similarity

Figure 7 shows the 5-shot classification performance on the Hotels-8k dataset as a function of the part similarity between the query and the correct support set class. Part similarity, which ranges from 0 to 1, is calculated as the overlap coefficient between the matching part types from each set.

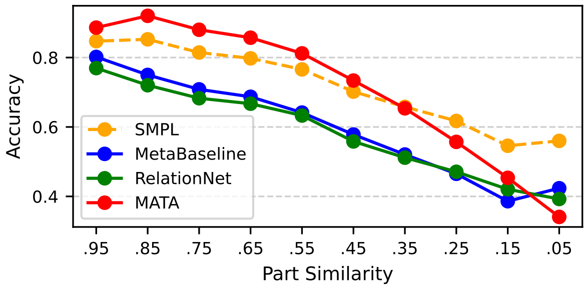

Figure 7. 5-shot classification accuracy on Hotels-8k dataset as a function of query-support part similarity.

As part similarity decreases, all methods experience a reduction in accuracy, but the rate of decline varies greatly. The performance of SMPL is least affected, highlighting its robustness in handling complex intra-class variations. Comparatively, the performance of MATA is quite high when part similarity is high, but rapidly declines, reaching the lowest accuracy among all methods for the most dissimilar cases. SMPL outperformed these methods in aggregate; this analysis indicates that the greatest benefit was obtained for the most difficult, visually dissimilar cases.

4.4. Ablation Analysis

We conducted a comprehensive ablation analysis to investigate feature extraction, encoding, and multiple aspects of the SMPL training process. Unless otherwise specified, the few-shot classification values were obtained on the Hotels-8k dataset using a DINO-pretrained ViT backbone.

4.4.1. Input Features

The ablation study, summarized in Table 3, highlights the individual and combined contributions of key components

	Component			Accuracy	
#	Part Feat	Img Feat	Class Enc	1-shot	5-shot
1		✓		.377	.604
2	✓			.481	.708
3		✓	✓	.384	.613
4	✓		✓	.504	.727
5	✓	✓		.493	.722
6	✓	✓	✓	.515	.737

Table 3. Ablation study showing the impact of feature encoding components on 1-shot and 5-shot accuracy on Hotels-8k.

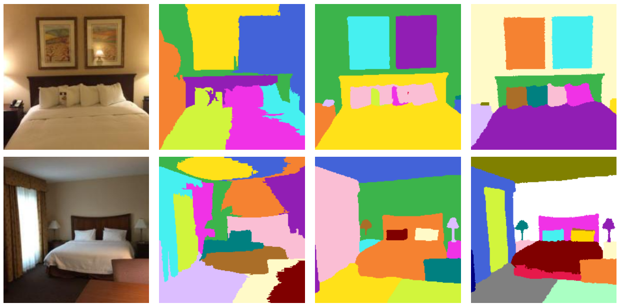

Figure 8. Each row shows (L-R) an image and segmentation map produced by SuperPixels, Swin-Tiny and GroundedSAM.

of part encoding: *part features*, *(global) image feature*, and *class encoding*. For clarity, we refer to each configuration by its row number (e.g., #3). Simply using a global image feature to represent the image (#1) establishes a modest baseline with a 1-shot accuracy of 0.377 and a 5-shot accuracy of 0.604. This configuration shows that even though DINO features provide strong image-level representations, their effectiveness is limited in the few-shot setting alone. Alternatively, using only part features (#2), we see the model's performance improve significantly for both tasks, with a 27.6% and 17.2% increase over using only the image feature, respectively. This highlights the importance of localized part information, where specific part details can offer distinct cues. For each combination of feature types (part-only, image-only, and both), the addition of class encoding yields an improvement, ranging from 1.8% to 4.7% for 1-shot and 1.4% to 2.6% for 5-shot. Class encoding is particularly beneficial when combined with part features. The combination of all three components in #6 produces the highest accuracy, indicating that the interplay of global image context, part information, and class encoding enhances the model's robustness in the few-shot setting.

4.4.2. Part Segmentation and Encoding

To analyze the sensitivity of SMPL to the accuracy of part segmentation or the representative power of the part features, we evaluated multiple combinations of segmentation methods and feature representations. We evaluated three segmentation methods: SuperPixels [1], Swin-Tiny [23] trained ADE-20k [54], and GroundedSAM. Figure 8 shows representative examples of the segmentation maps generated by each method. For part feature representation, we compare a (Supervised) ViT-Base pretrained on ImageNet [35], as well as DINO and DINOv2 [29], which are unsupervised methods.

	Supervised		DINO		DINOv2	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
SuperPixels	.464	.694	.473	.704	.576	.805
Swin-Tiny	.481	.709	.482	.715	.598	.813
GroundedSAM	.494	.719	.515	.737	.604	.819

Table 4. Comparison of few-shot performance on the Hotels-8k dataset using part features generated with different segmentation methods (rows) and feature representations (columns).

Table 4 shows the few-shot performance on Hotels-8k. We notice two, perhaps unsurprising, trends: better segmentation and better features lead to better few-shot performance for SMPL. The effect of feature representation appears to be stronger than that of the segmentation method. DINO and supervised features were trained on the same dataset with the same architecture; the modest increase in performance using DINO aligns with recent work suggesting that unsupervised methods generally produce more salient local representations [38]. The DINOv2 features are the best performers with SMPL and least affected by semantic segmentation quality, particularly in the 5-shot setting. The modular SMPL approach can take advantage of advances in segmentation or feature representations.

4.4.3. Attention Schemes

Selective attention has been shown to be beneficial in terms of computational efficiency and model performance [12]. Here, we analyze the effect of the following selective attention schemes: (1) Full: The default setting for full pairwise attention between all token types. (2) Class-Selective: Part and image tokens fully attend to each other, but class tokens only attend to part and image tokens of the corresponding class. (3) Intra-Class: Tokens only attend to tokens in the same class. (4) Part-Selective: Tokens only attend to tokens of the same part category and class tokens attend to the tokens in the same class. For all schemes, *class* refers to the (known) support classes and (unknown) query class. Figure 9 shows the diagrams for the four schemes (a-d) and their performance on Hotels-8k.

The class-selective approach of SMPL outperforms the other schemes in both the 1-shot and 5-shot settings. The full attention scheme is underconstrained and no learning takes place; the classification accuracy of 0.050 corresponds to chance performance on the 20-way task. A variant of the full scheme with masked attention between the class tokens produces the same result. Of the viable alternatives, the intra-class scheme results in the worst performance; this

	P	I	C
P	✓	✓	✓
I	✓	✓	✓
C	✓	✓	✓

(a) Full

	P	I	C
P	✓	✓	*
I	✓	✓	*
C	*	*	

(b) Class-Selective

	P	I	C
P	*	*	*
I	*	*	*
C	*	*	

(c) Intra-Class

	P	I	C
P	*		*
I		✓	*
C	*	*	

(d) Part-Selective

Scheme	1-shot	5-shot
Full	.050	.050
Class-Selective	**.515**	**.737**
Intra-Class	.484	.708
Part-Selective	.495	.724

(e) Performance

Figure 9. Selective Attention. For each scheme (a-d), the chart shows the attention relationship between the part (P), image (I), and class (C) tokens as ✓(full) or * (selective), with blank indicating no attention. (e) Few-shot performance on Hotels-8k.

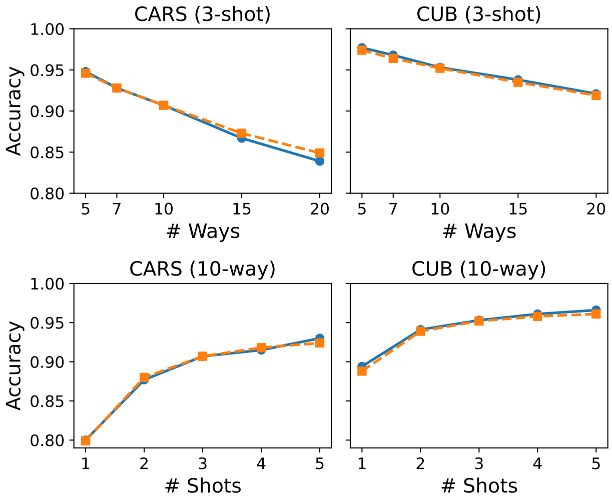

Figure 10. Multi-task training. SMPL performance for specific (blue) and multi-task (orange) training on CARS and CUB.

suggests that cross-class matching is integral to training. The part-selective scheme works nearly as well as the class-selective scheme, suggesting that the interplay of different part types possible in the class-selective approach provides additional contextual cues for matching.

4.4.4. Multi-task Training

Most of the competing few-shot approaches follow a dedicated training scheme where the number of shots and ways must be specified prior to meta-training and the model can only be deployed for that setting. For example, for MetaBaseline, deploying a model trained for 1-shot on a 5-shot task results in a 14.6% drop in classification accuracy compared to using the matching model. SMPL employs multi-task training, where the training episode task sizes are randomly sampled. Figure 10 compares the performance of specific versus random multi-task meta-training using SMPL on two datasets. The blue points show the performance with the model trained to the corresponding N-way, K-shot setting and the orange points show the performance using the single model trained with the number of shots and ways uniformly sampled between N_{min} and N_{max} and K_{min} and K_{max}, respectively. Multi-task SMPL performs as well the dedicated models, including, surprisingly, outperforming the dedicated model for the 20-way, 3-shot experiment on CARS.

5. Conclusion

We introduced SMPL, a simple combination of semantic segmentation, attention, and multi-task training for few-shot learning that outperformed existing methods across multiple datasets, varying in domain, part overlap, and intra-class similarity. Through extensive experiments, we demonstrated that SMPL is highly effective, achieving state-of-the-art performance in various few-shot settings.

A limitation of our approach is part map generation, which can involve prompting text-guided segmentation models to produce part labels at the required level of granularity and may require careful tuning to ensure consistent and accurate part segmentation, which can be challenging for complex or highly variable objects. While the results of Section 4.4.2 suggest that SMPL is robust to inaccurate segmentation, this dependency may limit the generalizability of the approach, particularly to specialized domains for which vision-language models are not yet grounded.

Future work will focus on reducing the level of supervision for part map generation and exploring ways to dynamically adapt the level of part granularity based on task complexity and/or dataset characteristics. Additionally, we plan to extend SMPL to multi-modal settings, where it can leverage both visual and textual data to enhance part-based representations, potentially improving performance in cases where visual data alone is insufficient.

Acknowledgements

This research includes calculations carried out on HPC resources supported in part by the National Science Foundation through major research instrumentation grant number 1625061 and by the US Army Research Laboratory under contract number W911NF-16-2-0189. Special thanks to Amir Aghdam for his help with designing figures.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 7

[2] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. 2, 5

[3] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4080–4088, 2018. 2

[4] Kaidi Cao, Maria Brbic, and Jure Leskovec. Concept learners for few-shot learning. In *International Conference on Learning Representations*, 2021. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. International Conference on Computer Vision*, pages 9650–9660, 2021. 5

[6] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Multi-scale adaptive task attention network for few-shot learning. In *Proc. International Conference on Pattern Recognition*, pages 4765–4771. IEEE, 2022. 2, 5

[7] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proc. International Conference on Computer Vision*, pages 9062–9071, 2021. 2, 4, 5

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 4

[9] Thomas Elsken, Benedikt Sebastian Staffler, Jan Hendrik Metzen, and Frank Hutter. Meta-learning of neural architectures for few-shot learning. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 12362–12372, 2019. 2

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 2

[11] Fusheng Hao, Fengxiang He, Liu Liu, Fuxiang Wu, Dacheng Tao, and Jun Cheng. Class-aware patch embedding adaptation for few-shot image classification. In *Proc. International Conference on Computer Vision*, pages 18905–18915, 2023. 2

[12] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023. 7

[13] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[14] Huaxi Huang, Junjie Zhang, Jian Zhang, Jingsong Xu, and Qiang Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. *IEEE Transactions on Multimedia*, 23:1666–1680, 2020. 2

[15] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1182, 2015. 1

[16] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019. 2

[17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proc. European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2, 5

[18] Rashmi Kamath, Gregory Rolwes, Samuel Black, and Abby Stylianou. The 2021 hotel-id to combat human trafficking competition dataset. *arXiv preprint arXiv:2106.05746*, 2021. 2, 4

[19] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *Proc. International Conference on Computer Vision*, pages 8822–8833, 2021. 2

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshop*, pages 554–561, 2013. 2, 4

[21] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 2, 5

[22] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 2, 5

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. International Conference on Computer Vision*, pages 10012–10022, 2021. 7

[24] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. 2

[25] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proc. International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017. 2

[26] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *Proc. International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018.

[27] A Nichol. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2

[28] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2, 4

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez,

Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 7

[30] Archit Parnami and Minwoo Lee. Learning from few examples: A summary of approaches to few-shot learning. *ArXiv*, abs/2203.04291, 2022. 2

[31] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[32] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 2

[33] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5

[34] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 7

[36] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. 2

[37] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proc. International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1842–1850, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[38] Michal Shlapentokh-Rothman, Ansel Blume, Yao Xiao, Yuqun Wu, Sethuraman TV, Heyi Tao, Jae Yong Lee, Wilfredo Torres, Yu-Xiong Wang, and Derek Hoiem. Region-based representations revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17107–17116, June 2024. 1, 3, 7

[39] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 5

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 5

[41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2, 5

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 5

[43] Eleni Triantafillou, Hugo Larochelle, Jake Snell, Josh Tenenbaum, Kevin Jordan Swersky, Mengye Ren, Richard Zemel, and Sachin Ravi. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. 2

[44] Peter Uršič, Rok Mandeljc, Aleš Leonardis, and Matej Kristan. Part-based room categorization for household service robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2287–2294, 2016. 1

[45] Robert van der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F. Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1866–1876, 2023. 1

[46] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 4

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016. 2

[49] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds 200. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 4

[50] Haoqing Wang, Shibo Jie, and Zhi-Hong Deng. Focus your attention when few-shot classification. In *Advances in Neural Information Processing Systems*, 2023. 2, 5

[51] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognit.*, 76:704–714, 2018. 1

[52] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020. 2

[53] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, A. Elgammal, and Dimitris N. Metaxas.

Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1143–1152, 2016. 1

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 7

[55] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *Proc. International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019. 2