Supplementary Material for Fine-grained Few-shot Recognition by Deep Object Parsing

Ruizhao Zhu
rzhu@bu.edu
Pengkai Zhu
zpk@bu.edu
Samarth Mishra
samarthm@bu.edu
Venkatesh Saligrama
srv@bu.edu

Boston University
Boston, MA, US

Abstract

This supplementary document details DOP's derivation and algorithm for few-shot learning, along with further comparisons with other methods. Additional experimental details, visualizations, and results are included. We provide code that reproduces our findings, to be published alongside the paper.

A. Additional Details of DOP

A1. Derivation of PARSE

As mentioned in Sec. 3 of the main paper, estimating part expression and location leads to two coupled optimization problems.

$$z_p(\mu) = \arg\min_{\beta} \sum_{c\in C} \|\phi_{c,M(\mu)} - D_{p,c}\beta_c\|^2 + \lambda\|\beta\|_1. \tag{1}$$

$$\mu_p = \arg\min_{\mu\in[G]\times[G]} \left[L_p(\mu) \triangleq \sum_{c\in C} \|[\phi_c]_{M(\mu)} - D_{p,c} z_{p,c}(\mu)\|^2 + \lambda\|z_p(\mu)\|_1 \right] \tag{2}$$

For solving the above, we first approximate the solution to Equation (1) by optimizing the reconstruction error and subsequently thresholding. As mentioned in the main paper, this is closely related to thresholding methods employed in LASSO [3]. So, first we solve

$$z'_p(\mu) = \arg\min_{\beta} \sum_{c\in C} \|\phi_{c,M(\mu)} - D_{p,c}\beta_c\|^2$$

As a reminder, the subscript $M(\mu)$ refers to the projection of ϕ_c onto the support of $M(\mu)$, which is an $s \times s$ grid centered at μ. The quadratic form of the above optimization problem, gives us an explicit solution.

$$z'_{p,c}(\mu) = \frac{(D_{p,c} * \delta_\mu) : \phi_c}{\|D_{p,c}\|^2} = \frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2} \tag{3}$$

where $\delta_\mu(v) = \delta(\mu - v), v \in [G] \times [G]$ is a dirac delta centered at μ, $*$ is a convolution[1] : $D_{p,c} * \delta_\mu(v) = \sum_w D_{p,c}(w - v)\delta_\mu(v)$ and ':' is the double-dot product or the sum of all elements of an element-wise/Hadamard product.

For estimating location, we substitute z'_p into Equation (2) resulting in an upper bound for $L_p(\mu)$, which we denote as $L'_p(\mu)$.

$$\begin{aligned}
L_p(\mu) \leq L'_p(\mu) &= \sum_{c\in[C]} \|[\phi_{c,M(\mu)} - D_{p,c} z'_{p,c}(\mu)\|^2 + \lambda \|z'_p\|_1 \\
&= \sum_{c\in[C]} \left[\|\phi_{c,M(\mu)}\|^2 - 2(\phi_{c,M(\mu)} : D_{p,c}) z'_{p,c} + \|D_{p,c} z'_{p,c}\|^2 + \lambda |z'_{p,c}|\right] \\
&\overset{(1)}{=} \sum_{c\in[C]} \left[\|\phi_{c,M(\mu)}\|^2 - 2(\phi_{c,M(\mu)} : D_{p,c}) \frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2} \right. \\
&\qquad \left. + \|D_{p,c}\|^2 . \frac{(D_{p,c} * \phi_c)(\mu)^2}{\|D_{p,c}\|^4} + \lambda \frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2}\right] \\
&\overset{(2)}{=} \sum_{c\in[C]} \left[\|\phi_{c,M(\mu)}\|^2 - \frac{(D_{p,c} * \phi_c)(\mu)^2}{\|D_{p,c}\|^2} + \lambda \frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2}\right] \\
&= \sum_{c\in[C]} \left[\|\phi_{c,M(\mu)}\|^2 - \frac{(D_{p,c} * \phi_c)(\mu)^2}{\|D_{p,c}\|^2} + \lambda \frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2} \right. \\
&\qquad \left. - \frac{\lambda^2}{4\|D_{p,c}\|^2} + \frac{\lambda^2}{4\|D_{p,c}\|^2}\right]
\end{aligned} \tag{4}$$

For step (1) above, we substitute $z'_{p,c}$ from Equation (3). For step (2), note that $D_{p,c} : \phi_{c,M(\mu)} = (D_{p,c} * \phi)(\mu)$, since $M(\mu)$ is an $s \times s$ attention map centered at μ.

From Equation (4), by ignoring the first and the last terms and contracting the binomial squares, we get the following as our estimate for μ_p. Note that the last term is ignored because it does not depend on μ. Also, the first term $\sum_{c\in[C]} \|\phi_{c,M(\mu)}\|^2$, which is the energy across all channels varies little for different values of μ.

$$\begin{aligned}
\mu_p = \underset{\mu\in[G]\times[G]}{\arg\min}\ L'_p(\mu) &= \underset{\mu\in[G]\times[G]}{\arg\min} - \sum_{c\in C} \left[\frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|} - \frac{\lambda}{\|D_{p,c}\|}\right]^2 \\
&= \underset{\mu\in[G]\times[G]}{\arg\max} \sum_{c\in C} ((\theta_{p,c} * \phi_c)(\mu) - \lambda_c)^2
\end{aligned} \tag{5}$$

$\theta_{p,c} = D_{p,c}/\|D_{p,c}\|$, and $\lambda_c = \lambda/2\|D_{p,c}\|$ becomes a channel dependent constant. The location estimate in Equation (5), is thus, in the form of template matching per channel.

[1] Note that following terminology from signal processing this is not actually a convolution but a cross-correlation. However, the way we use this term has been accepted in literature surrounding convolutional neural networks.

Differentiable Estimates. As mentioned in the main paper, the above estimate (Equation (5)) for μ_p does not provide any gradients for the parameters in $\theta_{p,c}$ or those involved in computing ϕ_c. We make the estimate differentiable in its parameters by approximating the argmax as the expectation of a softmax distribution ν_p over $[G] \times [G]$ with a low temperature T.

$$\nu_p(\mu) = \text{softmax}\left(\frac{1}{T}\sum_{c\in C}((\theta_{p,c} * \phi_c)(\mu) - \lambda_c)^2\right); \quad \mu_p = \mathbb{E}_{\mu\sim\nu_p}\mu \tag{6}$$

Substituting back the estimate of μ_p into Equation (3) again makes z_p unusable to get gradients (since μ_p is an index in a non-continuous domain $[G] \times [G]$). One workaround is estimating z_p as an expectation over ν_p of Equation (3) (similar to how μ_p is estimated).

$$z'_{p,c} = \mathbb{E}_{\mu\sim\nu_p}\left[\frac{(D_{p,c} * \phi_c)(\mu)}{\|D_{p,c}\|^2}\right]; \; z_{p,c} = S_\zeta(z'_{p,c}) \tag{7}$$

However, we found a different estimate turns out to be more accurate and performs better in practice. Using the first expression from Equation (3)

$$z'_{p,c} = \frac{(D_{p,c} * \delta_{\mu_p}) : \phi_c}{\|D_{p,c}\|^2} \approx \frac{(D_{p,c} * \widehat{\delta}_{\mu_p}) : \phi_c}{\|D_{p,c}\|^2} \tag{8}$$

We make this estimate of $z_{p,c}$ differentiable by using a differentiable approximation of δ_{μ_p}, $\widehat{\delta}_{\mu_p}$ which is a low-radius ($\sigma^2 = 0.25$) gaussian centered at μ_p. With the DOP model with 1 part, this estimate (Equation (8)) achieves an accuracy of 90.56% on 5-way 5-shot classification on the CUB dataset, while the estimate from Equation (7) achieves an accuracy of 89.46% on the same task.

A2. Few-shot Learning

Algorithm outlines the loss computation for a single query q. In Algorithm , η is a tunable parameter controlling the weight of the prior. In each episode, we use an average of the loss output over multiple query examples, which results in an end-to-end differentiable criterion in all trainable parameters, allowing us to optimize using gradient descent.

Algorithm Training loss for Few-shot recognition with DOP (single episode, single query)

Given: requirements for Algorithm 1 `PARSE`, weighting function α, tunable parameter γ
Input: Query example with ground truth label $q, y^{(q)} \in \mathcal{X} \times [N]$. Support examples $I = \bigcup_{y\in[N]}\left[I_y = \{x^{(i,y)}\}_{i\in[M]}\right]$
Trainable parameters: Convolutional backbone f, part templates $\{D_{s,p,c}\}_{s\in\mathcal{S},p\in[K],c\in[C]}$, weighting function α
Compute parses $\texttt{PARSE}(q) = (\{\mu^{(q)}_{s,p}\}, \{z^{(q)}_{s,p}\})$, $\texttt{PARSE}(x^{(i,y)}) = (\{\mu^{(i,y)}_{s,p}\}, \{z^{(i,y)}_{s,p}\}); s \in \mathcal{S}, p \in [K]$
Compute distances $d(q,y)$ for $y \in [N]$ using Equation (7)
Output: loss $\ell(q) = \ell_{CE}(q) + \eta\frac{1}{|I|+1}\sum_{x\in I\cup\{q\}}\ell_{div}(x)$ using Equation (8)(9)

B. More on Compared Methods

Prior works on FSC [2, 20, 21, 26, 29] have also focused on combining parts, albeit with different notions of the concept. As such, the term part is overloaded and is unrelated to our notion. DeepEMD [29] focuses in the image-distance metric based on an earth mover's distance between different parts. Parts are simply different physical locations in the image and not a compact collection of salient parts for recognition. [20] uses salient object parts for recognition, while [21] attempts to encode parts into image features. However, both these methods require additional attribute annotations for training, which are expensive to gather and not always available. [2] and [26] discover salient object parts and use them for recognition via attention maps similar to our method.

We compare DOP to state-of-the-art few-shot learning methods, including RENet [7], FRN [25],TDM [12] and DeepEMD[29] and also to methods like FOT [23], VFD [27], DN4 [14] and TDM [12], which are dedicated to the fine-grained setting. To highlight the contribution of DOP , we tabulate in Table 1 the differences of the model design compared to prior works [2, 21, 26, 29] in few-shot learning that also use part composition.

While there are prior works that learn recognition via object parts, and use instance-dependent reweighting, DOP is unique since it uses reconstruction with templates (RwT) as a criterion, uses a prior on the geometry of parts using part-locations and uses this geometry for comparing instances. See Table 1 for a tabulated comparison.

Note : There are some prior works where the notion of the term part is overloaded and is unrelated to our notion. Hence DeepEMD [29] and LCR [21] do not have a ✓under "Parts". LCR [21] attempts to encode parts into image features. DeepEMD [29] focuses in the image-distance metric based on an earth mover's distance between different parts. Here, parts are simply different physical locations in the image and not a compact collection of salient parts for recognition.

Again, FRN [25] does not have a ✓under "RwT". It uses a reconstruction objective, but attempts to reconstruct query features from support. While this helps in determining belongingness to a class based on how well the support features reconstruct query, the method does not use templates that are shared across all image instances, reconstruction using which allows for low noise representations.

Table 1: Similarities and differences in high-level use of components by DOP and prior work. Parts: recognition using parts; RwT: Reconstruction with Templates; Geo: using geometry of parts for instance comparison, and incorporating prior on geometry.; Reweighting: instance dependent reweighting of matching scores.

Methods	Parts	RwT	Geo	Reweighting
LCR [21]				
SAML [2]	✓			✓
DeepEMD [29]				✓
FRN [25]				
TPMS [26]	✓			✓
TDM [12]				✓
DOP (ours)	✓	✓	✓	✓

C. Additional Experiments

C1. Details on Dataset Settings

We compare DOP on four fine-grained datasets: Caltech-UCSD-Birds (CUB) [22], Stanford-Dog (Dog) [8] Stanford-Car (Car) [10] and Aircraft [17] against state-of-the-art methods.
Caltech-UCSD-Birds (CUB) [22] is a fine-grained classification dataset with 11,788 images of 200 bird species. Following convention[4], the 200 classes are randomly split into 100 base, 50 validation and 50 test classes.
Aircraft contains 100 classes of aircrafts and 10,000 images in total. Following recent benchmark [12, 25], we processes all images based on bounding box. And the 100 classes are split into 50, 25 and 25 classes for training, validation and test.
Stanford-Dog [8] is a dataset for fine-grained classification. Dog contains 120 dog breeds with a total number of 20,580 images. For few-shot learning evaluation, we follow the benchmark protocol proposed in [14]. Specifically, 120 classes of Dog are split into 70, 20, and 30 classes, for training, validation, and test, respectively. Similarly, Car is split into 130 train, 17 validation and 49 test classes.
Stanford-Car [10] is a dataset for fine-grained classification. Car consists of 16,185 images from 196 different car models. For few-shot learning evaluation, we follow the benchmark protocol proposed in [14]. Similarly to previous datasets, Car is split into 130 train, 17 validation and 49 test classes.

C2. Training details

Our model is trained with 10,000 episodes on CUB and 30,000 episodes on Stanford-Dog/Car for experiments with both ResNet12 and ResNet18. In each episode, we randomly select 10 classes and sample 5 and 10 samples as support and query data. The weight on the geometric prior η is set to 1.0 on CUB and 0.1 on Stanford-Dog/Car, respectively. We train from scratch with Adam optimizer [9]. The learning rate starts from 5e-4 on CUB and 1e-3 on Stanford-Car/Dog, and decays to 0.1x every 3,000 episodes on CUB and 9,000 episodes on Dog/Car. On CUB, objects are cropped using the annotated bounding box before resizing to the input size. On Stanford-Car/Dog, we use the resized raw image as the input. We employed standard data augmentations, including horizontal flip and perspective distortion, to the input images.

C3. What parts does DOP detect?

We visualize the locations μ_p learned by DOP in Figure 1. DOP is able to detect consistent parts for the same task and often finds semantically meaningful parts like head and torso/breast in birds and dogs and wheels and doors/windows on cars. Figure 1 also shows some failure cases of DOP , where it might fail to locate parts on the object if similar visual signatures appear in the background.

C4. Visualizing Templates and Part Expressions

Some templates of the learned dictionary D_p are visualized in Figure 2. Our model uses each template to reconstruct the original feature in the corresponding channel. We see diverse visual representations in different channels, implying that DOP learns diverse visual templates

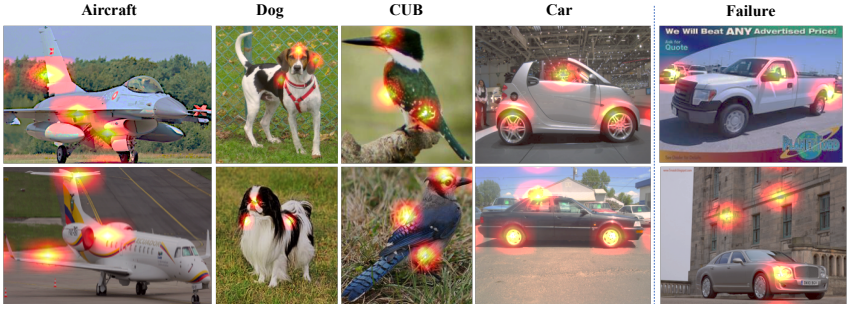

Figure 1: Exemplar part locations learned by DOP when $K = 3$. From left to right: Aircraft, CUB, Dog, Car, and failure cases. DOP can fail and locate parts on the background if it has visual signatures similar to an object.

from the training set to express objects. Figure 3 shows the activated templates for different objects. The model uses the same templates to express the same class.

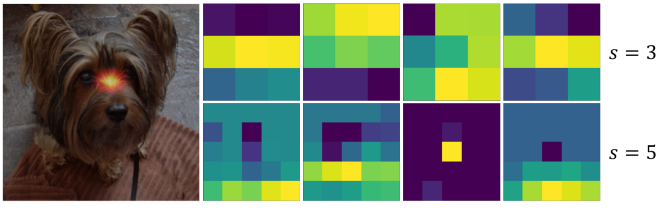

Figure 2: Exemplar templates of learned dictionary D_p. The templates shown are for randomly sampled channels for scale 3 (top) and 5 (bottom).

C5. Full Results on CUB

Many existing methods have been implemented on the CUB dataset (Table 2). We can reach comparable state-of-the-art performance.

References

[1] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[2] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8460–8469, 2019.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

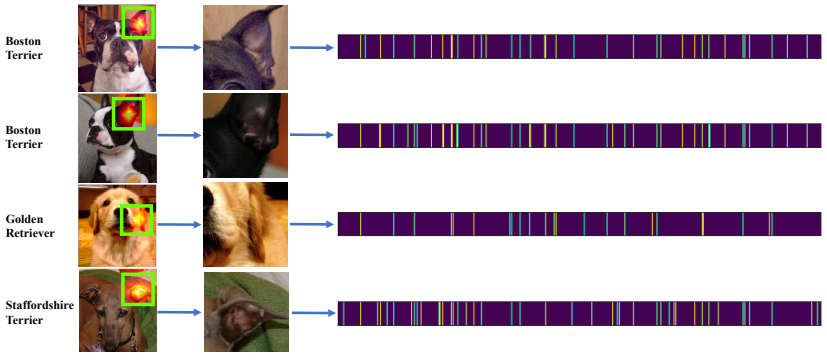

Figure 3: Template coefficients z_p of the same part for two Boston Terriers (top 2 rows), a Golden Retriever (3rd row) and a Staffordshire Terrier (4th row). Template coefficients for images of the same class are similar. Visually-similar classes (Boston Terrier and Staffordshire Terrier) share some of the same activated templates, while visually distinct classes (Golden Retriever) differ a lot on their selection of active templates.

[4] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018.

[5] Jie Hong, Pengfei Fang, Weihao Li, Tong Zhang, Christian Simon, Mehrtash Harandi, and Lars Petersson. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 913–923, 2021.

[6] Huaxi Huang, Junjie Zhang, Litao Yu, Jian Zhang, Qiang Wu, and Chang Xu. Toan: Target-oriented alignment network for fine-grained image categorization with few labeled samples. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2): 853–866, 2021.

[7] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *ICCV*, 2021.

[8] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[11] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

Table 2: Few-shot accuracy in % on CUB (along with 95% confidence intervals). If not specified, the results are those reported in the original paper. †: results are obtained by running the public implementation released by authors.

Methods	**Backbone**	*1-shot*	*5-shot*
ProtoNet[19]	ResNet18	71.88±0.91	87.42±0.48
MTL[16]	ResNet12	73.31±0.92	82.29 ±0.51
Δ-encoder [18]	ResNet18	69.80±0.46	82.60±0.35
Baseline++ [1]	ResNet18	67.02±0.90	83.58±0.54
SimpleShot[24]	ResNet18	62.85±0.20	84.01±0.14
DN4[14]†	ResNet18	70.47±0.72	84.43±0.45
MetaOptNet[11]†	ResNet12	75.15±0.46	87.09±0.30
AFHN[13]	ResNet18	70.53±1.01	83.95±0.63
BSNet[15]	ResNet18	69.61±0.92	83.24±0.60
DeepEMD[29]	ResNet12	75.65±0.83	88.69±0.50
FOT[23]	ResNet18	72.56±0.77	87.22±0.46
VFD [27]	ResNet12	79.12±0.83	91.48±0.39
FRN[25]	ResNet12	83.16 ± 0.19	92.59 ± 0.23
RENet[2]	ResNet12	79.49±0.44	91.11±0.24
TOAN[6]	ResNet12	67.17± 0.81	82.09±0.56
RAP[5]	ResNet18	83.59±0.18	90.77±0.10
TDM[12]	ResNet12	83.36 ± 0.22	92.08 ± 0.13
HelixFormer[28]	ResNet12	81.66±0.30	91.83±0.17
DOP	ResNet18	82.62±0.65	**92.61±0.38**
DOP	ResNet12	**83.39±0.82**	**93.01±0.43**

[12] SuBeen Lee, WonJun Moon, and Jae-Pil Heo. Task discrepancy maximization for fine-grained few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5331–5340, 2022.

[13] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13470–13479, 2020.

[14] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[15] Xiaoxu Li, Jijie Wu, Zhuo Sun, Zhanyu Ma, Jie Cao, and Jing-Hao Xue. Bsnet: Bi-similarity network for few-shot fine-grained image classification. *IEEE Transactions on Image Processing*, 30:1318–1331, 2020.

[16] Q Sun Y Liu, TS Chua, and B Schiele. Meta-transfer learning for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[18] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *arXiv preprint arXiv:1806.04734*, 2018.

[19] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[20] Luming Tang, Davis Wertheimer, and Bharath Hariharan. Revisiting pose-normalization for fine-grained few-shot recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14352–14361, 2020.

[21] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6372–6381, 2019.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[23] Chaofei Wang, Shiji Song, Qisen Yang, Xiang Li, and Gao Huang. Fine-grained few shot learning with foreground object transformation. *Neurocomputing*, 466:16–26, 2021.

[24] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[25] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8012–8021, 2021.

[26] Jiamin Wu, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Task-aware part mining network for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8433–8442, 2021.

[27] Jingyi Xu, Hieu Le, Mingzhen Huang, ShahRukh Athar, and Dimitris Samaras. Variational feature disentangling for fine-grained few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8812–8821, 2021.

[28] Bo Zhang, Jiakang Yuan, Baopu Li, Tao Chen, Jiayuan Fan, and Botian Shi. Learning cross-image object semantic relation in transformer for few-shot fine-grained image classification. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2135–2144, 2022.

[29] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020.