Exploring Tuning Characteristics of Ventral Stream's Neurons for Few-Shot Image Classification

Lintao Dong, Wei Zhai, Zheng-Jun Zha*

University of Science and Technology of China
{dlintao, wzhai056}@mail.ustc.edu.cn, zhazj@ustc.edu.cn

Abstract

Human has the remarkable ability of learning novel objects by browsing extremely few examples, which may be attributed to the generic and robust feature extracted in the ventral stream of our brain for representing visual objects. In this sense, the tuning characteristics of ventral stream's neurons can be useful prior knowledge to improve few-shot classification. Specifically, we computationally model two groups of neurons found in ventral stream which are respectively sensitive to shape cues and color cues. Then we propose the hierarchical feature regularization method with these neuron models to regularize the backbone of a few-shot model, thus making it produce more generic and robust features for few-shot classification. In addition, to simulate the tuning characteristic that neuron firing at a higher rate in response to foreground stimulus elements compared to background elements, which we call belongingness, we design a foreground segmentation algorithm based on the observation that the foreground object usually does not appear at the edge of the picture, then multiply the foreground mask with the backbone of few-shot model. Our method is model-agnostic and can be applied to few-shot models with different backbones, training paradigms and classifiers.

Introduction

Few-shot image classification aims to learn new visual concepts from extremely limited examples, which can't be achieved without prior knowledge. As for human beings, we can learn a novel object quickly with only a glance at it. This remarkable ability implies that there is a generic and robust encoder in our brain which can extract features to effectively represent the visual information of an object. According to the two-streams hypothesis (Schneider 1969; Mishkin and Ungerleider 1982; Goodale et al. 1991), human's brain possesses two distinct pathways for visual perception - the ventral stream and the dorsal stream. Of these two pathways, the ventral stream is thought to be responsible for transform-invariant visual object and face recognition, which can be seen as the generic and robust encoder mentioned previously. As a result, tuning characteristics of ventral stream's neurons can be useful prior knowledge to improve existed few-shot models with different backbones, training paradigms and classifiers, which are not fully explored in few-shot image classification.

Briefly speaking, tuning characteristics of ventral stream's neurons can be divided into 5 groups (Kruger et al. 2012) - tuning characteristics about shape, color, motion, depth and belongingness. We don't consider tuning characteristics about motion and depth in this paper, as the data used in image classification is static and monocular.

For tuning characteristics about shape and color, we computationally model the neurons in ventral stream tuned to shape cues and color cues, which are summarized in Figure 1. With these neuron models, we can regularize the backbone of a few-shot model to make it produce more generic and robust features. To the best of our knowledge, we are the first to utilize the computational models of neurons to benefit few-shot learning, thus building a bridge to fertilize few-shot learning from the development of computational neuroscience. To achieve this, one way is to use distillation: firstly we use neuron models' responses to images and their corresponding labels to train a classification model. Then we distill the knowledge from this classification model to a normally trained classification model fed by original image data to get the improved few-shot model. While this distillation method can successfully transfer the knowledge from ventral stream to few-shot model, it omits the hierarchy of ventral stream's neurons (V1→V2→V4). To also exploit the hierarchy information, we propose the hierarchical feature regularization method, as shown in the lower part of Figure 2. Firstly we group the neuron models according to the visual areas they lie in. Then we use the responses of neurons from V1, V2, V4 areas to respectively regularize features of different layers in the backbone of few-shot model and these layers regularized should have the same hierarchy as the neurons, by which the backbone not only captures the tuning characteristics of ventral stream's neurons, but also maintains the same hierarchical structure of the ventral stream. If this hierarchy is reversed, the effect of feature regularization will be discounted, which can be seen from the experiment section.

For tuning characteristic of belongingness, existed method (Luo et al. 2021) has explored the similar problem of detecting the foreground object without supervision. Their method is based on the prior knowledge that features of foreground objects in images should be the most common fea-

*Corresponding author.

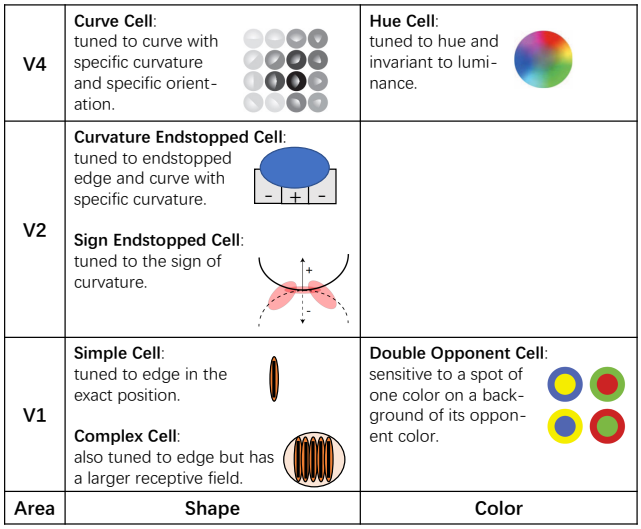

Figure 1: Ventral stream's cells utilized in this paper and their tuning characteristics.

tures within one class, thereby can be identified via a clustering algorithm. However, this method is not applicable in one-shot setting because it is hard to cluster foreground object's features with only one image in each class. Applicable for both one-shot and few-shot settings, our method is based on another prior knowledge that the foreground object usually does not appear at the edge of the picture, which means the image feature from the edge region can be used to find background region. Based on this observation, we use HSV color model and K-means clustering to get the representative image features of the edge region and use these features to pick background region, then the foreground region will be the rest of the image. The procedure is shown in the upper part of Figure 2. Our main contributions in this paper can be summarized as follows:

- We propose the novel hierarchical feature regularization method with computational models of ventral stream's neurons to improve few-shot classification models, thus building a bridge to fertilize few-shot learning from the development of computational neuroscience.
- We propose a foreground segmentation algorithm to simulate the tuning characteristic of belongingness by multiplying the segmentation mask with the feature maps of the few-shot model's backbone, which can further improve few-shot models.
- We apply our method on few-shot models with different backbone, classifier and training setting. The consistent improvements over these models demonstrate the effectiveness and generality of our method.

Related Works

Few-shot classification. Recent few-shot classification methods can be roughly grouped into three categories. Hallucination-based method learns a generator from data in the base classes and use the learned generator to generate new novel class data for data augmentation (Chen et al. 2019; Hariharan and Girshick 2017; Li et al. 2019, 2020). Metric-based method aims to learn an embedding function that maps images to a metric space such that the relevance between a pair of images can be measured based on their distance (Oreshkin, Rodríguez López, and Lacoste 2018; Snell, Swersky, and Zemel 2017; Vinyals et al. 2016; Zhang et al. 2014, 2020). Optimization-based method aims to learn how to optimize the gradient descent procedure so that the learner can have a good initialization or update direction or learning rate (Finn, Abbeel, and Levine 2017; Jamal and Qi 2019; Munkhdalai and Yu 2017; Zheng et al. 2021; Zhu et al. 2021; Rajeswaran et al. 2019). Our method is in spirit similar to the optimization-based method, in that we tend to alter the gradient descent procedure to find a better update direction by using regularization techniques.

Regularization techniques. Regularization techniques have been widely used in the deep learning community for training deep neural network to prevent it from overfitting and improve their generalization performances (Srivastava et al. 2014; DeVries and Taylor 2017). Overfitting is a typical issue for few-shot models, due to the extremely limited examples of novel class and the disjoint distribution between training data and testing data. In this sense, regularization techniques are suitable to be used on few-shot classification. In fact, there are quite a few methods utilizing regularization to improve few-shot model (Yoo et al. 2018; Mangla et al. 2020; Osahor and Nasrabadi 2022). In this paper, we present a novel regularization technique which exploits the tuning characteristics and the hierarchy of ventral stream's neurons to benefit few-shot models.

Approach

Modelling Ventral Stream's Neurons

Ventral stream is a multistage pathway with the hierarchy of V1→V2→V4→IT. The learning of novel object is thought to be achieved in the inferotemporal cortex (IT) through the experience-induced changes such that neurons become more selective for learned visual object (Hasegawa and Miyashita 2002; Sakai and Miyashita 1991; Sigala and Logothetis 2002). Compared to the function of IT, the function of areas before IT (V1, V2, V4) is more like feature extractor in the sense that they have persistent selectivity for specific visual cues such as local orientation, curvature and color contrast (Pasupathy 2006; Kruger et al. 2012). To benefit few-shot models from these generic and robust features used by our brain's visual system, we manage to computationally model neurons in V1, V2, V4 areas firstly.

V1 area. The most typical cells found in V1 are simple cell and complex cell which are found by Hubel and Wiesel (Hubel and Wiesel 1959). They are both tuned to oriented edge or bar, while the complex cell has a degree of spatial invariance. Fisrtly we use Difference of Gaussians

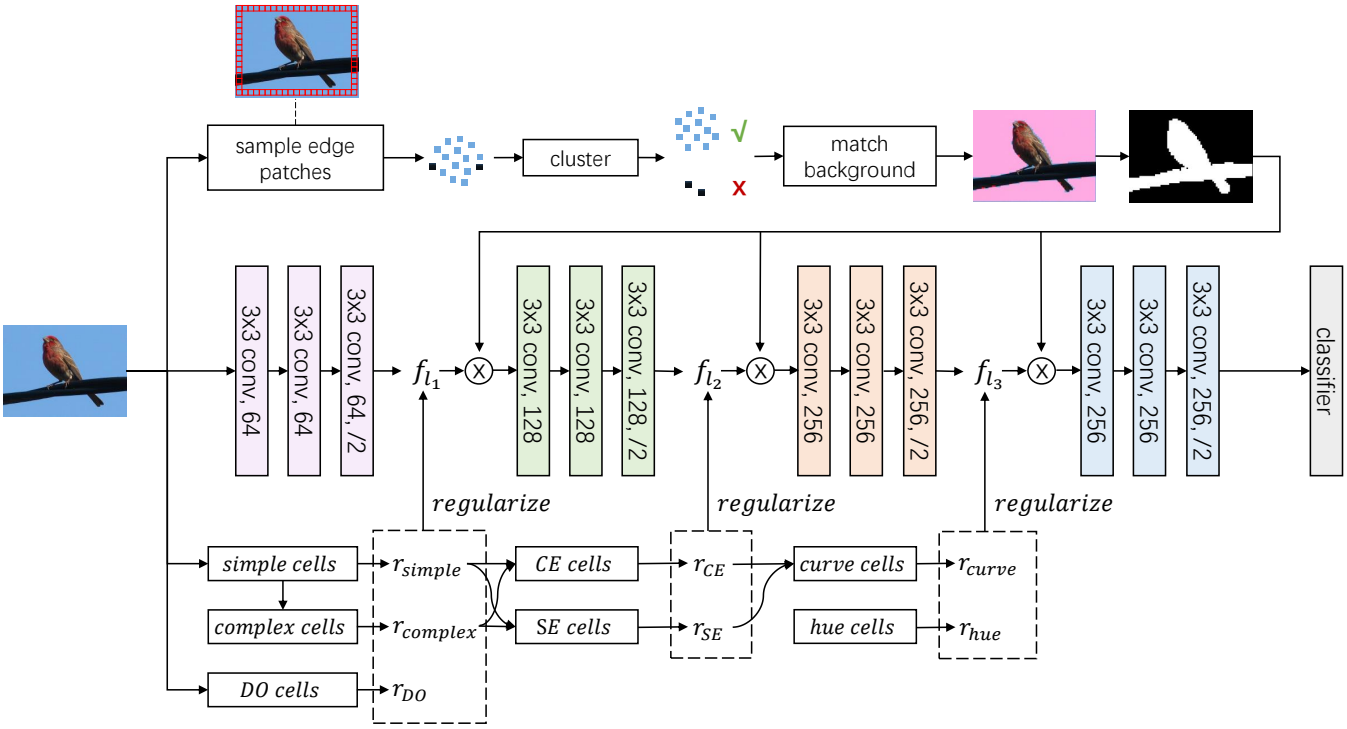

Figure 2: Illustration of our proposed method, which comprises two parts: one (bottom) is the hierarchical feature regularization method using computational models of ventral stream's neurons tuned to shape cues and color cues to regularize the backbone of few-shot model; the other (top) is a foreground segmentation algorithm to simulate the tuning characteristic of belongingness. Here we show the methods in fine-tuning setting with backbone of ResNet-12. To avoid clutter, we simplify the training procedure by only showing the pre-training stage and omitting the fine-tuning stage and classification loss.

to model the odd simple cell:

$$G_{simple}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}((\frac{x_1'}{\sigma_x})^2+(\frac{y_1'}{\sigma_y})^2)} - \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}((\frac{x_2'}{\sigma_x})^2+(\frac{y_2'}{\sigma_y})^2)}, \tag{1}$$

$$\begin{aligned}(x_1', y_1') &= Rotate(x-x_1, y-y_1, \theta_s),\\ (x_2', y_2') &= Rotate(x-x_2, y-y_2, \theta_s),\end{aligned} \tag{2}$$

$$\begin{aligned}(x_1, y_1) &= Rotate(-\frac{S}{2}, 0, \theta_s),\\ (x_2, y_2) &= Rotate(\frac{S}{2}, 0, \theta_s).\end{aligned} \tag{3}$$

$Rotate(x, y, \theta)$ is an operation which rotates the point (x, y) around origin by θ degree clockwise and obtain a new point (x', y'):

$$\begin{aligned}x' &= xcos(\theta) + ysin(\theta),\\ y' &= -xsin(\theta) + ycos(\theta).\end{aligned} \tag{4}$$

σ_x and σ_y control the width and height of gaussian function. θ_s is the orientation of simple cell. S is the distance between centers of two gaussian functions. We set $S = 4\sigma_x$. For RGB image, we should integrate the information from three color channels:

$$r_{simple} = \sqrt{{r_{simple_R}}^2 + {r_{simple_G}}^2 + {r_{simple_B}}^2}, \tag{5}$$

$$\{r_{simple_{E_i}} = G_{simple} \otimes c_{E_i}\}_{E=\{R,G,B\}}, \tag{6}$$

where $r_{simple_R}, r_{simple_G}, r_{simple_B}$ are the responses of odd simple cells to R, G, B color channels c_R, c_G, c_B. r_{simple} is the response of simple cells to a RGB image. $\otimes$ represents convolution operation.

The complex cell receives signals from simple cells and integrates them. The typical model of complex cell is the sum of several laterally displaced simple cells (Spitzer and Hochstein 1985). Following this idea, we model the complex cell as the summation over responses of simple cells with gaussian weights:

$$r_{complex} = \frac{G_{complex} \otimes r_{simple}}{\rho}, \tag{7}$$

$$G_{complex}(x,y) = e^{-\frac{1}{2}((\frac{x'}{\sigma_x'})^2+(\frac{y'}{\sigma_y'})^2)}, \tag{8}$$

$$(x', y') = Rotate(x, y, \theta_c + \frac{\pi}{2}), \tag{9}$$

where σ_x' is a small value which we set as 0.5. As a result, complex cell is actually the sum of several laterally displaced simple cells. θ_c is the orientation of complex cell which is also the orientation of simple cells and perpendicular to the orientation of elongated gaussian function. ρ is a normalization factor which we set as $\rho = \frac{Max(G_{complex} \otimes r_{simple})}{Max(r_{simple})}$. Max is the operation of taking maximum value.

Besides shape, another important cue for visual perception is color. One type of color-coding cell named double-opponent cell is found exist widely in V1 (Gegenfurtner 2003; Livingstone and Hubel 1984), which encodes color contrast on two color axises of blue-yellow and red-green. We follow Gao et al. (2013) to model double-opponent cells. Signals are first transformed from the RGB space to the single-opponent space according to Ebner (2007):

$$\begin{aligned} o_{RG} =& \frac{c_R - c_G}{\sqrt{2}}, o_{GR} = -o_{RG}, \\ o_{YB} =& \frac{c_Y - 2c_B}{\sqrt{6}}, o_{BY} = -o_{YB}, \end{aligned} \tag{10}$$

where c_Y represents the yellow channel of input color image, given by $c_Y = (c_R + c_G)/2$, which is constructed for the computation of blue-yellow (B-Y) opponency. The responses of simple-opponent cells (SO) can be computed as:

$$r_{SO_{R+G-}}(\sigma) = G_{SO}(\sigma) \otimes o_{RG}, \tag{11}$$

$$G_{SO}(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{12}$$

where σ controls the scale of simple-opponent cell's receptive field we set as 1. Other SO cells $r_{SO_{G+R-}}$, $r_{SO_{B+Y-}}$, $r_{SO_{Y+B-}}$ can be obtained in the similar way. Note that in the expression of $A + B-$, the sign $+$ and $-$ denote the excitation and inhibition, respectively. Then the responses of double-opponent cells (DO) can be computed as:

$$r_{DO_{R+G-}} = r_{SO_{R+G-}}(\sigma) + k * r_{SO_{G+R-}}(\lambda\sigma), \tag{13}$$

where σ and $\lambda\sigma$ respectively define the scales of the receptive field center and its surround of a double-opponent cell. k is a weight to control the contribution of receptive field surround. We set $\lambda = 3$ and $k = 0.8$. Other DO cells $r_{DO_{G+R-}}$, $r_{DO_{B+Y-}}$, $r_{DO_{Y+B-}}$ can be obtained in the similar way.

V2 area. V2 contains endstopped cells (hypercomplex cells) which are characterized by inhibitory areas (end-zones). End-zones endow endstopped cells with the capacity to identify edges with limited length and curves (for cells stopped at both ends). Apart from the degree of curvature, another contour characteristic that V2 cells seem to encode is the sign of curvature (Dobbins, Zucker, and Cynader 1987; Hegdé and Van Essen 2000). To simulate these tuning characteristics, we use the computational models of curvature endstopped cell and sign endstopped cell from Rodríguez-Sánchez and Tsotsos (2012). Curvature endstopped cell (CE) is modelled as an excitatory center simple cell with two inhibitory displaced complex cells:

$$\begin{aligned} r_{CE}(x, y) =& \Phi(\alpha_c * r_{simple}(x, y) - \\ & \alpha_{d1} * r_{complex}(x_{d1}, y_{d1}) - \\ & \alpha_{d2} * r_{complex}(x_{d2}, y_{d2})), \end{aligned} \tag{14}$$

$$\Phi(r) = \frac{1 - e^{-r/\epsilon}}{1 + 1/\Gamma e^{-r/\epsilon}}, \tag{15}$$

where $\alpha_c, \alpha_{d1}, \alpha_{d2}$ are the gains for the center and displaced cells. $r_{simple}(x, y)$, $r_{complex}(x_{d1}, y_{d1})$, $r_{complex}(x_{d2}, y_{d2})$ are the responses of the center simple cell and two displaced complex cells. These cells have the same orientation and two complex cells are placed on either side of the center simple cell with the same distance $d = 2\sigma_y$. Φ is the rectification function and ϵ is the maximum response of the set of neurons for a given scale divided by 8.5, a factor that provided a good normalization approximation for this rectification.

sign endstopped cell (SE) uses the same structure as curvature endstopped cell except that two displaced complex cell are rotated by 45 degrees with opposite direction (Rodríguez-Sánchez and Tsotsos 2012):

$$\begin{aligned} r_{SE+}(x, y) =& \phi(r_{simple}(x, y) - \\ & r_{complex_{45}}(x_{d1_{45}}, y_{d1_{45}}) - \\ & r_{complex_{135}}(x_{d2_{135}}, y_{d2_{135}})), \end{aligned} \tag{16}$$

$$\begin{aligned} r_{SE-}(x, y) =& \phi(r_{simple}(x, y) - \\ & r_{complex_{135}}(x_{d1_{135}}, y_{d1_{135}}) - \\ & r_{complex_{45}}(x_{d2_{45}}, y_{d2_{45}})). \end{aligned} \tag{17}$$

V4 area. By combining responses of curvature endstopped cells and sign endstopped cells, we obtain the curve cell tuned to curve with specific orientation and curvature, which is able to represent various contour fragments. We adopt the same method as Rodríguez-Sánchez and Tsotsos (2012) to model curve cell:

$$\begin{aligned} r_{curve_i} &= r_{CE_i} \cap (r_{SE+} > r_{SE-}), \\ r_{curve_{i+n}} &= r_{CE_i} \cap (r_{SE-} > r_{SE+}), \end{aligned} \tag{18}$$

where n is the number of curvature endstopped cell types (with different orientation). The number of curve cell types is $2n$ (with different orientation and sign).

Color coding cells in V4 differ from those in V1, V2 in that they code for hue, instead of color opponency along the two principal color axes, and that the tuning to hue is invariant to luminance (Conway, Moeller, and Tsao 2007). To encode hue representation, we firstly transform RGB image into HSV image with channels of hue c_H, saturation c_S and value (brightness) c_V. Then the hue of a pixel can be represented by its distance to the pixels of seven 'standard colors' (from red to purple). To achieve luminance invariance, only hue and saturation are considered:

$$r_{color_i} = e^{-\frac{1}{2}\left(\frac{(c_H - \eta_i)^2}{\sigma_h^2} + \frac{(c_S - 1)^2}{\sigma_s^2}\right)}, \tag{19}$$

where r_{color_i} represents one hue from {red, orange, yellow, green, cyan, blue, purple}. η_i is the standard hue of one color. In addition, we add white and black as extra hues. Implementation Details are described in appendix. All hue representations are combined to obtain the response of hue cells r_{hue} with 9 channels.

Hierarchical Feature Regularization

Since we have the computational models of neurons, we can use them to regularize the backbone of few-shot model so that the backbone can produce more generic and robust features for few-shot classification. In addition, neurons from different visual areas have the hierarchical structure of V1→V2→V4. To capture both the tuning characteristics and the hierarchy of ventral stream's neurons, we propose

the hierarchical feature regularization (HFR) method. Hierarchy makes sense in that neurons in different visual areas have different complexity. Tuning characteristics of lower areas should be learned in shallow layer to prevent overfitting. Tuning characteristics of higher areas should be learned in deep layer in case the network has enough capacity.

For each image in training set, we can get the responses of V1 neurons: $\{r_{simple_i}\}_{i=1}^{i=n_s}$, $\{r_{complex_i}\}_{i=1}^{i=n_c}$, $r_{DO_{R+G-}}$, $r_{DO_{G+R-}}$, $r_{DO_{B+Y-}}$, $r_{DO_{Y+B-}}$. In our experiments, we use 4 orientations and 3 different sizes, thus $n_s = 12$ and $n_c = 12$. Combining all responses of V1 neurons, we obtain the V1 feature $f_{V1} \in \mathbb{R}^{W \times H \times n_{V1}}$, where W and H are the width and height of input image, n_{V1} is the number of channels and $n_{V1} = 28$. In the same way, we can obtain $f_{V2} \in \mathbb{R}^{W \times H \times n_{V2}}$ and $f_{V4} \in \mathbb{R}^{W \times H \times n_{V4}}$, where $n_{V2} = 36$ and $n_{V4} = 33$. Because all computational models can be transformed into convolution layers with fixed kernel, the computations of these models can be accelerated by GPU so that the use of these neuron models will not cost much.

Then we choose three layers l_1, l_2, l_3 in the backbone of few-shot model to do feature regularization, where l_1 is lower than l_2 and l_2 is lower than l_3. For convolutional networks (CNN), the model can be divided into several modules. Each module consist of several repetitive building blocks with a max pooling layer, especially for ResNets, which are widely used by few-shot classification task (Oreshkin, Rodríguez López, and Lacoste 2018; Rusu et al. 2018). In our experiments, we choose outputs of these modules to regularize. Take the example of ResNet-12, it has four modules that each consist of three conv layers with 3x3 kernel and a 2x2 max pooling layer applied at the end of each module. We choose the first three modules' outputs to regularize, as shown in Figure 2. We use f_{V1}, f_{V2}, f_{V4} to regularize l_1, l_2, and l_3 respectively, with Mean Squared Error (MSE) loss:

$$\begin{aligned} \mathcal{L}_{V1} &= \|f_{l_1}^p - Pool_1(f_{V1})\|^2, \\ \mathcal{L}_{V2} &= \|f_{l_2}^p - Pool_2(f_{V2})\|^2, \\ \mathcal{L}_{V4} &= \|f_{l_3}^p - Pool_3(f_{V4})\|^2, \end{aligned} \tag{20}$$

where $f_{l_n}^p$ is part of the output of l_n. Because the number of channels is not matched between f_{l_n} (the full output of l_n) and its corresponding ventral feature, we only regularize part of f_{l_n}, which is represented as $f_{l_n}^p$. Regularizing only part of the feature in the backbone is reasonable in the sense that neurons modelled by us is only part of the neurons in ventral stream which can't represent all aspects of visual objects. $Pool$ is average pooling function used to down-sample ventral feature so that their size can match the corresponding backbone feature. We combine these losses into one:

$$\mathcal{L}_{ventral} = \mathcal{L}_{V1} + w_1\mathcal{L}_{V2} + w_2\mathcal{L}_{V4}, \tag{21}$$

where w_1, w_2 are hyper-parameters to adjust the weights of each loss. We simply set both w_1, w_2 as 1. To use hierarchical feature regularization (HFR), you just need to simply combine the cross-entropy classification loss with $\mathcal{L}_{ventral}$.

Foreground Segmentation for Belongingness

Experiments have shown that some neurons in ventral stream fire at a higher rate in response to foreground stimulus elements compared to background elements (Lamme 1995). We call this tuning characteristic belongingness. To simulate belongingness, we propose a foreground segmentation algorithm without any supervision. According to Luo et al. (2021), although background knowledge has positive impact on the performance of in-class classification tasks, it serves as a source of shortcut knowledge which harms the evaluation performance in few-shot classification. As a result, belongingness serves to rectify the shortcut learning of background for few-shot learning.

For the task of image classification, the object usually does not appear at the edge of the picture. If we can obtain the representative image features of the edge region, then we can use these features to pick background region and the rest of the image will be foreground region. To achieve this, we firstly divide the input image into little square patches with length of $L_{patch} = \lfloor \frac{Max(W,H)}{K} \rfloor$, where K is set as 32. We only use the image patches at the edge of the image. Therefore, we obtain $n_{patch} = 2 * (\lfloor \frac{W}{L_{patch}} \rfloor + \lfloor \frac{H}{L_{patch}} \rfloor)$ patches. For each patch, we transform the color space from RGB into HSV, with channels of hue, saturation and value (brightness). Then we use the channels of hue and saturation to represent this image patch, thus obtaining the patch feature invariant to luminance:

$$H_{patch} = Pool(c_H), S_{patch} = Pool(c_S), \tag{22}$$

where $c_H \in \mathbb{R}^{L_{patch} \times L_{patch}}$ and $c_S \in \mathbb{R}^{L_{patch} \times L_{patch}}$ are the channels of hue and saturation of this image patch. $Pool$ is the average pooling function. Next, we use K-means algorithm to cluster patches at the edge of the image based on (H_{patch}, S_{patch}). We choose the clusters with more than τ_g patches to represent background. The centers of these clusters $\{z_i\}_{i=1}^{i=n_g}$ can be seen as the features of background. We set $\tau_g = \lfloor \frac{n_{patch}}{6} \rfloor$. Finally, foreground region can be extracted utilizing $\{z_i\}_{i=1}^{i=n_g}$. We use a sliding window with size of $L_{patch} \times L_{patch}$ and stride of $\lfloor \frac{L_{patch}}{2} \rfloor$. For each window, we extract its hue-saturation feature and compute its euclidean distance to $\{z_i\}_{i=1}^{i=n_g}$. If one of these distances is less than the threshold τ_d, then this window is classified as background, otherwise it is classified as foreground. The overall foreground region is the union of all foreground windows. By multiplying the segmentation mask with the feature maps of few-shot model's backbone, we can simulate the tuning characteristic of belongingness and rectify the shortcut learning of background for few-shot learning. Notice that we use the segmentation mask both in training and evaluation, which can maximize the effectiveness according to Luo et al. (2021). In fact, the edge region of image can be represented by more complicated neural representation, which we leave for future works.

Experiments

Implementation Details

We implement our method over advanced few-shot models of RENet (Kang et al. 2021) and Distribution Calibration (DC) (Yang, Liu, and Xu 2021), which use different paradigms of meta-learning and fine-tuning. Besides,

method	CUB		miniImageNet		CIFAR-FS		tieredImageNet	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
RENet	79.49	91.11	67.60	82.58	74.51	86.60	71.61	85.28
RENet-ventral	83.33 (+3.84)	92.97 (+1.86)	69.71 (+2.11)	84.23 (+1.65)	75.82 (+1.31)	87.45 (+0.85)	73.94 (+2.33)	87.15 (+1.87)
KNN-DC	79.88	88.73	67.07	79.17	73.23	83.91	76.18	87.41
KNN-DC-ventral	82.41 (+2.53)	90.75 (+1.92)	69.10 (+2.03)	81.00 (+1.83)	74.63 (+1.40)	85.01 (+1.10)	78.35 (+2.17)	89.08 (+1.67)
SVM-DC	79.49	90.26	67.31	82.30	74.55	86.05	77.93	89.72
SVM-DC-ventral	82.12 (+2.63)	91.93 (+1.67)	69.12 (+1.81)	83.57 (+1.27)	75.69 (+1.14)	86.97 (+0.92)	79.27 (+1.34)	90.91 (+1.19)
LR-DC	79.56	90.67	68.57	82.88	74.71	86.35	78.19	89.90
LR-DC-ventral	82.26 (+2.70)	92.60 (+1.93)	69.98 (+1.41)	84.22 (+1.34)	75.78 (+1.07)	87.09 (+0.74)	79.48 (+1.29)	91.03 (+1.13)

Table 1: 5-way 1-shot and 5-shot classification accuracy (%) on four standard benchmarks before and after applying our method.

method	backbone	CUB		miniImageNet		tieredImageNet	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchNet (Vinyals et al. 2016)	ResNet-12	71.87 ± 0.85	85.08 ± 0.57	63.08 ± 0.80	75.99 ± 0.60	68.50 ± 0.92	80.60 ± 0.71
ProtoNet (Snell, Swersky, and Zemel 2017)	ResNet-12	66.09 ± 0.92	82.50 ± 0.58	62.39 ± 0.21	80.53 ± 0.14	68.23 ± 0.23	84.03 ± 0.16
DeepEMD (Zhang et al. 2020)	ResNet-12	75.65 ± 0.83	88.69 ± 0.50	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
RENet (Kang et al. 2021)	ResNet-12	79.49 ± 0.44	91.11 ± 0.24	67.60 ± 0.44	82.58 ± 0.30	71.61 + 0.51	85.28 ± 0.35
COSOC (Luo et al. 2021)	ResNet-12	-	-	69.28 ± 0.49	85.16 ± 0.42	73.57 ± 0.43	87.57 ± 0.10
CSEI (Li, Wang, and Hu 2021)	ResNet-12	-	-	68.94 ± 0.28	85.07 ± 0.50	73.76 ± 0.32	87.83 ± 0.59
SetFeat (Afrasiyabi et al. 2022)	ResNet-12	79.60 ± 0.80	90.48 ± 0.44	68.32 ± 0.62	82.71 ± 0.46	73.63 ± 0.88	87.59 ± 0.57
DeepBDC (Xie et al. 2022)	ResNet-12/18	**84.01** ± 0.42†	**94.02** ± 0.24†	67.83 ± 0.43	**85.45** ± 0.29	73.82 ± 0.47	89.00 ± 0.30
S2M2 (Mangla et al. 2020)	WRN-28-10	80.68 ± 0.81	90.85 ± 0.44	64.93 ± 0.18	83.18 ± 0.11	73.71 ± 0.22	88.59 ± 0.14
MetaQDA (Zhang et al. 2021)	WRN-28-10	-	-	67.83 ± 0.64	84.28 ± 0.69	74.33 ± 0.65	89.56 ± 0.79
LR-DC (Yang, Liu, and Xu 2021)	WRN-28-10	79.56 ± 0.87	90.67 ± 0.35	68.57 ± 0.55	82.88 ± 0.42	78.19 ± 0.25	89.90 ± 0.41
RENet-ventral	ResNet-12	**83.33** ± 0.40	**92.97** ± 0.24	69.71 ± 0.45	84.23 ± 0.29	73.94 ± 0.48	87.15 ± 0.35
LR-DC-ventral	WRN-28-10	82.26 ± 0.89	92.60 ± 0.34	**69.98** ± 0.56	84.22 ± 0.42	**79.48** ± 0.26	**91.03** ± 0.41

Table 2: Performance comparison on CUB, miniImage, and tieredImageNet, with 95% confidence intervals. † denotes the method uses ResNet-18 and higher resolution of 224×224 instead of 84×84.

shape	color	belongingness	CUB†	miniImageNet
×	×	×	79.56	67.60
✓	×	×	80.83 (+1.27)	68.73 (+1.13)
✓	✓	×	81.17 (+1.61)	68.98 (+1.38)
✓	✓	✓	82.26 (+2.70)	69.71 (+2.11)

Table 3: Effects of different tuning characteristics. †Here we use 'LR-DC-ventral' method on CUB dataset for clearer comparison, considering that RENet use cropped images on CUB which can't show the effect of belongingness.

they use different backbones of ResNet-12 and WRN-28-10, which are the most used in few-shot classification. We also implement different classifiers on DC model. We use three groups of parameters $\{\sigma_x^i\} = \{0.5, 1, 2\}$, $\{\sigma_y^i\} = \{1, 2, 4\}$, $\{\sigma_y^{'i}\} = \{1, 2, 4\}$ to encode shape cues with neurons of different scales. Orientations are set as $\{0°, 45°, 90°, 135°\}$. To use foreground masks, we adopt the same fusion sampling strategy as (Luo et al. 2021). We use the same data processing method as RENet and DC. Introduction to datasets and more implementation details are given in appendix.

Results and Analysis

Effectiveness of our method. We report the accuracy and the improvements after applying our method on 5-way 1-shot and 5-way 5-shot settings for CUB, miniImageNet, CIFAR-FS, and tieredImageNet, as shown in Table 1. 'KNN-DC', 'SVM-DC' and 'LR-DC' represent Distribution Calibration methods with classifier of KNN, SVM and Linear Regression respectively. Results of 'KNN-DC' are produced by us. The suffix '-ventral' means the methods benefit from knowledge of ventral stream, by using our hierarchical feature regularization (HFR) and foreground segmentation algorithm. As shown in Table 1, our method consistently improves all models with different backbones, training paradigms and classifiers, which demonstrate the effectiveness and generality of our method. Our method works best on CUB dataset, with a remarkable improvement on RENet (79.49%→83.33%). However, our method has relatively limited improvements on CIFAR-FS, due to the low resolution and already cropped images.

Comparison to State-Of-The-Art Methods. We choose 'RENet-ventral' and 'LR-DC-ventral' to compare with current few-shot classification methods on CUB, miniImageNet and tieredImageNet. As shown in Table 2, our methods achieve state-of-the-art performances on all datasets except the 5-way 5-shot setting of miniImageNet.

Ablation Studies. We use 'RENet-ventral' method to conduct extensive ablation studies on CUB and miniImageNet, either in the absence of each module or by replacing them with others and compare the results in 5-way 1-shot setting.

Firstly, we evaluate the effectiveness of each tuning characteristic of ventral stream's neurons we utilize. As shown

V1	V2	V4	reverse	CUB	miniImageNet
×	×	×	×	79.49	67.60
✓	×	×	×	81.76 (+2.27)	68.21 (+0.61)
✓	✓	×	×	82.57 (+3.08)	68.67 (+1.07)
✓	✓	✓	×	83.14 (+3.65)	68.98 (+1.38)
✓	✓	✓	✓	82.17 (+2.68)	68.34 (+0.74)

Table 4: Effects of different visual areas' neurons.

method	CUB	miniImageNet
original	79.49	67.60
fusion-input	77.65 (-1.84)	66.13 (-1.47)
fusion-fc	79.11 (-0.38)	67.26 (-0.34)
distillation	81.01 (+1.52)	68.14 (+0.54)
HFR (ours)	83.14 (+3.65)	68.98 (+1.38)

Table 5: Performance comparison of methods utilizing neuron computational models for few-shot classification.

in Table 3, the biggest performance gain is brought by the usage of tuning characteristics to shape cues.

Next, we evaluate the effectiveness of each visual area's neurons tuned to shape and color cues in Table 4. When the hierarchical structure of ventral stream's neurons utilized to regularize is reversed, which means we use V4, V2, V1 neurons to regularize l_1, l_2, l_3 respectively, the performance is degraded, which demonstrates the importance of hierarchy.

Finally, we evaluate various methods utilizing computational models of ventral stream's neurons for few-shot classification. In Table 5, 'fusion-input' represents the method fusing the input image with responses of all neurons and use it as new input to few-shot model for training and evaluation. 'fusion-fc' represents the method fusing the feature before fc layer with responses of all neurons. We use concatenation as the fusion method. 'distillation' represents the distillation method mentioned in introduction section. As we can see, simply combing responses of all neurons with few-shot model's input or feature doesn't make sense. On the contrary, it may harm the model's performance. Distillation can improve few-shot models by transferring knowledge from ventral stream's neurons. However, to use distillation, we need to firstly train a classification model using responses of neurons. Our HFR method can be directly used without extra training stage. In addition, our HFR method is more effective than distillation with a greater performance gain.

Visualization

Regularized features vs. non-regularized features. Figure 3 shows l_1 features regularized by simple cells (left) and l_2 features regularized by curvature endstopped cells (right). As we can see, regularized features have the property of representational adequacy, thanks to the various orientations (shown in different column) and scales (shown in different row) designed systematically. However, features without regularization are disorganized and biased to the training set, which are less robust to the disjoint test set in few-shot classification compared to regularized features.

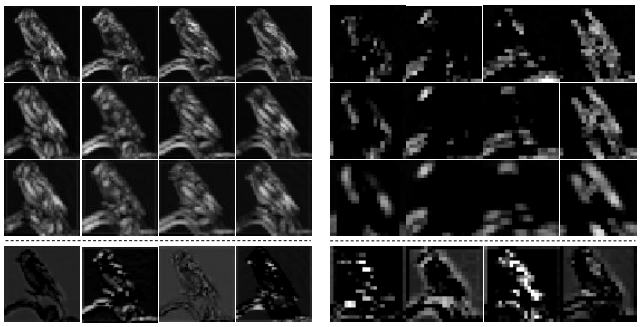

Figure 3: Regularized features of l_1 (left) and l_2 (right) with non-regularized features shown in the bottom line.

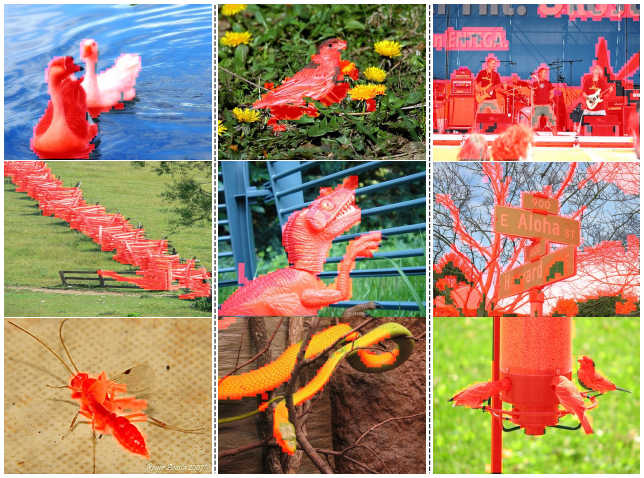

Figure 4: Example results of foreground segmentation.

Results of foreground segmentation. Left column of Figure 4 shows segmentation results of images with simple homochromatic background. Images in middle column have more complicated background. Foreground objects can't be extracted seperately in the images of right column, however part of their background can be removed.

Conclusion

In this paper, we study the problem of how to utilize tuning characteristics of ventral stream's neurons as prior knowledge to improve few-shot classification models. To utilize the tuning characteristics about shape and color, we propose the novel hierarchical feature regularization method with computational neuron models to regularize the backbone of few-shot model. To utilize the tuning characteristic of belongingness, we propose a foreground segmentation algorithm and multiply the segmentation mask with few-shot model's backbone, based on the prior knowledge that the foreground object usually does not appear at the edge of the picture. The consistent improvements over models with different backbone, training paradigm and classifier demonstrate the effectiveness and generality of our method.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, National Natural Science Foundation of China (NSFC) under Grant 62225207 and U19B2038, and the University Synergy Innovation Program of Anhui Province under Grants GXXT-2019-025.

References

Afrasiyabi, A.; Larochelle, H.; Lalonde, J.-F.; and Gagné, C. 2022. Matching Feature Sets for Few-Shot Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9014–9024.

Chen, Z.; Fu, Y.; Wang, Y.-X.; Ma, L.; Liu, W.; and Hebert, M. 2019. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8680–8689.

Conway, B. R.; Moeller, S.; and Tsao, D. Y. 2007. Specialized color modules in macaque extrastriate cortex. *Neuron*, 56(3): 560–573.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dobbins, A.; Zucker, S. W.; and Cynader, M. S. 1987. End-stopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329(6138): 438–441.

Ebner, M. 2007. *Color Constancy*. John Wiley & Sons.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Gao, S.; Yang, K.; Li, C.; and Li, Y. 2013. A color constancy model with double-opponency mechanisms. In *Proceedings of the IEEE international conference on computer vision*, 929–936.

Gegenfurtner, K. R. 2003. Cortical mechanisms of colour vision. *Nature Reviews Neuroscience*, 4(7): 563–572.

Goodale, M. A.; Milner, A. D.; Jakobson, L.; and Carey, D. 1991. A neurological dissociation between perceiving objects and grasping them. *Nature*, 349(6305): 154–156.

Hariharan, B.; and Girshick, R. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE international conference on computer vision*, 3018–3027.

Hasegawa, I.; and Miyashita, Y. 2002. Categorizing the world: expert neurons look into key features. *Nature Neuroscience*, 5(2): 90–91.

Hegdé, J.; and Van Essen, D. C. 2000. Selectivity for complex shapes in primate visual area V2. *Journal of Neuroscience*, 20(5): RC61–RC61.

Hubel, D. H.; and Wiesel, T. N. 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3): 574.

Jamal, M. A.; and Qi, G.-J. 2019. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11719–11727.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8822–8833.

Kruger, N.; Janssen, P.; Kalkan, S.; Lappe, M.; Leonardis, A.; Piater, J.; Rodriguez-Sanchez, A. J.; and Wiskott, L. 2012. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1847–1871.

Lamme, V. A. 1995. The neurophysiology of figure-ground segregation in primary visual cortex. *Journal of neuroscience*, 15(2): 1605–1615.

Li, H.; Eigen, D.; Dodge, S.; Zeiler, M.; and Wang, X. 2019. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1–10.

Li, J.; Wang, Z.; and Hu, X. 2021. Learning intact features by erasing-inpainting for few-shot classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8401–8409.

Li, K.; Zhang, Y.; Li, K.; and Fu, Y. 2020. Adversarial feature hallucination networks for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13470–13479.

Livingstone, M. S.; and Hubel, D. H. 1984. Anatomy and physiology of a color system in the primate visual cortex. *Journal of Neuroscience*, 4(1): 309–356.

Luo, X.; Wei, L.; Wen, L.; Yang, J.; Xie, L.; Xu, Z.; and Tian, Q. 2021. Rectifying the shortcut learning of background for few-shot learning. *Advances in Neural Information Processing Systems*, 34: 13073–13085.

Mangla, P.; Kumari, N.; Sinha, A.; Singh, M.; Krishnamurthy, B.; and Balasubramanian, V. N. 2020. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2218–2227.

Mishkin, M.; and Ungerleider, L. G. 1982. Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural brain research*, 6(1): 57–77.

Munkhdalai, T.; and Yu, H. 2017. Meta networks. In *International Conference on Machine Learning*, 2554–2563. PMLR.

Oreshkin, B.; Rodríguez López, P.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31.

Osahor, U.; and Nasrabadi, N. M. 2022. Ortho-Shot: Low Displacement Rank Regularization with Data Augmentation for Few-Shot Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2200–2209.

Pasupathy, A. 2006. Neural basis of shape representation in the primate brain. *Progress in brain research*, 154: 293–313.

Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32.

Rodríguez-Sánchez, A. J.; and Tsotsos, J. K. 2012. The roles of endstopped and curvature tuned computations in a hierarchical representation of 2D shape. *PloS one*, 7(8): e42058.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*.

Sakai, K.; and Miyashita, Y. 1991. Neural organization for the long-term memory of paired associates. *Nature*, 354(6349): 152–155.

Schneider, G. E. 1969. Two visual systems: Brain mechanisms for localization and discrimination are dissociated by tectal and cortical lesions. *Science*, 163(3870): 895–902.

Sigala, N.; and Logothetis, N. K. 2002. Visual categorization shapes feature selectivity in the primate temporal cortex. *Nature*, 415(6869): 318–320.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Spitzer, H.; and Hochstein, S. 1985. A complex-cell receptive-field model. *Journal of Neurophysiology*, 53(5): 1266–1286.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Xie, J.; Long, F.; Lv, J.; Wang, Q.; and Li, P. 2022. Joint Distribution Matters: Deep Brownian Distance Covariance for Few-Shot Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7972–7981.

Yang, S.; Liu, L.; and Xu, M. 2021. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*.

Yoo, D.; Fan, H.; Boddeti, V.; and Kitani, K. 2018. Efficient k-shot learning with regularized deep networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2020. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12203–12213.

Zhang, H.; Zha, Z.-J.; Yang, Y.; Yan, S.; and Chua, T.-S. 2014. Robust (semi) nonnegative graph embedding. *IEEE transactions on image processing*, 23(7): 2996–3012.

Zhang, X.; Meng, D.; Gouk, H.; and Hospedales, T. M. 2021. Shallow bayesian meta learning for real-world few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 651–660.

Zheng, K.; Lan, C.; Zeng, W.; Zhang, Z.; and Zha, Z.-J. 2021. Exploiting sample uncertainty for domain adaptive person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3538–3546.

Zhu, K.; Cao, Y.; Zhai, W.; Cheng, J.; and Zha, Z.-J. 2021. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6801–6810.