F-SCP: An Automatic Prompt Generation Method for Specific Classes based on Visual Language Pre-training Models

Baihong Han[a], Xiaoyan Jiang[a,*], Zhijun Fang[a], Hamido Fujita[b,c,d], Yongbin Gao[a]

[a]*Shanghai University of Engineering Science, Shanghai, China*
[b]*Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur, 54100, Malaysia*
[c]*Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain*
[d]*Regional Research Center, Iwate Prefectural University, Takizawa, 020-0693, Japan*

Abstract

The zero-shot classification performance of large-scale vision-language pre-training models (e.g., CLIP, BLIP and ALIGN) can be enhanced by incorporating a prompt (e.g., "a photo of a [CLASS]") before the class words. Modifying the prompt slightly can have significant effect on the classification outcomes of these models. Thus, it is crucial to include an appropriate prompt tailored to the classes. However, manual prompt design is labor-intensive and necessitates domain-specific expertise. The CoOp (Context Optimization) converts hand-crafted prompt templates into learnable word vectors to automatically generate prompts, resulting in substantial improvements for CLIP. However, CoOp exhibited significant variation in classification performance across different classes. Although CoOp-CSC (Class-Specific Context) has a separate prompt for each class, only shows some advantages on fine-grained datasets. In this paper, we propose a novel automatic prompt generation method called F-SCP (Filter-based Specific Class

*Corresponding author

Email addresses: m320121323@sues.edu.cn (Baihong Han), xiaoyan.jiang@sues.edu.cn (Xiaoyan Jiang), zjfang@sues.edu.cn (Zhijun Fang), fujitahamido@utm.my, HFujita-799@acm.org (Hamido Fujita), gaoyongbin@sues.edu.cn (Yongbin Gao)

Prompt), which distinguishes itself from the CoOp-UC (Unified Context) model and the CoOp-CSC model. Our approach focuses on prompt generation for low-accuracy classes and similar classes. We add the Filter and SCP modules to the prompt generation architecture. The Filter module selects the poorly classified classes, and then reproduce the prompts through the SCP (Specific Class Prompt) module to replace the prompts of specific classes. Experimental results on six multi-domain datasets shows the superiority of our approach over the state-of-the-art methods. Particularly, the improvement in accuracy for the specific classes mentioned above is significant. For instance, compared with CoOp-UC on the OxfordPets dataset, the low-accuracy classes, such as, Class21 and Class26, are improved by 18% and 12%, respectively.

Keywords:
Multi-modal, Vision Language Model, Prompt Tuning , Large-scale Pre-training Model

---

1. Introduction

Recently, various large-scale pre-training models enjoy widespread popularity and have extensive applications across various downstream domains due to their remarkable capabilities. These models for example are BERT [1], GPT3 [2], InstructGPT [3], in the NLP domain, and CLIP [4], ALIGN [5], and BLIP [6] in the multi-modal field. The primary advantage of large-scale pre-training models stems from their extensive range of understanding capabilities, acquired through training on extensive pre-training datasets. Additionally, these models can be directly applied to downstream datasets through the utilization of zero-shot learning. Compared with unimodal vision models, multi-modal models, e.g., CLIP, exhibit enhanced power in image classification tasks. CLIP excels in leveraging textual information, specifically class words which accomplishes image classification by mapping image features and text features into a shared feature space for similarity matching. In contrast, visual models necessitate training on specific datasets, limiting their processing capabilities to trained classes and restricting them to predefined labels, making them incapable of handling novel class samples.

By incorporating a prompt alongside simple class words, the input text information in CLIP can be enriched manually. The influence of a well-suited prompt on a model's classification ability is highly significant. For example,

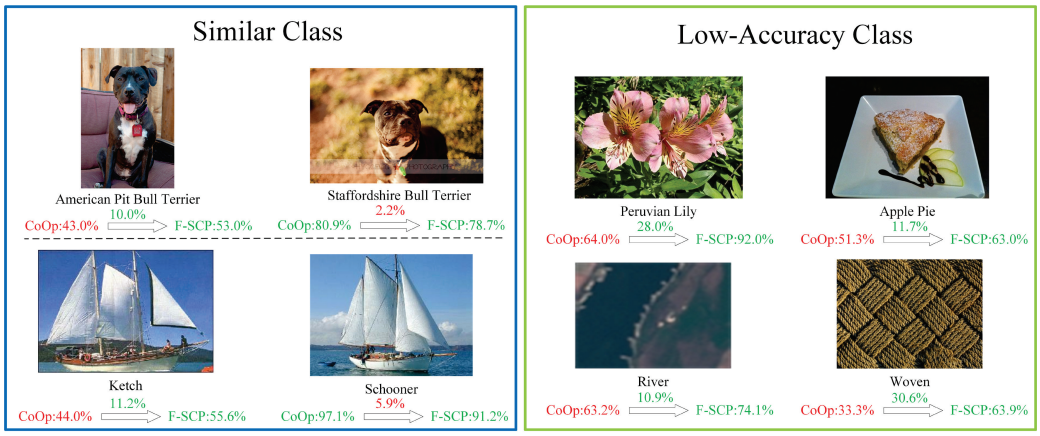


Figure 1: **Sample images.** Compared with CoOp, our F-SCP model demonstrates a notable improvement in accuracy for specific classes. The specific classes comprise similar classes and low-accuracy classes. In the case of similar classes, a lot of images of B are misclassified as A, resulting in a low accuracy of B. Our approach can substantially enhance the model's effectiveness specifically for the low-accuracy B, while maintaining the performance of the high-accuracy class A. Additionally, for the low-accuracy class, we can achieve a significant improvement in the accuracy of it, resulting in an overall enhancement in accuracy of image classification task.

when CLIP includes a straightforward prompt: "A photo of a {CLASS}" in the ImageNet dataset, it demonstrates a notable 1.3% enhancement in classification accuracy. In the case of the Oxford-IIIT Pet dataset, incorporating "a type of a pet" after the aforementioned prompt template can lead to a 1% increase in accuracy. CLIP employed the prompt engineering and ensembling approach for experimentation across 36 datasets, resulting in an average improvement of 5%.

Zhao et al. [7] proposed the CoOp model, which realized the automatic prompt generation for CLIP. This method substantially improves the image classification ability of CLIP model. CoOp comprises two designs: unified context mode, where all classes in the dataset employ the same prompt, and class-specific context mode, where each class in the dataset utilizes a distinct prompt. In our study, we observe that CoOp may exhibit limited effectiveness in addressing certain specific classes. This limitation may arise due to the presence of similar classes, for which the prompts generated by CoOp had no effective discrimination. As shown in Figure 1, CoOp results in large accuracy difference between similar classes, such as, American Pit Bull Terrier and Staffordshire Bull Terrier. This is because one of the classes

is heavily misclassified to its similar classes.

In CoOp-UC, all classes in the dataset use the same prompt. However, the identical prompt hardly guarantees to fit all classes in the dataset. In CoOp-CSC, each class in the dataset has its own prompt, which may contain many common features between classes. This will affect the classification results of their respective classes with each other. Moreover, experiments prove that CoOp-CSC is only suitable for fine-grained datasets and is not as effective as CoOp-UC on most of the datasets. To improve the poor performance of specific class models, we introduce a validated approach to tackle the problem in CoOp and other Vision Language Pre-training (VLP) models.

In this paper, we propose a flexible prompt generation architecture that targets specific hard classes. Two modules, that is, Filter and SCP, are incorporated into the prompt generation process. The Filter module figures out the hard samples and similar samples in the downstream dataset. The SCP module enhances the cross-entropy loss function by incorporating a weight matrix based on the ordinal number of the specific class. Afterwards, we retrain the model to generate prompt for the specific hard classes prompt. Experimental results show that F-SCP gains significant accuracy improvements both for specific classes and the overall model accuracy.

The major contribution of our work to pattern recognition is presenting an effective targeted prompt tuning framework to improve the classification accuracy of challenging low-accuracy classes. It focuses on enhancing the vision language large-scale pre-training model performance which can be directly transferred to multiple downstream applications through zero-shot learning. Through the introduction of Filter and SCP modules to selectively identify and generate tailored prompts for certain classes, we provide a systematic approach to leverage prompt tuning in a focused manner.

The substantial improvements achieved by tuning prompts for specific classes evidence the efficacy of this methodology in advancing pattern recognition systems. While prior works also explored prompt tuning, they adopt a generic way for all classes, which is not the best solution. Our work shows that not all classes benefit equally from the same prompt. By targeting prompt generation to specific classes, we can improve the overall performance.

To summarize, the work has the following contributions:

1. To the best of our knowledge, we are the first to identify the limitation

of prompt generation in current large-scale pre-training models. To boost the performance for challenging specific classes, we present a systematic framework for conducting effective targeted prompt tuning.

2. We design a set of criteria to identify and pick out specific classes. Moreover, we propose a method to generate more appropriate prompts for these specific classes while ensuring minimal impact on the accurate classification of other classes, particularly similar classes. We generate specific classes prompts by augmenting the weights in the loss function's weight matrix that corresponds to the selected classes.

3. Experiments on six classification datasets with distinct domains, show the superiority of our approach over both CoOp-UC and CoOp-CSC methods. We demonstrate the potential of applying targeted prompt tuning across diverse recognition domains like fine-grained classification, texture recognition, satellite image labeling, and etc.

2. Related Work

2.1. Vision Language Pre-training Model

Initially, visual language pre-training models undergo self-supervised learning, typically employing a pre-training task to extract supervised signals from extensive unlabeled data. This training enables the model to acquire a universal representation. Subsequently, these models can be fine-tuned using a small amount of hand-labeled data specifically for downstream tasks, resulting in remarkable outcomes.

Earlier VLP models utilized pre-trained object detectors to extract visual features. This object detection-based approach to regional feature extraction has led to strong performance for VLP models. As the pioneering image-text pre-training model, VisualBERT [8] , adopted Faster R-CNN [11] for visual feature extraction , which was pre-trained on the COCO dataset. VisualBERT demonstrated robust performance on language and image tasks (e.g. VQA and VCR). Similarly, VL-BERT [9] utilized the same feature extraction architecture. It enhanced the original BERT by taking RoI (Region of Interest) and text as inputs, and was pre-trained on the larger CC3M dataset. This led to enhanced performance over VisualBERT. Additionally, VL-BERT employed MVM (Masked Vision Modeling) as the pre-training task. Previously, most VLP models used pre-training tasks inspired by MLM (Masked

Language Modeling) in NLP models like BERT, which did not effectively leverage visual information.

While object detection-based regional features yield remarkable performance, extracting such features can be computationally expensive. To address this, SimVLM [12] developed an efficient visual language pre-training framework using CNNs for image feature extraction. Unlike previous methods utilizing object detection and auxiliary losses, SimVLM treated the entire image as a patch and used a consistent language modeling objective for end-to-end training. Similarly, PREVALENT [10] also used CNN to extract image features for end-to-end training, presented a pioneering pre-training and fine-tuning algorithm for the VLN (Vision Language Navigation) task. PREVALENT involved training on a vast collection of graph-text-action triads using a self-supervised learning method, and fuller use of visual information resulting in exceptional performance. The SmiDocNADE [13] model integrated and augmented features extracted from multimodal data. It achieved extensive multimodal data fusion and association mining, resulting in state-of-the-art performance for marketing intent analysis.

The scale of pre-training dataset is a major factor influencing VLP model performance. CLIP [4] , an influential recent VLP model, adopted a simple architecture and utilized a pre-training dataset consisting of more than 400 million image-text pairs collected from the Web. CLIP used contrastive language and image pre-training, achieving remarkable results that rival those of supervised models in zero-shot scenarios. CLIP demonstrated the importance of large-scale pre-training datasets for model capability improvement. Inspired by CLIP, ALIGN [5] was pre-trained on 1.8 billion noisy image-text pairs, achieving excellent performance and showing dataset noise did not impede effectiveness. In order to improve the training effectiveness of the self-supervised model, unlike CLIP which only used contrastive language and image pre-training, the ALBEF [14] proposed a new ITC (Image-Text contrast Learning) pre-training task and utilized knowledge distillation to enhance the training process.

However, most existing pre-trained models only excel at image-text retrieval or text generation tasks, not both. BLIP [6] employed an innovative model architecture of MED (Multi-modal mixture of Encoder-Decoder) for efficient multi-task pre-training and adaptive transfer learning. And most of the existing pre-training models extend the dataset with noisy image-text pairs collected from the web in order to improve the performance, but it is obvious that this noisy supervised signal is certainly not optimal, so BLIP

used a dataset cleaning method called CapFilt (Captioning and Filtering).

Since VLP models rely on image retrieval for classification, enhancing retrieval also improves classification capability. Unlike cross-modal retrieval through binary hashing in DMVH [16] , the VLP model is based on the image-text matching approach by learning the relationship between different modal features of the same instance to infer the potential alignment between sentence fragments and image regions, thus realizing image-text matching. In order to be able to better utilize the VLP model for image retrieval tasks without re-training the VLP model, AGREE [15] method was based on the VLP model, and the image-text entities were aligned in the fine-tuning and reordering phases through the three modules of VEA (Visual Entity-Image Alignment), TEA (Textual Entity-Image Alignment), and TIA (Textual-Image Entity Alignment), which improved the VLP model's image retrieval performance. To be able to better capture the interaction between text and image, the VLCDoC [17] model captured interactions between image-text through the InterMCA and IntraMSA attention modules, and obtained a significant improvement compared to the unimodal model on the document classification task.

2.2. Prompt Tuning

Prompt Tuning was initially employed in the field of **NLP** to address the disparity between pre-training data and downstream datasets, facilitating the adaptability of large-scale pre-training datasets to specific tasks. This approach focuses on minimizing the discrepancy between the pre-training and downstream phases, enhancing the applicability of large-scale pre-training datasets.

In the field of NLP, Petroni F et al.[18] proposed a manual prompt design method, which aligns the structure of downstream task datasets with the MLM pre-training dataset format. This approach aims to minimize the discrepancy between the pre-training task and the downstream task, thereby enhancing the applicability of large-scale pre-training datasets. Manual prompt design lacks stability in performance and requires substantial manual effort. To tackle this limitation, AUTOPROMPT [19] integrated the original task inputs with generated trigger tokens using a gradient search approach to create prompts generalizable across all inputs. OptiPrompt [20] method further enhanced this technique by not restricting the search space of the cues to discrete tokens, but rather optimizing the cues directly on the continuous.

Inspired by prompts used in NLP, Jia et al. [21] presented VPT, a novel fine-tuning approach for large-scale pre-training models in the vision domain. VPT introduced a small number of learnable parameters for fine-tuning and achieved remarkable outcomes. Similarly, ZegCLIP [22] integrated a set of learnable vectors as visual cues into each layer of the fixed CLIP image encoder to solve the zero-shot semantic segmentation task. Prompt tuning can also be applied to the pre-trained generative model. Xiao et al. [23] proposed a novel text-guided image framework that enables stable training of multi-text image. It can edit different images conditioned on text prompts within one model, without needing separate models for each text. Extending CLIP to video, X-CLIP [24] utilized cross-frame attention and video-specific cueing techniques. These techniques enable the use of semantic information from class-tagged text when modeling video frame timing information. To bridge the gap between visual and textual representations, UPT [25] combined a network of shared cues to generate visual cues and textual cues.

To enable automatic prompt generation and improve the adaptability of CLIP to downstream tasks, CoOp [7] added learnable word vectors in front of the input text, automatically updated based on context to obtain optimal prompts. Subsequently CoCoOp [26] added a lightweight network called Meta-Net to solve the overfitting problem, which can transfer input information to learnable cue vectors. Compared to CoOp, CoCoOp improved the accuracy of the model for new classes. To further mitigate CoOp overfitting, SubPT [27] projected the gradients in backpropagation onto the low-rank subspace spanned by the early gradient flow feature vectors throughout the training process, improving the model's image classification ability based on CoOp. For generalized performance on new domains, the TPT [28] model obtains excellent performance by randomly cropping visual cues from the input images. Table 1 provides clear comparative results, including our proposed F-SCP. Compared with the other methods, the proposed F-SCP focuses prompt tuning on low-accuracy categories while retaining the initial unified prompt for other categories.

3. Methodology

Figure 2 shows the architecture of F-SCP, which is built upon the foundations of both the CLIP model and the CoOp model.

Table 1: **Comparison of related works**

Model	Vison FE	Hand-Crafted Prompt	Automated Prompt Engineering	Unified Prompt	Class-Specific Prompt
VisualBERT	OD-RFs	-	-	-	-
VL-BERT	OD-RFs	-	-	-	-
CLIP	CNN/Xformer	✓	✗	✓	✗
ALBEF	Xformer	✓	✗	✓	✗
BLIP	CNN/Xformer	✓	✗	✓	✗
CoOp-UC	CNN/Xformer	✓	✓	✓	✗
CoOp-CSC	CNN/Xformer	✓	✓	✗	✓
F-SCP	CNN/Xformer	✓	✓	✓	✓

3.1. Contrastive Language-Image Pre-training (CLIP)

Large-scale pre-training datasets play a crucial role in ensuring the outstanding performance of VLP models. The CLIP model utilizes a pre-training dataset consisting of over 400 million image-text pairs sourced from the Internet. The CLIP model consists of two encoders: an image encoder, which can employ backbones like ResNet [29], ViT [30], and a text encoder, which can use backbones like CBOW [31], Transformer [32]. CLIP applied in the domain of image classification is similar to the image retrieval method. It involves extracting image features using the image encoder and extracting the features of each class word using the text encoder. The image encoder extracts the image features denoted as I and the text encoder extracts the features of each class word represented as $\{T_i\}\left(_{i=1}^{K}\right)$(with K classes). These features are used to calculate cosine similarity. The class with the highest similarity to the image features I is then output, determined by the following formula:

$$P\left(t|I\right) = \frac{exp\left(sim\left(I, T_t\right)/\tau\right)}{\sum_{i=1}^{K} exp\left(sim\left(I, T_t\right)/\tau\right)}, \tag{1}$$

where τ is a learnable temperature parameter, and $sim(\cdot, \cdot)$ denotes the cosine similarity.

To enhance the model's performance, the input text can be augmented by including a prompt preceding the class word, such as "a photo of a [CLASS]".

3.2. CoOp

CoOp enhances the hand-crafted prompt template method based on the CLIP model. Hand-crafted prompt templates demand substantial effort and suffer from instability, making it challenging to find the optimal prompt template for the downstream dataset. Additionally, even a slight modification to the prompt can significantly affect the model's results. Consequently, CoOp

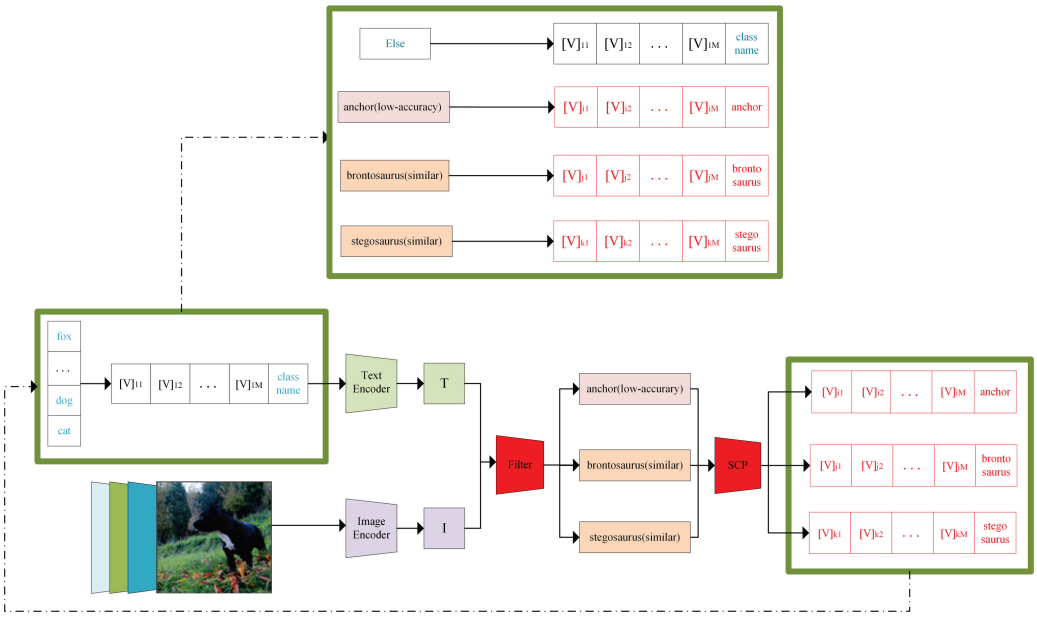


Figure 2: **The architecture of the F-SCP model.** Based on CoOp, classes are selected as specific ones if they are low-accuracy classes or similar classes with low-accuracy. Subsequently, the class number obtained from the Filter module is inputted into the SCP module, where a weight matrix is incorporated based on the class number within the cross-entropy function to retrain and obtain new prompts for the specific classes. Lastly, the prompts obtained from the SCP module are embedded within the original prompts array to obtain the final prompts.

presents an automated prompt design method that transforms the original manual prompts into M learnable word vectors:$\{v_1, v_2, v_3, \ldots, v_M\}$, CoOp further presents two designs for different datasets: the UC mode, which employs the same prompt for all classes in the dataset, and the CSC mode, which utilizes separate prompts for each class, making it more suitable for fine-grained classification tasks. In the UC mode, the text is fed into the CLIP text encoder as $T_i = \{v_1, v_2, v_3, \ldots, v_M, c_i\}$, where c_i denotes the name of the i-th class. The subsequent calculation follows the same procedure as CLIP, using the following formula:

$$P\left(t|I\right) = \frac{exp\left(sim\left(I, g\left(T_t\right)\right)/\tau\right)}{\sum_{i=1}^{K} exp\left(sim\left(I, g\left(T_i\right)\right)/\tau\right)}, \tag{2}$$

where $g(\cdot)$ denotes the text encoder of CLIP, τ represents a learnable temperature parameter, and $sim(\cdot, \cdot)$ represents the cosine similarity.

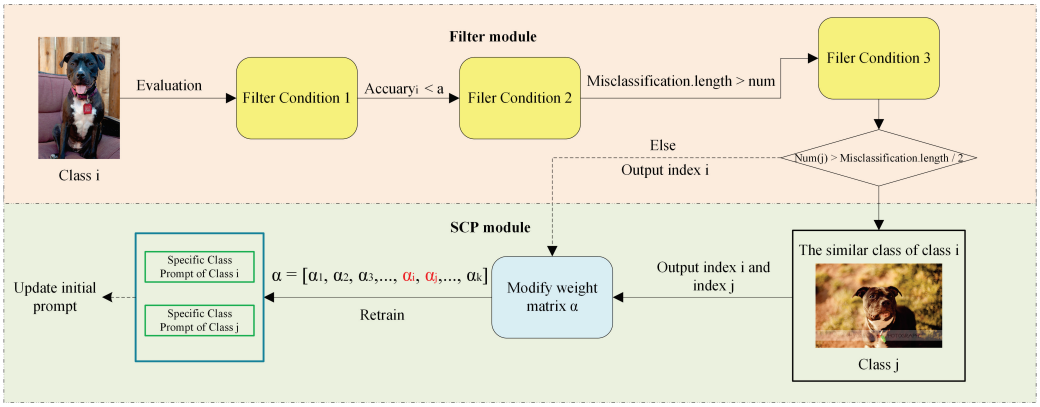


Figure 3: **The flow of the Filter module and the SCP module.** Misclassification represents the class to which the images in class i are misclassified, e.g. Misclassification = [2, j, j, j, j], representing that five images in class i are misclassified, of which one image is misclassified as class 2 and four images are misclassified as class j. The a, num and α in the figure can be set with reference to Equation (3), Equation (5), Equation (9) and Equation (10).

In the CoOp-CSC mode, each class has a different prompt. Therefore, the prompt for the i-th class is $\{v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iM}\}$, and the text input to the CLIP text encoder is $T_i = \{v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iM}, c_i\}$. The calculation formula used is the same as Equation (2).

3.3. The proposed F-SCP

CoOp often experiences lower accuracy for certain classes compared to others when performing image classification tasks. We attribute this issue to the requirement of CoOp's UC mode where all classes in the dataset share the same automatically generated prompt. It is challenging for a single prompt to adequately represent all classes in the dataset. On the other hand, CoOp's CSC mode generates a separate prompt for each class, but it overlooks the impact of one class's prompt on other classes. Consequently, the CSC mode is more suitable for fine-grained datasets but generally less accurate than the UC model across most datasets. To address this, we propose the F-SCP model, which combines the strengths of the UC mode and CSC mode. It selects the specific classes using the Filter module and modifies prompts of specific classes to enhance model performance. The detailed flow of Filter module and SCP module is shown in Figure 3.

3.3.1. Filter

The primary filter condition for the Filter module is accuracy. After generating the initial prompt through CoOp, model evaluation is performed to generate the accuracy results for each class. We aim to identify and pick out poorly classified classes before passing them to the SCP module for individual prompt generation. Since the average accuracy varies across datasets, it is challenging to establish a fixed threshold for low accuracy. Therefore, we propose setting the threshold manually based on the dataset results. The filter condition 1 for the Filter module is defined as follows:

$$Accuracy_i < a, \tag{3}$$

where $Accuracy_i$ represents the classification accuracy of class i, and a represents the manually set accuracy threshold (see Section 4.5).

The low accuracy of a class is attributed not only to the difficulty of the samples but also to the insufficient sample size. Therefore, it is not practical to generate a separate prompt for classes with inadequate samples. Generating prompts for classes with a limited number of samples not only has a negligible impact on improving the overall accuracy but also has the potential to affect the correct classification of a few images in other classes. Consequently, we introduce the filter condition 2 to the Filter module:

$$num = \frac{Total}{CK} \tag{4}$$

$$Total_i - Correct_i > num, \tag{5}$$

where $Total$ represents the overall sample size, C is a constant with a default value of 4 (see Section 4.5), K denotes the number of classes, $Total_i$ refers to the total sample size within the class i, and $Correct_i$ represents the number of correctly classified samples of the class i, and num is the threshold value for the number of images to be corrected, if the number of images to be corrected is greater than this value, the class is picked out. We recommend configuring the value of num based on Equation (4). It is also permissible to adapt this value according to individual preferences.

Selecting based solely on accuracy is inadequate. We encounter another issue related to the presence of similar classes within the dataset. Some classes exhibit low accuracy due to a significant number of images being misclassified as another similar class. We find the problem when two similar classes face, that is, one class is classified with a high accuracy, but the other

with low accuracy. Moreover, creating a separate prompt only for the low-accuracy class would flip the accuracy result of the involved classes. To address this, we introduce an additional constraint in the Filter module. We select out classes that exhibit low accuracy and examine the misclassified class to which they belong. If more than 50% of the misclassified images belong to the same class, we consider both classes to be similar and select them out, subsequently outputting them to the SCP module for separate prompt generation. Hence, the filter condition 3 of the Filter module is as follows:

$$\begin{aligned} \text{if } \max\left(\text{bincount}\left(x_1, x_2, x_3, \ldots, x_m\right)\right) > \frac{m}{2}, \\ x = \text{mode}\left(x_1, x_2, x_3, \ldots, x_m\right) \end{aligned} \tag{6}$$

where the set $\left\{x_1, x_2, x_{3,\ldots,} x_m\right\}$ represents the class serial numbers in which the sample was misclassified. m refers to the number of misclassified samples. $bincount(\cdot)$ denotes the frequency count of each element in the statistical array, while $mode(\cdot)$ signifies the value that occurs most frequently in the array.

Overall, filter condition 1 and filter condition 2 ensure the selection of low-accuracy classes with an adequate sample size. Filter condition 3 identifies similar classes for the low-accuracy ones. Following these three filter conditions, we select the low-accuracy classes with sufficient sample sizes, along with their similar classes. Afterwards, new prompts tailored for these specific categories are generated in the following SCP module.

3.3.2. Specific Class Prompt

The SCP module generates separate new prompts for the classes selected by the Filter module and replaces those initial prompts generated by CoOp. Inspired by Focal Loss [33], we introduce a weight matrix α to modify the cross-entropy loss function. The updated cross-entropy loss function is given by Equation (8).

$$CE(x) = -\sum_{i=1}^{C} y_i \log f_i(x), \tag{7}$$

$$NCE(x) = -\sum_{i=1}^{C} \text{softmax}(\alpha)_i y_i \log f_i(x), \tag{8}$$

where y_i is the true label of the i-th class, the $f_i(x)$ is the output value of the model, and α is a $1 \times K$ weight matrix, where K signifies the number of classes. The default value of the weight matrix α is the all-one matrix.

In the weight matrix α, it is necessary to assign higher weights to the corresponding classes during training to generate prompts. The weight matrix only modifies the weights of one class in each training round, except for the similar classes mentioned at filter condition 3 which the weights need to be adjusted together during training. The generated prompts are then extracted based on the class number and incorporated into the original prompts array. The value of the weight of class i can be determined using Equation (9) or manually set.

$$\alpha_i = \frac{AN\left(1 - Accuracy_i\right)}{n_i}, \tag{9}$$

where N represents the average sample size of the classes in the dataset, and n_i represents the number of samples in the i-th class. The default value of A is 40. In section 4.6, we discuss the selection of the variable A.

In addition, when modifying similar classes together in the weight matrix, the two classes exhibit varying accuracy levels and different numbers of images requiring correction, so it is inappropriate to assign the same weight, and the weights of classes with low accuracy and a high number of images requiring correction should be slightly larger. The weight for the low-accuracy class can be calculated using Equation (9), while its similar class should have weights determined based on the guidelines outlined in Equation (10).

$$\alpha_j = \frac{\alpha_i Accuracy_i \left(Total_j - Correct_j\right)}{Accuracy_j \left(Total_{\,i} - Correct_i\right)}, \tag{10}$$

where the class i and the class j are considered to be similar classes. The equation regulates the weight for the similar class based on the classification accuracy of both the low-accuracy class and its similar class, as well as the number of misclassified samples.

4. Experiments and Discussions

4.1. Datasets

We perform comparative experiments using CLIP, CoOp, and our proposed method on six publicly available datasets from various domains: Caltech101 [34], OxfordPets [35], Flowers102 [36], Food101 [37], DTD [38], and EuroSAT [39]. In order to demonstrate the generalization ability of our approach, these datasets cover a range of domains, including fine-grained classification, radar satellite imagery, generic class classification, and texture classification. The detailed information of each dataset is shown in Table 2.

Table 2: **Detailed data of each dataset**

Dataset	Classes	Train	Val	Test	Prompt
Caltech101	100	4128	1649	2465	"a photo of a [CLASS]."
DTD	47	2820	1128	1692	"[CLASS] texture."
EuroSAT	10	13500	5400	8100	"a centered satellite photo of [CLASS]."
Flower102	102	4093	1633	2463	"a photo of a [CLASS], a type of flower."
Oxfordpets	37	2944	736	3699	"a photo of a [CLASS], a type of pet."
Food101	101	50500	20200	30300	"a photo of [CLASS], a type of food."

4.2. Training Details

Our model is based on the open-source code of CoOp. We utilize the ResNet-50 image encoder backbone for all experimental purposes. The initial prompt is generated by CoOp-UC or CoOp-CSC. By default, we set the number of word vectors to 16. Similar to CoOp, our context vectors are randomly initialized using a Gaussian distribution with a standard deviation of 0.02. For training, we employ SGD (Stochastic Gradient Descent) with a learning rate decayed according to the cosine annealing rule. We use a value of 16 for shots and train for a total of 200 epochs.

4.3. Baseline

Our model is compared with two baseline models. The first baseline model is zero-shot CLIP, where the prompts in our training CLIP are manually tailored for specific datasets, For instance, the OxfordPets dataset prompt is set as "a type of pet," while the Flower102 dataset prompt is "a type of flower," and so on. The second baseline model is CoOp, and our model builds upon the improvements made in CoOp. We specifically evaluate our model against both the UC mode and CSC mode of CoOp.

4.4. Comparison

In this section, we compare F-SCP with CoOp-UC, CoOp-CSC, and zero-shot CLIP on image classification datasets from six different domains. Our objective is to demonstrate that our model significantly enhances the classification performance of the CLIP model. The experimental results are presented in Table 3.

Table 3: **Comparison of experimental results**

	Caltech-101	DTD	EuroSAT	Flower102	Oxfordpets	Food101
CLIP	86.1	42.6	38.2	66.1	86.2	**77.8**
CoOp-UC	91.7	63.2	81.8	94.4	86.9	74.8
CoOp-CSC	90.3	63.1	82.8	96.0	79.7	70.4
Ours(F-SCP)	**92.0**	**65.2**	**85.8**	**96.2**	**87.6**	76.3

Table 4: **We show four classes for each dataset, where red denotes the selected low-accuracy classes and blue signify the selected similar classes.** F-SCP effectively mitigates the issue of low accuracy in certain classes, leading to an improvement in the overall model accuracy on the dataset.

	EuroSAT	Class 0	Class 3	Class 6	Class 8
CoOp	81.8	76.6	67.3	70.8	63.2
Ours(F-SCP)	85.8	87.1	69.1	81.2	74.1
Improvement	4.0	10.5	1.8	10.4	10.9

	DTD	Class 1	Class 4	Class 10	Class 44
CoOp	63.2	36.1	27.8	30.6	33.3
Ours(F-SCP)	65.2	52.8	44.4	55.6	63.9
Improvement	2.0	16.7	16.6	25.0	30.6

	OxfordPets	Class 2	Class 34	Class 21	Class 26
CoOp	86.9	43.0	80.9	71.0	63.0
Ours(F-SCP)	87.6	53.0	78.7	89.0	75.0
Improvement	0.7	10.0	-2.2	18.0	12.0

	Caltech101	Class 54	Class 78	Class 26	Class 59
CoOp	91.7	97.1	44.4	47.6	60.0
Ours(F-SCP)	92.0	91.2	55.6	52.4	70.0
Improvement	0.3	-5.9	11.2	4.8	10.0

	Flowers102	Class 17	Class 40	Class 42	Class 88
CoOp	94.4	64.0	73.7	79.5	74.5
Ours(F-SCP)	96.2	92.0	92.1	82.1	96.4
Improvement	1.8	28.0	18.4	2.6	21.9

	Food101	Class 0	Class 15	Class 37	Class 77
CoOp	74.8	51.3	50.3	49.7	51.0
Ours(F-SCP)	76.2	63.0	54.3	78.7	35.7
Improvement	1.4	11.7	4.0	29.0	-15.3

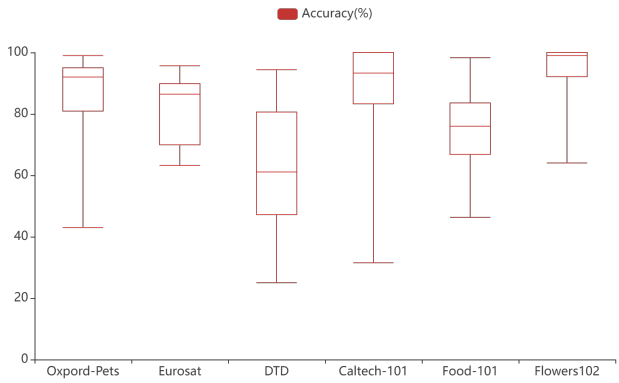


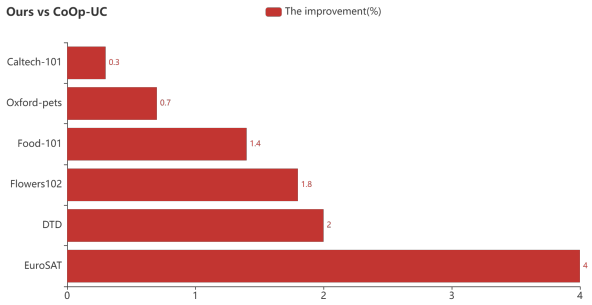
Figure 4: **Box plots of the accuracy of each class in the six datasets.** The accuracies shown in the figure are from the CoOp-UC model. The box plots demonstrate that each dataset contains certain classes with significantly lower accuracy compared to the majority of classes.

4.4.1. F-SCP vs CoOp

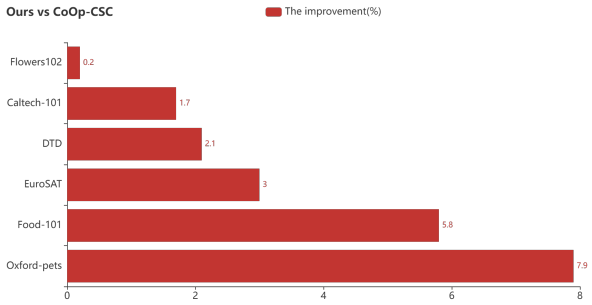
In the experiments, our model outperforms both CoOp-UC and CoOp-CSC across the six datasets. Such as our model achieves a 4-point improvement over CoOp-UC on the EuroSAT dataset. When compared to CoOp-CSC on all six datasets, except for a relatively minor improvement on Flowers-102, our model had a greater advantage (refer to Figure 5). CoOp faces the issue of certain classes being significantly less accurate than others across most datasets (refer to Figure 4). F-SCP demonstrates substantial improvements in all selected low-accuracy classes, resulting in an overall enhancement in accuracy (refer to Table 4). One reason for the limited improvement observed on the Caltech101 dataset is its inherent class-imbalance issue, where the number of samples in the low-accuracy categories is relatively small. Consequently, generating specific prompts for these categories has less impact on overall accuracy enhancement.

4.4.2. F-SCP vs zero-shot CLIP

Our model demonstrates a substantial improvement compared to the zero-shot CLIP model using hand-crafted prompts across five datasets. Figure 6 illustrates the achieved improvements on six datasets, with an average improvement of 17.3%. Notably, DTD and Flowers102 exhibit improvements of more than 20%, while the EuroSAT dataset demonstrates a significant improvement of nearly 50%.



(a) Ours vs CoOp-UC



(b) Ours vs CoOp-CSC

Figure 5: **Our model has different magnitudes of improvement for both CoOp-CSC and CoOp-UC on the six datasets.**

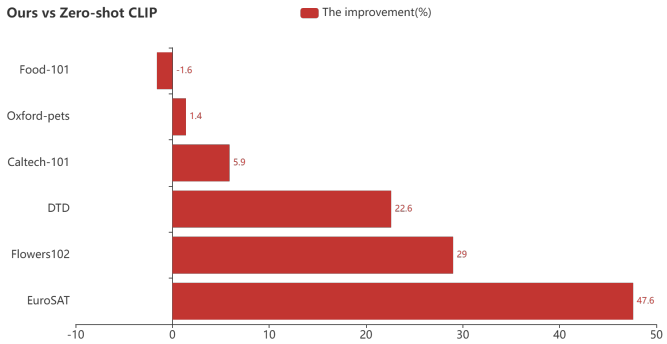


Figure 6: **Our model shows a significant improvement compared to the CLIP model on most of the data sets.**

However, similar to the CoOp model, our model's performance in the fine-grained dataset is subpar. It slightly underperforms the zero-shot CLIP on

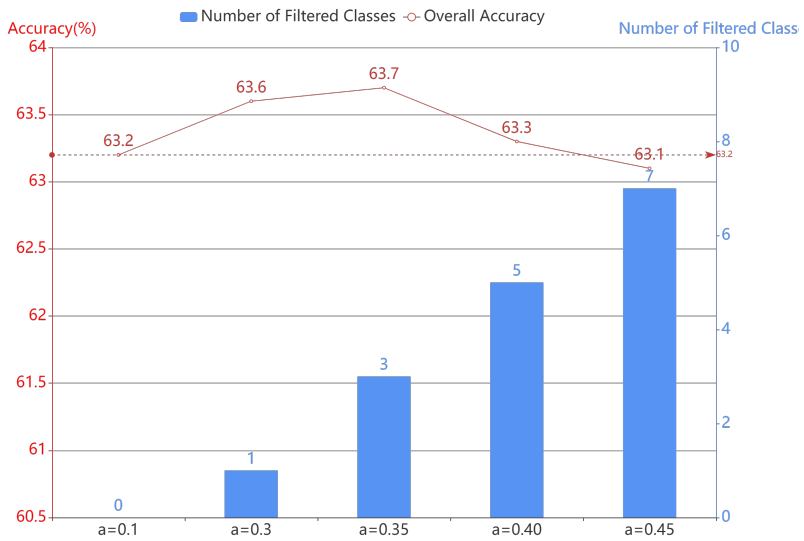


Figure 7: **The impact of variable a on accuracy and the number of selected classes.** We conducted experiments using the DTD dataset as an example, focusing solely on filter condition 1.

the Food-101 dataset and only narrowly outperforms the zero-shot CLIP on the Oxford-Pets fine-grained dataset by 1.4%. CoOp analyzed this issue as a result of model-generated prompt overfitting, which our approach partially mitigates.

4.5. Ablation study on Filter conditions

For the filter condition 1, we recommend adapting the accuracy threshold to different datasets for optimal performance. The value of a is associated with the accuracy of the categories in the dataset. To determine the optimal value, it is advisable to begin with the lowest accuracy and gradually increase it (see Figure 7). The value of a should be carefully chosen to avoid excessive selecting of categories. High values of a lead to an increased number of selected classes, resulting in a negative impact on overall accuracy (see Section 4.7).

Filter condition 2 holds significant importance, particularly in datasets with class-imbalance, as illustrated by the Caltech101 dataset. Refer to Figure 8, certain classes have significantly larger sample sizes others. Consequently, we examine cases where low-accuracy classes have a limited number of images in the dataset itself. As shown in Table 5, Class 58 in the Caltech-

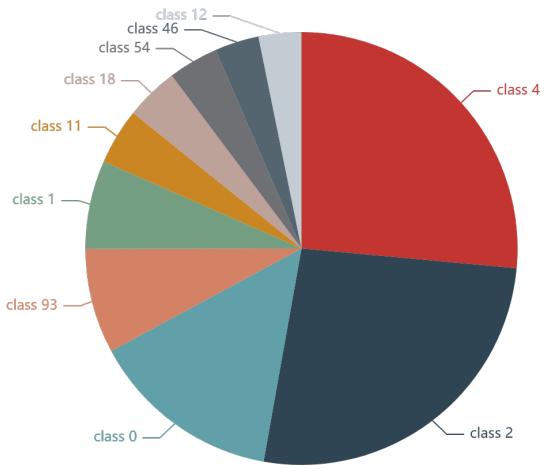


Figure 8: **The ten categories in the Caltech101 dataset, the area presents the number of class samples, highlight the presence of class-imbalance within the dataset.**

101 dataset has only 53% accuracy and belongs to the low-accuracy class. However, this class comprises only thirteen images. Even if the accuracy of this class can be improved to 100% after prompt generation, the impact on the overall dataset accuracy would be insignificant. Moreover, focusing on improving the accuracy of such a small class may adversely affect the correct classification of other classes (refer to Section 4.7). Table 6 emphasizes the significance of filter condition 2 and provides validation for the default value of C in Equation (4).

For the filter condition 3, we conduct experiments to validate its necessity. We use the Oxford-Pets dataset as an example (refer to Table 7). In this dataset, Class 2 was identified as a low-accuracy class and was frequently misclassified as Class 34, indicating a similarity between the two classes. If prompt generation is performed exclusively for Class 2, it would result in a significant drop in the accuracy of Class 34. However, when both classes are input into the SCP module for prompt generation, Class 2 improves by 11 percentage points while Class 34 only decreases by two points, which is an acceptable trade-off.

Table 5: **A part of CoOp-UC evaluation results for low-accuracy classes in the Caltech-101 dataset.**

Caltech101	Accuracy(%)	Total	Correct
Class 58	53.0	13	7
Class 27	40.0	15	6

Table 6: **The effect of C on the accuracy results with the Caltech101 dataset.** We conduct experiments using the Caltech101 dataset, configuring a as 0.75 in filter condition 1, and observe that the highest accuracy is achieved when C approached the value of 4.

C	Number of Filtered Classes	Accuracy(%)
1	0	91.7
2	1	91.8
4	7	92.0
8	15	91.2

Table 7: **Results of similar class experiments for the Oxford-Pets dataset.** In the table, "CoOp-UC" is the result of CoOp-UC, "Only Class2" and "Class2 and Class34" are the results of F-SCP.

OxfordPets	CoOp-UC(%)	Only Class2(%)	C2 and C34(%)
Class2	43.0	61.0(+18.0)	54.0(+11.0)
Class34	80.9	60.7(-20.2)	78.7(-2.2)

4.6. Weight setting of the SCP module

The SCP module adjusts the weight matrix based on the class serial number obtained from the Filter module, and there is no definitive standard for determining the weight size, weight assignment is influenced by the number of images in a specific class and its accuracy, we suggest using Equation (9) and Equation (10) which we derive for weight assignment, or you have the flexibility to set the weights according to your preference. Experiments were performed on the Class 37 which is a low-accuracy class of Food-101 dataset using various weight values and observed that the experiment yielded optimal results with α_{37} weight value of approximately 30 (refer to Figure 9).

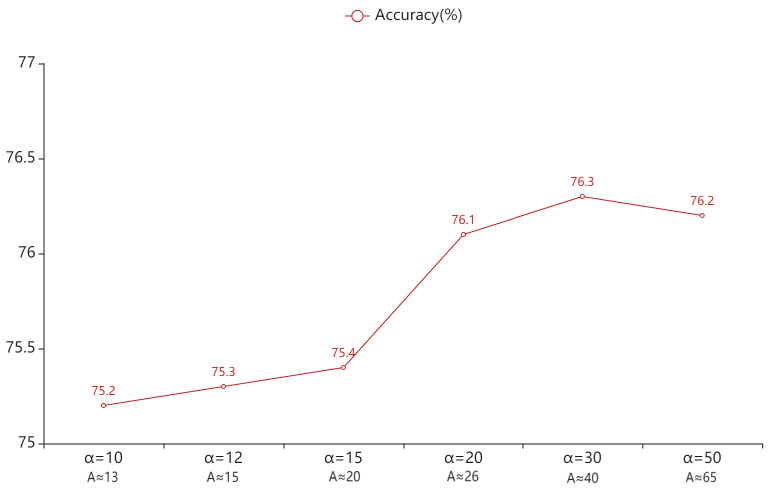


Figure 9: **The effect of weight α_{37} on the accuracy results with the Food101 dataset.** Experimentally, the best result is obtained when α_{37} is about 30. The value is close to the result of Equation (9) when A is approximately 40.

4.7. The effect of specific class prompt on other classes

After applying the Filter module, the classes are divided into two groups: selected classes and unselected classes. The prompts generated for the selected classes have a minimal impact on the classification of the other unselected classes. While the accuracy of the selected classes is greatly improved, there may be a slight decrease in accuracy for the unselected classes in the dataset. We attribute this decrease to the presence of few images in the unselected classes and the prompts generated by the selected classes containing common features. As a result, some images of unselected classes may be misclassified into the selected classes. Of course, there are also cases where images in the unselected class that would have been misclassified are correctly classified due to specific class prompt.

Take the EuroSAT dataset as an example, the accuracy improvement of each class in this dataset is shown in Figure 10. The selected classes in this dataset are Class 0, Class 3, Class 6, and Class 8, which show a significant improvement in accuracy. Class 5, and Class 7 also demonstrate a slight increase in accuracy. However, Class 2, Class 4, and Class 9 experience a minor decrease in accuracy. This can be attributed to specific class prompt also had an impact on the classification of unselected classes, a few pictures in Class 2 are misclassified as Class 6, some images in Class 4 are misclassified as Class 3, and a few images in Class 9 are misclassified as Class 8. Detailed

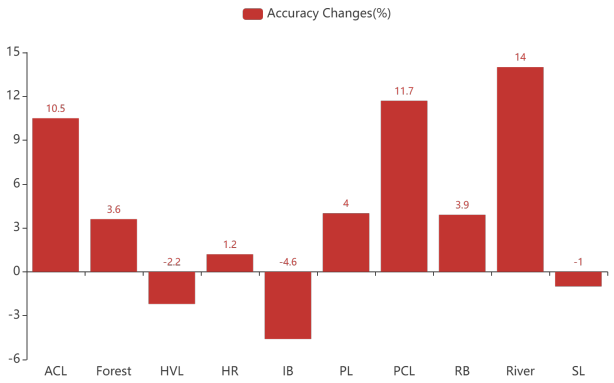


Figure 10: **The accuracy changes of F-SCP compared with zero-shot CLIP for each class in the EuroSAT dataset.** Among them, ACL, HVL, HR, IB, PL, PCL, RB, and SL correspond to "Annual Crop Land", "Herbaceous Vegetation Land", "Highway or Road", "Industrial Buildings", "Pasture Land", "Permanent Crop Land", "Residential Buildings" and "Sea or Lake". The class numbers start from 0 and correspond to the classes on the x-axis from left to right in the figure.

information is shown in Figure 11. This is consistent with our previous analysis that few images in these classes have visual features that match specific class prompt, thus affecting the correct judgement of the model, and this phenomenon is the reason why CoOp-UC performs better than CoOp-CSC on most of the datasets.

5. Conclusion

In this paper, we propose an automatic efficient prompt generation approach to leverage large-scale pre-training models performance in downstream tasks. We utilize targeted learning to address the limitations of CLIP models. The proposed model picks out specific classes and subsequently enhances the model's capabilities based on generating specific classes prompts by increasing the weights in the loss function's weight matrix that corresponds to the selected classes. When applying large-scale pre-training models to domain-specific downstream tasks, our approach can be adapted to enhance the model's capabilities and compensate for its limited classification performance on specific classes.

The proposed F-SCP can be applied to enhance the performance of pattern recognition systems in applications, such as, autonomous driving, where

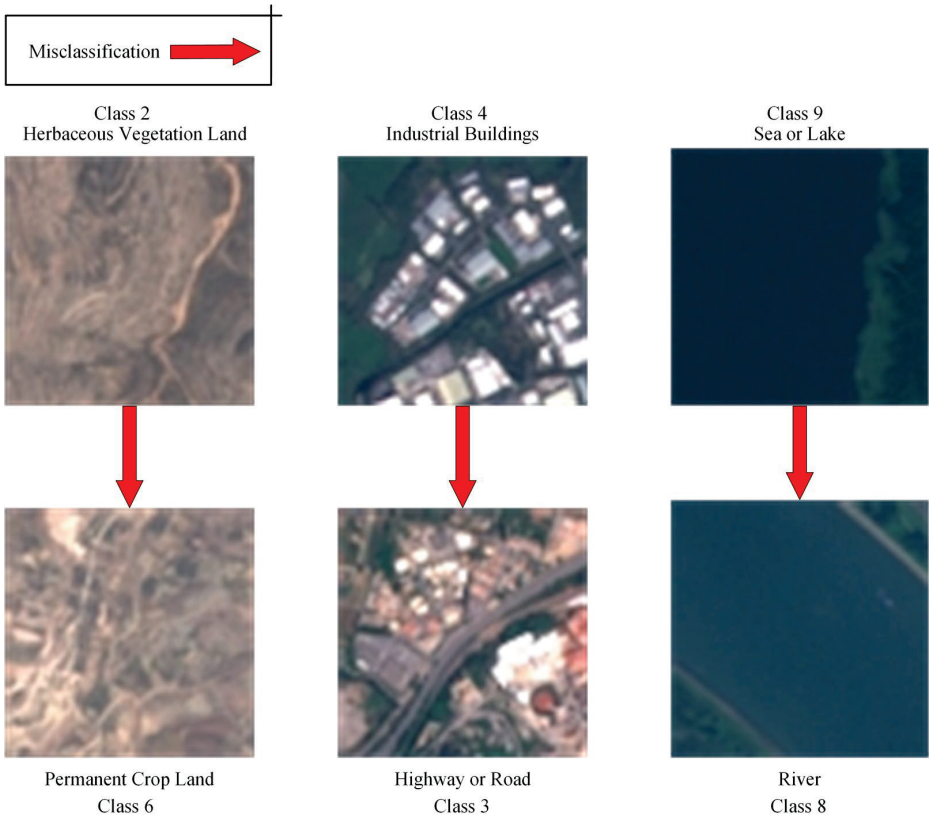


Figure 11: **Misclassified images.** Few images in unselected classes can be misclassified into selected classes by specific class prompt influence.

prompts can be tuned for cases that pose challenges to the system, thereby improving the safety and reliability. In commercial contexts, prompt tuning could improve accuracy for challenging products or conditions that an image classification system may struggle with, such as identifying damaged packages in logistics or obscured objects in surveillance. In classification tasks like species identification, targeted tuning can help distinguish species with which existing models struggles. Prompt tuning provides a way to tailor a general pre-trained model to specific applications that were not its original focus, using a small, application-specific dataset. This enhances the flexibility of deploying pattern recognition systems. Additionally, the methodology can be extended to conditional prompt tuning, dynamically adjusting prompts at test time to improve accuracy on challenging samples. In summary, F-SCP

presents a new capability for targeted accuracy improvements with potential benefits for real-world pattern recognition systems across various industries and applications. The ability to fine-tune existing large-scale pre-trained models for specific classes opens up new possibilities.

Nevertheless, our approach still struggles to perform effectively on fine-grained datasets, which is related to the overfitting problem present in CoOp. More advanced prompt tuning techniques could be explored for fine-grained distinctions. Moreover, the setting thresholds requires manual tuning on each dataset to achieve optimal performance. An adaptive, data-driven way of setting the thresholds could improve the model generalization ability.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC), Essential project, Nr.: U2033218.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1(Long and Short Papers) Association for Computational Linguistics, Minneapolis, Minnesota, 2018, pp. 4171—4186.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in neural information processing systems(NIPS), Vol. 33, 2020, pp. 1877–1901.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, in: Advances in Neural Information Processing Systems(NIPS), Vol. 35, 2022, pp. 27730–27744.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning(ICML), 2021, pp. 8748–8763.

[5] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International Conference on Machine Learning(ICML), 2021, pp. 4904–4916.

[6] J. Li, D. Li, C. Xiong, S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: International Conference on Machine Learning(ICML), 2022, pp. 12888–12900.

[7] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Learning to prompt for vision-language models, International Journal of Computer Vision(IJCV) 130 (9) (2022) 2337–2348.

[8] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, K.-W. Chang, Visualbert: A simple and performant baseline for vision and language, in: Annual Meeting of the Association for Computational Linguistics(ACL), 2019.

[9] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, J. Dai, Vl-bert: Pre-training of generic visual-linguistic representations, in: International Conference on Learning Representations(ICLR), 2020.

[10] W. Hao, C. Li, X. Li, L. Carin, J. Gao, Towards learning a generic agent for vision-and-language navigation via pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2020, pp. 13137–13146.

[11] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[12] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, Y. Cao, SimVLM: Simple visual language model pretraining with weak supervision, in: International Conference on Learning Representations(ICLR), 2022.

[13] L. Zhang, J. Shen, J. Zhang, J. Xu, Z. Li, Y. Yao, L. Yu, Multimodal marketing intent analysis for effective targeted advertising, IEEE Transactions on Multimedia 24 (2021) 1830–1843.

[14] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, S. C. H. Hoi, Align before fuse: Vision and language representation learning with momentum distillation, in: Advances in neural information processing systems(NIPS), Vol. 34, 2021, pp. 9694–9705.

[15] X. Wang, L. Li, Z. Li, X. Wang, X. Zhu, C. Wang, J. Huang, Y. Xiao, Agree: Aligning cross-modal entities for image-text retrieval upon vision-language pre-trained models, in: International Conference on Web Search and Data Mining(WSDM), 2023, p. 456–464.

[16] L. Xie, J. Shen, J. Han, L. Zhu, L. Shao, Dynamic multi-view hashing for online image retrieval, in: International Joint Conference on Artificial Intelligence(IJCAI), 2017.

[17] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol, O. R. Terrades, Vlcdoc: Vision-language contrastive pre-training model for cross-modal document classification, Pattern Recognition(PR) 139 (2023) 109419.

[18] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, in: In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP), 2019, p. 2463–2473.

[19] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing(EMNLP), 2020, pp. 4222–4235.

[20] C. D. ZHONG Z, FRIEDMAN D, Factual probing is [mask]: Learning vs. learning to recall, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT), 2021, pp. 5017—-5033.

[21] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.-N. Lim, Visual prompt tuning, in: European Conference on Computer Vision(ECCV), 2022, pp. 709–727.

[22] Z. Zhou, Y. Lei, B. Zhang, L. Liu, Y. Liu, Zegclip: Towards adapting clip for zero-shot semantic segmentation, in: Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition(CVPR), 2023, pp. 11175–11185.

[23] C. Xiao, Q. Yang, X. Xu, J. Zhang, F. Zhou, C. Zhang, Where you edit is what you get: Text-guided image editing with region-based attention, Pattern Recognition(PR) 139 (2023) 109458.

[24] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, H. Ling, Expanding language-image pretrained models for general video recognition, in: European Conference on Computer Vision(ECCV), 2022, pp. 1–18.

[25] J. Wang, C. Wang, F. Luo, C. Tan, M. Qiu, F. Yang, Q. Shi, S. Huang, M. Gao, Towards unified prompt tuning for few-shot text classification, in: Findings of the Association for Computational Linguistics(EMNLP), 2022, pp. 524–536.

[26] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2022, pp. 16816–16825.

[27] C. Ma, Y. Liu, J. Deng, L. Xie, W. Dong, C. Xu, Understanding and mitigating overfitting in prompt tuning for vision-language models, IEEE Transactions on Circuits and Systems for Video Technology(TCSVT) (2023).

[28] M. Shu, W. Nie, D.-A. Huang, Z. Yu, T. Goldstein, A. Anandkumar, C. Xiao, Test-time prompt tuning for zero-shot generalization in vision-language models, in: Advances in Neural Information Processing Systems(NIPS), 2022.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2016, pp. 770–778.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations(ICLR), 2021.

[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Computer Science (2013).

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems(NIPS), Vol. 30, 2017.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2017, pp. 2980–2988.

[34] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: 2004 conference on computer vision and pattern recognition workshop, 2004, pp. 178–178.

[35] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2012, pp. 3498–3505.

[36] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 2008, pp. 722–729.

[37] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: European Conference on Computer Vision(ECCV), 2014, pp. 446–461.

[38] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2014, pp. 3606–3613.

[39] P. Helber, B. Bischke, A. Dengel, D. Borth, Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (2019) 2217–2226.