**Semantic Prompt for Few-Shot Image Recognition - Appendix**

**First Author**
Institution 1
Institution 1 address
firstauthor@i1.org

**Second Author**
Institution 2
First line of institution2 address
secondauthor@i2.org

---

**Figure 1. Network structures.** We adopt Visformer [1] as the backbone, which consists of a convolution stem and three feature extraction stages. The first stage consists of seven convolution blocks, and the second and the third stages consist of four Transformer blocks. Patch embedding blocks are inserted between stages. (a) Transformer block. (b) Convolution block. (c) Patch embedding block. (d) The whole structure.

1. **Image encoder structure**

   We take Visformer [1] as our image encoder. It is built on ViT [2], and includes more CNN structures in the network. The framework of Visformer is shown in Figure 1. Totally, it contains three stages, where the first stage is composed by convolution blocks and the remaining stages are composed by Transformer blocks. A CNN stem is included at the bottom layer to stabilize network training. To reduce spatial resolution and inject position information, patch embedding blocks are inserted to the network between stages.

1.1. **Transformer block**

   The Transformer block of Visformer is illustrated in Figure 1 (a), which is similar to the Transformer block in ViT but replacing Layernorm (LN) with Batchnorm (BN). Specifically, a Transformer block consists of Multi-Head Self-Attention (MSA), two Batchnorm layers and a two-layer MLP. Denoting $Z_{l-1} \in \mathbb{R}^{M \times C_z}$ as the input to the Transformer block, the output $Z_l \in \mathbb{R}^{M \times C_z}$ can be written as:

   \[
   Z'_l = MSA(BN(Z_{l-1})) + Z_{l-1}, \\
   Z_l = MLP(BN(Z'_l)) + Z'_l,  \tag{1}
   \]

   where $M$ is the number of patch tokens, and $C_z$ is the number of channels.
**Base classes**

![w/o SP](image1.png) ![w/ SP](image2.png)

**Novel classes**

![w/o SP](image3.png) ![w/ SP](image4.png)

Figure 2. t-SNE results of feature distributions. We randomly select images from seven base classes (left) and seven novel classes (right). The image features are extracted by directly using the pre-trained feature extractor (w/o SP) or with the guidance of semantic prompts (w/ SP).

**Multi-Head Self-Attention.** The key to the Transformer block is the Multi-Head Self-Attention module, which aggregates relevant information in the global view into each image patch according to their attention weights. Let $Z \in \mathbb{R}^{M \times C_z}$ denote the input to a MSA module. MSA first maps the input into three values, i.e., $q, k, v \in \mathbb{R}^{N_h \times M \times C_h}$, with linear projection parameterized by $W_{qkv}$

$$[q, k, v] = Z W_{qkv}, \quad (2)$$

where $N_h$ is the number of heads and $C_h$ is the number of channels for each head. It then computes the attention matrix $A \in \mathbb{R}^{N_h \times M \times M}$ by taking the inner product between $q$ and $k$ and performing softmax along the spatial dimension (i.e., the last dimension):

$$A = softmax(qk^T/C_h^{\frac{1}{4}}). \quad (3)$$

The attention weights are used to aggregate $v$ within each head via weighted sum. The final output is obtained by concatenating outputs of all heads and performing linear projection parameterized by $W_{out}$:

$$MSA(Z) = (Av) W_{out}. \quad (4)$$

**1.2. Convolution block**

Visformer replaces Transformer blocks in the first stage with convolution blocks, as they can reduce computation costs without accuracy degradation. The convolution block is illustrated in Figure 1 (b). It follows an inverse bottleneck structure, where the first and third convolution layers are used to increase and decrease channel dimensions, respectively. The second layer is a group convolution layer with 8 groups and a kernel size of 3. The input is added to the output via a residual connection.

**1.3. Patch embedding**

The structure of the patch embedding block is illustrated in Figure 1 (c). Let $x \in \mathbb{R}^{H \times W \times C}$ denote the input to the patch embedding block, where $H \times W$ denotes the spatial resolution and $C$ is the number of channels. The patch embedding block transforms the input size into $H/P \times W/P \times CP^2$ via a convolution layer with the stride and kernel equal to $P$. This reduces the spatial resolution by $1/P^2$ and increases the channels by $P^2$. Then, learnable position embeddings are added to the normalized input to inject position information.

**2. Visualization of feature distributions**

To evaluate the effectiveness of the proposed Semantic Prompt (SP) approach for improving image representations, we visualize the feature distributions in Figure 2 with t-SNE [3], where ‘w/o SP’ means we directly use the pre-trained Visformer to capture image features, and ‘w/ SP’ means we capture image features with semantic prompts. It can be seen that most base classes have clear classification boundaries after pre-training, but there are still some classes (e.g., red and grey) are very close. After meta-trained with SP, these classes can be clearly separated with a large margin. More evident results can be found in novel classes, where there is no clear boundaries without using SP. This result indicates that image representations obtained with the pre-trained feature extractor are hard to capture the intrinsic class-specific features for novel classes. In contrast, using SP can effectively alleviate this problem and leads to a well-separated feature space.

**References**

