Weakly-supervised Spatially Grounded Concept Learner for Few-Shot Learning (Supplementary)

BMVC 2023 Submission # 858

In this Appendix, we include the following details, which we could not include in the main paper due to space constraints:

- Visualizing patches associated with concepts.
- Visualizing grounding concepts learned by VGCoL.
- Visualization of Similarity Matrix M on SUN dataset.
- Visualization of patches around semantic prototypes on SUN dataset.
- Ablation studies on CUB dataset.
- Visualization of patches around semantic prototypes from different layers on CUB dataset.
- Failure Cases for VGCoL.
- Data pre-processing for our experiments.

1 Visualizing patches associated with concepts

To demonstrate the visual semantics achieved by VGCoL, we visualize the patches around the prototypes obtained from the last module of our method on the AWA2 dataset (shown in Figure 1). We also use the same setup to visualize the patches around the prototypes of ConstellationNet [1]. We extract cell features that are nearest to the prototypes and then retrieve corresponding patches of the original image. Figures 1 (a), (b), and (c) show the patches extracted by ConstellationNet. The patches show some similarity but fail to demonstrate any coherence in terms of semantics. For instance, Figure 1 (a) depicts a mixture of *stripes* and *spots* while 1 (b) and 1 (c) depict the *lower body* and *mouth* of animals respectively (these differences are highlighted in red for easier visualization). In contrast, the proposed VGCoL model extracts more coherent patches in terms of semantics. As shown in Figure 1 (d) corresponds to patches around semantic concept *spots*, 1 (e) illustrates to *stripes* and 1 (f) shows the color *white*. Further, the ConstellationNet method is *non-identifiable*, which means a human is needed to interpret the learned prototypes. Unless the patches are extracted and visualized in an above-said manner, one cannot identify the prototype that was

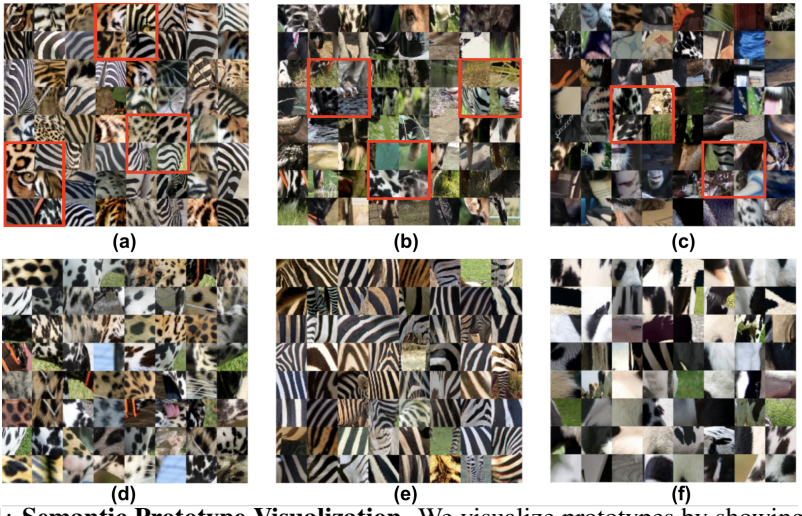

Figure 1: **Semantic Prototype Visualization.** We visualize prototypes by showing patches pertaining to the cell features that are nearest to a particular prototype. Three Constellation-Net [] prototypes are in (a), (b) and (c). Figures (d), (e) and (f) depict patches obtained by VGCoL that corresponds to *spots*, *stripes* and color *white* concepts respectively. The prototypes from our method correspond to the attributes present in the vocabulary for the AWA2 dataset.

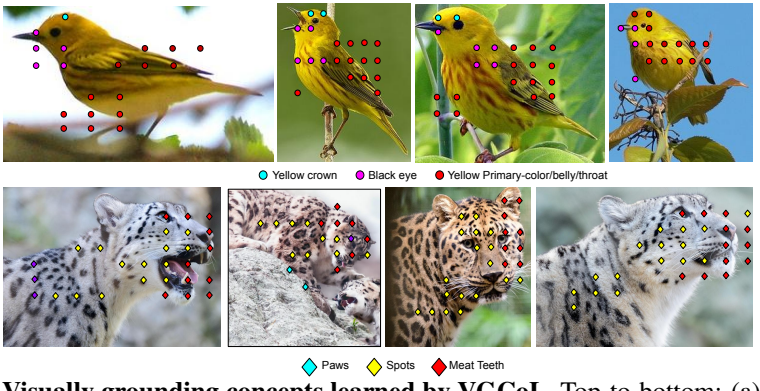

Figure 2: **Visually grounding concepts learned by VGCoL.** Top-to-bottom: (a) shows the grounding of different concepts to the spatial location of an image sample from the CUB dataset; (b) demonstrates the visual grounding of semantic concepts on a sample from the AWA2 dataset.

learned. However, the proposed method allows a one-to-one correspondence between the attributes and the learned concept prototypes. During training, each attribute gets associated with a particular concept prototype which makes VGCoL prototypes *identifiable* and easy to interpret.

2 Visualizing concept activation maps

In Figure 2 we visualize spatial grounding of three different semantic prototypes from CUB and AWA2 datasets, respectively. Figure 2 (a) shows four instances of the Yellow Warbler bird from the CUB dataset. We highlight the cell features nearest to the prototypes *yellow crown*, *black eye*, *yellow belly*, and *yellow throat*. Since the CUB dataset is fine-grained in attributes, we grouped the three attributes under a single name *yellow primary-color/belly/throat* and highlighted it with a red dot. Similarly, in Figure 2 (b) we highlight the attributes *paws*, *spots*, and *meat teeth* from the AWA2 dataset on the cell features that are closest to them (we do not group attributes for AWA2 since it is a coarse-grained dataset in terms of attributes). Interestingly, we observe that cell features near the head are closest to the prototypes for *eye* and *crown* in birds and *teeth* in leopards, despite us not providing any localization information for attributes while training. This grounding is learned implicitly.

3 Visualizing Similarity Matrix M on SUN dataset

Figure 3 visualizes the similarity matrix M for four different classes (*fences*, *canal*, *desert* and *canteen*) from the SUN dataset. We show the similarity maps corresponding to the top 4 concepts for each class. We can see strong activation in the different regions for different concepts. For example, in the third row, for the class desert, we can see that the model learns to pay attention to the region with the tree for grounding the concept *Tree*, while it looks at the sky region for grounding the concept *Clouds*. This suggests that the model is learning to associate semantics with different image regions.

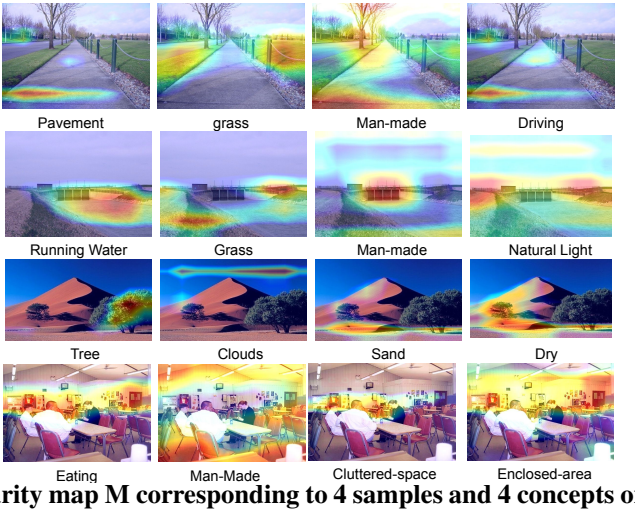

Figure 3: **Similarity map M corresponding to 4 samples and 4 concepts on SUN dataset**. Each row corresponds to a class and caption denotes concept.

Figure 4: **Semantic Prototype Visualization on SUN dataset.** We visualize prototypes by showing patches pertaining to the cell features that are nearest to a particular prototype. Figures (a), (b), and (c) depict patches obtained by VGCoL that correspond to *flowers*, *sports*, and *railroads* concepts, respectively.

4 Visualizing patches around semantic prototypes on SUN dataset

Figure 4 shows the patches pertain to three separate concepts (*flowers*, *sports* and *railroads*) from the SUN dataset. We can see that the prototypes learned by our model have a robust semantic correspondence with the attributes present in the data.

5 Ablation Studies

We conduct ablation studies on the CUB dataset using our VGCoL conv4 model. Figure 5 (a) shows the 1-shot and 5-shot accuracies when we introduce semantic decoders at different network positions. We observe that only one semantic decoder after the fourth convolutional layer attains the best accuracy. We make use of this model in all our experiments. Although having a semantic decoder after each convolutional layer also gives us a similar performance, we use the architecture with only one semantic decoder as this model is computationally faster. In Figures 5 (b) and 5 (c) we vary the values of beta and alpha and subsequently note

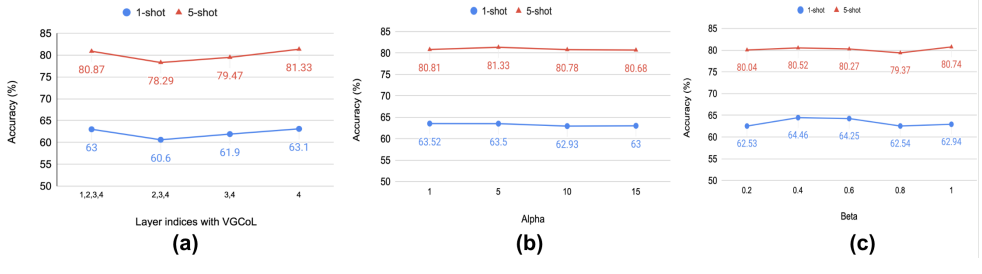

Figure 5: **Abaltions on CUB dataset using our VGCoL conv4 backbone architecture.** (a) shows the effect of having a semantic decoder at different positions in the network while (b) and (c) show the 1-shot and 5-shot accuracy for different values of alpha and beta.

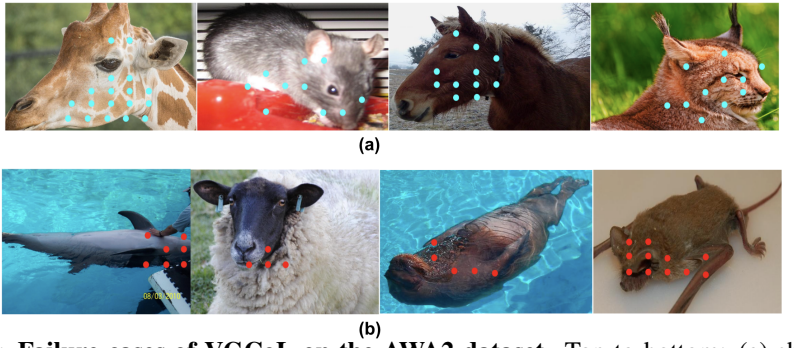

Figure 6: **Failure cases of VGCoL on the AWA2 dataset.** Top-to-bottom: (a) shows the visual grounding of imbalanced *quadra-pedal* concept; (b) demonstrates grounding of visually indiscernible and imbalanced *new-world* concept.

the 1-shot and 5-shot performances. We observe that the beta value in the range of 0.4 and 0.6 gives us the best performance. Similarly, when we set alpha to 5 we observe the best accuracies on 1-shot and 5-shot tasks.

6 Visualization of Patches for Different Layers

As mentioned in the previous section, we performed an experiment where we introduced a semantic decoder after each conv layer. The purpose was to induce semantics at each layer. In Figure 10 we show the prototype visualization corresponding to semantic clustering modules after the 2nd, 3rd and 4th conv layers. We demonstrate patches corresponding to the nearest cell features for 4 separate concepts, namely *upper part brown*, *red color bill*, *throat color blue*, and *crown color white*. We can see from the patches that the prototypes learned by our model have strong correspondence with the attributes of the CUB dataset and such correspondence can be easily induced at each layer of the network.

6.1 Failure Cases

We observe that imbalance among concepts is a challenge for VGCoL as our method relies on class-specific attribute information. For instance, the concept *quadra-pedal* (meaning walks on four legs) is present for most of the animal classes in the AWA2 dataset. The VGCoL model has to visually ground this attribute even if the legs are not visible in the image (as shown in Figure 6 (a)). This causes the VGCoL model to wrongly localize the spatial region for an imbalanced attribute such as *quadra-pedal*. Another failure case arises from some visually indiscernible attributes such as *new-world*, which is difficult to semantically align with a spatial region (shown in Figure 6 (b)). The *new-world* is also a high-occurring attribute that makes it difficult for VGCoL to ground this concept visually.

7 Data Pre-processing

For all our experiments, we resize the images to 84×84. We also use a simple data cropping strategy for CUB and AWA2 datasets where we use a pre-trained object detection model

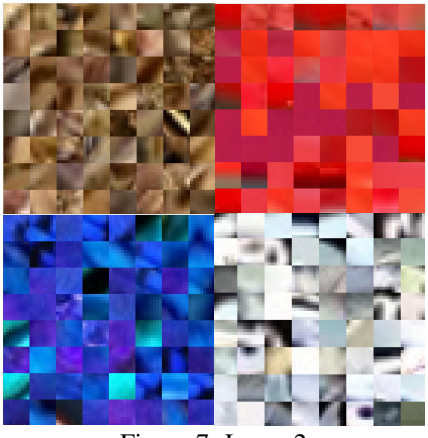

Figure 7: Layer 2

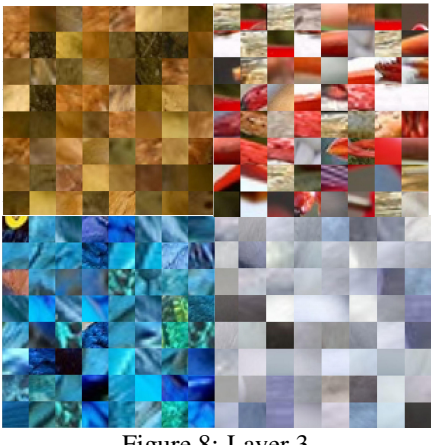

Figure 8: Layer 3

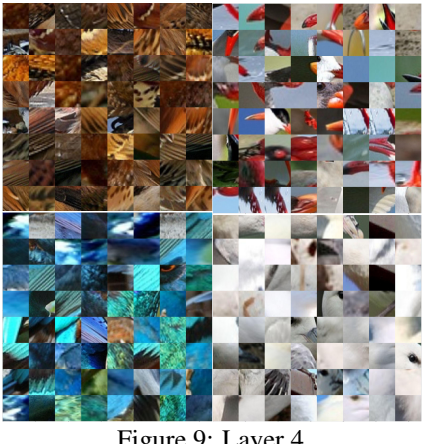

Figure 9: Layer 4

Figure 10: **Visualization of prototypes from semantic clustering module at different layers.** We show 4 prototypes and the patches nearest to them.

(YOLO_v5) [1] to extract the most dominant object (bird/animal) in the image. If no object is found by the detection model for a given image, then we keep the original figure. This is to ensure that the seen/unseen split has consistent samples. We observe that this data pre-processing helps in smooth convergence and better performance on the visual grounding of concepts. Once cropped, the images are resized to 84×84. For the semantic prototypes, we experiment with pre-trained word-vector embedding such as GloVe [2]. However, we observed no difference in performance. We speculate that since the cell features and semantic prototypes are in different spaces, initializing semantic prototypes with word-vector embedding is analogous to random initialization.

References

[1] Weijian Xu, Yifan xu, Huaijin Wang, and Zhuowen Tu. Attentional constellation nets for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[1] https://github.com/ultralytics/yolov5
[2] https://nlp.stanford.edu/projects/glove/