Evolving Interpretable Visual Classifiers with Large Language Models

1 Full Table

Table 1: We report all of the accuracies per dataset, for our method and baselines, across various class name types and with/without templates. * denotes the results that are averaged over the CLIP predetermined templates.

	iNaturalist					Kiki-Bouba	
Method	Lichen	Wrasse	Wild Rye	Manzanita	Bulrush	KB1	KB2
CLIP Class Name (S)	18.3	24.0	32.0	26.0	20.0	N/A	N/A
CLIP Class Name (C)	16.7	28.0	20.0	22.0	20.0	38.8	38.8
CLIP Class Name (S+C)	21.7	32.0	30.0	26.0	26.0	N/A	N/A
CLIP Class Name (S)*	21.7	14.0	24.0	20.0	16.0	N/A	N/A
CLIP Class Name (C)*	16.7	22.0	22.0	22.0	22.0	25.6	25.6
CLIP Class Name (S+C)*	23.3	20.0	24.0	18.0	16.0	N/A	N/A
Zero-shot Attributes	28.3	12.0	18.0	18.0	16.0	20.6	19.2
Zero-shot Attributes*	25.0	16.0	22.0	18.0	24.0	20.0	19.1
Classification by Desc (S).	20.0	30.0	24.0	26.0	16.0	N/A	N/A
Classification by Desc (C).	30.0	30.0	30.0	26.0	18.0	22.8	36.8
Classification by Desc (S+C).	26.7	26.0	36.0	22.0	18.0	N/A	N/A
Classification by Desc (S*).	18.3	34.0	26.0	28.0	14.0	N/A	N/A
Classification by Desc (C)*.	30.0	32.0	30.0	28.0	20.0	28.8	21.2
Classification by Desc (S+C)*.	28.3	30.0	34.0	24.0	18.0	N/A	N/A
Gradient-based Approach	21.7	20.0	40.0	18.0	18.0	16.7	55.6
Gradient-based Approach*	23.3	20.0	24.0	20.0	20.0	14.4	51.2
Ours	48.3	44.0	46.0	40.0	40.0	73.8	59.3
Ours*	33.3	42.0	58.0	42.0	42.0	79.2	49.4

1.1 Training Details

Pre-training Initialization To initialize the attribute bank for pre-training, we used the individuals words generated by prompting GPT3 with imagenet classes. This provided us with generally descriptive visual attributes, as the ImageNet classes spans daily life, from objects to animals. When initializing the pre-training classifier bank, we randomly initialized 1000 different groups of attributes that serve as binary classifiers with a threshold. We then prune this classifier bank by only keeping the top 50 binary classifiers, according to a metric that measures the difference in scores between the attributes for positive class images and the negative class images.

Batch Size and Objective Function We used a batch size of the entire training dataset ($\approx$ 300 images for iNaturalist images, and $\approx$ 800 images for the KikiBouba datasets), since it fit into memory and we weren't computing gradients. We used a temperature of 0.07 for the cross-entropy loss. We trained on eight A100 GPUs, for roughly three hours per class for pre-training, and three hours for joint-training. The same process is applied to the joint-training initializaiton, except the attribute bank for the classifier bank initialization is now the unique words that are generated during pre-training.

Parallelization In practice, we use continuous batching to generate b mutations per class, therefore only adding the best out of the $M \cdot b$ classifier mutations per class. To implemenent continuous batching with Llama-2-70-B, we use the library VLLM [1]. We repeat this process for 500 iterations, or until the process converges (the training accuracy stops increasing).

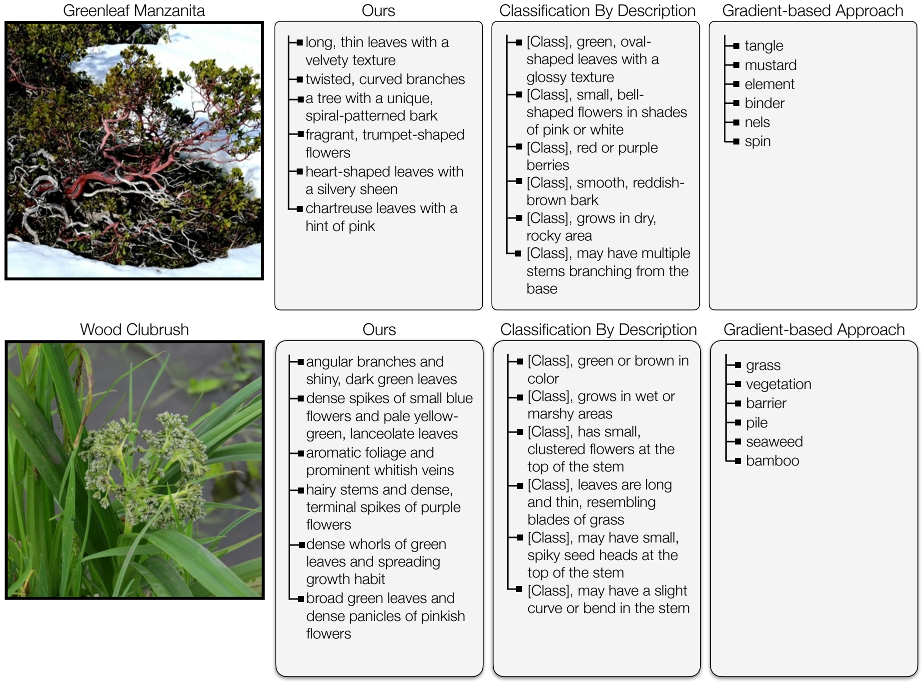
Prompting. Below is the prompt we used, followed by the ranked sets of attributes, along with their score:

"Here are some programs for class X. The programs are ranked according to average accuracy. We are playing a game of attribute discovery. Based on what you've seen below, propose a new program with diverse visual attributes that you think might achieve an even higher score. Please try to make new original attributes out of what you have seen, instead of just repeating."

1.2 Time Complexity

At each iteration of our algorithm, we generate a new set of attributes per class, b times, which each mutate the M sampled classifiers. This results in $O(b \cdot C)$ calls to the LLM, $O(b \cdot C)$ encodings of the generations, and $O(b \cdot C^2)$ multiplications within CLIP for the text-image similarity. As such, we opted for picking families of plants and animals that contained between five to six species for fine-grained classification. Further training details can be found in the supplementary.

1.3 Qualitative Results of our method, CBD, and Gradient-Based Approach



Ours

- long, thin leaves with a velvety texture
- twisted, curved branches
- a tree with a unique, spiral-patterned bark
- fragrant, trumpet-shaped flowers
- heart-shaped leaves with a silvery sheen
- chartreuse leaves with a hint of pink

Classification By Description

- [Class], green, oval-shaped leaves with a glossy texture
- [Class], small, bell-shaped flowers in shades of pink or white
- [Class], red or purple berries
- [Class], smooth, reddish-brown bark
- [Class], grows in dry, rocky area
- [Class], may have multiple stems branching from the base

Gradient-based Approach

- tangle
- mustard
- element
- binder
- nels
- spin

Ours

- angular branches and shiny, dark green leaves
- dense spikes of small blue flowers and pale yellow-green, lanceolate leaves
- aromatic foliage and prominent whitish veins
- hairy stems and dense, terminal spikes of purple flowers
- dense whorls of green leaves and spreading growth habit
- broad green leaves and dense panicles of pinkish flowers

Classification By Description

- [Class], green or brown in color
- [Class], grows in wet or marshy areas
- [Class], has small, clustered flowers at the top of the stem
- [Class], leaves are long and thin, resembling blades of grass
- [Class], may have small, spiky seed heads at the top of the stem
- [Class], may have a slight curve or bend in the stem

Gradient-based Approach

- grass
- vegetation
- barrier
- pile
- seaweed
- bamboo

Fig. 1: Comparison of Attributes by Method. We show qualitative examples of our learned attributes, classification by description's attributes (CBD), and our gradient-based approach attributes. CBD often produces reasonable attributes, but they are not discriminative, resulting in poor recognition accuracy. Gradient-based methods often produce poor attributes due to optimization difficulties.

References

1. W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, “Efficient memory management for large language model serving with pagedattention,” in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.