Learning to Predict Visual Attributes in the Wild – Supplementary Material

Khoi Pham[1] Kushal Kafle[2] Zhe Lin[2] Zhihong Ding[2] Scott Cohen[2]
Quan Tran[2] Abhinav Shrivastava[1]
[1]University of Maryland, College Park [2]Adobe Research
[1]{khoi,abhinav}@cs.umd.edu [2]{kkafle, zlin, zhding, scohen, qtran}@adobe.com

Contents

1. Example Predictions from our SCoNE Model

Figure 1 shows more qualitative results of our SCoNE algorithm together with the object localizer result as well as one (out of three total) attention maps from the multi-attention module. Throughout all examples, we can see that our model performs robustly for various attribute types. Our object localizer can correctly infer the object region and help alleviate the object occlusion problem, which is particularly a challenge for models that use global average pooling for feature aggregation. For example, in figure 2e, the object *table* is partially occluded by a lot of clutter which can distract a model that relies on global average pooling. Here, our object localizer clearly isolates the parts of *table*, making it easier to predict its attributes. However, object attributes are not always dependent on the foreground or surface features. Many attributes depend on the context, *e.g.*, *parked* vs. *running car*. Therefore, we designed our multi-attention module to complement the object localizer by *allowing* it to attend to image regions outside the object. This can be clearly seen in figure 2a, where attributes such as *sunny* or *bright* can be hard to infer by simply looking at the given object *tree*. Our multi-attention module looks at the sunny spots on the pavement which can help our model infer the presence of *sunny* and *bright* attributes. However, the multi-attention module is also free to attend to regions in the object to further supplement object localizer's attention to specific parts of object. For example, in figure 2d, our multi-attention module attends to the hind legs of *dog*, which supplements the object localizer's attention map and can provide additional information to help the model infer that the dog is *jumping*.

2. Additional Details for the VAW Dataset

In figure 2, we show examples of images and their attribute annotations from the VAW dataset. The images show both positive and negative annotations from our dataset as well as a subset of the result of our negative label expansion scheme, which is a rule-based system derived on the premise of mutual exclusivity of certain attributes. For example, if an object is annotated with positive attribute *empty*, the attribute *filled* can be auto-annotated as a negative attribute for the same object.

In figure 3, we show the distribution of top-15 attributes in various attribute categories arranged in descending order according to the number of available positive annotations. The diagram clearly shows the long-tailed nature of our VAW dataset, with some categories showing highly skewed distributions (*color, material*) and others have a more evenly balanced distribution (*texture, others*). For example, in the *material* category, the annotations for top-2 attributes (*metal* and *wooden*) consist of over 30.91% of total number of annotations (41.4% of positives and 23.5% of negatives). Reassuringly, our strong baseline as well as SCoNE model works almost equally well for more balanced categories (*e.g.*, *texture*) as well as a skewed category (*e.g.*, *material*).

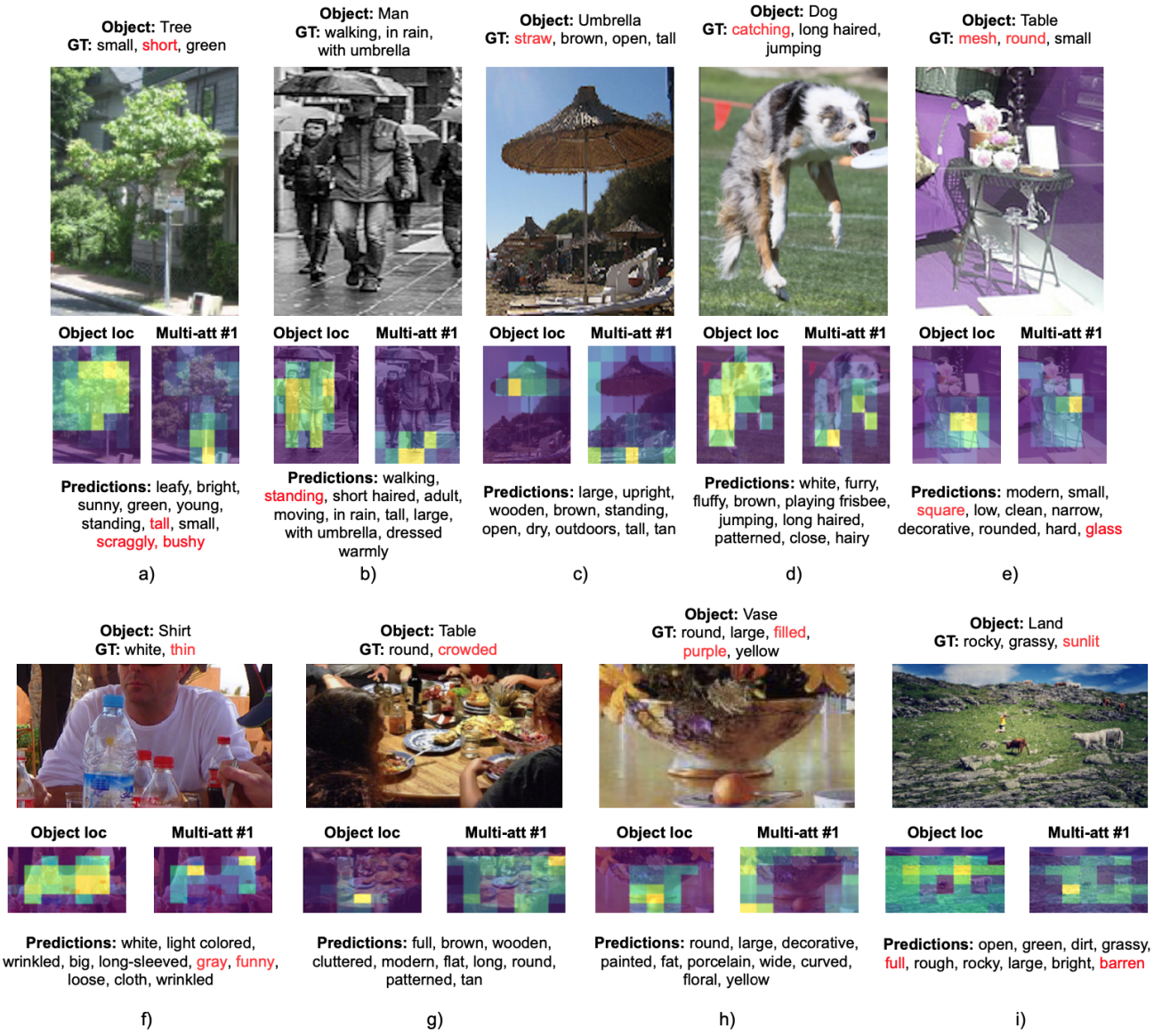

Figure 1. **Qualitative results.** Examples of predictions from SB+SCoNE. We show the object name and its ground truth positive attribute labels above the image. The object localized region, attention map #1, and model top-10 predictions are shown below. Red text represents missed or incorrect predictions.

3. Additional Ablation Studies

3.1. Study of different reweighting and resampling methods

Our VAW dataset, by nature, has a large amount of data imbalance which is further exacerbated after our negative label expansion. Hence, we also studied various reweighting and resampling techniques to tackle this issue. Here, we show results for different methods that we considered. These methods include: (1) Class-aware sampling (CAS) [13]: fill classes in a training batch as uniform as possible; (2) Inverse frequency (IF) [17]: assign weight of each class to be inversely proportional to its frequency (applied with smoothing factor $\alpha = 0.1$); (3) Class-balanced (CB) [1]: also a class-wise reweighting method similar to [17] but uses *effective number* instead of actual number of positives $E_n = (1 - \beta^n)/(1 - \beta)$ with $\beta = 0.999$ and n is the number of positives; (4) RW-BCE: our proposed reweighting scheme presented in section 4.4 in the main paper; ours is the only method among these that explicitly assigns different weight for the positive and negative label of every class (5) Repeat factor sampling (RFS) [9, 4]: a resampling trick that oversamples images that contain the tail classes.

The results are reported in table 1. All these techniques

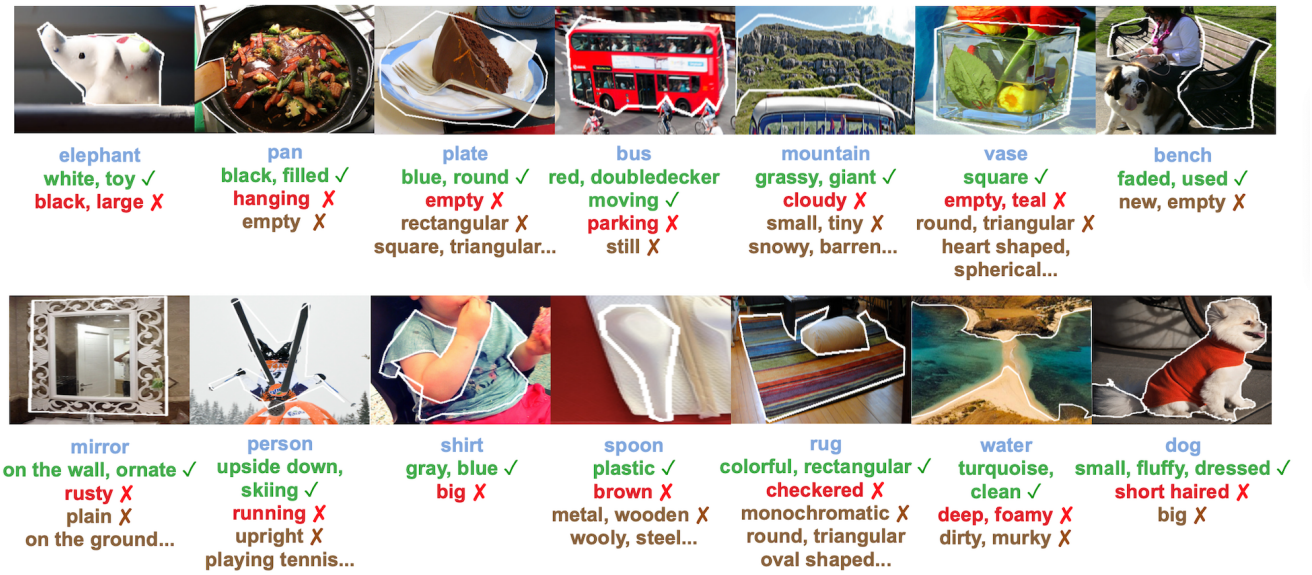

Figure 2. Examples of images and their annotations from the VAW dataset. Object names, positive attributes, explicitly labeled negative attributes, and negative labels from our negative label expansion are shown in corresponding colors for each example.

Figure 3. Distribution of positive and negative annotations for attributes in different categories. We show the top-15 attributes with the most number of positive annotations in each category sorted in descending order.

are implemented on top of the ResNet-Baseline model and trained on the training data after negative label expansion.

CAS achieves low mAP score while still having decent mR@15 and mA. Because the VAW dataset is extremely imbalanced, applying CAS can lead to severe undersampling (or oversampling) of the head (or tail) classes. In addition, since CAS maintains a uniform distribution of classes in a training batch, no classes are dominant by the others as well as the negative examples do not dominate the positive examples. Hence, CAS still achieves good mean recall and

Methods (+negative label expansion)	mAP	mR@15	mA	F1@15
ResNet-Baseline	65.6	53.8	69.4	68.6
+ Class-aware sampling (CAS) [13]	63.5	56.6	70.2	65.4
+ Class-balanced (CB) [1]	65.7	54.7	69.6	68.4
+ Inverse frequency (IF) [17]	65.8	54.8	69.6	68.4
+ RW-BCE	**66.0**	56.1	70.3	**68.8**
+ Repeat factor sampling (RFS) [9] [4]	65.6	55.2	70.0	68.2
+ RW-BCE + RFS	**66.0**	**57.0**	**70.6**	**68.8**

Table 1. Investigation of different reweighting and resampling methods.

mean accuracy.

IF and CB both assign higher weights for tail classes and achieve better results across all metrics over the baseline. Our formulation, RW-BCE, aims at (1) mitigating the over-suppression and rarity of negative examples and (2) highlighting the rare classes using the same weighting as in IF, hence, it achieves better results in all metrics over IF and CB. Finally, RFS is a resampling trick that does not rely on undersampling the head classes, thereby addressing one of the weaknesses of CAS, resulting in better performance in the VAW dataset. Because RFS is a sampling technique, it can be used in conjunction with any reweighting methods. Therefore, we use RW-BCE along with RFS (referred as RR) in our main paper whose results are better than the others across most metrics as shown in table 1.

3.2. Components of the Strong Baseline

In the main paper, we presented ablations for our SB+SCoNE model, which is comprised of Strong Baseline, Negative label Expansion, and Supervised Contrastive Learning. However, Strong Baseline itself is comprised of many sub-components which extends the ResNet-Baseline: the object localizer, the multi-attention module, and the usage of low-level features. In this section, we will dissect how each of these components affect our Strong Baseline model.

We ablate our Strong Baseline model with each component and train on our training data after negative label expansion. We report results in table 2.

Removing each sub-component has a negative effect on the performance of the Strong Baseline model. For example, removing the use of low-level features not only lowers mAP in *color* and *material* attributes but it also lowers it for higher-level attributes (*e.g.*, *action*). This is likely due to the absence of clearly defined low- and high-level features, which forces a 'single' feature to represent both low- and high-level features. This adversely affects the network's ability to learn high-level attributes (*e.g.*, *action*) as well as low-level (*color, texture*), thus lowering performance for both.

Interestingly, removing the object localizer does not result in a drastically diminished performance. Visualizing the multi-attention output of our full model (Fig. 2) reveals that even without object mask supervision, the model is still able to differentiate between object and background/distractors with the multi-attention maps which are trained with weak supervision from the attribute labels. However, removing all components, which is devoid of any form of attention, severely hampers model performance across all categories.

In general, all sub-components are necessary for our model to perform well across different attribute types.

3.3. Interaction of SupCon and Attention module

As presented in the main paper, using SupCon as a pretraining scheme can be at odds with the attention module. To investigate this issue, we compare between the following models that are trained on the training set after negative label expansion: (1) Strong Baseline, (2) Strong Baseline + SupCon pretraining, (3) Strong Baseline without multi-attention, (4) Strong Baseline without multi-attention + SupCon pretraining, and (5) Strong Baseline with SupCon joint training. The results are reported in table 3.

From the results, we can see that SupCon pretrained helps improve model performance for our strong baseline model variant without multi-attention. This clearly shows that SupCon is an effective technique. However, we can also see that the mAP score drops when using SupCon pretraining for the unmodified Strong Baseline model, which consists of multi-attention module. We conjecture that SupCon pretraining being incompatible with multi-attention is largely because SupCon pretraining uses global average pooling (GAP) for feature aggregation which encourages the feature extractor to ignore the surrounding context (the majority of attribute features lie on the object foreground which is in the image center), whereas the multi-attention module aims to detect features at different locations including the surrounding.

To alleviate this issue, we jointly train the supervised contrastive loss with our whole model. Results from table 3 shows that SupCon joint training no longer experiences the above problem while improves almost all overall metrics. The benefit of SupCon joint training is even more evident in the tail classes.

Supervised contrastive learning is still a new learning approach with very few exploration in the community. We believe there will be better ways to incorporate supervised contrastive learning in a multi-label setting such as ours.

4. Evaluation Metrics

In this section, we present details about the different evaluation metrics that we use. We have used mAP as our primary metric, since it describes the quality of the model to rank correct images higher than the incorrect ones for each attribute label. mR@15 is also important as it shows how well the model manages to output the ground truth positive

Methods (+Neg)	Overall				Class imbalance (mAP)			Attribute types (mAP)					
	mAP	mR@15	mA	F1@15	Head	Medium	Tail	Color	Material	Shape	Texture	Action	Others
Strong Baseline	67.7	54.3	70.0	69.6	75.9	64.3	46.9	68.8	73.9	67.0	69.4	60.2	69.1
w/o Multi-attention (MA)	67.4	53.5	69.7	69.7	75.9	63.8	46.4	67.8	74.7	66.9	68.5	58.0	69.0
w/o Low-level feature (LL)	67.3	53.7	69.9	69.4	75.4	63.8	48.4	68.5	73.6	66.1	67.5	59.3	68.9
w/o Object localizer (OL)	66.9	53.1	69.6	69.1	75.3	63.4	45.5	67.5	73.8	66.5	68.4	58.9	68.3
w/o OL, MA and LL	65.6	53.8	69.4	68.6	74.8	62.3	43.2	67.3	73.3	66.3	67.7	56.0	67.4

Table 2. Ablation study on the three components of the Strong Baseline model by removing each one. The last row also corresponds to the ResNet-Baseline model.

Methods (+Neg)	Overall				Class imbalance (mAP)			Attribute types (mAP)					
	mAP	mR@15	mA	OV-F1	Head	Medium	Tail	Color	Material	Shape	Texture	Action	Others
Strong Baseline	67.7	54.3	70.0	69.6	75.9	64.3	46.9	68.8	73.9	67.0	69.4	60.2	69.1
+ SupCon pretraining	67.3	54.8	70.0	69.5	75.7	63.8	45.5	67.8	73.1	66.8	69.2	59.6	68.8
+ SupCon joint training	68.2	55.2	70.3	70.0	76.1	64.7	47.8	69.1	75.0	67.3	69.8	60.0	69.4
SB w/o Multi-attention	67.4	53.5	69.7	69.7	75.9	63.8	46.4	67.8	74.7	66.9	68.5	58.0	69.0
+ SupCon pretraining	67.6	54.1	69.7	69.8	75.9	63.9	46.6	67.7	75.0	67.0	68.3	57.7	69.1

Table 3. Experiments to show the incompatibility between SupCon pretraining and multi-attention used by both our Strong Baseline and SCoNE model. The top section shows results that accuracy decreases when using SupCon pretraining with multi-attention in Strong Baseline model, which can be alleviated by switching to jointly training. The bottom section shows that SupCon pretraining works well on its own when multi-attention is not being used.

attributes in its top 15 predictions in each image. In addition, mA and F1@15 can also be used to evaluate model performance in a different light.

mAP: similar to [16], the mAP score is computed by taking the mean of the average precision of all C classes

$$mAP = \frac{1}{C}\sum_{c} AP_c, \tag{1}$$

in which the average precision of each class is computed as

$$AP_c = \frac{1}{P_c}\sum_{k=1}^{P_c} \text{Precision}(k, c) \cdot \text{rel}(k, c), \tag{2}$$

where P_c is the number of positive examples of class c, $\text{Precision}(k, c)$ is the precision of class c when retrieving the best k images, $\text{rel}(k, c)$ is the indicator function that returns 1 if class c is a ground-truth positive annotation of the image at rank k. Note that due to VAW being partially labeled, we compute this metric only on the annotated data similar as in [16]. This evaluation scheme is also similar to what is used in [4], where the authors introduce the definition of federated dataset. In this federated dataset setup, we only need for each label a positive and a negative set, then average precision for each label can be computed on these 2 sets.

mA: as in [8, 15], we compute the mean balanced accuracy (mA) to evaluate all models in a classification setting, using 0.5 as threshold between positive and negative prediction. Because our dataset is highly unbalanced between the number of positive and negative examples for some attributes, balanced accuracy is a good metric as it calculates separately the accuracy of positive and negative examples then take the average of them. In concrete, the mA score can be computed as follows

$$mA = \frac{1}{C}\sum_{c}\Big(\frac{TP_c}{P_c} + \frac{TN_c}{N_c}\Big)/2, \tag{3}$$

where C is the number of attribute classes, P_c and TP_c are the number of positive examples and true positive predictions of class c, and N_c and TN_c are defined similarly for the negative examples and predictions. Because mA uses threshold 0.5, models that are not well-balanced between positive and negative prediction tend to receive low score.

mR@15 and F1@15: we follow [3] to compute the precision, recall and F1 score. For each image, we consider the top 15 predictions of the model as its positive predictions. These predictions are then compared with the ground-truth annotations to compute the metrics. Because VAW dataset is partially labeled, we only consider the predictions of labels that have been annotated, *i.e.* if class c is predicted on an image but that image is unannotated for class c, then the prediction is ignored. The overall precision and recall are computed as follows

$$\text{OV-Precision} = \frac{\sum_c TP_c}{\sum_c N_c^p}, \ \text{OV-Recall} = \frac{\sum_c TP_c}{\sum_c P_c}, \tag{4}$$

where TP_c is the number of true positives for attribute class c, N_c^p is the number of positive predictions of class c, and P_c is the number of ground truth positive examples of class c. With the same notations, the per-class precision and recall

are computed as

$$\text{PC-Precision} = \frac{1}{C}\sum_c \frac{TP_c}{N_c^p},\ \text{PC-Recall} = \frac{1}{C}\sum_c \frac{TP_c}{P_c}. \tag{5}$$

The F1 score is the harmonic mean of precision and recall, which is defined as

$$\text{OV-F1} = \frac{2 \times \text{OV-Precision} \times \text{OV-Recall}}{\text{OV-Precision} + \text{OV-Recall}}, \tag{6}$$

$$\text{PC-F1} = \frac{2 \times \text{PC-Precision} \times \text{PC-Recall}}{\text{PC-Precision} + \text{PC-Recall}}. \tag{7}$$

In our paper, we report the per-class recall and overall F1 score, and we refer to them respectively as mR@15 and F1@15 throughout the text and our tables.

5. Implementation Details

We use the ImageNet-pretrained [2] ResNet-50 [6] as the feature extractor, and use the output feature maps from ResNet block 2 and 3 as low-level features. For the object name embedding, we use the pretrained GloVe [12] 100-d word embeddings. We do not finetune these word embeddings during training as we want our model to generalize to unseen objects during test time.

We implement our model in PyTorch [11] and train using Adam optimizer with the default setting, batch size 64, weight decay of $1e-5$, an initial learning rate of $1e-5$ for the pretrained ResNet and 0.0007 for the rest of the model. We train for 12 epochs and apply learning rate decay of 0.1 every time the mAP on the validation set stops improving for 2 epochs. We use image size 224x224 as input and basic image augmentations which include random cropping around object bounding box, random grayscale when an instance is not labeled with any color attributes, minor color jittering, and horizontal flipping. For each object bounding box in the dataset, we expand its width and height by $\min(w, h) \times 0.3$ to capture more context. For the hyperparameters, we set $\lambda_{\text{fg}} = 0.25, \lambda_{\text{div}} = 0.004$. In the multi-attention module, we select $D_{\text{proj}} = 128$ and use $M = 3$ attention maps. Regarding reweighting and resampling, we use $t = 0.0006$ for RFS and $\alpha = 0.1$ for smoothing in the RW-BCE reweighting terms.

For SupCon pretraining, we pretrain on top of ImageNet-pretrained ResNet for 10 epochs with batch size 384 (768 views per batch), and initialize all matrices A_c with the identity matrix. In the contrastive loss, we set temperature $\tau = 0.25$. We believe using a larger batch size will greatly benefit supervised contrastive pretraining as suggested by the authors [7]. For SupCon joint training with the other losses of the Strong Baseline model, we keep batch size as 64, we add $\lambda_{\text{sup}}\mathcal{L}_{\text{sup}}$ to the loss where $\lambda_{\text{sup}} = 0.5$, and all other hyperparameters are the same as above.

6. Additional Details for Negative Label Expansion

We classify the attributes into types and construct their overlapping and exclusive relations using existing ontology from a related work [5], WordNet [10], and the relation edges from ConceptNetAPI [14]. Specifically:

- Attribute categories: are automatically derived from WordNet hypernyms and ConceptNetAPI *IsA* relation edge. These are then manually verified.
- Overlapping relations: from WordNet, we check if two attributes share the same synset (*e.g.*, *muddy* and *dirty* share WordNet synset *dirty.s.06*). From ConceptNetAPI, we use the following relation edges: *Synonym, SimilarTo, DerivedFrom*.
- Exclusive relations: From WordNet, we use antonyms retrieved from the synsets' lemmas. From ConceptNetAPI, we use the following relations: *Antonym, DistinctFrom*.

7. Image Search Results from our SCoNE Model

We show from Figure 4 to Figure 8 our image search (ranking) results when searching for specific attributes. Our model is able to search for images that exhibit one to multiple attributes, as demonstrated in Figure 5 where we search for multiple colors at a time. In addition, the results in Figure 8 also show that our model is able to differentiate between objects with different size (*e.g.*, *small* vs. *large bird*, *small* vs. *large phone*).

References

[1] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 2, 4

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[3] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013. 5

[4] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 2, 4, 5

[5] Chi Han, Jiayuan Mao, Chuang Gan, Joshua B. Tenenbaum, and Jiajun Wu. Visual Concept Metaconcept Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 6

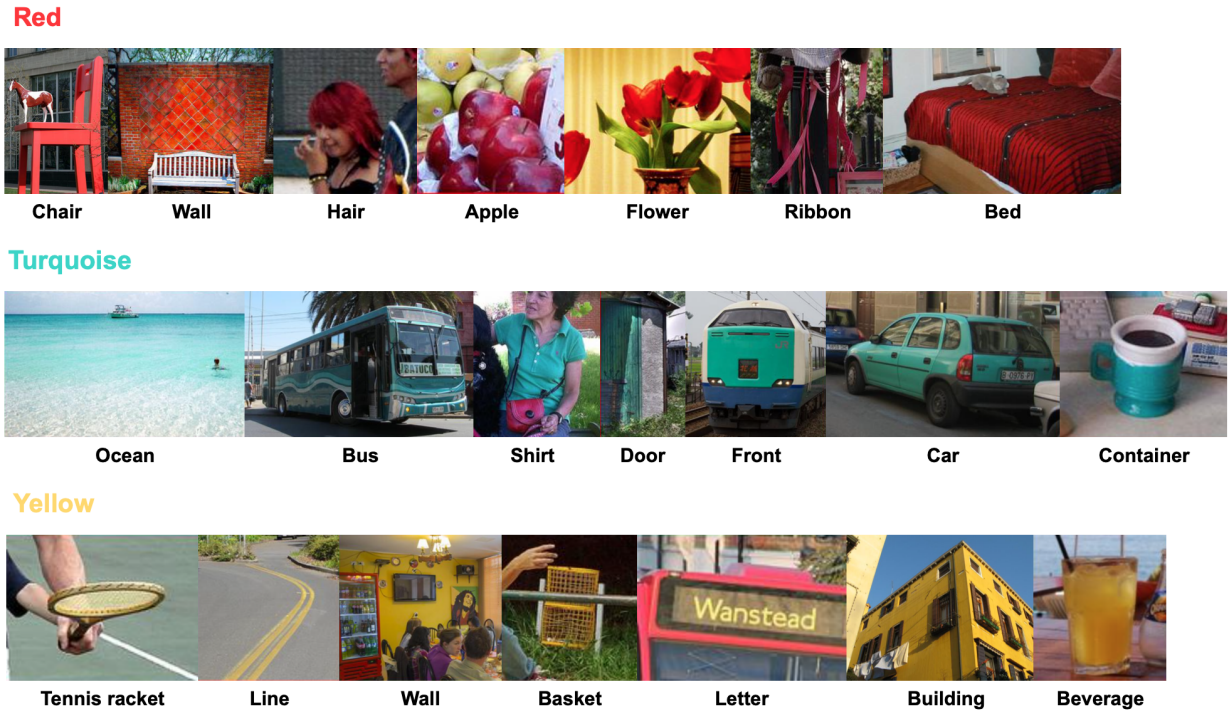

Figure 4. **Image search results.** We show the top retrieved images of SB+SCoNE when searching for some *color* attributes.

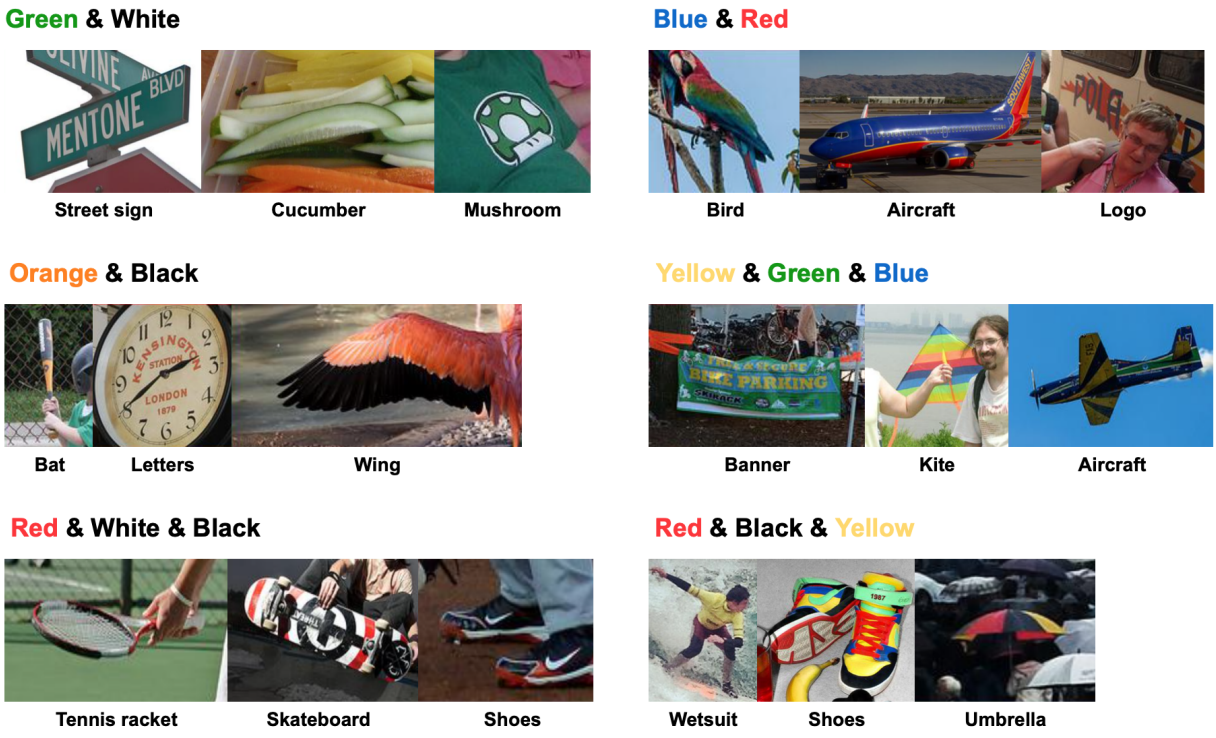

Figure 5. **Image search results.** We show the top retrieved images of SB+SCoNE when searching for images that exhibit multiple *color* attributes.

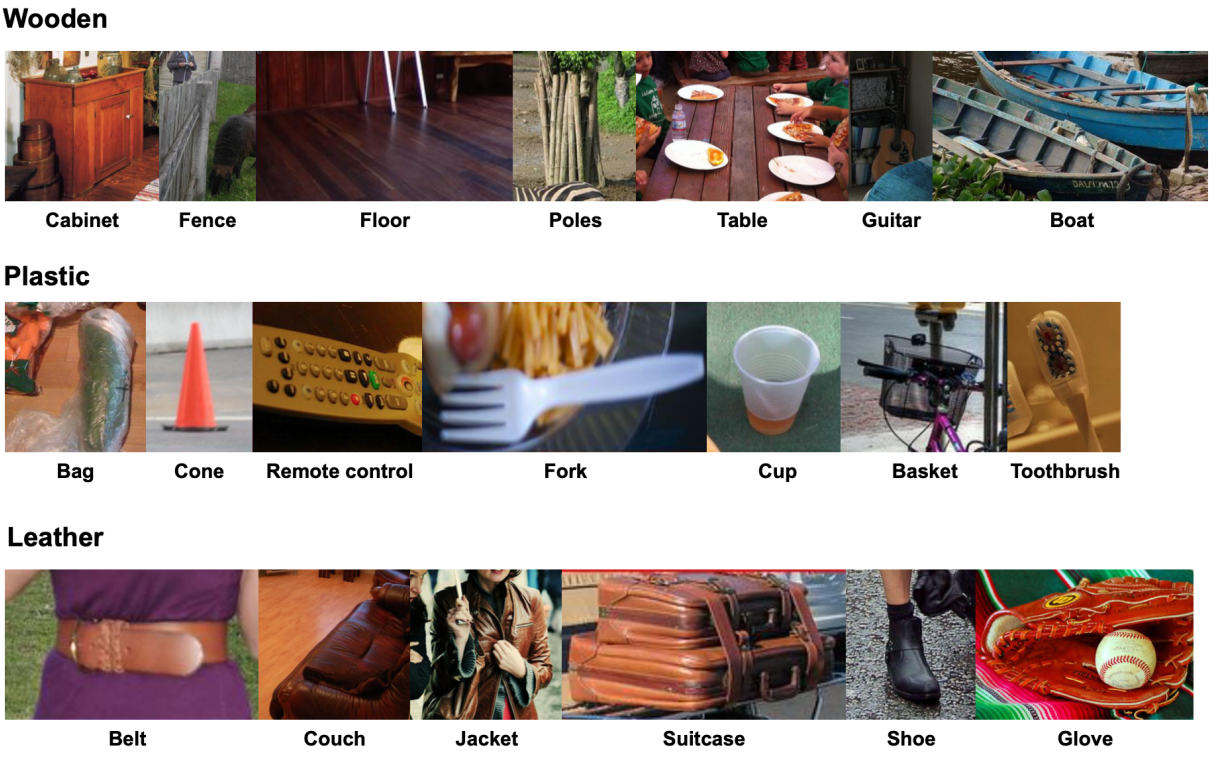

Figure 6. **Image search results.** We show the top retrieved images of SB+SCoNE when searching for some *material* attributes.

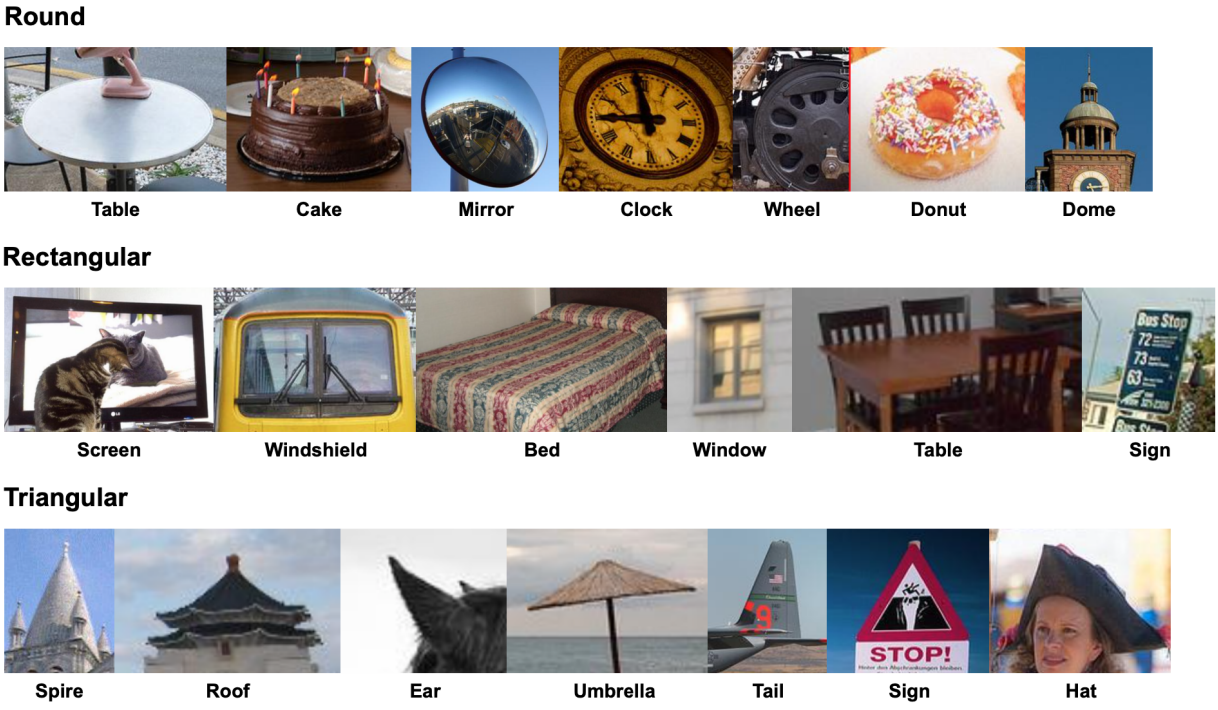

Figure 7. **Image search results.** We show the top retrieved images of SB+SCoNE when searching for some *shape* attributes.

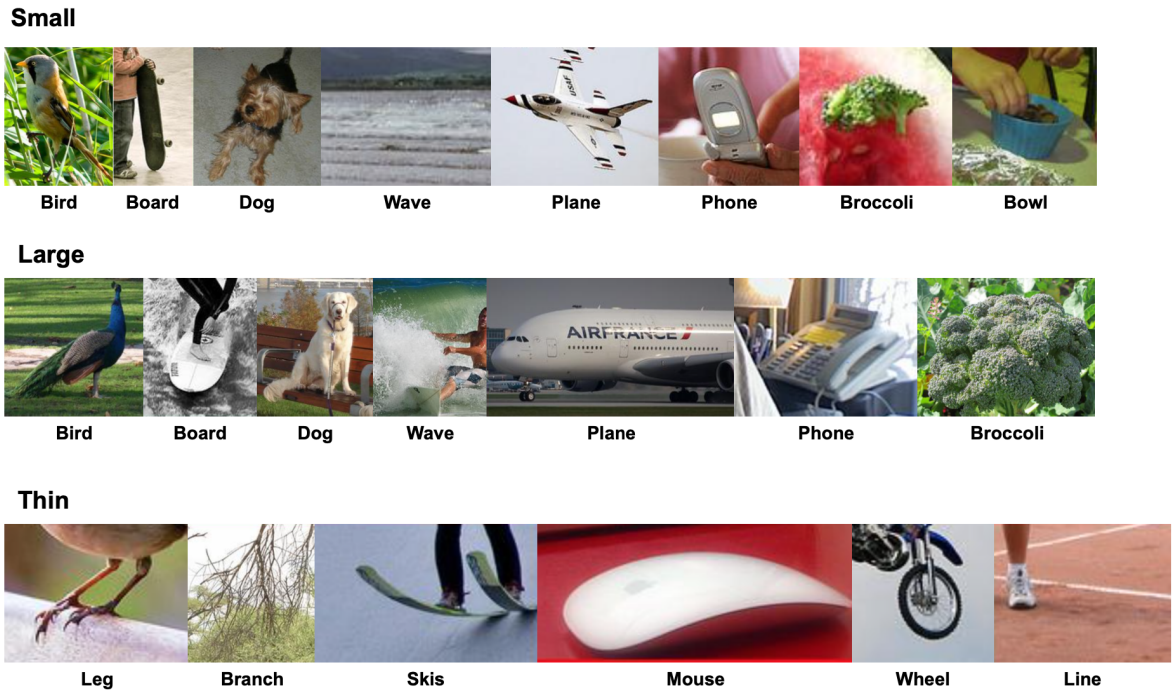

Figure 8. **Image search results.** We show the top retrieved images of SB+SCoNE when searching for some *size* attributes. We deliberately show the same object categories between the 1st and 2nd row to show how our model is able to make distinction between a *small bird* vs. *large bird*, *small plane* vs. *large plane*, *etc*.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[7] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 6

[8] Dangwei Li, Zhang Zhang, Xiaotang Chen, Haibin Ling, and Kaiqi Huang. A richly annotated dataset for pedestrian attribute recognition. *arXiv preprint arXiv:1603.07054*, 2016. 5

[9] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 2, 4

[10] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995. 6

[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 6

[13] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. 2, 4

[14] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *arXiv preprint arXiv:1612.03975*, 2016. 6

[15] Chufeng Tang, Lu Sheng, Zhaoxiang Zhang, and Xiaolin Hu. Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4997–5006, 2019. 5

[16] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847, 2017. 5

[17] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017. 2, 4