MCPNet: An Interpretable Classifier via Multi-Level Concept Prototypes

Supplementary Material

6. Layer selections

Layer	ResNet50	Inception v3	ConvNeXt tiny
1	layer1	Conv2d_4a_3x3	stages[0]
2	layer2	Mixed_6a	stages[1]
3	layer3	Mixed_7a	stages[2]
4	layer4	Mixed_7c	stages[3]

Table 6. Selected layers for the three backbone models used in this paper.

In Table 6, we present the selected layers for ResNet50, InceptionV3, and ConvNeXt-tiny, with the layer names following the PyTorch code. In the case of ResNet50 and ConvNeXt, the selected layers are based on the original architecture design, where each layer corresponds to one convolutional block. For InceptionV3, we partition the model into four parts, treating each as a distinct layer.

7. Training process

Our complete training workflow is depicted in Algorithm 1. We initiate the process by extracting the multi-level concept prototypes (MCP) and class-specific multi-level concept prototype distributions (MCP distribution), where the backbone model is initialized by using the weights pre-trained on ImageNet. The term "CalMCP" in Lines 2 and 11 of Algorithm 1 denotes the process of computing the MCP using weighted Principal Component Analysis (PCA) to capture the global semantic information from the segments, as detailed in Section 3.3. The term "CalMCPDist" in Lines 3 and 12 of Algorithm 1 represents the procedure of averaging the MCP distribution across images within the same class, yielding the class-specific MCP distribution that encapsulates the predominant concept distribution, as explained in Section 3.4. Throughout the training process, these extracted MCP and MCP distributions are employed to calculate the loss and we update them every epoch to incorporate the latest features learned by the model.

8. More benchmarks

In Table 8, we present additional performance comparisons between our Multi-Level Concept Prototypes Classifier (MCPNet) and the ProtoPNet series methods [2, 14, 15, 29] across three datasets with different backbone architectures. We retrained all models using their source code (i.e. their models are also pretrained on ImageNet dataset with batch size set to 64, following the same setting as ours) and full images (i.e. no cropping) for fair comparisons. The results indicate that MCPNet achieves comparable performance in various scenarios by offering multi-level explanations, thereby providing a more comprehensive understanding of the deep-learning black box.

Please kindly note that, the results of our retraining ProtoPNet series methods on ResNet50 differ from the reported results in their papers since their original experimental settings adopt the pretraining on iNaturalist 2017 [26] with some of them utilizing cropped images or different batch size during training. The notably low performance of PIP-Net [15] on ResNet34 may be attributed to their method not being optimized for a smaller model (while the training of ConvNeXt based on our experimental setting does achieve the similar result as in the original PIP-Net [15] paper, hence verifying the correctness of our implementation).

Algorithm 1: Training a MCPNet

Input: Training set $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}, (x_1^{\mathcal{T}}, y_1^{\mathcal{T}}) = \mathcal{T}_1$, validation set $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_n\}, (x_1^{\mathcal{V}}, y_1^{\mathcal{V}}) = \mathcal{V}_1$, Epochs E, Concept segment size C'.

1 Initial model f weight w_f with pre-trained on ImageNet and remove the fully connected layer.

/* Compute the multi-level concept prototypes (MCP). Mentioned in Section 3.3. */

2 $\mathcal{MCP}$ = CalMCP(f, $\mathcal{T}$);

/* Compute the class-specific MCP distribution $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^{y_n}\}$. Mentioned in Section 3.4. */

3 $\mathcal{D}$ = CalMCPDist($\mathcal{T}$, $\mathcal{MCP}$);

4 **for** *epoch* $\in$ {*1,..., E*} **do**

/* Train phase */

5 **for** $(x_b, y_b) \in \{\mathcal{T}_1, ..., \mathcal{T}_n\}$ **do**

6 $F_1, F_2, F_3, F_4 = f(x_b)$

7 split segments $S_l \leftarrow F_l, l \in \{1, 2, 3, 4\}$

8 compute loss $\mathcal{L}^{\mathbf{CKA}}(S_l)$

9 compute loss $\mathcal{L}^{\mathbf{CCD}}(x_b)$ with class-specific MCP distribution $\mathcal{D}$

10 Minimize losses by updating w_f

/* Re-calculate MCP */

11 $\mathcal{MCP}$ = CalMCP(f, $\mathcal{T}$)

/* Re-calculate class-specific MCP distribution */

12 $\mathcal{D}$ = CalMCPDist($\mathcal{T}$, $\mathcal{MCP}$)

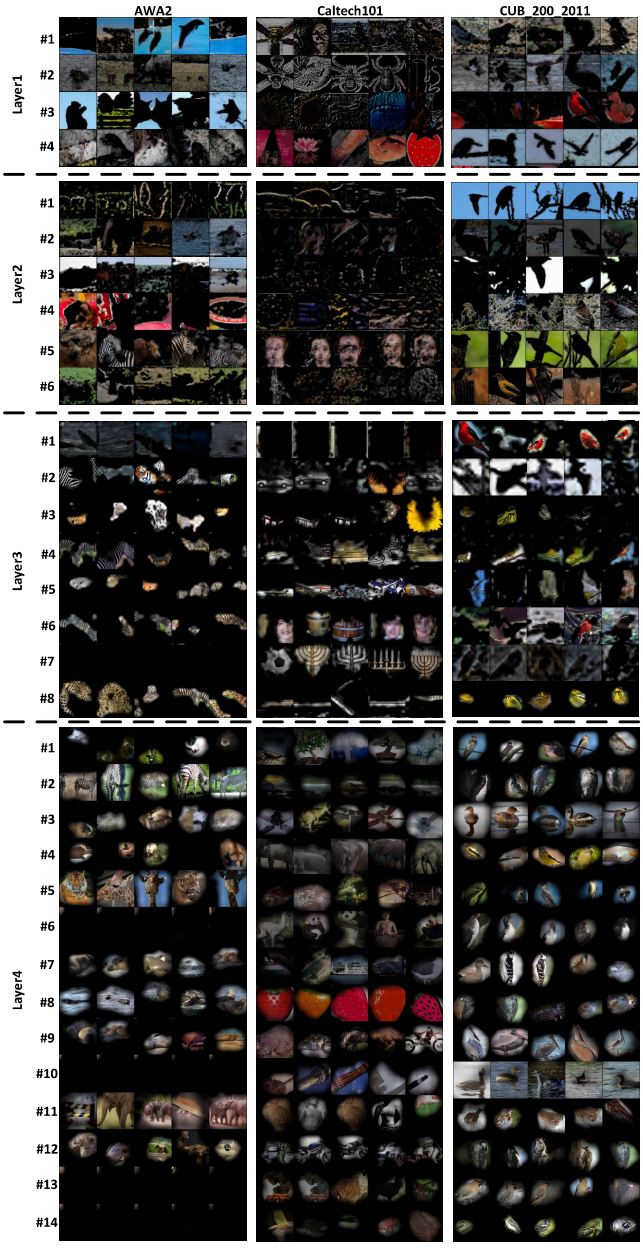

Figure 7. Multi-level concept prototype samples on three datasets. (Backbone: ResNet50)

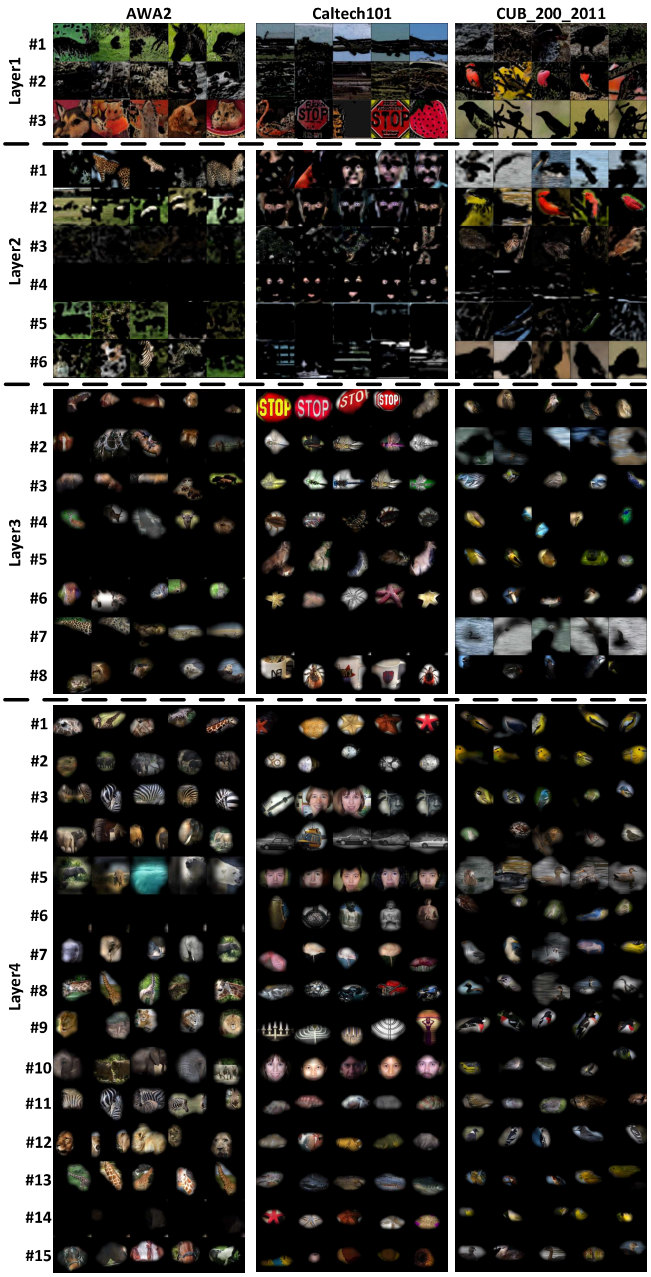

Figure 8. Multi-level concept prototype samples on three datasets. (Backbone: Inception V3)

Regarding ProtoTree [14] on CUB_200_2011, the performance appears to be particularly sensitive to pretraining on iNaturalist 2017, as we could not achieve satisfactory results with pretraining on ImageNet alone. It is worth noting that the original paper of ProtoTree [14] only provides quantitative results for ResNet50.

We present the performance comparison with the ProtoPNet series methods on the CUB_200_2011 dataset in Table 7, where ResNet50 is adopted as backbone while being now pretrained on the iNaturalist 2017 dataset (i.e. in comparison to the results in Table 8 where the pretraining is based on ImageNet dataset, here we follow the common setting in most of ProtoPNet series methods to have ResNet50 pretrained on iNaturalist 2017 dataset). We can observe that, even with different pretrained weights for the ResNet50 backbone, our proposed MCPNet manages to maintain performance that is on par with other methods.

Large-scale dataset We also evaluate our model on the large-scale dataset, denoated as **sampled ImageNet-1K**,

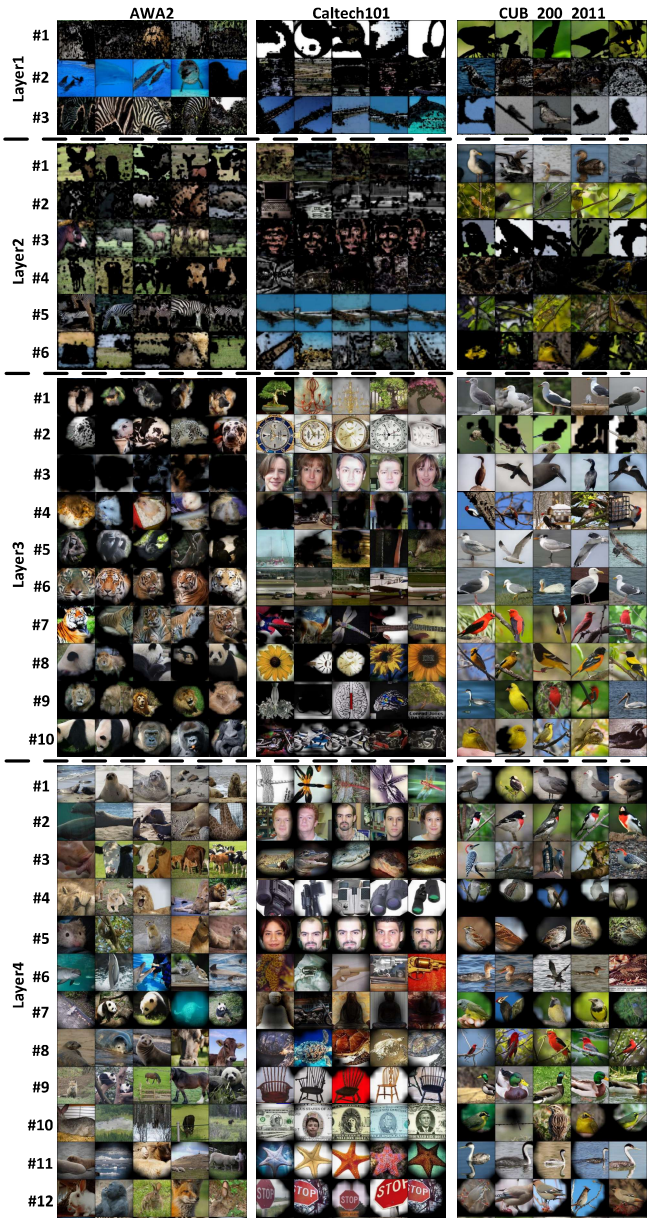

Figure 9. Multi-level concept prototype samples on three datasets. (Backbone: ConvNeXt-tiny)

which is built by randomly sampling one-tenth of the images for each category in the ImageNet training set while keeping the full ImageNet evaluation set. As shown by the results summarized in Table 7, our method is capable of learning discriminative MCP distributions and meaningful MCPs on such large-scale dataset. In Figure 11, we also provide example visualizations of multi-scale prototypes learnt by MCPNet on the large-scale sampled ImageNet-1K dataset. It is worth noting that, while most previous prototype-based methods focus on learning class-specific prototypes with often having their inter-class prototypes enforced to be orthogonal, such orthogonality becomes quite challenging to maintain when the number of classes increases thus typically leading to diminished performance.

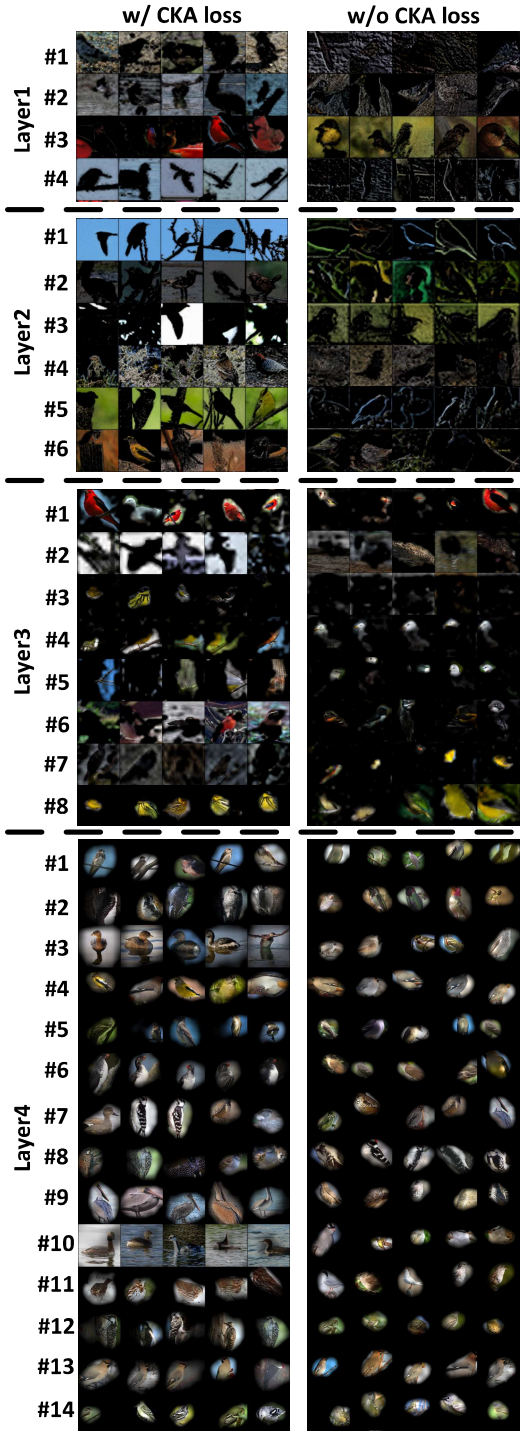

Figure 10. More samples of w/ and w/o CKA loss(Backbone: ResNet50; Dataset: CUB_200_2011).

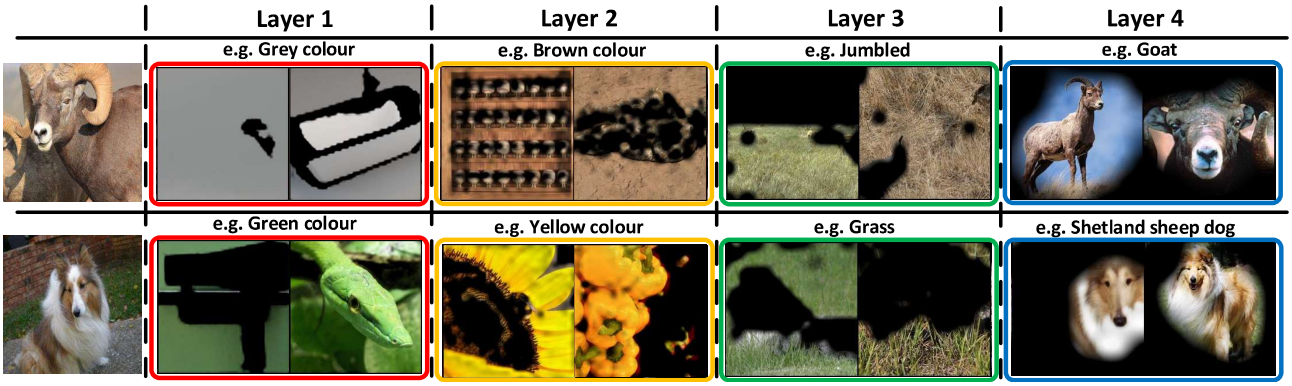

Figure 11. The interpretation of images (leftmost column) based on the multi-scale prototypes (extracted on different layers of the classification model) learnt from the large-scale sampled ImageNet-1K dataset by our MCPNet (using ResNet50 as the backbone), e.g. identifying colors such as white and green from the lower-level layers, and recognizing entities like goats and Shetland sheepdogs from the higher-level layers.

Method	CUB (iNaturalist)	Sampled ImageNet-1K (None)
ProtoTree [13]	77.22%	9.07%
Deformable ProtoPNet [2]	85.66%	1.48%
ST-PrototPNet [28]	87.63%	58.15%
PIP-Net [14]	82.33%	0.10%
MCPNet (Ours)	86.28%	61.73%

Table 7. We here provide two experimental results: 1) The middle column reveals how MCPNet (as well as the ProtoPNet series methods), while the backbone is pretrained on the iNaturalist dataset, performs on the CUB dataset; 2) The rightmost column details the performance when various models are trained on a sampled subset of ImageNet-1K and then evaluated on the entire ImageNet-1K validation set. (Backbone : ResNet50).

9. Prototype Visualizations

To grasp the meaningful concepts represented by the concept prototypes in MCPNet, we visualize each prototype with the top-5 response images from the dataset, as illustrated in Figure 7, Figure 8, and Figure 9 (respectively for the backbones of ResNet50, Inception V3, and ConvNeXt-tiny). Each row of images corresponds to a single concept prototype, showcasing prototypes from multiple levels that represent explanations of different scales for the model.

It is noteworthy that *null* concept prototypes exist in some experiments, as exemplified by the $\#7$ prototype in layer 3 and the $\#6, \#10, \#13, \#14$ prototypes in layer 4 from AWA2, which are shown in Figure 7. The presence of *null* concepts may suggest that the model has learned a sufficient number of concept prototypes, which is fewer than our predefined number of concepts in that layer, to effectively perform the image classification. Moreover, as the dataset complexity increases, concept prototypes are encouraged to learn more discriminative concepts for accurate image classification thus leading to the decrease for the number of *null* prototypes, as observed in the concept prototypes from Caltech101 and CUB_200_2011 in Figure 7.

9.1. The effect of CKA loss

To assess the disentangling effect of the Centered Kernel Alignment (CKA) loss, we observe that, despite a slight performance degradation when combined with the Class-aware Concept Distribution (CCD) loss, the disentanglement among concept prototypes becomes more significant due to the presence of the CKA loss, as illustrated in Figure 10. In the absence of the CKA loss, prototypes such as $\#1$ and $\#5$ in layer 2 seem to repeatedly learn the meaning of edges, while prototypes like $\#4$ and $\#5$ in layer 3 repeatedly focus on the bird's head with white color (i.e. there are prototypes being less disentangled).

10. More explanation samples

10.1. Image-level explanations

In Figure 12, we illustrate how MCPNet classifies an image by the MCP distribution. We compute the MCP distribution for each image, reflecting the presence of concept prototypes. By aligning the image's MCP distribution with the class-specific MCP distribution, which signifies the strength of concept prototypes across the majority of images in the same class, the image is classified into the category most resembling its concept prototype strength. Beyond image classification, the MCP distribution is also employed to generate explanations for individual instances, as depicted in Figure 13. Each computed concept prototype strength signifies the identified concept prototype on the image. Analyzing the strength of each concept prototype unveils how the model arrives at the classification decision for each image.

Backbone	Methods	Explanation	Accuracy		
			AWA2	Caltech101	CUB_200_2011
ResNet34	Baseline	N/A	94.60%	94.78%	77.89%
	ProtoTree [14]	Single-Scale	90.33%	77.71%	15.79%
	Deformable ProtoPNet [2]	Single-Scale	90.55%	95.95%	74.53%
	ST-ProtoPNet [29]	Single-Scale	93.56%	96.24%	77.65%
	PIP-Net [15]	Single-Scale	8.00%	44.52%	7.65%
	MCPNet (Ours)	Multi-Scale	93.02%	93.32%	76.98%
ResNet152	Baseline	N/A	95.76%	95.76%	78.82%
	ProtoTree [14]	Single-Scale	92.48%	87.72%	20.88%
	Deformable ProtoPNet [2]	Single-Scale	91.16%	93.65%	76.27%
	ST-ProtoPNet [29]	Single-Scale	93.95%	96.09%	79.01%
	PIP-Net [15]	Single-Scale	64.44%	69.88%	26.87%
	MCPNet (Ours)	Multi-Scale	94.28%	94.54%	80.79%

Table 8. The comparison between our MCPNet and various methods on three benchmark datasets: AWA2, Caltech101, CUB_200_2011.

Several example results of further comparisons with PIP-Net [15] in terms of explanations are shown in Figure 16, where PIP-Net [15] produces object-centric explanations thus leading to misclassification when images lack the object-centric prototypes related to the correct class or have confusion across object-centric prototypes from various classes. Conversely, MCPNet bases its classification on a multi-level concept response that incorporates both high-level and low-level concepts, in which such approach ensures that the entire spectrum of concept responses contributes to the variation in the MCP distribution.

10.2. Class-level explanations

We will delve deeper into how our Multi-Level Concept Prototype Classifier (MCPNet) provides class-level explanations of the model through class-specific MCP distributions. The class-specific MCP distribution summarizes the distribution of concept prototype strengths within images of the same class, while our Class-aware Concept Distribution (CCD) loss also encourages the difference in such MCP distributions across classes. The class-wise explanations are derived by examining the high or low responses of the concept prototype in class-specific MCP distribution, which indicate the presence or absence of specific concept prototypes for the corresponding class, as illustrated in Figure 15.

We also illustrate the distinctions between the two MCP distributions for the fine-grained dataset and showcase the concept prototypes contributing to the differences between these distributions, as depicted in Figure 14.

11. Concept prototype importance and discriminativeness

Our MCPNet excels at revealing the importance or discriminativeness of each concept prototype, both at the image and class levels. For the image level, the importance of a concept prototype is directly inferred from the concept's strength, demonstrating its impact on the final prediction of classification. For the class level, we calculate the standard deviation of the strength for each concept within the class-specific MCP distribution to assess its discriminativeness across classes, as shown in Figure 17.

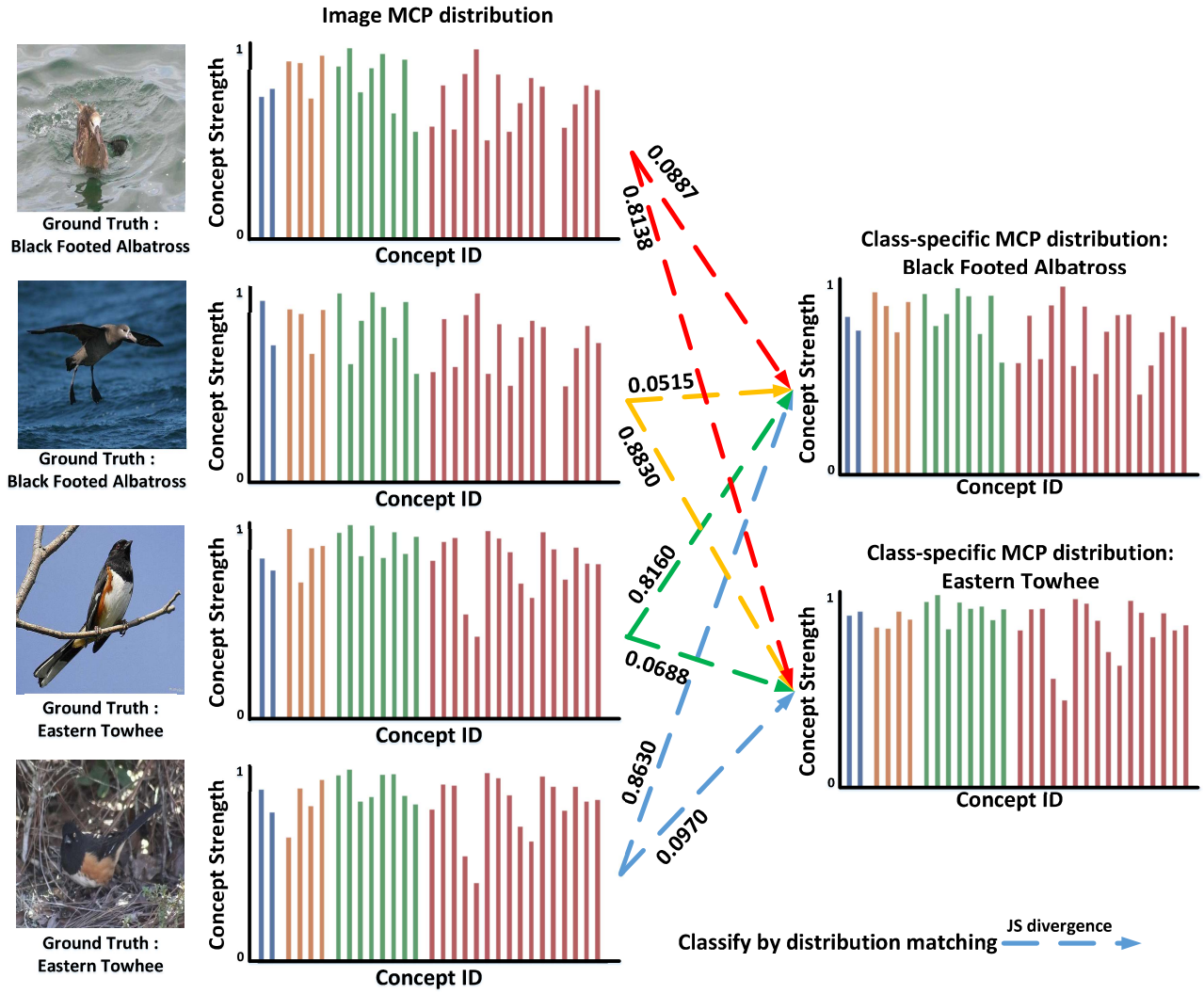

Figure 12. The demonstration of the image classification process in MCPNet, which involves matching image MCP distributions to the class-specific MCP distribution.

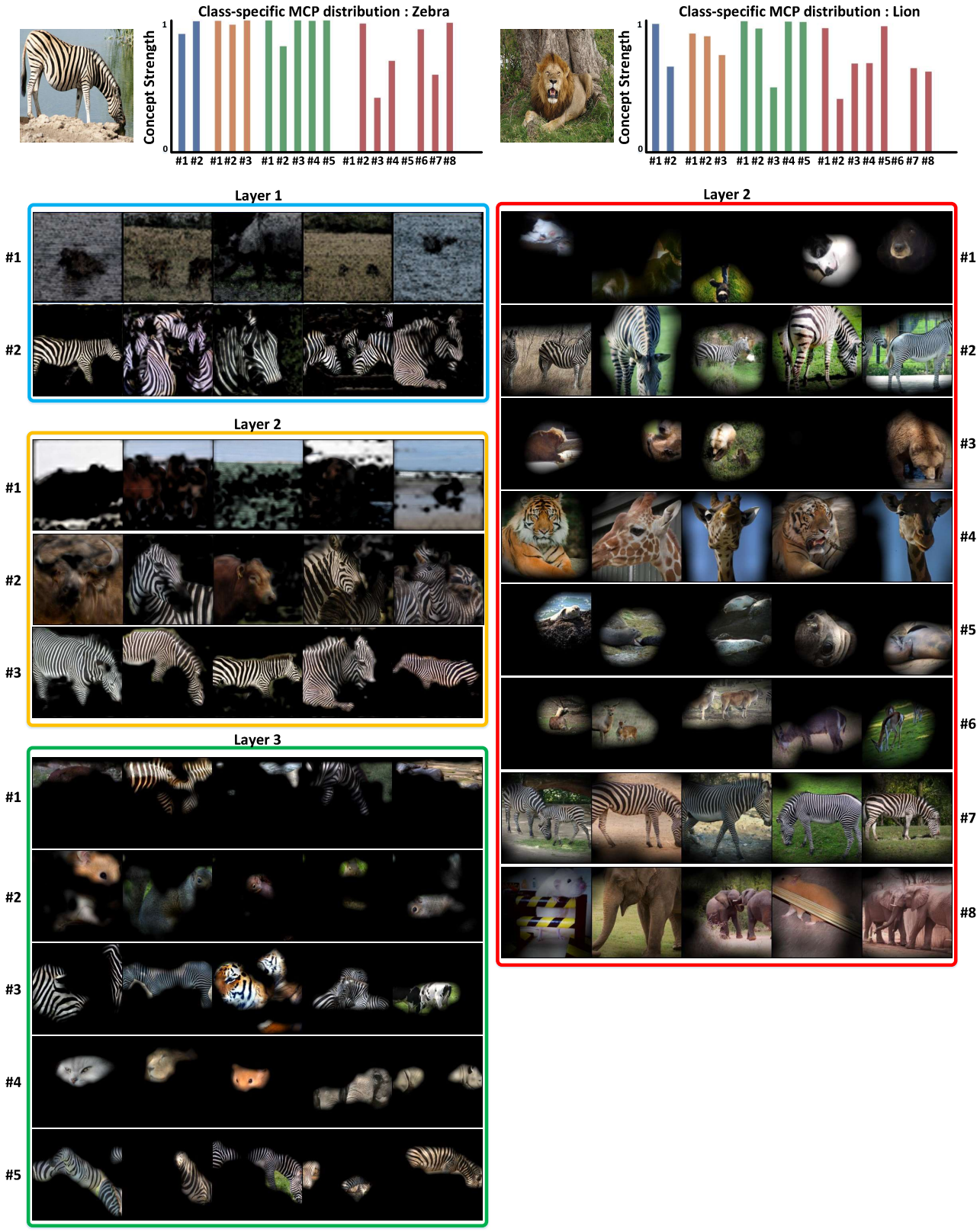

Figure 13. Image-wise explanation. For each image, the explanation is derived by calculating the concept strength to form the MCP distribution. The concept strength indicates whether the concept prototypes are prevalent in the image.

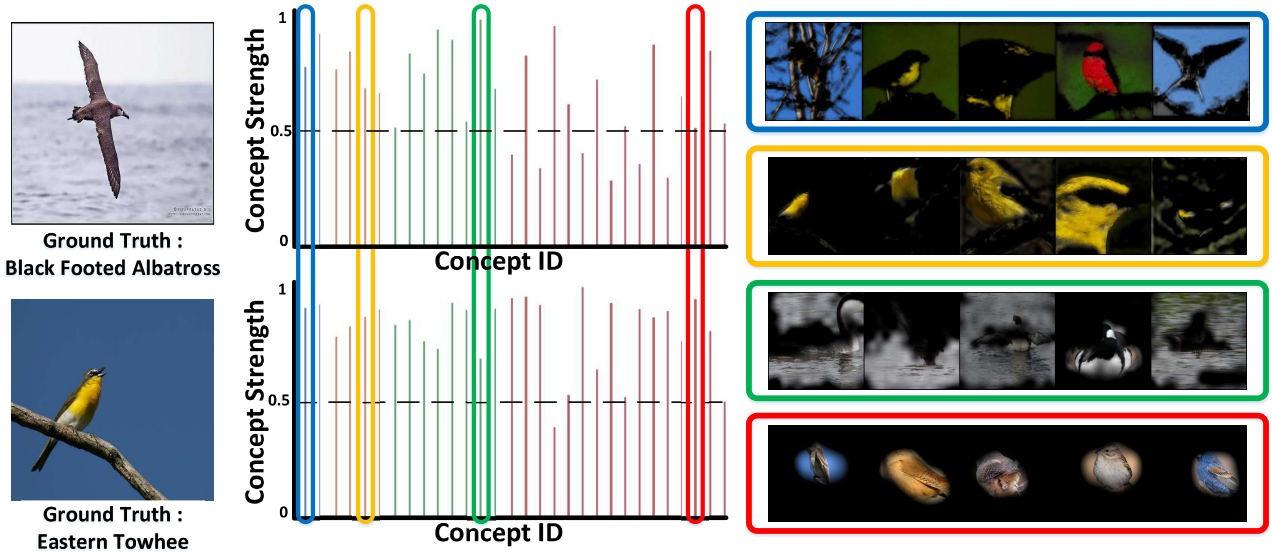

Figure 14. Illustrates the disparity between class-specific MCP distributions on CUB_200_2011 (backbone: ResNet50). We select concepts that elicit significantly different responses between the two class-specific MCP distributions to showcase the class-wise explanation based on the presence or absence of concept prototypes. The colors (blue, yellow, green, and red) represent layers 1 to 4, respectively. The boxes shown on the right side highlight the concept prototypes which contribute to the substantial differences between the two class-specific MCP distributions

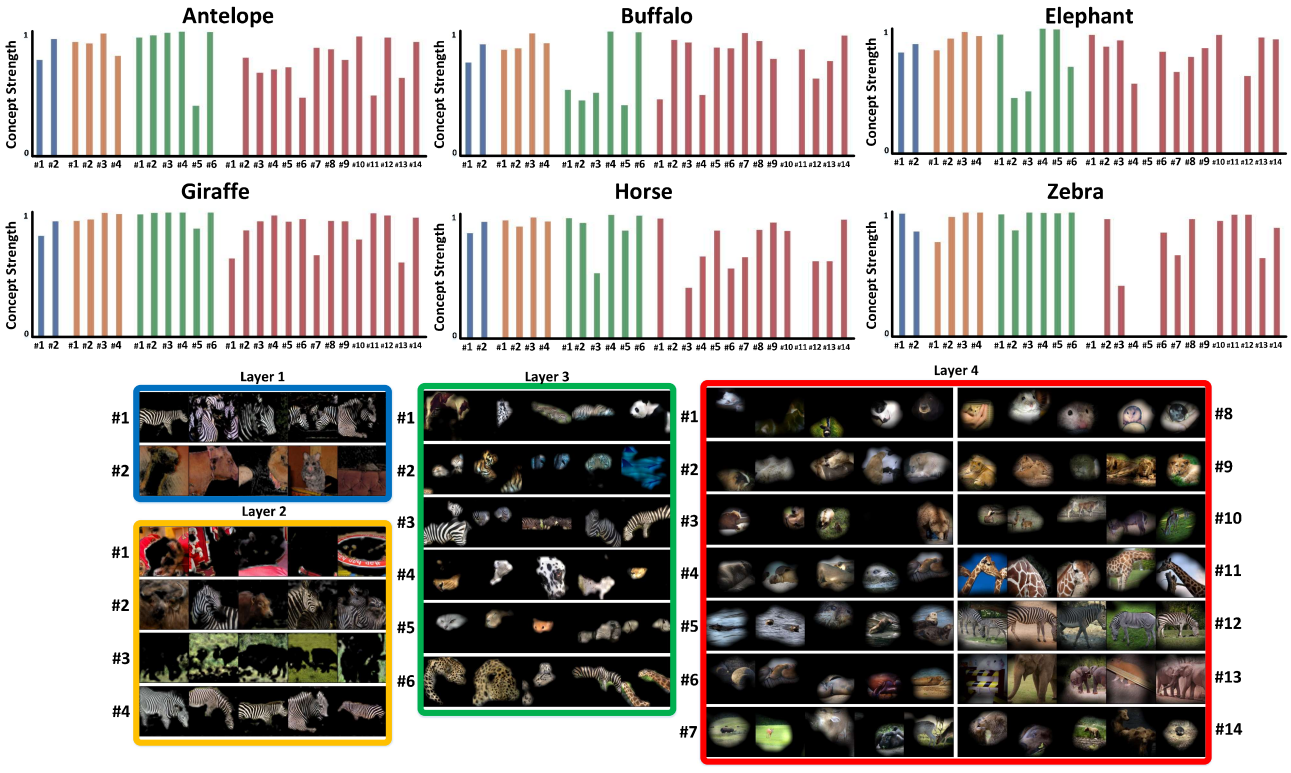

Figure 15. Class-wise explanation. For each category, the explanation is derived by aggregating the MCP distributions for the images of the target class. The concept strength indicates whether the concept prototypes are prevalent in most images belonging to the same class in the dataset.

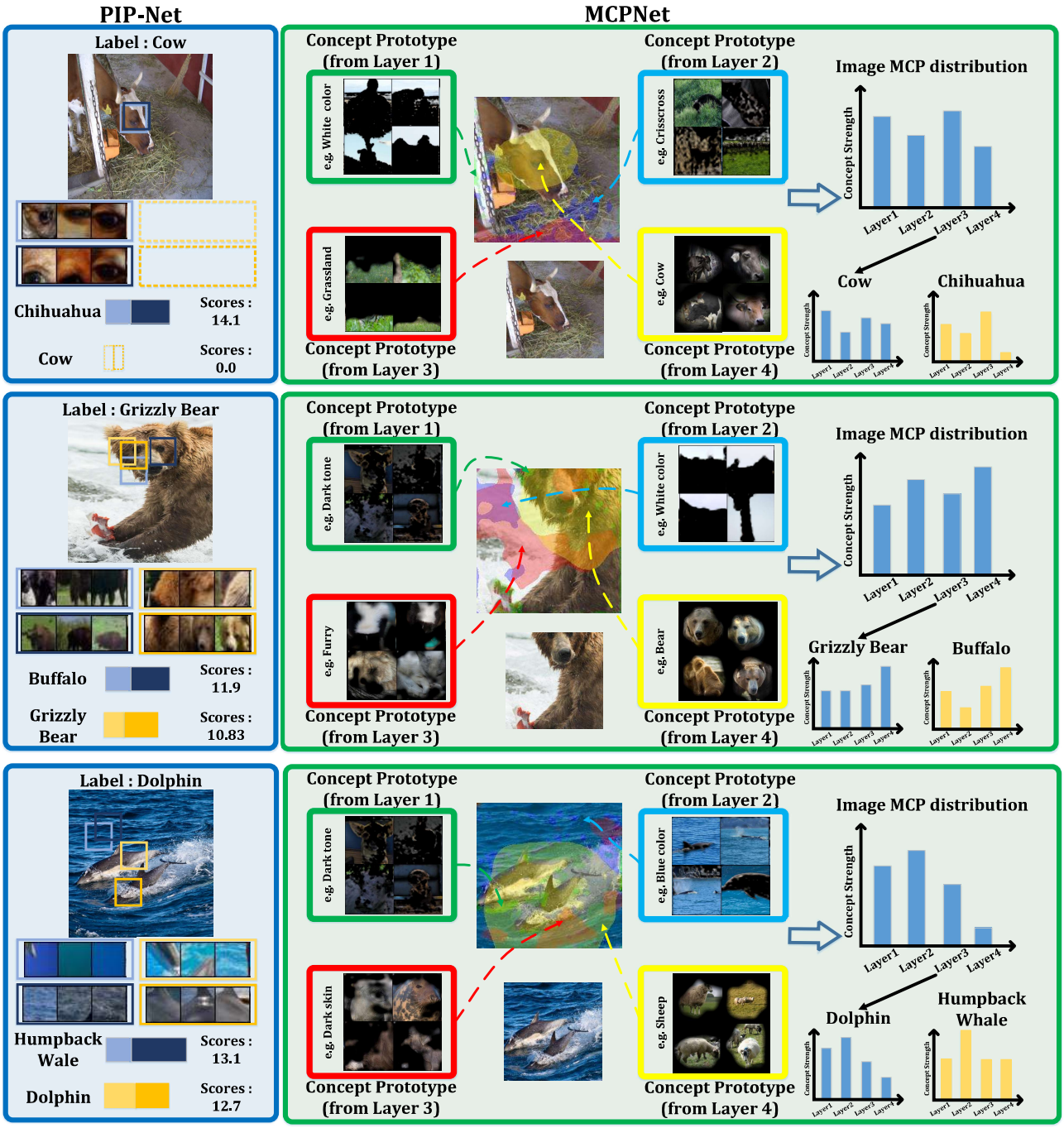

Figure 16. More explanation comparisons between MCPNet and PIP-Net [15]. PIP-Net [15] approaches all three scenarios with object-focused explanations, which results in inaccurate classifications. In the first scenario, PIP-Net fails to identify any matching prototypes from the correct class within the image. For the second and third scenarios, despite PIP-Net recognizing concepts from the correct class in the image, concepts from other classes receive higher responses, thus leading to confusion. Conversely, MCPNet employs multi-scale concept explanations as the foundation for accurate classification. In particular, for the second scenario, the high responses to both Grizzly Bear and buffalo classes in terms of high-level concept would lead to confusion if the classification is based solely on the high-level responses, while such confusion can be resolved with the incorporation of low-level concept responses. Moreover, in the third scenario, even without a direct concept match in the image – such as the concept from layer 4, potentially interpreted as sheep – MCPNet accurately interprets the image using the constructed MCP distribution based on the holistic consideration over the distribution of concept responses across multiple scales.

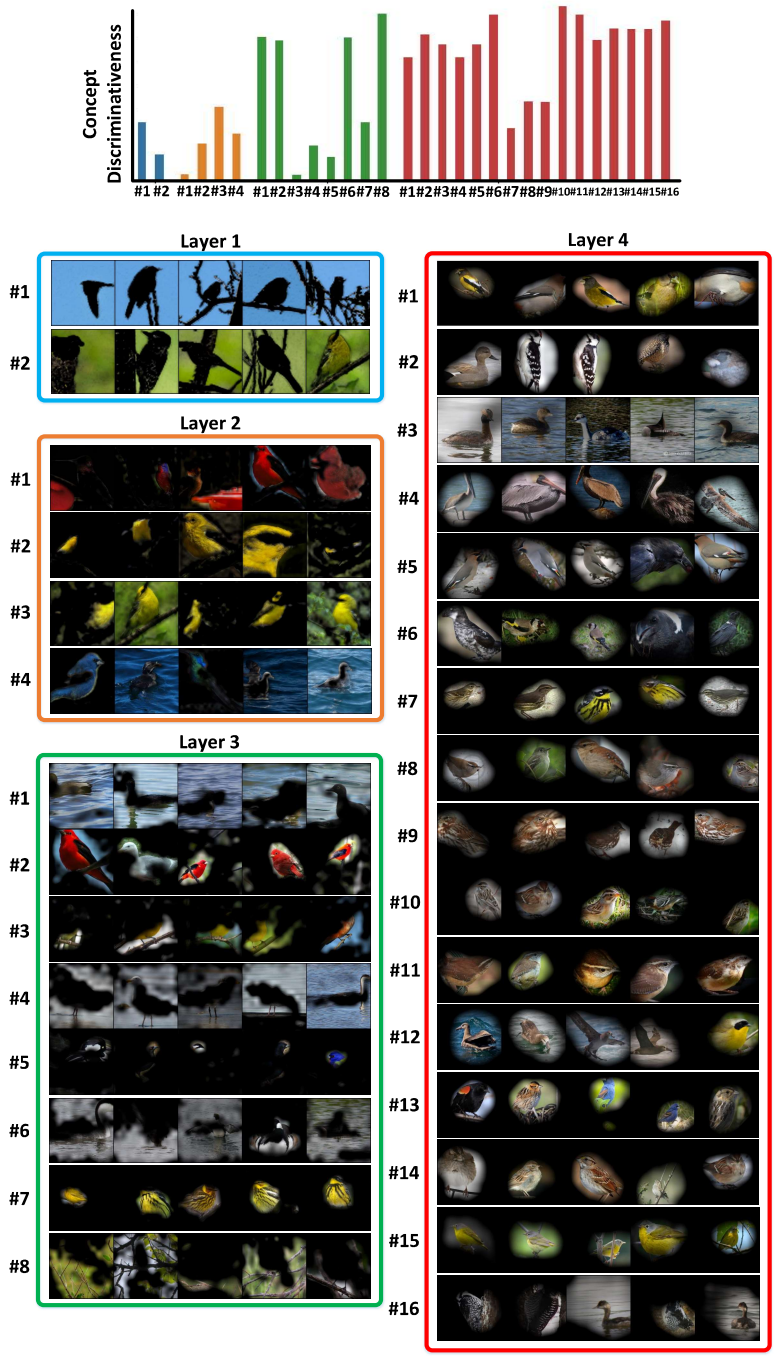

Figure 17. Concept discriminativeness at the class level is depicted in the upper part of the figure, showcasing the standard deviation of concept strength across classes. The higher variance in concept strength signifies that the corresponding concept prototypes play a more discriminative role in distinguishing images.

References

[1] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 2, 3

[2] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10265–10275, 2022. 2, 3, 6, 7, 1, 5

[3] Alexandros Doumanoglou, Stylianos Asteriadis, and Dimitrios Zarpalas. Unsupervised interpretable basis extraction for concept–based visual explanations. *IEEE Transactions on Artificial Intelligence*, 2023. 2

[4] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 1, 2, 3

[5] Jindong Gu, Rui Zhao, and Volker Tresp. Semantics for global and local interpretation of deep convolutional neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 6

[7] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 1, 2, 3

[8] Siwon Kim, Jinoh Oh, Sungjin Lee, Seunghak Yu, Jaeyoung Do, and Tara Taghavi. Grounding counterfactual explanation of image classifiers to textual concept space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10942–10950, 2023.

[9] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2, 3

[10] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 4

[11] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 5

[12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 6

[13] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3

[14] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021. 2, 3, 6, 7, 1, 5

[15] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023. 1, 2, 3, 6, 7, 5, 9

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1, 2, 3

[17] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019. 2

[18] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototype sharing for interpretable image classification and similarity discovery. *arXiv preprint arXiv:2011.14340*, 2020. 2, 3

[19] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision*, pages 351–368. Springer, 2022. 2, 3

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 3

[22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 3

[23] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(5), 2012. 4

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 3

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[26] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 8769–8778, 2018. 1

[27] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. 2011. 5

[28] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10962–10971, 2023. 1

[29] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis McCarthy, Helen Frazer, and Gustavo Carneiro. Learning support and trivial prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2062–2072, 2023. 2, 3, 6, 7, 1, 5

[30] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021. 2, 3

[31] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 4, 5

[32] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6261–6270, 2019. 1