MCPNet: An Interpretable Classifier via Multi-Level Concept Prototypes

Bor-Shiun Wang[†] Chien-Yi Wang[*‡] Wei-Chen Chiu[*†]
[†]National Yang Ming Chiao Tung University [‡]NVIDIA Research
eddiewang.cs10@nycu.edu.tw, walon@cs.nctu.edu.tw, chienyiw@nvidia.com

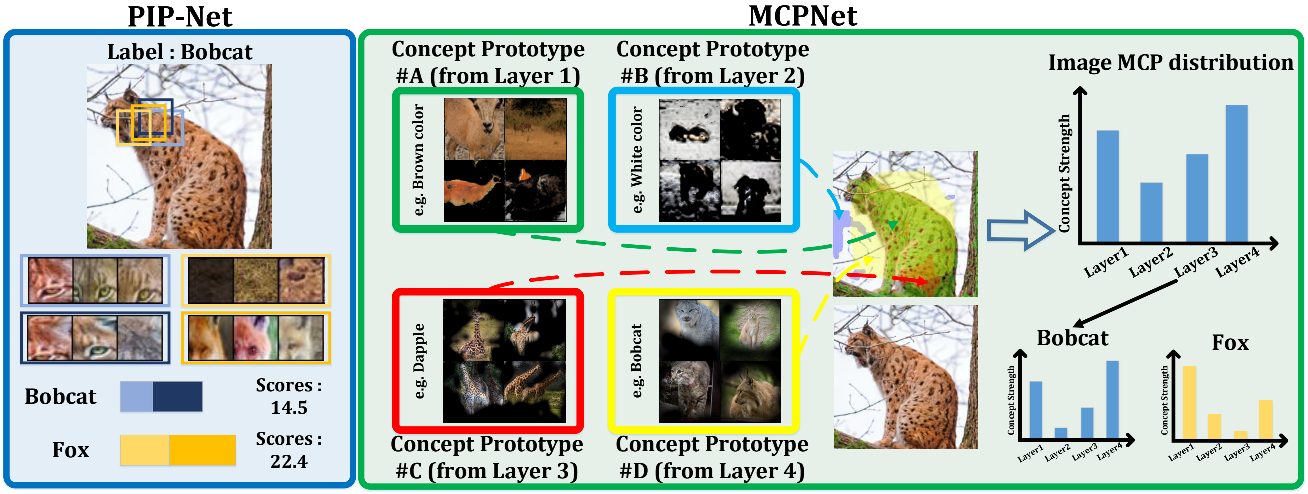

Figure 1. **The "Bobcat" was correctly classified by our MCPNet but incorrectly classified as a "Fox" by PIP-Net [15].** On the right side, we provide an illustration of using our proposed Multi-level Concept Prototype (MCP) distribution to classify and explain the input image. In particular, our concept prototypes are extracted from multiple layers of the classification model (thus having low-level to high-level concepts). In comparison with a recent state-of-the-art baseline, PIP-Net [15] shown on the left side which only adopts single-level explanations (symbolized as colorful boxes on the bottom portion, they are usually extracted from the last model layer), our proposed MCPNet provides more comprehensive explanations as well as better classification performance.

Abstract

Recent advancements in post-hoc and inherently interpretable methods have markedly enhanced the explanations of black box classifier models. These methods operate either through post-analysis or by integrating concept learning during model training. Although being effective in bridging the semantic gap between a model's latent space and human interpretation, these explanation methods only partially reveal the model's decision-making process. The outcome is typically limited to high-level semantics derived from the last feature map. We argue that the explanations lacking insights into the decision processes at low and mid-level features are neither fully faithful nor useful. Addressing this gap, we introduce the Multi-Level Concept Prototypes Classifier (MCPNet), an inherently interpretable model. MCPNet autonomously learns meaningful concept prototypes across multiple feature map levels using Centered Kernel Alignment (CKA) loss and an energy-based weighted PCA mechanism, and it does so without reliance on predefined concept labels. Further, we propose a novel classifier paradigm that learns and aligns multi-level concept prototype distributions for classification purposes via Class-aware Concept Distribution (CCD) loss. Our experiments reveal that our proposed MCPNet while being adaptable to various model architectures, offers comprehensive multi-level explanations while maintaining classification accuracy. Additionally, its concept distribution-based classification approach shows improved generalization capabilities in few-shot classification scenarios. Project link is available at https://eddie221.github.io/MCPNet/

1. Introduction

The rapid integration of deep learning across various domains has brought to the forefront a critical question: what underlying mechanisms drive the decisions of these models? This query has led to the emergence of Explainable Artificial Intelligence (XAI), a field dedicated to demystifying the operations of opaque 'black box' models.

In XAI, numerous methods have been developed, addressing different aspects of model interpretability [4, 5, 7–9, 13, 15, 16, 27, 31]. These methods generally fall into

[*]Equal advising

	MCPNet (Ours)	ProtoPNet [1, 2, 15]	Concept Bottleneck [9]	TCAV [7]	CRAFT [4]
Explanation Type	Inherently	Inherently	Inherently	Post-hoc	Post-hoc
Explanation Scale	Multi-Level	Single-Level	Single-Level	Single-Level	Single-Level
w/o Concept Labels	✓	✓	✗	✗*	✓
w/o Modifying Models	✓	✗	✗	✓	✓**

Table 1. We compare our method with four distinct lines of explainable approaches. Our MCPNet inherently offers multi-level explanations without the need for model modifications or concept labels, making it competitive compared to methods that achieve only partial properties. *It's noteworthy that TCAV requires dataset preparation both with and without specific concepts. **Only for non-negative features due to the limitation of Non-Negative Matrix Factorization.

two categories: 1) post-hoc methods, and 2) inherently interpretive methods. Post-hoc methods focus on elucidating model behaviours either locally [16] or globally [13], offering explanations for predictions without necessitating model retraining. While being valuable, these methods often provide explanations that lack coherence with the models' decision-making processes, leading to potential issues of unfaithfulness in interpretation [17].

To address the limitations of unfaithfulness in post-hoc methods, there has been a growing emphasis on inherently interpretable models featuring built-in, case-based reasoning processes. In contrast to post-hoc methods, these models generate explanations that are integral to the classification process. As one of the pioneering work, Concept Bottleneck Model (CBM) [9] which first translates an image into features signifying the presence or absence of predefined concepts, and then bases its decision-making on these conceptual representations. Recognizing the challenge of acquiring pre-defined concept labels, subsequent research has shifted towards the autonomous identification of concepts during training. Notable among these are ProtoPNet [1] and its derivatives [2, 14, 15, 18, 19, 28, 29], which build inherently interpretable classifier models with a predetermined number of prototypical (concept) parts that are learned automatically during the training process.

While recent advancements in inherently interpretable methods have significantly improved explanations of black box classifier models, a common limitation persists: these methods typically derive explanations from a single part of the model. Most of the previous studies (e.g., ProtoPNet series) have concentrated on extracting human-understandable concepts from the last feature map, just before the fully connected (FC) layer, to elucidate model behavior (noting that here we take the classification models as the representative example, without loss of generality). This line of approaches transform features into explanations to inform outcomes, yielding understandable concepts but only illuminating the last model layer with high-level semantics, leaving much still obscured as a 'black box'.

In this paper, we present MCPNet, a novel **hierarchical explainable classifier** designed for more comprehensive multi-level model explanations. By eliminating the FC layer, MCPNet encourages learning of more distinctive features across various model layers. Unlike previous methods that used entire channels to represent concept features, often facing challenges in establishing orthogonality among similar concepts [3], MCPNet partitions feature maps into distinct **segments**. Each segment focuses on learning unique concepts, facilitated by our proposed Centered Kernel Alignment (CKA) loss. These segments are further differentiated using a weighted Principal Component Analysis (PCA), which prioritizes pixel importance in extracting the concept prototype (CP). For each image, MCPNet calculates the concept response using the CP and its corresponding concept segment, forming what we term the Multi-level Concept Prototype distribution (MCP distribution). Additionally, we introduce the Class-aware Concept Distribution (CCD) loss. This loss function enhances the distinction of the MCP distribution between different classes while minimizing it within the same class. To classify images without the conventional FC layer, MCPNet compares the image's MCP distribution with the class-specific centroid MCP distribution, which is an average of the MCP distributions across instances in the same category, to identify the most similar class. The primary contributions of our work are outlined as follows:

- We introduce a novel hierarchical explainable classifier MCPNet that offers in-depth, multi-level explanations of model behavior. This advancement marks a significant shift from traditional models, which primarily focus on high-level semantics, to a more comprehensive approach that includes insights from various layers of the model.
- The proposed inherently multi-level interpretable paradigm can be seamlessly integrated with multiple convolution-based model architectures without additional modules or trainable parameters while maintaining comparable classification performance.
- Evaluation on several benchmark datasets verifies that our method can provide richer concept-based explanations across low-to-high semantic levels and exhibit better generalization ability towards unseen categories.

2. Related Works

Post-hoc Interpretation Methods In the realm of post-hoc explanations, a variety of methods have been developed to elucidate model behaviors without necessitating retraining. These approaches predominantly utilize extracted features from specific instances. A notable category within this field is *attribution methods* which are initially proposed by [21] and subsequently inspiring further research [20, 22, 24]. These methods primarily generate heatmaps to highlight the impact of individual pixels on model outcomes.

Alternatively, *concept-based methods* aim to derive human-understandable concepts from model features. By using the predefined concept, TCAV [7] measures the concept impact on models' outputs. A recent advancement in this area is CRAFT [4], which employs Non-Negative Matrix Factorization (NMF) to deconstruct target features and iteratively clarify ambiguous concepts from lower-layer features. Model-agnostic methods also play a pivotal role, with LIME [16] introducing a local explanation technique that assesses the influence of feature presence or absence on model results through input perturbation. In contrast, SHAP [13] utilizes Shapley values from game theory to provide individual prediction explanations on a global scale. Despite their utility, these methods have significant limitations, primarily in providing explanations that may not align with model predictions. This discordance raises concerns about the reliability of explanations, as it becomes challenging to discern whether inaccuracies lie in the explanation or stem from reliance on spurious data in predicting outcomes.

Inherently Interpretable Methods Inherently interpretable methods, which learn explanations during the training process, result in outcomes more closely aligned with the model's decision-making process due to these integrated explanations being more faithful and reliable. The Concept Bottleneck Model (CBM) [9] exemplifies this by mapping images to features that represent the presence or absence of predefined concepts, subsequently utilizing these concepts for decision-making. However, this method's reliance on predefined concept labels restricts its ability to discriminate concepts not explicitly provided in the data.

Alternatively, ProtoPNet [1] learns class-specific prototypes, akin to concepts, with a set number per class. It classifies by calculating responses from each class's prototype and summarizing these responses using a fully connected layer. Explanations are derived as a weighted sum of all prototypes. Deformable ProtoPNet [2] enhances this approach by training the prototypes to be orthogonal, aiding in clarity of interpretation. Similarly, TesNet [29] incorporates an additional module to separate prototypes on the Grassmann manifold.

To optimize the number of prototypes, ProtoPShare [18] merges similar prototypes, while ProtoPool [19] adopts a soft-assignment approach for prototype sharing across classes. ProtoTree [14] utilizes a binary decision tree, allowing prototypes to be shared across all classes, thereby reducing their number. This process resembles traditional decision trees, where image features traverse the tree, aggregating logits at leaf nodes and considering probabilities from the root to the leaves. More recent approaches, such as PIP-Net [15], strive for prototypes that align more closely with human perception by bridging the gap between latent and pixel spaces. ST-ProtoPNet [28] introduces support prototypes that position near the classification boundary to enhance discriminative capabilities.

Despite these methods significantly enhancing explanations for black-box models, their focus on high-level features leaves the behaviors of low and mid-layer features largely opaque. Our proposed MCPNet addresses this gap by providing explanations via multi-level concept prototypes. Table 1 compares our MCPNet with four representative lines of approaches, which summarizes the difference in multiple critical aspects.

3. Method

3.1. Overall Framework

As motivated previously, in this paper we introduce an inherently interpretable framework named MCPNet to reveal multi-level global concepts throughout different model layers. This marks a departure from previous approaches which usually provide only single-level explanations. Moreover, our framework has the capability to classify images via adopting the distribution of multi-level concept prototypes instead of relying on the typical fully connected (FC) classifier, in which it ensures the model to consider not only the features from the last layer but also from mid- and even low-layers, thus providing better generalizability towards unseen categories. In the following we sequentially detail the main components and important designs of our proposed MCPNet framework.

3.2. Centered Kernel Alignment (CKA) Loss

Given a feature map $F_l \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$ of dimension (width $H_l\times$ height $W_l\times$ channels C_l) and batch size B obtained from the l-th layer of a deep model, as the general architecture design of deep neural networks by nature extracts different characteristics of the input data into the features placed along the channels (i.e. each channel stands for a certain data characteristic), we now would like to partition the feature map F_l along the channel dimension into several distinct segments where each segment of size $\mathbb{R}^{B\times C'_l\times H_l\times W_l}$ groups C'_l channels to form a more semantic-meaningful component (representing a specific combination of data characteristics), termed as "concept". These concepts in results serve as a bridge/proxy to provide more interpretable explanations upon the input data samples towards their corresponding categories/classes. Moreover, the concepts ideally should be diverse and discriminative from each other in order to describe the data from different aspects, forming a more concise but representative basis

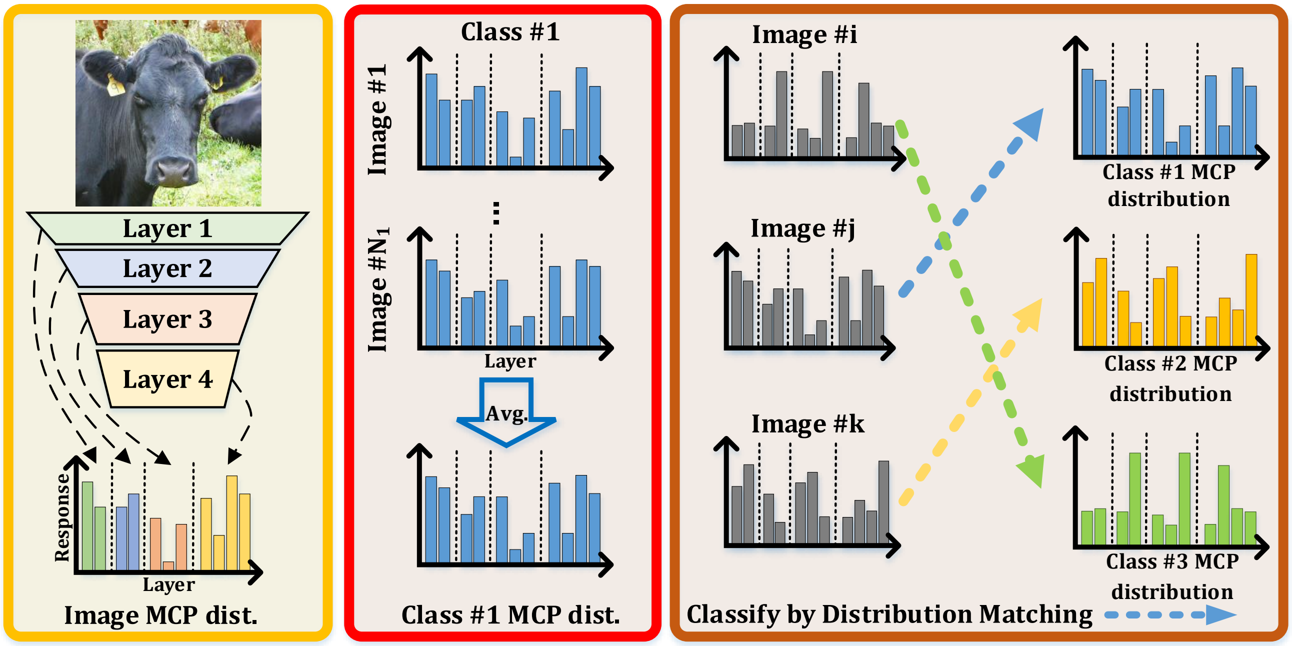

Figure 2. The overall workflow of our MCPNet. To classify the image, we first calculate the class-specific MCP distribution (yellow box) by averaging the MCP distributions from instances of specific classes in the training set (red box). Utilizing the class-specific MCP distribution, images are classified by identifying the most similar class via calculating the Jensen-Shannon (JS) divergence (brown box).

of interpretation. To this end, we introduce Centered Kernel Alignment (CKA) loss $\mathcal{L}^{\mathbf{CKA}}$, which leverages CKA metric [10] to measure the similarity among segments and its minimization brings the distinct concepts (i.e. the concepts are learned to be dissimilar and independent from each other).

Basically, given two concept segments $\mathcal{X}$ and $\mathcal{Y}$, their CKA similarity $\mathbf{CKA}(\mathcal{X},\mathcal{Y})$ is defined as:

$$\mathbf{CKA}(\mathcal{X},\mathcal{Y}) = \frac{\mathbb{H}(\mathcal{X},\mathcal{Y})}{\sqrt{\mathbb{H}(\mathcal{X},\mathcal{Y})}\sqrt{\mathbb{H}(\mathcal{Y}\mathcal{Y})}}, \tag{1}$$

where operator $\mathbb{H}$ stands for the unbiased Hilbert-Schmidt independence criterion proposed by [23] in which $\mathbb{H}(\mathcal{X},\mathcal{Y})$ is formulated as:

$$\frac{1}{B(B-3)}(tr(\tilde{K}\tilde{L}) + \frac{1^T\tilde{K}11^T\tilde{L}1}{(B-1)(B-2)} - \frac{2}{B-2}1^T\tilde{K}\tilde{L}1) \tag{2}$$

where $\tilde{K}$ and $\tilde{L}$ are stemmed from the kernerl matrices of $\mathcal{X}$ and $\mathcal{Y}$ respectively with having $\tilde{K}_{i,j} = (1-\mathbb{1}_{i=j})K_{ij}$ and $\tilde{L}_{i,j} = (1-\mathbb{1}_{i=j})L_{ij}$. Noting that here the variable B stands for the number of samples involved into the computation of $\mathbb{H}$, which is exactly the batch size in our application.

Based on the CKA similarities between segments from F_l, the CKA loss $\mathcal{L}^{\mathbf{CKA}}$ of the l-th layer is defined as:

$$\mathcal{L}^{\mathbf{CKA}}(S_l) = \frac{2}{M_l(M_l-1)}\sum_{i=1}^{M_l}\sum_{j=i}^{M_l}\mathbf{CKA}(S_{l,i}, S_{l,i}), \tag{3}$$

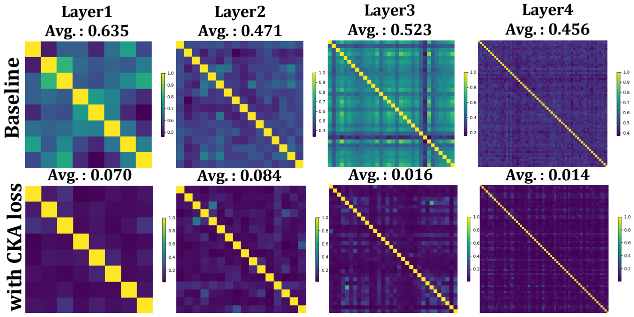

Figure 3. Visualization of CKA similarities for each pair of segments from different model layers (model backbone: ResNet50; dataset: AWA2 [30]). The average for the upper triangular portion of each CKA similarity matrix is also provided accordingly. With apply our proposed CKA loss, the similarities between segments are clearly reduced, i.e. leading to more distinct concept segments.

where $S_{l,i}$ and $M_l = C_l/C'_l$ denotes the i-th concept segment and the total number of segments of l-th layer respectively. In Figure 3 we visualize the CKA similarities between segments for different layers, highlighting the noticeable decrease in terms of CKA similarity brought by applying our proposed CKA loss.

3.3. Multi-level Concept Prototype Extraction

For a single input data (i.e. B = 1), each concept segment $S_{l,i}$ in the l-th layer contains $H_l \cdot W_l$ feature vectors of length C'_l. When it comes to the entire dataset of having N data samples, there will be in total $N \cdot H_l \cdot W_l$ feature vectors for $S_{l,i}$, and every feature vector serves as an instance of the corresponding semantic behind the concept segment. In order to better outline such semantic globally behind $S_{l,i}$, we propose to identify the principal direction of all these feature vectors (i.e. analogous to the common ground among the feature vectors) by using the weighted Principal Component Analysis technique (where we weight each feature vector according to its L2-norm to account for the degree of importance).

The resultant principal direction (i.e. the eigenvector corresponding to the largest eigenvalue of the covariance matrix built upon $N \cdot H_l \cdot W_l$ feature vectors of length C'_l for $S_{l,i}$) is termed as the **concept prototype**. Please note that the concept prototype is globally defined and shared within the entire dataset, while the concept segments in turn act more likely as the sample-wise or batch-wise instantiation of the corresponding semantic. Hence, we can further compute the **prototype response** of a concept segment $S_{l,i}$ (which is based on the input of a single sample or a batch of samples) with respect to the corresponding concept prototype $P_{l,i}$, following a simple procedure: Firstly the response map which records the cosine similarities of $P_{l,i}$ at each position on $S_{l,i}$ is computed, then the max-pooling is applied on the response map to obtain the prototype response. Such prototype response stands for the degree of agreement between concept segments and the concept prototype, hence signifies how likely the input sample(s) contributing to the concept segments would own the particular semantic of the target concept prototype. It is worth noting that, while the numerical range of original prototype response is $[-1.0, 1.0]$, we linearly map it to the range of $[0.0, 1.0]$ for better usage in the later computation.

The extraction of concept prototypes is thoroughly applied on all the concept segments from all the model layers, leading to **multi-level concept prototypes**, in which it is one of the key factors differentiating our MCPNet from the others (where most previous works only provide single-level explanation, typically extracted from the last layer).

3.4. Class-aware Concept Distribution Loss

From cognitive perspective, the samples belonging to the same class ideally should have similar combination of concepts. Hence, we build upon such idea to propose the Class-aware Concept Distribution (CCD) loss, which encourages the samples of the same class to have similar distribution of concept prototypes (i.e. the distribution upon the prototype responses with respect to all the multi-level concept prototypes) while enlarging the distribution distance across different classes. In other words, such loss helps to realize the classification via leveraging our Multi-level Concept Prototype distribution (named as **MCP distribution**), while forsaking typical fully-connected-layer-based classifier with even providing better interpretability.

Basically, with denoting the MCP distribution of an input sample/image x_i as D_i and the class label of x_i as $y(x_i)$, we first compute the class-specific centroid MCP distribution $D^{\mathbf{c}}$ via averaging D_i of all the samples x_i belonging to the same class $\mathbf{c} = y(x_i)$. Then our CCD loss is defined as:

$$\begin{aligned}\mathcal{L}^{\mathbf{CCD}}(x_i) = \; & \mathbb{J}(D_i, D^{\mathbf{c}=y(x_i)}) \\ & + \sum_{\mathbf{c}' \neq y(x_i)} \max(\mathbf{m} - \mathbb{J}(D_i, D^{\mathbf{c}'}), 0),\end{aligned} \tag{4}$$

where $\mathbb{J}$ stands for the Jensen-Shannon divergence (a common metric used for evaluating the distance between distributions), while $\mathbf{m}$ is the margin such that $\mathbb{J}(D_i, D^{\mathbf{c}'})$ contributes to the loss only if it is smaller than $\mathbf{m}$, and it helps avoiding a collapsed solution (which is a typical technique used the contrastive loss of metric learning).

The overall objective function $\mathcal{L}$ to train our model is basically the combination of both CKA and CCD losses:

$$\mathcal{L} = \sum_{l=1}^{L} \mathcal{L}^{\mathbf{CKA}}(S_l) + \lambda_{CCD} \sum_{x_i \in \mathbf{X}} \mathcal{L}^{\mathbf{CCD}}(x_i), \tag{5}$$

where L denotes the number of layers in our model, $\mathbf{X}$ denotes the training dataset, and λ_{CCD} denotes the weight of CCD loss. It is worth noting that all the concept prototypes and all the class-specific centroid MCP distributions are updated after every epoch on the training set to reflect the newest features learned by the model.

3.5. Multi-Level Concept Prototypes Classifier

As mentioned in the previous subsection, our MCPNet does not require the FC layer attached to the model end for performing classification. Instead, our MCPNet is able to classify the input sample x_i simply via searching for the closest class-specific centroid MCP distribution to D_i:

$$\tilde{y}(x_i) = \arg\min_{\mathbf{c}} \mathbb{J}(D_i, D^{\mathbf{c}}). \tag{6}$$

4. Experiments

Datasets. Three datasets are adopted for our evaluation:

- **AWA2** [30] is dataset which was originally proposed for evaluating the zero-shot classification task. It consists of 37322 images with 50 categories, each is additionally annotated with 85 attributes. We split a quarter of the entire dataset as the test set, while the rest is taken as the training set. Please note that the attribute labels are not used in our model training.
- **Caltech101** [11] is a classification dataset that has 9146 images from 101 distinct categories, in which each class has roughly 40 to 800 images. We randomly draw a quarter of images of each class to form the test set, while the rest becomes the training set.
- **CUB_200_2011** [26] is a fine-grained image classification dataset that contains 11788 images of 200 bird classes/species (and additional has 312 binary labels of

Backbone	Methods	Explanation	Accuracy		
			AWA2	Caltech101	CUB_200_2011
ResNet50	Baseline	N/A	94.92%	94.21%	77.94%
	ProtoTree [14]	Single-Scale	90.60%	72.19%	18.00%†
	Deformable ProtoPNet [2]	Single-Scale	85.51%	93.88%	73.42%†
	ST-ProtoPNet [28]	Single-Scale	93.76%	95.95%	76.34%†
	PIP-Net [15]	Single-Scale	85.99%	87.86%	70.99%†
	MCPNet (Ours)	Multi-Scale	93.92%	93.88%	80.15%
Inception V3	Baseline	N/A	95.47%	96.42%	79.43%
	ProtoTree [14]	Single-Scale	92.29%	86.02%	13.03%
	Deformable ProtoPNet [2]	Single-Scale	92.68%	97.22%	72.99%
	ST-ProtoPNet [28]	Single-Scale	93.60%	96.99%	75.25%
	PIP-Net [15]	Single-Scale	43.82%	45.04%	6.76%
	MCPNet (Ours)	Multi-Scale	94.62%	95.76%	78.94%
ConvNeXt-tiny	Baseline	N/A	96.55%	96.56%	84.55%
	ProtoTree [14]	Single-Scale	94.00%	78.82%	21.57%
	Deformable ProtoPNet [2]	Single-Scale	91.94%	93.65%	35.05%
	ST-ProtoPNet [28]	Single-Scale	94.22%	97.17%	81.84%
	PIP-Net [15]	Single-Scale	93.80%	96.61%	82.74%
	MCPNet (Ours)	Multi-Scale	95.61%	95.95%	83.45%

Table 2. The classification performance evaluation on AWA2, Caltech101, and CUB_200_2011 benchmarks with different backbone choices. The baseline represents the typical classification without any explanation capability. †The discrepancies with respect to the accuracies reported in their original papers are caused by using different pretrained weights (here we adopt the pertaining on ImageNet).

attributes), where 5,994 images are used for training while the other 5,794 images are for testing. Please note that, we only use the training images and corresponding class/species labels for performing our model training without using the additional attribute annotations.

4.1. Implementation Details

Layer Selections. We adopt three off-the-shelf models, ResNet50 [6], and Inception v3 [25], ConvNeXt-tiny [12] to do the experiments. The layers selected for each model are shown in the supplementary.

Training hyper-parameters. We conduct training for ResNet50 and InceptionNet V3 over 100 epochs, using Adam optimization with a weight decay set to 1e-4. The learning rate follows a decay rate of 10 every 40 epochs, beginning at 1e-4. For ConvNeXt-tiny, we utilized the official training code with 100 epochs. The dimension for concept segments, denoted as C', is set to 32 for ResNet50 and InceptionNet v3 and 16 for ConvNeXt-tiny. The Class-aware Concept Distribution (CCD) loss margin is set to 0.01 for ResNet50 and InceptionNet v3 and 0.05 for ConvNeXt-tiny. As the original numerical range of CCD loss is way smaller than that of CKA loss, we set λ_{CCD} to 100 to balance between these two loss functions.

4.2. Quantitative Results

This section explains how MCPNet provides explanations at multiple levels without compromising performance, comparing to the typical training paradigm coupled with a fully connected (FC) classifier. Basically, the conventional approach trains the model using cross-entropy loss, focusing the model's learning on features in the final layer for image differentiation – a technique widely used in prior methods. In contrast, MCPNet classifies images based on the distribution formed by the response of multi-level concept prototypes, considering concepts of various scales.

In Table 2, we present the primary quantitative outcomes across three classification datasets, comparing MCPNet with the baseline and other methods in the ProtoPNet series. MCPNet matches or surpasses their performance by categorizing images through alignment with the nearest class-specific MCP distribution and delivers explanations across multiple levels. Conversely, alternative methods offer explanations at a single scale, primarily concentrating on the model's final layer to produce object-centric explanations. The findings further demonstrate MCPNet's versatility, as it can be integrated into diverse models to yield robust performance and multi-level explanations.

5-shot Classification. In the 5-shot experiments, the datasets are divided into seen and unseen sets. The model is trained on the seen set, from which we derive global concept prototypes. Within the unseen set, 5 images from each category are randomly selected to create the class-specific MCP distributions for classes in the unseen set. The remaining images are utilized to assess the model's efficacy in 5-shot classification.

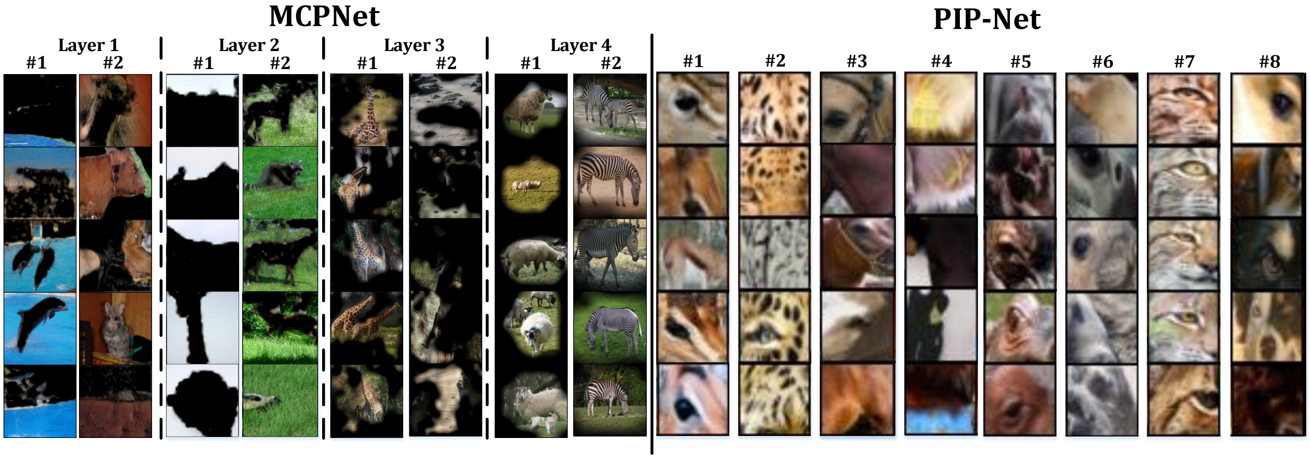

Figure 4. Concept prototype examples from MCPNet and PIP-Net [15]. We show the top-5 responses for the sampled concept prototypes. For MCPNet, the concept prototype from various layers generates explanations in different scales, e.g. color-like explanations in low-layer and object-like explanations in high-layer. On the contrary, PIP-Net [15] only provides single-scale patch-level explanations.

Dataset	Method	Accuracy
AWA2	Baseline	60.55%
	ProtoTree [14]	33.68%
	Deformable ProtoPNet [2]	19.71%
	ST-ProtoPNet [28]	30.15%
	PIP-Net [15]	26.17%
	MCPNet (Ours)	**73.79%**

Table 3. The 5-shot classification performance with ResNet50 backbone on AWA2 datasets. The baseline represents the typical classification without any explanation capability.

We evaluate the accuracy of our approach against the baseline and methods from the ProtoPNet series. For the baseline, we apply the same classification mechanism used in MCPNet to images in the unseen set. For the ProtoPNet series methods, the model was trained on the seen set using default settings, followed by a single epoch of training on the images which are arranged similarly to those used for creating the MCPNet's class MCP distribution. This one-epoch training is chosen to align with MCPNet for its processing the images in a single pass.

In Table 3, we demonstrate our method achieves competitive outcomes in the 5-shot experiments. This indicates that the concepts we extract are sufficiently generalized to discern between classes, including those not involved in the concept extraction process.

4.3. Qualitative Results.

In this section, we introduce how we recognize the meaning of concept prototypes and how MCPNet explains the images or even the classes with the MCP distributions.

Concept Prototype meanings. To intuitively grasp each concept prototype's essence, we visually depict them through images that elicit the highest top-5 responses for a given concept within the dataset. By overlaying these images with the response map described in Section 3.3, the visible regions help us identify the features highlighted by the concept prototype. Our findings are contrasted with those from PIP-Net [15] in Figure 4 – unlike PIP-Net [15] where the explanations are typically tied to specific object parts, MCPNet offers explanations on multiple levels derived from various layers of the model. These explanations range from high-level concepts, such as parts of an object, to low-level attributes, such as color. Additional examples are provided in Figure 5 and in the supplementary material.

MCP distribution explanations. Here we provide explanation upon how our MCP distribution is used to interpret the model's classification of images. For each image, we obtain the corresponding MCP distribution by evaluating the interactions between the extracted concept prototype and its related concept segment, which shows the concepts contained in the image. As depicted in Figure 1, we determine the image's classification by comparing its distribution to the class-specific MCP distribution. This process involves identifying concept responses that show a high degree of similarity to a particular category, suggesting that the image's concepts resemble those of the target category. This forms the basis of our explanation for why an image is categorized into a specific class. For each class, the class-specific MCP distribution – which averages the MCP distributions of images within that class – highlights prevalent concept responses, with higher responses denoting concepts frequently associated with that class.

4.4. Ablation Study

The effect of the different losses. Table 4 shows the performance impacts of including or excluding each proposed loss. While the CKA loss is dedicated to disentangling concept segments, its singular use results in performance that

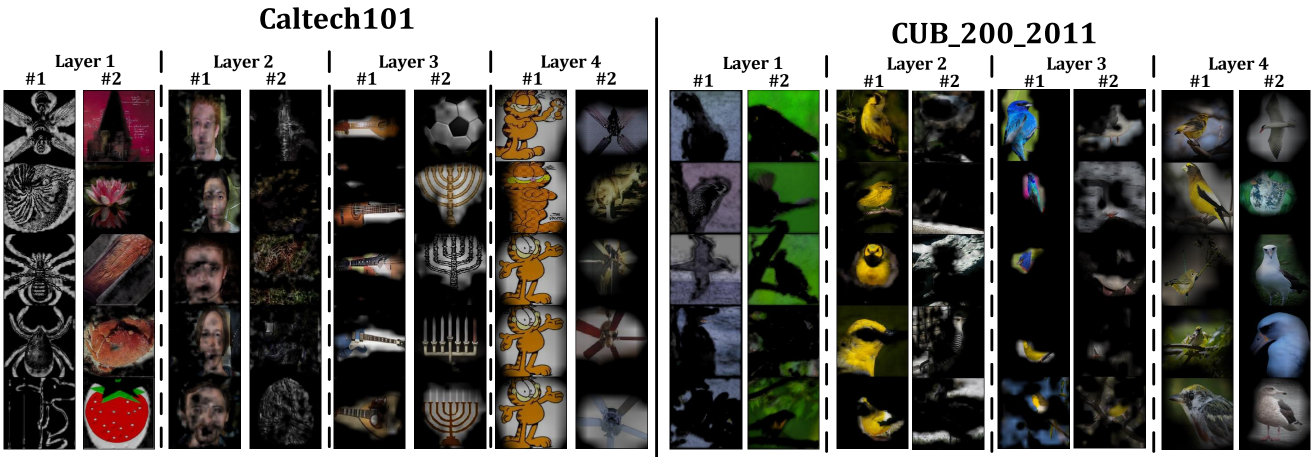

Figure 5. The sampled multi-level concept prototypes learnt by our proposed MCPNet. (Backbone : ResNet50)

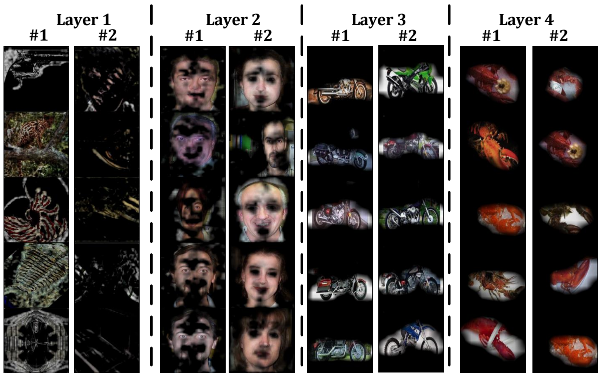

Figure 6. Showing the multi-level concept prototypes learnt by MCPNet variant with only the CCD loss, in which there are more duplicate prototypes than the ones learnt by the full MCPNet (i.e. the CKA loss is also adopted).

CKA loss	CCD loss	Accuracy
✓		44.85%
	✓	94.21%
✓	✓	93.88%

Table 4. Ablation study on the effect of CKA loss and CCD loss. It shows the MCPNet classification accuracy on Caltech101 with ResNet50 backbone.

falls below the baseline, as it lacks the capability to classify images effectively. The integration of the proposed CCD loss, on the other hand, leads to an improvement in accuracy over the baseline. However, without the CKA loss, there is an observed increase in similarity among concept segments, leading to duplicated concept prototypes, a phenomenon depicted in Figure 6. The combined application of CCD and CKA losses outperforms the exclusive use of CCD loss, achieving more distinct concept prototypes with only a slight compromise in performance.

The effect of channel size. Table 5 presents the impact of different channel sizes on the performance of our model, with a comparison among 32, 16, and 8 channels. The model with 32 channels offers half the concept segments of the 16-channel setup and a quarter of those in the 8-channel configuration. On coarse-grained datasets like AWA2 and Caltech101, the performances of models with 32 and 8 channels are comparable, indicating that the number of concept segments with 32 channels suffice for capturing class distinctions in these scenarios. In contrast, the 8-channel setup shows enhanced performance on the fine-grained CUB dataset, likely attributable to the higher number of concept segments that better capture the nuanced differences within the dataset.

Dataset	Channel	Accuracy
AWA2	32	93.92%
	16	93.95%
	8	93.58%
Caltech101	32	93.88%
	16	93.79%
	8	93.51%
CUB_200_2011	32	80.15%
	16	80.19%
	8	81.22%

Table 5. The ablation study with different channel sizes of the concept segments (Backbone: ResNet50).

5. Conclusion

In this paper, we propose Multi-Level Concept Prototype Classifier (MCPNet), an inherently interpretable method which learns multi-layer concept prototypes without reliance on predefined concept labels. In addition to having more comprehensive multi-level model explanations, our MCPNet is experimentally shown to provide a classification paradigm which is able to achieve comparable performance as the typical fully-connected-layer-based classifier while achieving better generalizability upon unseen classes.

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

[3] Alexandros Doumanoglou, Stylianos Asteriadis, and Dimitrios Zarpalas. Unsupervised interpretable basis extraction for concept-based visual explanations. *arXiv preprint arXiv:2303.10523*, 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

[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

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

[27] 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

[28] 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

[29] 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

[30] 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

[31] 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