Multi-Grained Interpretable Network for Image Recognition

Peiyu Yang, Zeyi Wen and Ajmal Mian
University of Western Australia, Perth, Australia
Email: peiyu.yang@research.uwa.edu.au; {zeyi, ajmal}@uwa.edu.au

***Abstract*—Given a classification problem with a large number of classes, humans often compare features at different granularities from coarse to fine to gradually recognize an object. However, current deep models are generally trained to directly make the final prediction, focusing on improving the ability of the network to extract features without considering the interpretability of the model. In this paper, we propose a multi-grained interpretable network to imitate the reasoning process of humans. The proposed network is equipped with techniques to assign images with multi-grained labels, so as to train a tree-structured classifier that learns features at different levels of granularity. The proposed method can hierarchically classify objects in images at different granularities, while providing a decision pathway with multi-grained explanations for practitioners. Experimental results demonstrate that our method achieves competitive prediction accuracy on CUB-200-2011 and Stanford Cars datasets, and simultaneously produces high-quality explanations of its decisions. Moreover, our method shows higher robustness of the learned features to adversarial examples generated by the FGSM and PGD attacks.**

I. Introduction

Despite their great success, the decision-making process of current deep models lacks interpretability, which hinders their applicability to high-stake problems in areas such as healthcare and finance. To provide explanations for deep models, many explainability methods [1], [2], [3] have been proposed to decipher the predictions made by deep models. However, these post-hoc methods are unable to provide sufficient details for explaining the complicated decision pathway of the black-box model. Instead of explaining a black-box model, many research works [4], [5], [6] aim to construct a self-explainable model. Chen et al. [6] proposed a transparent model by replacing the conventional extractive reasoning process with a case-based reasoning process, which compares the similarity between the input features and learned visual feature vectors called "prototypes" to make predictions. Due to the transparency of the case-based reasoning architecture, the prototypes are also extended to other problems including hierarchical classification and zero-shot classification [7], [8]. However, humans tend to hierarchically compare features of different granularities to recognize objects [9], [10] as shown in the top of Figure 1, which most of the current deep models fail to imitate. Current deep learning models often make their predictions for all the classes in a single go, as shown in the bottom of Figure 1. Predicting classes in one layer without distinction impedes the model to extract distinctive features and hinders humans from understanding the decision-making process of models.

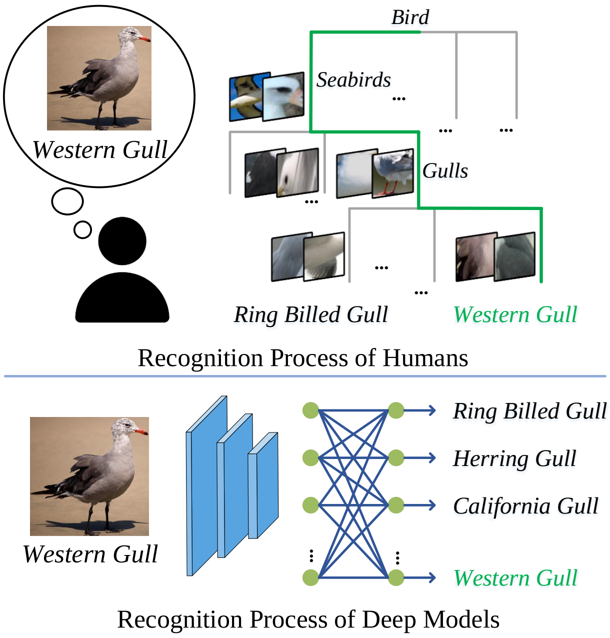


Fig. 1. Comparison of the image recognition process performed by humans and the deep model.

In this paper, we propose a multi-grained interpretable network to hierarchically classify objects at different granularities. For an arbitrary dataset with a larger number of classes, our method organizes the classes into different granularities within a class hierarchy tree. In this tree, classes with similar features are abstracted under the same parent node as a parent class label of the input image. To simulate the process of hierarchical recognition performed by humans, we construct a tree-structured classifier based on the class hierarchy tree. We initialize the prototype feature vectors for each node in the class hierarchy tree to separately learn features at different granularities. For each input example, our method looks for the node where the input example belongs at each level of the tree-structured classifier, and calculates the corresponding loss at different granularities simultaneously during training. When performing inference, our method first compares the coarse-grained features to predict a high-level class label. Then the fine-grained features are gradually exploited to achieve the final precise prediction. Moreover, our method provides a decision pathway with different granularity explanations for practitioners.

We evaluate our method on two challenging fine-grained classification datasets including CUB-200-2011 [11] and Stanford Cars [12]. Experimental results show that our method can achieve competitive performance compared to the state-of-the-art methods. Moreover, our model is robust when encountering adversarial examples generated by two popular attacks: PGD [13] and FGSM [14]. To summarize, we make the following three major contributions:

- We propose techniques to organize classes into different granularities within a class hierarchy tree. In this tree, we can assign multi-grained labels to input images.
- We develop a tree-structured classifier to hierarchically classify input images, enabling the features to be learned at different granularities. The prediction process provides a decision pathway with different granularity explanations for practitioners to interpret the decision-making process of our model.
- We conduct experiments on two challenging datasets and their corresponding adversarial examples to evaluate our method. Experimental results show that our method improves the predictive accuracy and provides high-quality explanations, while learning robust features for resilience to adversarial attacks.

II. Related Work

Explainable artificial intelligence (XAI) has attracted much attention within the deep learning community in recent years. Many XAI research studies aim to develop approaches that provide reasons behind a decision-making process. We can categorize the XAI approaches into (i) post-hoc explainability techniques and (ii) transparent models.

A. Post-hoc Explainability Techniques

Post-hoc techniques design algorithms to explain black-box deep learning models including model-specific and model-agnostic approaches. Model-agnostic approaches can be extended for applicability to any model. LIME [2] learns a linear model locally around the prediction for its explainability. SHAP [3] employs the game theory to assign each feature an importance value for an individual prediction. In addition to these local explanation methods, another method [15] trains a transparent model, such as a decision tree, to approximate a complex black-box model while maintaining high accuracy, which is also model-agnostic. On the other hand, model-specific approaches are proposed to explain specific models. For example, DeconvNet was proposed to construct the maximum activation maps to locate the most effective parts of an image for decisions made by a CNN [16]. To visualize the strongest activations of the input image, many approaches [1], [17], [18], [19] have been developed to generate a saliency map by assigning an importance score for each input to explain the prediction of deep models.

However, due to the complicated decision pathway of the black-box model, it is difficult for the post-hoc methods to provide sufficient details in their explanations. Therefore, more research works focus on the development of transparent models compared to post-hoc techniques.

B. Transparent Models

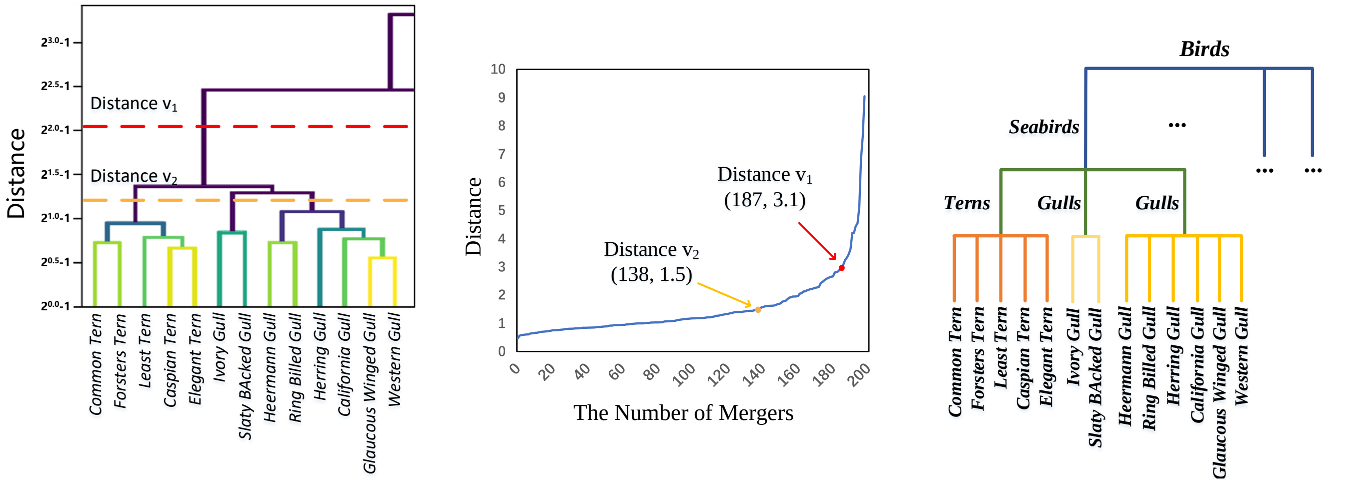
Transparent models aim to design transparent rules or architectures to make the model self-explainable. Ross et al. [4] trained models with input gradient penalties to change the decision boundary. Thus, the method ensures that it can make the right prediction based on the right reasons. Similarly, Schramowski et al. [20] proposed an interactive learning method to correct the model's ambiguity related errors. To construct a model with transparent rules, another line of work [21] incorporates prior knowledge into the model training process which enables the model to have strong generalization ability and interpretability.

In addition to transparent rules, many research works focus on improving the conventional architecture of deep models to imitate the human reasoning process. Inspired by the human vision system, a model named Saccader [22] combines both the BagNet [23] and hard attention mechanism for classification. To construct a model with reasoning process in a human-understandable way, an autoencoder network [5] uses case-based reasoning instead of conventional extractive reasoning to generate explanations. Compared with the autoencoder network, ProtoPNet [6] performs classification by learning prototype feature vectors containing the local features in a more flexible way. ProtoTree [24] is a variant of ProtoPNet, which arranges the prototypes in a binary tree structure to improve the interpretability. Inspired by the interpretability of the prototype vectors, other methods [8], [7] extend the case-based reasoning networks to other tasks including hierarchical classification and zero-shot learning.

Our proposed method further improves the case-based reasoning process to simulate the hierarchical reasoning process of humans. Although existing works [25], [26] also employ cascade structure to improve the decision pathway of deep models, our model trained with multi-grained labels exhibits strong interpretability to provide hierarchical explanations.

III. Methodology

In this section, we elaborate the details of our proposed multi-grained interpretable network. For obtaining multi-grained labels to train a hierarchical classification network, our method first organizes the classes into different granularities with a class hierarchy tree. In this tree, the classes are abstracted into parent classes with different granularities as nodes in the tree, which enables an input image with a fine-grained label to have multi-grained labels. Moreover, we initialize the visual features named prototypes for each node of the class hierarchy tree to construct a tree-structured classifier. The prototypes of our model are separately supervised by multi-grained labels, which simultaneously decreases the



(a) The merging process of the agglomerative clustering is represented by a binary tree.

(b) The curve shows the relationship between the distance and the number of mergers.

(c) The constructed class hierarchy tree on CUB-200-2011 dataset.

Fig. 2. A class hierarchy tree with a depth of 4 is built by merging the clustering binary tree with two distances.

coupling between features and enables prototypes to learn more distinctive features.

During inference, our method first exploits the coarse features to predict a high-level class label, and then gradually absorbs finer features to reach a detailed level prediction. In the process, we can find a branch of the tree that corresponds to a decision pathway of the prediction. Consequently, our method provides a decision pathway with multi-grained explanations for practitioners to better understand the inference process of the model. Below, we provide details of each key component of our proposed network.

A. *Class Hierarchy Tree Construction*

Current deep models are often trained to output an accurate prediction from a large number of classes in one layer, which not only degrades their performance but also increases the coupling between the learned features. To train a network to imitate the reasoning process of humans, we organize the classes into different granularities to obtain multi-grained labels for each input image.

To organize the classes into different granularities, the representation of each class needs to be obtained first. Similar to the ProtoPNet [6], we train prototypes in a classification task to learn the local features of a class. The prototype can be viewed as a generalized convolution, which computes distance instead of the inner product of the conventional convolution. Compared with the general convolution, the prototype is trained to be close to the local features of a certain class, which more directly represents the corresponding class in a latent space. Therefore, we initialize m prototypes from $\{p_i\}_{i=1}^{m}$ to learn the local features of a class c. We employ the weights of the last fully connected layer w_{fc} as the feature importance to aggregate the prototypes. For the prototype p_i of a class c, we compute the centroid of prototypes to represent the class as

$$r^c = \frac{\sum_{i=c\cdot m}^{(c+1)\cdot m} p_i \cdot w_{fc}^{(i,c)}}{\sum_{i=c\cdot m}^{(c+1)\cdot m} w_{fc}^{(i,c)}}, \quad p_i \in P_c, \tag{1}$$

where the representation r^c is defined for the class c.

After obtaining the representations of classes, we organize the classes into different granularities based on their similarity. We employ the agglomerative clustering method to group the classes. Agglomerative clustering is one of the most widely used hierarchical clustering algorithms where clustering is performed using a bottom-up approach. Agglomerative clustering successively merges two clusters with the smallest distance and the final clusters are obtained by constraining the number of merges by a given distance. The merging process of the agglomerative clustering can be represented by a binary tree as shown in Figure 2a. The color of the node containing two branches from light to dark indicates the distance from small to large when the two clusters are merged.

The distance of the agglomerative clustering can be regarded as the tolerance of the example dissimilarity within the cluster. The clusters generated by different distances can be considered to belong to different levels of granularities. To determine the distances for organizing classes, we draw a curve (see Figure 2b) to show the relationship between the increasing distance and the number of clusters. Since choosing the points around the *knee of a curve* is a common heuristic in mathematical optimization, we select two distances v_1 and v_2 around the inflection point which means that shrinking rewards are not worth the additional cost.

Next, we construct a class hierarchy tree through the interrelationship between the clusters. We can observe a property of the agglomerative clustering from Figure 2a that cluster results built by a small distance v_2 (orange dashed line) is the subset of the results built by v_1 (red dashed line). Thus, we can transform the binary tree of Figure 2a to a non-binary class hierarchy tree by gradually merging the classes into a root

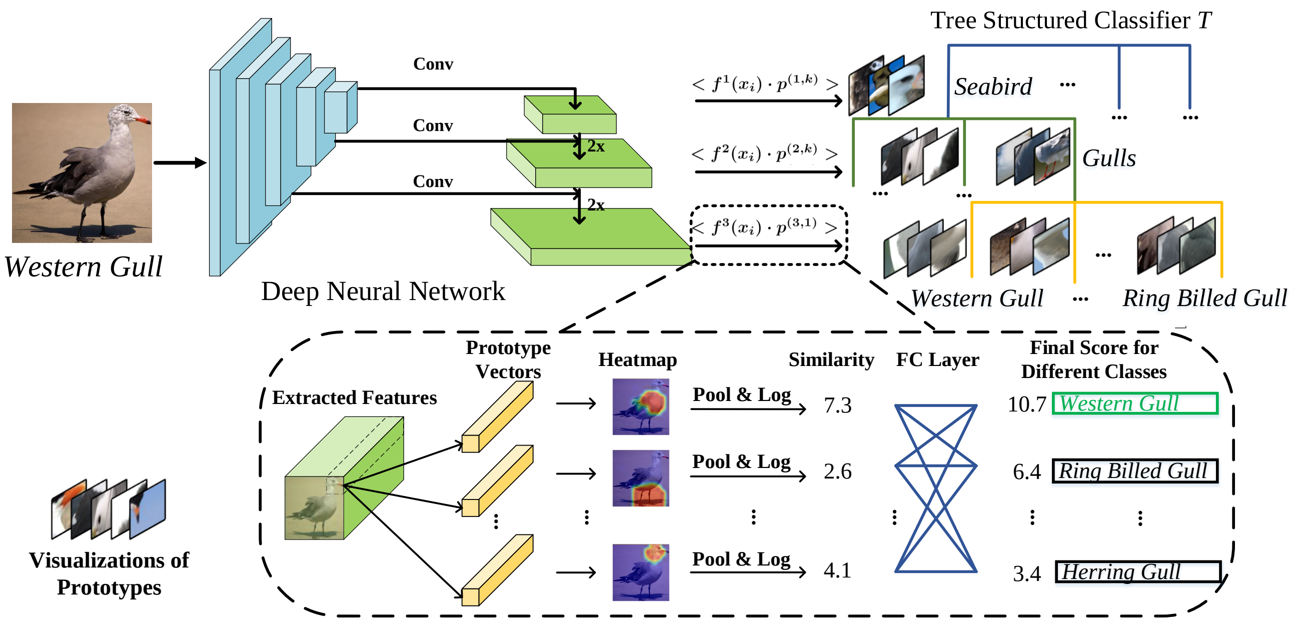


Fig. 3. Architecture of the proposed multi-grained interpretable network.

class, as shown in Figure 2c. It can be observed from the label names that the clustering results can effectively group similar classes into one cluster on CUB-200-2011 [11]. For example, the birds named *Tern* and *Gull* are grouped into different parent nodes respectively.

In the class hierarchy tree, we organize classes into different granularities. For each fine-grained class, we can find a pathway from a coarse granularity to a fine granularity from the tree. As a result, we can assign each fine-grained class with labels at different levels of granularity.

B. Multiple Grained Tree-Structured Classifier

Considering the recognition process where humans hierarchically compare features at different granularities, our method mimics this decision-making process by constructing a multi-grained tree-structured classifier. Similar to the decision-making process for a classification problem of humans, the proposed classifier gradually absorbs finer information to reach a more detailed level prediction.

The architecture of the proposed multi-grained interpretable image recognition network is shown in Figure 3. Given an input image containing a bird named *Western Gull*, a deep convolutional neural network f is first employed to extract deep features $f^h(x)$ in different scaled feature maps for training features in the h-th granularity. We further construct a top-down architecture to fuse different scaled feature maps for extracting richer features. Then, we initialize the prototypes for each node of the class hierarchy tree to construct a tree-structured classifier T. For the k-th node in the h-th level of the classifier T, we initialize m prototypes $\{p_i^{(h,k)}\}_{i=1}^{m}$. These prototypes only learn features at the granularity of the h-th level of the tree to refine the input to a finer granularity.

The detailed reasoning process is shown in the bottom of Figure 3. Firstly, the distance between the prototype $p_i^{(h,k)}$ with each patch of the extracted features $f^h(x)$ is computed to produce a heatmap. Instead of computing the inner product, the heatmap is computed by calculating the Euclidean distance between the different patches of the features $f^h(x)$ and the prototype $p_i^{(h,k)}$. Then a pooling operation is used to find patches that have the closest distance with different prototypes. An activation function is defined as follows to convert a small distance to a large similarity score $s_{p_i^{(h,k)}}$.

$$s_{p_i^{(h,k)}} = \max_{f_i^h(x) \in patch(f^h(x))} log(\frac{(\|f_i^h(x) - p_i^{(h,k)}\|_2^2 + 1)}{(\|f_i^h(x) - p_i^{(h,k)}\|_2^2 + \epsilon)}) \quad (2)$$

At last, a fully connected layer of the k-th node in the h-level of classifier T gathers the similarity score as the final prediction score for a precise class.

To summarize, we construct a tree-structured classifier to imitate the hierarchical reasoning process of humans. Given an input image containing a bird *Western Gull*, the network classifies the input into a rough child class *Seabird* in the first level of T. Then the input is further classified into a more precise type named *Gulls* by the child node classifier of *Seabird*. In the last level, the network distinguishes *Western Gull* from *Gulls* by comparing the fine-grained features. Since the prototype learns features in a set of similar classes, the prototypes are able to learn discriminative features to distinguish similar classes.

C. Training Algorithm

Here, we provide the details of the training algorithm for our proposed multi-grained explainable network. In our method, the loss function is calculated at all levels of the multi-grained network. For a batch of n images with labels $\{(x_i, y_i)\}_{i=1}^{n}$, the loss function ℓ that we aim to optimize defined as follows.

$$\ell = \min_{c_k^h(y_i) \in c(h,k)} (\frac{1}{n} \sum_{i=1}^{n} E(f^h(x_i) \cdot p^{(h,k)}, c_k^h(y_i))) \quad (3)$$

where E denotes a cross-entropy loss in the h-th level, which is used to punish the error between the prediction results and labels in each hierarchy. $c_k^h()$ is a function to generate the multi-grained label of y_i for training the h-th level and the k-th node of the classifier T. The term $c(h, k)$ is a target label set for the branch in the h-th level and the k-th node. If the input label $c_k^h(y_i)$ belongs to this branch, a cross-entropy loss is computed between the prediction $f^h(x_i) \cdot p^{(h,k)}$ and the label $c_k^h(y_i)$.

For the optimization process, we employ a two-stage optimization. In the first stage, we load the pre-trained parameters of the backbone network and learned prototype vectors from ProtoPNet [6] to initialize our tree-structured classifier. Then we fix the backbone network and train the prototypes and newly added layers. In the second stage, we jointly optimize the prototypes and the convolution neural network. In this way, the prototypes can be calibrated by using multi-grained labels.

IV. Experiments

We conduct experiments on two widely used datasets to evaluate the predictive accuracy and interpretability of our proposed method. Then we conduct experiments on adversarial examples to study the robustness of our method.

A. *Experiment on CUB-200-2011*

Caltech-UCSD Birds dataset (CUB-200-2011) [11] is a challenging fine-grained classification dataset containing 11,788 images of 200 bird species. We crop the images using the provided bounding boxes and resize them to 224×224. For a fair comparison, all compared methods are trained and tested on the same images. In addition, we set the same number of prototypes for each class for both our model and ProtoPNet [6].

To evaluate the performance of our method, we compare its accuracy with both ProtoPNet and ProtoTree [24]. ProtoTree is also an interpretable network that trains prototypes to explain the reasoning process of the model. For a fair comparison, all the models are both trained and tested on the same cropped images. Table I shows the accuracy comparison between our method and the other two explainable models on both VGGNet-19 [27] and ResNet-34 [28]. We can observe that our method outperforms ProtoPNet by 2.0% and 0.8% on the two backbone networks, respectively. Compared with ProtoPNet, the prototypes of our model are separately learned in different granularities to hierarchically classify objects, which effectively improves the performance of our model. Although ProtoTree employs additional datasets iNaturalist [29] and more powerful network ResNet-50, our method can also achieve competitive results. We also achieve a competitive accuracy of 81.2% in comparison with the state-of-the-art black-box models such as PA-CNN [30] and MG-CNN [31] with the accuracy of 82.8% and 83.0% respectively. In addition, compared with the long decision pathway of ProtoTree, the shallow non-binary tree structure of our model is easier for humans to understand.

TABLE I
Accuracy comparison on CUB-200-2011 Dataset.

Model	Backbone	Accuracy	Depth
Our Model	VGG-19	78.8±0.1%	h=4
ProtoPNet	VGG-19	78.0±0.2%	h=1
Our Model	ResNet-34	81.2±0.1%	h=4
ProtoPNet	ResNet-34	79.2±0.1%	h=1
ProtoTree (+iNaturelist)	ResNet-50	82.2±0.7%	h=9

B. *Experiment on Stanford Cars*

The Stanford Cars is another challenging fine-grained classification dataset [12] containing 16,185 images of 196 types of cars. Table II shows the accuracy comparison between our model and another two explainable models including ProtoPNet and ProtoTree. It can be seen that our method outperforms ProtoPNet and ProtoTree on both ResNet-34 and VGGNet-19. Compared with two state-of-the-art black-box models such as MDTP [32] and PA-CNN with the accuracy of 92.5% and 92.8%, our model can achieve a competitive accuracy of 89.2% without using any other part level annotations.

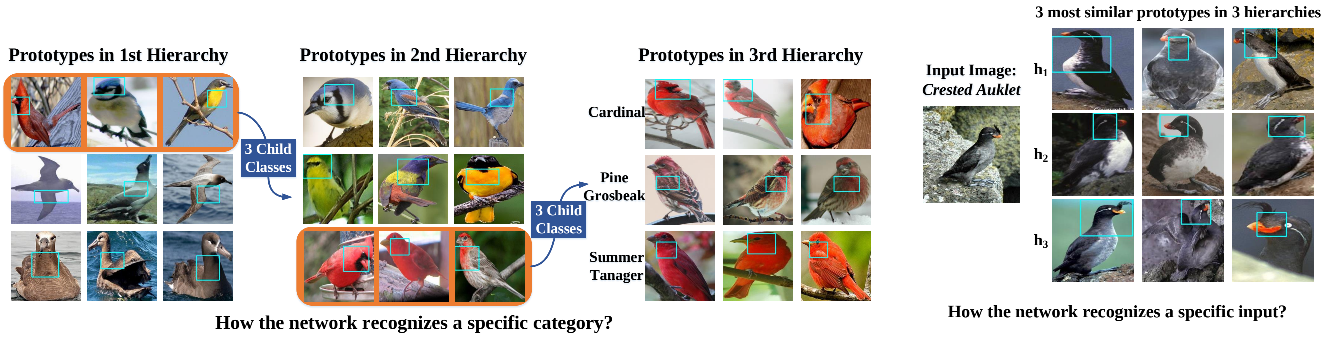
TABLE II
Accuracy comparison on Stanford Car Dataset.

Model	Backbone	Accuracy	Depth
Our Model	VGGNet-19	89.2±0.2%	h=4
ProtoPNet	VGGNet-19	87.4±0.3%	h=1
Our Model	ResNet-34	87.4±0.1%	h=4
ProtoPNet	ResNet-34	86.1±0.1%	h=1
ProtoTree	ResNet-50	86.6±0.2%	h=11

C. *Experiment on Adversarial Examples*

The susceptibility of deep networks to adversarial attacks is a serious concern [33]. Adversarial examples contain carefully crafted perturbations that are quasi-imperceptible to the human observer but drastically change the network decisions to incorrect labels. Hence, it is important to test the robustness of deep models to adversarial examples.

Here, we examine two popular attack methods FGSM [14] and PGD [13] to generate the adversarial examples. FGSM and PGD both need to access the gradients of the model to generate the examples. Unlike FGSM, which uses the gradient of one iteration to perturb the original example, PGD uses multiple iterations to generate perturbations. PGD is considered as one of the strongest attacks [33]. For the adversarial examples generated by FGSM, we set the value of perturbation magnitude ϵ to 0.02 for both our model and ProtoPNet on CUB-200-2011 and Stanford Cars respectively. For the PGD attack method, we set ϵ to 0.03 and 0.001 to generate the perturbation on CUB-200-2011 and Stanford Cars respectively. Accuracy comparison on adversarial examples generated by both FGSM and PGD attacks is presented in Table III. After


Prototypes in 1st Hierarchy
Prototypes in 2nd Hierarchy
Prototypes in 3rd Hierarchy
3 Child Classes
3 Child Classes
Cardinal
Pine Grosbeak
Summer Tanager
How the network recognizes a specific category?
3 most similar prototypes in 3 hierarchies
Input Image: Crested Auklet
h1
h2
h3
How the network recognizes a specific input?

(a) The global explanation.

(b) The local explanation.

Fig. 4. The visual examples of multi-grained global and local explanations provided by our model.

the attack by FGSM and PGD, ProtoPNet degrades accuracy by 29.3% and 26.5%, which is more significant than our method at 25.5% and 24.4% on CUB-200-2011 dataset. On Stanford Cars dataset, our method degrades accuracy by 29.8% and 23.9%, which outperforms the accuracy decay of the ProtoPNet by 68.5% and 51.6% by a large margin. Since both our model and the ProtoPNet are all trained on original images of the Stanford Cars dataset, the images contain more noise from larger background regions in comparison with the cropped images of the CUB-200-2011 dataset. Therefore, the accuracy of the ProtoPNet degrades significantly on the Stanford Cars Dataset. However, our multi-grained network enables the model to learn more robust features for resilience to adversarial attacks. Moreover, the structure of our model does not depend on the single last layer to make a prediction, which is more robust for the gradient-based attack.

TABLE III
EXPERIMENTAL RESULTS ON ADVERSARIAL EXAMPLES GENERATED BY TWO ATTACKS ON RESNET-34.

Model	Attack Method	Bef. Attack	Aft. Attack	Dataset
Our Model	FGSM	81.2%	55.7%	CUB
Our Model	PGD	81.2%	56.8%	CUB
ProtoPNet	FGSM	79.2%	49.9%	CUB
ProtoPNet	PGD	79.2%	52.7%	CUB
Our Model	FGSM	87.4%	57.6%	Cars
Our Model	PGD	87.4%	63.5%	Cars
ProtoPNet	FGSM	86.1%	17.6%	Cars
ProtoPNet	PGD	86.1%	34.5%	Cars

D. *Providing Explanations*

We visualize the prototypes to provide both global and local explanations of our model. For each prototype, we first scan the images of the training dataset to find which image patch is the most similar to the prototype in the latent space. Then we use the most similar patch of the image in the RGB space to represent the visualization of a prototype. The reason is that if an image patch is the most similar to the prototype in the latent space, the patch of the image can be highly activated by the prototype in the latent space during the inference process. Specifically, we employ a bounding box to mark the top 5% activated regions in the heatmap to represent the visualization of a prototype.

Global Explanations. The global explanations aim to provide an understanding of how the network recognizes a specific category. Figure 4a shows visualization of prototypes in three hierarchies for three species of birds including *Cardinal*, *Pine Grosbeak* and *Summer Tanager*. We can observe that our method learns prototypes with distinctive features in different hierarchies. For the prototypes in the first hierarchy, The feathers of the birds in the first column have more vivid colors. In its three child classes, different birds are divided into different groups according to their colors. Finally, the prototype can learn their own distinctive features among the three similar red birds. For example, the prototypes of *Cardinal* show the head in red and black. On the other hand, prototypes of *Pine Grosbeak* show bodies with patterns of red and black.

Local Explanations. The local explanation aims to provide explanations for an individual prediction. Figure 4b shows three most similar prototypes on three hierarchies for predicting the input *Crested Auklet*. We can observe that our model can gradually learn more detailed features in different hierarchies for the input *Crested Auklet*.

V. CONCLUSION

In this paper, we have proposed a multi-grained interpretable network to imitate the hierarchical reasoning process of humans. The proposed model can gradually absorb finer information to make a final prediction. We further propose a method to organize classes into different granularities to assign each input image with multi-grained class labels enabling the features to be optimized in different granularities. Thus, our method can provide practitioners with multi-grained explanations of the decision-making process. Experimental results have shown that our method can learn robust features in different granularities to improve classification accuracy while providing high-quality explanations as well as higher resistance to adversarial attacks.

ACKNOWLEDGMENT

Professor Ajmal Mian is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government.

References

[1] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *International Conference on Knowledge Discovery & Data Mining*, 2016.

[3] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[4] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," in *International Joint Conference on Artificial Intelligence*, 2017.

[5] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[6] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. Su, "This looks like that: Deep learning for interpretable image recognition," in *Advances in Neural Information Processing Systems*, 2019.

[7] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable image recognition with hierarchical prototypes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[8] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, "Attribute prototype network for zero-shot learning," in *Advances in Neural Information Processing Systems*, 2020.

[9] B. A. Purcell and R. Kiani, "Hierarchical decision processes that operate over distinct timescales underlie choice and changes in strategy," *Proceedings of the National Aacademy of Sciences*, 2016.

[10] G. Wan, S. Pan, C. Gong, C. Zhou, and G. Haffari, "Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning," in *International Joint Conferences on Artificial Intelligence*, 2021.

[11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," *California Institute of Technology*, 2011.

[12] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks." in *International Conference on Learning Representations*, 2018.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[15] O. Bastani, C. Kim, and H. Bastani, "Interpretability via model extraction," *arXiv preprint arXiv:1706.09773*, 2017.

[16] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[17] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.

[19] M. A. A. K. Jalwana, N. Akhtar, M. Bennamoun, and A. Mian, "CAMERAS: enhanced resolution and sanity preserving class activation mapping for image saliency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[20] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting, "Making deep neural networks right for the right scientific reasons by interacting with their explanations," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 476–486, 2020.

[21] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee, "Improving performance of deep learning models with axiomatic attribution priors and expected gradients," *Nature Machine Intelligence*, pp. 1–12, 2021.

[22] G. F. Elsayed, S. Kornblith, and Q. V. Le, "Saccader: Improving accuracy of hard attention models for vision," in *Advances in Neural Information Processing Systems*, 2019.

[23] W. Brendel and M. Bethge, "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," in *International Conference on Learning Representations*, 2019.

[24] M. Nauta, R. van Bree, and C. Seifert, "Neural prototype trees for interpretable fine-grained image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[25] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," in *Pattern Recognition*, 2003, pp. 297–304.

[26] L. D. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[29] G. V. Horn, O. M. Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. J. Belongie, "The inaturalist species classification and detection dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[30] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[31] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[32] Y. Wang, J. Choi, V. Morariu, and L. S. Davis, "Mining discriminative triplets of patches for fine-grained classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[33] N. Akhtar and A. S. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, 2018.