DProtoNet: Decoupling the inference module and the explanation module enables neural networks to have better accuracy and interpretability

Yitao Peng, Yihang Liu, Longzhen Yang, Lianghua He*
College of Electronic and Information Engineering Tongji University
4800 Cao’an Highway, Shanghai, China 201804
{pyt, 2111131, yanglongzhen, helianghua}@tongji.edu.cn

Abstract
The interpretation of decisions made by neural networks is the focus of recent research. In the previous method, by modifying the architecture of the neural network, the network simulates the human reasoning process, that is, by finding the decision elements to make decisions, so that the network has the interpretability of the reasoning process. The specific interpretable architecture will limit the fitting space of the network, resulting in a decrease in the classification performance of the network, unstable convergence, and general interpretability. We propose DProtoNet (Decoupling Prototypical network), it stores the decision basis of the neural network by using feature masks, and it uses Multiple Dynamic Masks (MDM) to explain the decision basis for feature mask retention. It decouples the neural network inference module from the interpretation module, and removes the specific architectural limitations of the interpretable network, so that the decision-making architecture of the network retains the original network architecture as much as possible, making the neural network more expressive, and greatly improving the interpretability. Classification performance and interpretability of explanatory networks. We propose to replace the prototype learning of a single image with the prototype learning of multiple images, which makes the prototype robust, improves the convergence speed of network training, and makes the accuracy of the network more stable during the learning process. We test on multiple datasets, DProtoNet can improve the accuracy of recent advanced interpretable network models by 5% to 10%, and its classification performance is comparable to that of backbone networks without interpretability. It also achieves the state of the art in interpretability performance.
1 Introduction

With the continuous development of neural networks [17, 20, 6, 7, 1, 10, 11], the interpretability of neural networks is a research direction that has received extensive attention. How to make neural networks have good classification performance and good interpretability at the same time is a challenging task. A large number of interpretability methods have been proposed, they are saliency map methods [16, 1, 22, 15] and interpretability models [2, 19, 18, 8] methods. The saliency map method generates the areas that the neural network pays attention to when making decisions by comprehensively analyzing the activation information of each convolutional layer when the neural network predicts. These methods are post-hoc attempts to explain already-trained models [12], which are normally interpretable and not easily understood by humans. They enable the network to learn feature templates for each category in the dataset, which are called prototypes, and predict the corresponding category by finding prototypes that are similar to a category. This kind of network belongs to the network with an interpretable architecture, it makes the network simulate the human way of thinking, which in turn makes the network interpretable, and the learning of these networks only requires the category labels of the images.

ProtoPNet, Gen-ProtoPNet and XProtoNet use $1 \times 1 \times D$, $h \times w \times D$, and Occurrence map as prototypes, respectively, to correspond to the feature patch in the original image. The above method is based on an assumption: in the feature map generated after the original image is encoded by a neural network, a tensor composed of elements of all channels at the same length and width positions represents a certain patch of the original image. However, there is no complete theory to support this hypothesis. The constraints of this hypothesis limit the learning ability of the neural network, and the network performance is generally lower than the unexplainable neural network [17, 6, 7] as the backbone.

We propose to use feature mask of the same type as feature map to mine prototypes in feature maps, each element of which consists of a value from 0 to 1, as shown in Figure 1. The prototypes mined by feature masks cover all the prototype cases that the above method [2, 18, 8] can produce. It makes the prototype of network learning not limited to a tensor composed of elements of all channels at the same position, but uses any combination of all elements of the entire feature map as a prototype to represent the corresponding feature patch on the original image. This greatly improves the expressiveness of the prototype, so that the fitting ability of the network is not constrained by the above assumptions, and the accuracy of the interpretable network DProtoNet can be almost the same as or even exceed the classification accuracy of the uninterpretable network as a backbone.

The previous interpretable neural network [2, 19, 18, 8] upsamples the feature map and the activation map of the prototype to the size of the original image, and the position with a large activation degree is used as the corresponding feature patch area of the prototype when the neural network is classified. They assume that when the feature map is affinely transformed into the original image,
the information of a certain location area on the feature map can represent the information of the corresponding location area of the original image. However, there is no complete theory to support the above assumptions. Therefore, the interpretability of this method is poor, and the localization of saliency map regions pointed out by these methods is not accurate.

We propose using Multiple Dynamic Masks (MDM) [13] to decode feature maps and feature masks generated tensor prototypes. By detecting the activation of the feature mask at a specific location on the feature map, we can mine the regions of the original image that DProtoNet focuses on when making decisions. This separates the activation maps of features and prototypes, which are common modules that participate in the reasoning process and interpretation process in the neural network, avoiding the need to learn the location information on the original map and limit the network performance. The above design of DProtoNet enables the network to retain the classification performance of the original backbone unexplainable network. Due to the introduction of the MDM method, the network’s learning and finding of prototypes are interpretable. We pay attention to each element of the feature map when generating the prototype, which makes the prototype representation region localization mined by the network in the original image more accurate, and the saliency map has less noise.

The learning of the prototype by the interpretable network in the early stage is to mine the tensor most similar to the current prototype in the corresponding feature map of a single image in the entire dataset through the network as an update of the new prototype. Since a single image may contain noisy information, the learned prototype may learn noise, causing the network performance to plummet. Therefore, we propose to replace the prototype learning of a single image with the prototype learning of multiple images. We generate several enhanced versions of the same image through data enhancement, and they must have the same prototype features. By mixing the prototype information from these multiple images, the learned prototype has robustness, accuracy and generalization. Avoid prototype introduction of noisy information. It represents the distribution of prototypes of a certain type of feature, rather than the prototype corresponding to a specific single template. This solves the problem that the prototype may learn noise, and makes the learned prototype more accurate and generalized, which in turn makes the network training performance more stable and the network learning convergence speed improved.

As shown in Figure 1, the DProtoNet we designed simulates the reasoning process of human beings. People observe specific feature regions by adding a mask of the area of interest to the original image, and compare the similarity with the template known by their own cognition. Determines whether the region is similar to the template. The original image is network encoded to generate a feature map, and feature masks are used to mask the area of interest and pooled to generate a feature vector, which is compared with the template prototype stored by itself, and a decision is made according to whether the feature vector is similar to the prototype template. The introduced Multiple Dynamic Masks Decoder generates the area that the network pays attention to on the original
image as a decision explanation.

By introducing feature masks and MDM method, we minimize the constraints imposed by the interpretable architecture on the neural network to maximize the fitting performance of the network. Let the network not only have good classification accuracy, but also have good interpretability. Note that we use the neural network encoder as a black box, and any neural network of any structure can be loaded in this plate, so DProtoNet can be applied to neural networks of any structure and is universal.

- We propose to use the feature mask to extract the internal information of the feature map as the prototype tensor, so that the prototype can extract the information in the feature map as a whole, improve the representation ability of the prototype, and then improve the classification accuracy of the neural network based on the prototype decision.

- We introduce the Multiple Dynamic Masks method into DProtoNet as a Decoder to explain the decision region of the neural network. The feature-prototype activation diagram, which is used as a neural network reasoning module and an explanation module, is coupled, which improves the classification accuracy and interpretability of the network, and avoids the need for the feature map to learn the location information on the original image, resulting in limited network performance.

- We propose to replace prototype learning from single-image prototype learning to multi-image prototype learning to avoid the problem of learn-
ing noisy prototypes during prototype learning. Make the learned prototype robust, accurate and generalizable. It improves the convergence speed of the network and makes the network training more stable.

2 Related Work

2.1 Saliency Maps

People point out the areas of interest for network decisions by generating Class Activation Map (CAM) for the network as saliency maps. Zhou [24] used Global Average Pooling (GAP) to integrate the information of all features in the whole space to obtain CAM. Selvaraju [16] proposed Grad-CAM, which uses gradient information flowing to the last convolutional layer in a CNN to understand the importance of each neuron for class recognition. This method is more general in acquiring CAM. Chattopadhay [1] proposed Grad-CAM++, which adds an extra weight to measure the elements of the gradient map. It enables more precise positioning of the CAM. In 2020, in order to solve the problem of neural network gradient noise, saturation and easy detection of false confidence samples, Wang [22] proposed a gradient-free method, namely Score CAM, which bridges the gap between Perturbation-based and CAM-based methods, and represent the weights of the activation maps in an intuitive and understandable way. Ramaswamy [15] proposed Ablation-CAM, which analyzes the contribution of each factor to the network and mines the factors that affect the classification decision.

The derivation of the above method is not easy to understand for humans, and the model lacks generality. These methods are post-hoc attempts to interpret the trained model, the interpretation methods are not adaptive, and the CAM location of the search will be partially biased. To this end, we propose to use the MDM method to decode the feature map and the corresponding prototype mask, so as to make the searched CAM more accurate and more interpretable than explaining the region concerned by the network decision.

2.2 Interpretable Models

Setting the structure of the neural network to mimic the human reasoning process makes the network interpretable. Their prediction process itself is interpretable, and no additional operations are required to analyze and interpret the inference network. The interpretability network determines categories by finding regions that are partially similar to stored templates.

Chien [2] proposed ProtoPNet, which sets the prototype as a 1 × 1 patch on the feature map, which is taken as the local features of the image, and visualizes the prototype by replacing them with the most similar training data patches. Singh [19] proposes NP-ProtoPNet, which fixes the last classification layer and exploits negative reasoning to improve the classification performance of the explainability network. Singh [18] proposed Gen-ProtoPNet, which improves the representation ability of the prototype and the classification of the network
Figure 2: Overall architecture of DProtoNet. DProtoNet distinguishes image categories by comparing the features of an input image to the prototypes of each classification.

by setting the prototype as the h×w patch on the feature map. Kim [8] proposed XProtoNet, which sets the prototype as a feature vector with variable activation positions and sizes, which further improves the classification performance of the network. The above interpretability network upsamples the similarity activation between the feature map and the prototype to the size of the original image as a saliency map to represent the area that the neural network pays attention to when classifying. There is no complete theory to support the similarity activation map upsampling of the feature map prototype to represent the area of interest in the neural network, and this method does not have theoretical interpretability. The interpretable architecture of the neural network limits the fitting space of the network, the classification performance of the network will be worse than the network classification performance as a backbone.

Therefore, we generate the prototype by multiplying the feature map and the feature mask and then taking the global average to generate the prototype, so that the prototype utilizes all the feature map information, and then the representation ability of the prototype is improved, and the classification accuracy of the network is improved. It also provides more accurate prototype tensors for the analysis of subsequent decision-making regions, which improves the interpretability of the network.

3 Methodology

DProtoNet decides the object category by comparing the similarity between the input image and the specific representation features of each type of object. As shown in Figure 2, it consists of four parts: feature extractor \( f_{encoder} \), prototype layer \( g_p \), full connected layer \( h \), and output logits.

Feature extractor \( f_{encoder} \) is composed of a baseline network \( f_b \) and a shaping network \( f_a \). The add layer constitutes the shaping network. It converts the input image into a fixed-shape feature map. The prototype layer \( g_p \) compares the masked and pooled feature vector of the feature map with the prototype
DProtoNet uses feature mask and feature map to generate a feature vector $z_i(x)$ and compare it with the prototype.

corresponding to each category, and calculates the maximum similarity activation between it and each prototype. Finally, classify the maximum similarity activation in full connected layer $h$ to get output logits.

Let: the dataset has $m$ categories, and each category $c$ has $K_c$ prototype. The total number of prototypes is $K = \sum_{i=1}^{N} K_c$. Input image $x \in \mathbb{R}^{H \times W \times C}$, the neural network encoder consists of the encoder layer and the add layer of the selected backbone neural network. The add layer is the shape of the tensor after adjusting the encoder layer. Obtain feature map $R^{H_1 \times W_1 \times D}$ through neural network encoder. Set $n$ feature masks $\{M_i\}_{i=1}^{n}$, $M_i \in \mathbb{R}^{H_1 \times W_1 \times D}$, which are used to mine the prototype information inside the feature map. As shown in Figure 3, the feature map $F(x)$ is multiplied by the feature mask $M_i$ and then subjected to Global Average Pooling (GAP) to generate feature vector $z_i(x)$.

$$z_i(x) = \text{GAP}(M_i F(x)) \quad (1)$$

The prototype layer calculates a similarity score between $z_i(x)$ and $P_j$. Similarity score $s$ is calculated using square similarity as:

$$s(z_i(x), P_j) = ||z_i(x) - P_j||_2^2 \quad (2)$$

The similarity score is used to describe the similarity between the tensor extracted in the feature map and the prototype tensor. The smaller the similarity score, the more similar it is to the prototype, and the greater the activation
value for classification. So we set an activation function $g$. It represents the similarity of the feature tensor mined in $F(x)$ that is most similar to the prototype $P_j$, and it represents the similarity between the original image corresponding to the tensor $F(x)$ and the image corresponding to the prototype.

$$g(F(x), P_j) = \max_{1 \leq i \leq n} \log((s(z_i(x), P_j) + 1)/(s(z_i(x), P_j) + \epsilon))$$ (3)

After the feature map is multiplied by each mask, the tensor obtained through GAP and the most similar activation value of each prototype tensor are used as the input of the final classification layer, and they pass through the fully connected layer to obtain the final prediction value for each category.

$$p(y^c|x) = \sum_{j=1}^{K} w^c_j g(F(x), P_j)$$ (4)

Where similarity activation $g$ indicates how similar the feature of the input image is to each prototype, and weight $w^c_j$ indicates how important each prototype $P_j$ is for the class $c$.

After the inference process of the network, DProtoNet makes category prediction by judging the similarity between the feature map of a certain area in the original image and the prototype.

During the training process, prototype $P_j$ is replaced with the mixed tensor of the most similar feature vector $z_i(x)$ in multiple images from the training images. These prototypes and corresponding feature masks can be visualized as interpretable images through Multiple Dynamic Masks Decoder.

### 3.1 Training Scheme

Our training process are divided into four phases:

1) stochastic gradient descent (SGD) of all layers, called initial stage; 2) stochastic gradient descent (SGD) of all layers, called jointly stage; 3) projection of prototypes, called push stage; 4) convex optimization of last layer, called last stage. We train the DProtoNet by cycling these four stages. The algorithm flow is performed according to the [2] method.

Note: The neural network encoder is $f_{encoder}$ whose network parameters are recorded as $w_{encoder}$, the baseline network that constitutes $f_{encoder}$ is $f_b$, its network parameter is $w_h$, the shaping network is $f_a$, and its network parameter is $w_a$. The prototypes $P = \{p_j\}_{j=1}^{K}$ in the prototype layer is $g_p$. The parameter of the fully connected layer, that is, the last layer $f_h$, is $w_h$. The set training images is: $D = [X,Y] = \{(x_i,y_y)\}_{i=1}^{n}$, there are $m$ categories in total, $Q_k$ represents the set of prototype $p_j$ belonging to category $k$. $z_k(x)$ represents the tensor obtained after the product of the feature map $F(x)$ and the mask $M_k$ through GAP, $z_k(\cdot) = GAP(M_kF(\cdot))$. Let $w_h^{(u,v)}$ be the $(u,v)$-th entry in $w_h$ that corresponds to the weight connection between the output of the $v$-th prototype unit $g_{p_v}$ and the logit of class $u$. 

8
3.1.1 Initial stage and jointly stage

In initial stage, we retain the ability of the baseline network \( f_b \) to extract features. The parameter \( w_b \) of the fixed \( f_b \) is fixed, and the parameters \( w_h, w_p, w_a \) of the network \( f_h, g_p, f_b \) are trained by SGD. In jointly stage, we train the parameters \( w_h, w_p, w_b \) and \( w_a \) of all layers \( f_h, g_p, f_b \) and \( f_a \) with SGD. The objective function we want to optimize is:

\[
\text{In initial stage:} \quad \min_{w_p, w_a, w_h} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(f_h \circ g_p \circ F(x_i), y_i) + \lambda_1 Clst + \lambda_2 Sep + \lambda_3 l_{wh} \tag{5}
\]

\[
\text{In jointly stage:} \quad \min_{w_p, w_h, w_a, w_h} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(f_h \circ g_p \circ F(x_i), y_i) + \lambda_1 Clst + \lambda_2 Sep + \lambda_3 l_{wh} \tag{6}
\]

where \( Clst \), \( Sep \) and \( l_{wh} \) are defined by:

\[
Clst = \frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \in Q_{y_i}} \min_k \| z_k(x) - p_j \|_2^2 \tag{7}
\]

\[
Sep = -\frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \notin Q_{y_i}} \min_k \| z_k(x) - p_j \|_2^2 \tag{8}
\]

\[
l_{wh} = \sum_{u=1}^{m} \sum_{j: p_j \notin Q_k} | w_h^{(u,v)} | \tag{9}
\]

Cross-entropy loss (CrsEnt) penalizes misclassification of training data. Clustering cost minimization (Clst) encourages each training image to have some latent feature regions that are at least close to a prototype of its own class, while separation cost minimization (Sep) encourages each latent patch of the training image to be far away a prototype of a non-self class. By constraining \( l_{wh} \), we try to let the prototypes that only belong to the \( u \) class participate in the classification decision, while the prototypes that do not belong to the \( u \) class try not to participate in the classification decision, avoid negative reasoning, and make the network interpretable. The ultimate goal is to make the last layer connection \( w_h^{(u,v)} \approx 0 \) satisfying \( p_v \notin Q_u \) for \( u \) and \( v \).

For example, we don’t want the negative reasoning situation of “it is judged that the bird belongs to class \( k_0 \) because the bird does not belong to class \( k \)” (class \( k \) is the class whose whole is not \( k_0 \)).

For any class \( u \), we set \( w_h^{(u,v)} = 1 \) to satisfy \( p_v \in Q_u \) for all \( v \) and \( w_h^{(u,v)} = -0.5 \) to satisfy \( p_v \notin Q_u \) for all \( v \) in initialization phase before training starts. In push stage:
For the projection of the prototype $p_j$, our projection method is different from \cite{2, 19, 18, 8} projecting the area of a single image. If the image quality is not good or the network classification performance is poor, the network training will be unstable. Therefore, we choose to mix the patch features on multiple images to generate a projection of the prototype, as shown in Figure 4.

Let: image $x_i$ belongs category $C$, \{$x_1^i, x_2^i, \ldots, x_R^i$\} is a group of pictures generated by $x_i$ is the original image after data augmentation. Let $p_j \in Q_C$. We believe that the original data $x_i$ has the same characteristics as the data-augmented $x_r^i$($r \in \{1, 2, \ldots, R\}$).

We sum and project the patches most similar to the prototype $p_j$ in each $x_r^i$ as update of $p_j$. Mathematically, for prototype $p_j$ of class $k$, i.e., $p_j \in Q_k$, we perform the following update:

$$e_r = \arg\min_e ||z_e(x_r^i) - p_j||_2$$ \hspace{1cm} (10)

$$p_j = \arg\min_p \sum_{r=1}^R ||z_e(x_r^i) - p||_2$$ \hspace{1cm} (11)

Such a new prototype contains a data-augmented mixture of all $p_j$ prototypes, and the $p_j$ generated by mixing multiple images is more robust than the $p_j$ generated by a single image. (11) According to the derivation, it can be known that the $p_j$ update formula is:

$$p_j = \frac{1}{R} \sum_{r=1}^R z_e(x_r^i)$$ \hspace{1cm} (12)

In last stage:

After the operation of the prototype projection stage, the last stage is used to adjust the fully connected layer of the last layer to optimize the classification.
Figure 5: The Multiple Dynamic Masks Decoder architecture consists of two components: Multiple Dynamic Masks and Mask Generator. Multiple Dynamic Masks mines the decision area for images in the neural network according to feature masks and generates mask vectors. The Mask Generator determines the prototype area of the image through the mask vector generation network of multiple dimensions. From blue to red, the activation degree increase.

performance. We fix all the layers before the last layer, and only update the parameter \( w_{wh} \) of the last layer. The optimization function is as follows:

\[
\min_{w_{wh}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(f_h \circ g_p \circ F(x_i), y_i) + \lambda_1 \text{ Clst} + \lambda_2 \text{ Sep} + \lambda_3 l_{w_{wh}} \tag{13}
\]

3.2 Multiple Dynamic Masks Decoder

As shown in Figure 5, it consists of Multiple Dynamic Masks and Mask Generator. It can use the neural network encoder, the prototype tensor and its corresponding feature map mask to visualize the prototype as the image area represented in the original dataset. It can indicate that the feature map in the input image \( x \) that is most similar to the prototype tensor corresponds to the decision region in the original image, which tells us that the neural network makes predictions based on which regions in the original image are similar to the corresponding prototype template.

3.2.1 Multiple Dynamic Masks

Let: Mask vectors \( \{d_i\}_{i=1}^{P} \), \( d_i \in \mathbb{R}^{a_i \times b_i \times 1} \), \( d_i \) initialized to a fixed value \( \epsilon \). For any \( i, j \in \{1, 2, ..., N\} \), if \( i \neq j \) then \( a_i \neq a_j \) or \( b_i \neq b_j \). Upsample function \( g(\cdot) \),
\( g(d_i) \in R^{H \times W \times 1} \), \( d_i \) of different sizes is upsampled to \( g(d_i) \) as a mask to mask the picture, and the CAM is generated through mask stacking to represent the area that the network pays attention to when making decisions.

Find the area most similar to the prototype \( p_t \) in the image \( x \) through MDM, generate the original image corresponding to the prototype \( p_t \) and denote it as \( x^{p_t} \), generate the maximum activation feature map mask of the prototype and denote it as \( M^{p_t} \). The selected activation position is the tensor position of the prototype \( p_t \) in DProtoNet, which serves as the attention position of the consistent activation of the neural network.

Note: \( \{d^x_i\}_{i=1}^D \) and \( \{d^{x^{p_t}}_i\}_{i=1}^D \) represent the feature masks generated based on the input image \( x \) and the input image \( x^{p_t} \), respectively.

We constrain the value of the mask so that the mask focuses on the regions that are most important to the decision by keeping the network close to activations at selected locations \( p_t \). We are able to train mask vectors \( d^x_i \), \( d^{x^{p_t}}_i \) by minimizing the following losses.

\[
L_x = \min_{d^x_i} || \text{GAP}(F(g(d^x_i)x)M^{p_t}) - \text{GAP}(F(x)M^{p_t}) ||_2^2 + \eta_x \sum_{i=1}^{a_i} \sum_{v=1}^{b_i} \frac{d^x_{iuv}}{|a_ib_i|} \tag{14}
\]

\[
L_{x^{p_t}} = \min_{d^{x^{p_t}}_i} || \text{GAP}(F(g(d^{x^{p_t}}_ix^{p_t})x^{p_t})M^{p_t}) - p_t ||_2^2 + \eta_{x^{p_t}} \sum_{i=1}^{a_i} \sum_{v=1}^{b_i} \frac{d^{x^{p_t}}_{iuv}}{|a_ib_i|} \tag{15}
\]

Here \( \eta \) is a hyperparameter and we train mask vectors \( d^x_i \), \( d^{x^{p_t}}_i \) by minimizing \( L_x \) and \( L_{x^{p_t}} \). The masks generated after upsampling of these vectors can preserve the regions of the prototype in the original image, and remove the regions that are not related to the prototype.

### 3.2.2 Mask Generation

The mask sets \( L_x \) and \( L_{x^{p_t}} \) trained by the two loss functions \( d^x_i \) and \( d^{x^{p_t}}_i \) in (14) and (15). After upsampling and stacking, the normalized tensor and the original tensor are dot-multiplied to get CAM after removing the noise points below the threshold \( \gamma \). Let CAM of \( x \), \( x^{p_t} \) are \( A^x \), \( A^{x^{p_t}} \) respectively.

\[
A^x = \text{Normalize}(\{ \sum_{i=1}^{D} g(d^x_i) \geq \gamma \} \sum_{i=1}^{D} g(d^x_i)) \tag{16}
\]

\[
A^{x^{p_t}} = \text{Normalize}(\{ \sum_{i=1}^{D} g(d^{x^{p_t}}_i) \geq \gamma \} \sum_{i=1}^{D} g(d^{x^{p_t}}_i)) \tag{17}
\]

CAM \( A^x \) and \( A^{x^{p_t}} \) are generated by multiple mask blending. \( A^x \) and \( A^{x^{p_t}} \) respectively represent the CAM of the feature regions of the original image \( x \) and the original corresponding image \( x^{p_t} \) which are most similar to the original
Figure 6: DProtoNet inference process visualization.

\( P_t \). \( \gamma \) is threshold, \( \{ \cdot \} \) represents a truth-valued function, 1 if true, 0 otherwise. 

\[ \text{Normalize}(X) = \frac{X - \min(X)}{\max(X) - \min(X)} \]

is the normalization function.

Mask and heatmaps are obtained by overlaying or multiplying the CAM and the original image, which are used to explain the regions of interest for neural network classification.

\begin{equation}
A^x_h = \alpha x + \beta A^x, A^{p_t}_h = \alpha x^{p_t} + \beta A^{x^{p_t}}
\end{equation}

\begin{equation}
A^x_b = A^x x, A^{p_t}_b = A^{x^{p_t}} x^{p_t}
\end{equation}

Here \( A^x_h \) and \( A^{p_t}_h \) respectively represent the heat map of the feature area similar to the feature \( p_t \) in the original image \( x \) and the heat map of the feature area corresponding to the prototype \( p_t \) in the dataset. \( A^x_b \) and \( A^{p_t}_b \) respectively represent the corresponding binary mask map. \( \alpha, \beta \) are the hyperparameters used to mix the original image and the CAM to generate the heatmap.
Figure 7: Deletion and Insertion curves generated on above methods.

4 Experiments

4.1 Datasets

4.1.1 CUB200-2011

It [21] is a bird dataset for testing fine-grained classification, with a total of 11,788 bird images including 200 different classes of birds. Among them, there are 5994 images in the training set and 5794 images in the test set. Each image in the test set has its corresponding segmentation label image, and the bird is divided as the foreground, and the rest are the background. We use the data-augmented training set to train the model and test the classification accuracy of the model on the test set. We generate a binary mask for the explanatory regional saliency map found by each model according to the set threshold, and compare the binary mask with the segmentation foreground map of the original data on each saliency map evaluation index. Test the interpretability and localization performance of saliency maps for fundus pathological regions.

4.1.2 Stanford Cars

The car dataset [9] contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class is nearly split on both training and testing sets. We use the data-augmented
| Dataset                | CUB200-2011         |
|------------------------|---------------------|
| **Model**              | **Backbone**        | ResNet50 | VGG19 | DenseNet121 |
| ProtoPNet              | 62.6                | 64.2     | 67.7  |
| NP-ProtoPNet           | 60.8                | 61.7     | 52.6  |
| Gen-ProtoPNet          | 58.6                | 63.1     | 64.2  |
| XProtoNet              | 67.1                | 65.5     | 69.9  |
| DProtoNet(ours)        | **75.5**            | **67.3** | **76.3** |

| Dataset                | Stanford Cars       |
|------------------------|---------------------|
| **Model**              | **Backbone**        | ResNet50 | VGG19 | DenseNet121 |
| ProtoPNet              | 78.2                | 79.8     | 79.9  |
| NP-ProtoPNet           | 76.3                | 76.1     | 71.1  |
| Gen-ProtoPNet          | 77.1                | 75.2     | 77.8  |
| XProtoNet              | 77.3                | 79.3     | 80.8  |
| DProtoNet(ours)        | **85.5**            | **85.1** | **86.4** |

Table 1: The accuracy of each network with ResNet50, VGG19, DenseNet121 as the backbone is compared in the CUB200-2011 and Stanford Cars datasets.

training set to train the model and test the model classification performance on the test set.

### 4.1.3 RSNA Pneumonia Detection

The dataset [23] contains frontal-view CT images of lungs in DICOM format. The annotated part contains the bounding box of the pneumonia region. It contains 26,684 training data and 3,000 test data. We test the classification accuracy of the model on this dataset and compare the localization accuracy of the classified explanatory region saliency maps generated by each model in the bounding box of this dataset.

### 4.1.4 iChallenge-PM

It [5] is a medical dataset about Pathologic Myopia (PM) provided in the iChallenge competition jointly organized by Baidu Brain and Sun Yat-Sen University Sun Yat-Sen Eye Center. There are 400 test data sets each, and each pathological picture in the test set has a pathological segmentation map of the picture. We train the model on the training set with data augmentation, and test the accuracy of the model in classifying pathological myopia pictures on the test set. And compared the distribution of the saliency map searched by each model in the pathological segmentation map, and tested the localization performance of the saliency map to the pathological region of the fundus.
Figure 8: In ResNet50, nine methods corresponding to the curves of Dice Coefficient, IOU, PPV and Sensitivity when the importance percentile of the masked image pixels are masked from 0 to 99.

4.2 Evaluation

We compare the classification accuracy of our proposed DProtoNet with recent advanced interpretability networks on the above four datasets. In order to evaluate the interpretability and localization performance of the saliency maps found by each model. We use Average Drop, Average Increase by [1]; Deletion, Insertion by [14]; Proportion by [22]; Dice, IOU, PPV, Sensitivity to measure the solvability and positioning accuracy of the saliency map generated by each model.

Accuracy is expressed as: \[\frac{\text{number of correct predictions}}{\text{total number of cases}}.\] The Average Drop is expressed as: \[\frac{\sum_{i=1}^{N} \max(0, Y_i^c - O_i^c)}{Y_i^c} \times 100.\] The Average Increase is expressed as: \[\frac{1}{N} \sum_{i=1}^{N} \text{Sign}(Y_i^c < O_i^c).\] \(Y_i^c\) denotes the prediction score of class \(c\) in the original image \(i\), and \(O_i^c\) denotes the prediction score of class \(c\) in the explained map obtained after masking in the original image. The symbol represents an indicator function that returns 1 if the input is true. We removed certain percentile pixels of the original image to generate a explained map. Deletion and insertion measures are the deletion and insertion of pixels from the original image in descending order of CAM activation value, respectively, and generate the area under the probability curve (AUC) described by the predicted probability result of the deleted or inserted image.

Proportion is expressed as: \[\frac{\sum_{(i,j)\in \text{bbox}} L_{(i,j)}^c}{\sum_{(i,j)\in \text{bbox}} L_{(i,j)}^c + \sum_{(i,j)\in \text{bbox}} L_{(i,j)}^m},\] where \(\text{bbox}\) is the bound-
Figure 9: In DenseNet121, nine methods corresponding to the curves of Dice Coefficient, IOU, PPV and Sensitivity when the importance percentile of the masked image pixels are masked from 0 to 99.

We believe that the neural network should have larger activation for regions where classification predictions are valid and smaller activation for regions where classification predictions are invalid. We set the activation value of a certain percentile of the CAM as a threshold and use this threshold to generate a binary mask.

True (TP) is the number of items correctly marked as belonging to the positive class. True negatives (TN) measure the proportion of negatives that are correctly identified as negatives. False positives (FP) are the number of items incorrectly labeled as belonging to the positive category. False negatives (FN) are the number of items incorrectly marked as not belonging to the positive category, see [19].

We adopt Dice Coefficient \( = \frac{2TP}{2TP + FP + FN} \), IOU \( = \frac{TP}{TP + TP + FN} \), PPV = \( \frac{TP}{TP} \) and Sensitivity = \( \frac{TP}{TP} \) computed from foreground images and binary
| Dataset   | iChallenge-PM | RSNA |
|-----------|--------------|------|
| **Model** | **Evaluation** | **Accuracy** | **Proportion** |
| ProtoPNet | 97           | 73.2  |
| NP-ProtoPNet | 97.25   | 76.4  |
| Gen-ProtoPNet | 97.5     | 75.6  |
| XProtoNet | 98           | 77.1  |
| DProtoNet (ours) | **98.5** | **82.2** |

Table 2: The accuracy and proportion of each network with ResNet50 as the backbone is compared in the iChallenge-PM and RSNA datasets.

masks as evaluation metrics for saliency map interpretability. Accuracy can also be expressed as $\frac{TP + TN}{TP + TN + FP + FN}$.

### 4.3 Experimental Details

We set up four datasets with 10 prototypes for each category, $K_c = 10, c \in \{1, 2, ..., m\}$. We perform data enhancement on the training set in the dataset. In the above four datasets, each image is rotated, perspective, sheared, and distorted to generate 10 augmented images each, and a total of 40 data augmented images are generated for each image. All the above dataset images are cropped to 224x224. Add layer consist of two 1 x 1 convolutional layers with ReLU activation between them. Hyperparameters are derived using five-fold cross-validation, $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 3$, $\epsilon = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = -0.08$, $\lambda_3 = 1e-4$.

We set $D = 10$, $\{d_i\}_{i=1}^{10}$, $d_i \in R^{a_i \times b_i \times 1}$, $a_i = b_i = 5 + i$, $\eta_i = 10$, $i \in \{1, 2, ..., 10\}$. We use the Adam optimizer, the learning rates of backbone layer $f_b$, add layer $f_a$, prototype projection layer $g_p$, and last layer $f_h$ in the interpretability network are set to $1e-4$, $3e-3$, $3e-3$, $1e-4$, respectively. The parameters of the network with ResNet50, VGG19, DenseNet121 as the backbone are all initialized to the values pre-trained on imagenet. The number of prototype channels is set to 128. The number of train batch is set to 40. In the training phase of Multiple dynamic masks vector, each vector is trained for 800 iterations. The epoch of DProtoNet training is set to 100, and the initial stage is the first 5 epochs. After that, it is a joint stage, and a prototype projection is performed every 10 epochs. In examining the evaluation index Proportion, the binary mask thresholds are set to the top 50%, 30% and 5% on the CUB200-2011, RSNA.
| Mod   | Eva | AD(%) | AI(%) | Deletion | Insertion | Proportion(%) |
|-------|-----|-------|-------|----------|-----------|---------------|
| Grad-CAM | Eva | 27.8  | 14.2  | 0.134    | 0.339     | 86.1          |
| Grad-CAM++ | Eva | 67.3  | 3.7   | 0.078    | 0.359     | 88.4          |
| Score-CAM | Eva | 44.5  | 13.2  | 0.088    | 0.391     | 85.1          |
| Ablation-CAM | Eva | 82.4  | 4.9   | 0.286    | 0.312     | 55.6          |
| ProtoPNet | Eva | 75.1  | 3.2   | 0.038    | 0.349     | 62.6          |
| NP-ProtoPNet | Eva | 45.7  | 11.5  | 0.161    | 0.271     | 76.5          |
| Gen-ProtoPNet | Eva | 55.2  | 15.9  | 0.135    | 0.284     | 73.8          |
| XProtoNet | Eva | 76.1  | 4.7   | 0.121    | 0.292     | 69.1          |
| DProtoNet(ours) | Eva | 17.5  | 21.1  | 0.028    | 0.709     | 94.3          |

Table 3: Evaluated results on Recognition. Take ResNet50 as the backbone, comparative evaluation in terms of Average Drop(AD), Average Increase(AI), Deletion scores, Insertion scores and Proportion. Eva and Mod represent Evaluation and Model, respectively.

and iChallenge-PM datasets, respectively. All experimental data are the results of running all networks on the above datasets and configurations. All models are trained on 1 2080Ti GPU.

4.4 Comparison with Baselines

We use ResNet50 [6], VGG19 [17], DenseNet121 [7] as the network backbone respectively, and test the interpretable neural networks ProtoPNet [2], NP-ProtoPNet [19], Gen-ProtoPNet [18], XProtoNet [8] and our proposed DProtoNet on the CUB200-2011 [21] dataset and Stanford Cars [9] datasets, compare the classification accuracy of the networks and the performance of generating explanatory saliency maps. In RSNA [23] and iChallenge-PM [5] medical images, we use ResNet50 [6] as the backbone to compare the classification accuracy and saliency map localization ability of the above 5 networks.

We compare the saliency maps generated by the above five networks and the four methods of Grad-CAM [16], Grad-CAM++ [11], Score-CAM [22], Ablation-CAM [15] with ResNet50, DenseNet121 as the baseline network on the above nine evaluation indicators Contrast, compare the interpretability and localization ability of the saliency maps generated by the network.

4.5 Visualization

We use ResNet50 as the baseline network, saliency maps generated by Grad-CAM, Grad-CAM++, Score-CAM, Ablation-CAM and using ResNet50 as backbone: ProtoPNet, NP-ProtoPNet, Gen-ProtoPNet, XProtoNet, DProtoNet, five saliency maps generated by interpretable neural networks for visual comparison.

We mixed the activation heatmap generated by the above nine models with
Table 4: Evaluated results on Recognition. Take DenseNet121 as the backbone, comparative evaluation in terms of Average Drop(AD), Average Increase(AI), Deletion scores, Insertion scores and Proportion. Eva and Mod represent Evaluation and Model, respectively.

| Dataset         | CUB200-2011 | Stanford Cars |
|-----------------|-------------|---------------|
| Base            | DProtoNet   | Baseline      |
| ResNet50        | 75.5        | 75.2          | 85.5 | 84.7 |
| VGG19           | 67.3        | 68.3          | 85.1 | 84.2 |
| DenseNet121     | 76.3        | 75.5          | 86.4 | 85.8 |
| Dataset         | iChallenge-PM | RSNA          |
| Base            | DProtoNet   | Baseline      |
| ResNet50        | 98.5        | 98.25         | 82.2 | 78.6 |
| VGG19           | 98.25       | 97.5          | 79.3 | 72.2 |
| DenseNet121     | 99.25       | 98.75         | 80.6 | 78.3 |

Table 5: Accuracy comparison on CUB200-2011, Stanford Cars, iChallenge-PM, RSNA.

5 Discussion

In Figure 6, we visualize the inference process of the prototype, the network infers the category of the picture by finding some similar prototypes. It conforms to the structure designed by DProtoNet, by finding similar prototype tensors in the feature map, and classifying images through prototype similarity.

We compare the classification performance of DProtoNet with recent ad-
Figure 10: (a), (b), (c) and (d), (e), (f) are interpretable networks composed of ResNet50, VGG19, DenseNet121 as backbones on CUB200-2011 and Stanford Cars datasets, respectively. The accuracy of the training situation is compared. Advanced interpretability networks [2, 19, 18, 8]. Table 1 and Table 2 show the classification performance of each network when the classification accuracy of network training converges on the interpretable network composed of ResNet50 and DenseNet121 on two traditional image datasets and two medical image datasets. DProtoNet improves classification accuracy by 5% to 8% over recent advanced networks, and the classification accuracy is the best among interpretable neural networks. Since DProtoNet uses the feature mask to mine the prototype in the feature map, it reduces the spatial limitation of the interpretable architecture on the network, improves the expression ability of the prototype and the learning ability of the neural network, and improves the classification accuracy of the network.

From Table 2, we compare the pathological localization ability of DProtoNet with recent advanced interpretability networks generated by Proportion. Among the medical datasets iChallenge-PM, RSNA, DProtoNet achieves the best localization ability for pathology. This shows that DProtoNet uses the feature mask to learn the prototype to make the prototype more accurate, and uses the MDM method to decode the feature mask prototype information, so that the network can improve the accuracy of the feature map mining information. The network’s generated saliency map is more accurate in its ability to localize pathological regions that explain the network’s decisions.
Figure 11: (a), (b) respectively represent the saliency maps obtained by the nine models on the CUB200-2011 dataset with DenseNet121 and ResNet50 as Backbone. From blue to red, the activation degree increases. (c) represents the saliency map binary masks, original images and labels of the five models on the iChallenge-PM dataset. (d) represents the decision area of each model on the RSNA dataset, the yellow box is the lesion area corresponding to the model decision area, and the red box is the real pathological area.
In order to test the search performance of the network for decision regions. We compare the saliency maps generated by DProtoNet, XProtoNet, ProtoPNet, NP-ProtoPNet, Gen-ProtoPNet and the neural network as backbone using Grad-CAM, Grad-CAM++, Score-CAM, Ablation-CAM on each evaluation index. As shown in Table 3, Table 4 and Figure 7, on the CUB200-2011 test dataset, we compare the evaluation indicators Average Drop, Average Increase, Deletion Score, and Insertion Score. DProtoNet uses ResNet50 and DenseNet121 as the backbone to compare the above four evaluation indicators. The state-of-the-art models improved by 37.1%, 32.7%, 26.3%, 81.3% and 51.3%, 14.5%, 8.9%, 17.9%, respectively. By gradually adding or removing the currently most concerned regions in the original image, DProtoNet shows the most obvious change in image classification compared to other networks. The best performance is achieved in each evaluation index. This shows that the saliency map generated by DProtoNet has the best interpretability, which is in line with human cognition of network reasoning, that is, the more important picture information is given, the more accurate the network classification will be. It best satisfies the region where the network decision is more concerned, the greater the activation of the saliency map. The saliency map generated by DProtoNet can great reflect the importance of the regions that the neural network pays attention to when making decisions. The saliency map generated by DProtoNet can best reflect the importance of the area that the neural network pays attention to when making a decision, and it best satisfies the area that the network decision is more concerned about, and the greater the activation of the saliency map.

We measure the quality of the saliency maps generated by the above nine models by their localization ability. We examine how many ground-truth regions of the saliency map can fit into the bounding box or segmentation map of the target object in the binary map generated by a certain threshold mask. In Figures 8 and 9, in each network with ResNet50 and DenseNet121 as backbone, when the mask ratio is between 30% and 99%, 60% and 99% respectively, DProtoNet is in Dice Coefficient, IOU, PPV, Sensitivity The localization ability of the model is the best, and compared with the previous model, it has been greatly improved.

In Table 2, Table 3 and Table 4, we use Proportion to measure the localization ability of the saliency map on all four datasets. In the CUB200-2011 dataset, more than 90% of our generated saliency maps fall in the segmentation foreground map. In the iChallenge-PM dataset, since we set a high binary mask threshold, the saliency map localization reaches 20%, which is the best among all interpretability models. In the RSNA pneumonia detection dataset, more than 50% of the energy in the saliency map of DProtoNet falls within the ground truth bounding box of the target object, and the pathological region localization ability is superior to previous work. This confirms that the saliency map generated by DProtoNet has less noise, and the generated saliency map achieves the state of the art.

In Figure 10, we observe the classification accuracy of DProtoNet and recent interpretability networks at training time. DProtoNet has the fastest con-
vergence speed, and after the network performs prototype projection, there is almost no drop in accuracy, and the network performance remains relatively stable. However, after the prototype projection operation of other networks, the classification performance of the network decreases. This is because the prototype activation map acts as both an inference module and an explanation module, resulting in a balanced learning of location information and semantic information for feature maps. The expressive ability of the feature map decreases, which in turn leads to a decrease in the network classification performance.

DProtoNet separates the inference module from the explanation module through feature masks and MDM methods, which makes the network’s expressive ability and prototype learning effect stronger, and the network classification performance is almost unaffected. As shown in Figure 10, ProtoPNet in (a) and NP-ProtoPNet in (d), the accuracy decreases after each prototype projection. This is because it uses single-image prototype extraction, which may cause noise information in network learning. Learning as a prototype leads to progressively worse network performance. We adopt multi-image prototype learning, so that the network uses the noise information of the image as a prototype with a lower probability, and the prototype is used as a distribution containing a certain feature instead of a patch of a single image, which makes the prototype projection more robust and makes the network more robust. Training is more stable.

Note in Table 5, we compared DProtoNet and its corresponding backbone with three classical neural networks ResNet50, VGG19, DenseNet121 on the four datasets CUB200-2011, Stanford Cars, iChallenge-PM and RSNA. Baseline network, classification performance on these data. It is found that the classification performance of DProtoNet is similar to its baseline network classification performance on almost every data, and even exceeds the performance of the baseline network. DProtoNet uses the feature mask as a tensor that stores the spatial information of the prototype feature map. The structure of the neural network in the feature extraction layer is preserved as much as possible, and the representation of the prototype is generalized to the greatest extent. The fitting ability of the network is hardly affected, and the classification performance of the network can be greatly preserved.

As shown in Figure 11, we observe the visualization performance of each model. The saliency map found by DProtoNet on each model and task has good interpretability and localization ability.

6 Conclusion

DProtoNet decouples the activation map generated by the feature map and the prototype through the feature masks and MDM method, so that the reasoning process and the interpretation process are separated, so that the network can retain the classification performance of the baseline network as much as possible.

Its classification accuracy is among the state of the art among recent interpretable neural networks, and is comparable to the classification accuracy of uninterpretable neural networks as backbones.
DProtoNet uses feature masks to store prototype location information, which improves the generalization of prototypes, and uses MDM to mine prototype information more accurately, so that the saliency map generated by DProtoNet has the state of the art interpretability and localization performance.

DProtoNet adopts multi-image prototype learning instead of single-image prototype learning, which makes the network converge faster, the learned prototype can represent the distribution of certain features. The prototype training is more robust. The saliency maps generated by DProtoNet have less noise, better localization ability and interpretability. DProtoNet can put the encode module of any neural network into the network as a black box, so DProtoNet is universal. DProtoNet can have good classification accuracy and interpretability at the same time.

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE winter conference on applications of computer vision (WACV), pages 839–847. IEEE, 2018.

[2] 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.

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

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[5] Huazhu Fu, Fei Li, José Ignacio Orlando, Hrvoje Bogunovic, Xu Sun, Jinguo Liao, Yanwu Xu, Shaochong Zhang, and Xiulan Zhang. Palm: Pathologic myopia challenge. IEEE Dataport, 2019.

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

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.
[8] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: diagnosis in chest radiography with global and local explanations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 15719–15728, 2021.

[9] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision workshops, pages 554–561, 2013.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021.

[11] 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.

[12] Cristiano Patrício, João C Neves, and Luís F Teixeira. Explainable deep learning methods in medical diagnosis: A survey. arXiv preprint arXiv:2205.04766, 2022.

[13] Yitao Peng, Longzhen Yang, Yihang Liu, and Lianghua He. Mdm: Visual explanations for neural networks via multiple dynamic mask. arXiv preprint arXiv:2207.08046, 2022.

[14] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421, 2018.

[15] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 983–991, 2020.

[16] 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.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[18] Gurmail Singh and Kin-Choong Yow. An interpretable deep learning model for covid-19 detection with chest x-ray images. Ieee Access, 9:85198–85208, 2021.
[19] Gurmail Singh and Kin-Choong Yow. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access*, 9:41482–41493, 2021.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[22] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

[23] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[24] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.