Interpretable Image Recognition by Constructing Transparent Embedding Space

Jiaqi Wang, Huafeng Liu*, Xinyue Wang, Liping Jing*
School of Computer and Information Technology, Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
{jiaqi.wang, huafeng, xinyuewang, lpjing}@bjtu.edu.cn

Abstract

Humans usually explain their reasoning (e.g. classification) by dissecting the image and pointing out the evidence from these parts to the concepts in their minds. Inspired by this cognitive process, several part-level interpretable neural network architectures have been proposed to explain the predictions. However, they suffer from the complex data structure and confusing the effect of the individual part to output category. In this work, an interpretable image recognition deep network is designed by introducing a plug-in transparent embedding space (TesNet) to bridge the high-level input patches (e.g. CNN feature maps) and the output categories. This plug-in embedding space is spanned by transparent basis concepts which are constructed on the Grassmann manifold. These basis concepts are enforced to be category-aware and within-category concepts are orthogonal to each other, which makes sure the embedding space is disentangled. Meanwhile, each basis concept can be traced back to the particular image patches, thus they are transparent and friendly to explain the reasoning process. By comparing with state-of-the-art interpretable methods, TesNet is much more beneficial to classification tasks, esp. providing better interpretability on predictions and improve the final accuracy. The code is available at https://github.com/JackeyWang96/TesNet.

1. Introduction

Convolutional neural networks(CNNs) [20, 19, 29, 12, 14] have achieved surpassing performance in many visual tasks, such as image recognition and detection. However, besides the extraordinary discrimination power, CNNs and their corresponding results are still hard to explain, which severely limits their applications such as self-driving cars, diagnosis of cancer and etc. Recently, more and more interpretable methods have been proposed on CNNs, in order to open the black box of neural networks. Among them, an intuitive strategy is to visualize the feature representation hidden inside a CNN, but there is still a vast gap between network visualization and semantic interpretations for neural networks.

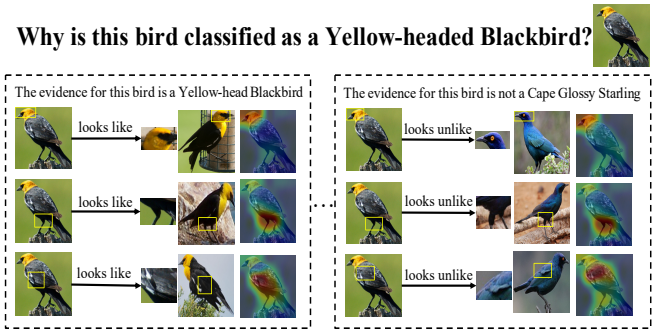

Figure 1: Image of a *Yellow-headed Blackbird* and humans usually explain their reasoning process through some parts of the image which looks like some learned concepts of the *Yellow-headed Blackbird* to classify the bird's species.

Considering the examples in Figure 1, how would you identify the bird image as a *Yellow-headed Blackbird* and not a *Cape Glossy Starling*? Maybe you find that the bird's head, legs, and feathers look like those concepts of *Yellow-headed Blackbird* rather than *Cape Glossy Starling*. In other words, you may gather the evidence in your mind and make a final decision. Specifically, humans usually explain their reasoning process by dissecting the image into object parts and pointing out the evidence from these identified parts to the concepts stored in his / her mind. Therefore, for the intelligent machine, it is an emergent issue to determine the object parts and construct the concepts in order to implement interpretable image classification.

To enforce the CNN-based classifier with interpretability, the basis concepts are introduced as a plug-in component in CNN architecture [1]. From the cognitive point of view [26], the interpretable concepts should cover the following characteristics: 1) Informative: the input data can be efficiently represented in the space spanned by the ba-

*Corresponding authors.

sis concepts, and its essential information is preserved in the new representation, 2) Diversity: each data point is related to only a few non-overlapping basis concepts, and the points belonging to one category are related to the similar subset of concepts, 3) Discriminative: the basis concepts are class-aware so that the categories can be separated well in the space of concepts.

For grasping the concepts, researchers take advantage of the high-level features in deep neural networks (e.g., CNN), on which the auto-encoding [1] or prototype learning [5, 15, 23] are operated. These existing methods can explicitly represent the inputs with basis concepts, i.e., they met the first requirement. Among them, some prior distribution such as *U-shaped Beta distribution* is adopted to limit the number of concepts to which the high-level features are related [15]. However, they suffer from the situation that the basis concepts may be entangled, which is not friendly to isolate the effect of individual concept to the input representation and the output category, and further destroys the classification performance.

In this study, thus, we focus on constructing the basis concepts simultaneously containing the above three characteristics. First, each category has its own basis concepts, and the corresponding concept subsets of different categories are as much different as possible. Second, a good mapping is built to provide a bridge between high-level features and basis concepts. Third, for the input image, the basis concepts are helpful to compute the final prediction score along all categories. To implement this, a Grassmann manifold is introduced to construct basis concepts. As shown in Figure 2, for each category, the subset of corresponding basis concepts is taken as a point on the Grassmann manifold. In this case, the basis concepts of one category are orthogonal to each other. Meanwhile, the class-aware concept subsets are part away from each other by constraining their projection metric. The above two constraints make sure that the basis concepts are disentangled. To improve the transparency of leaned basis concepts, the prototypical high-level patches of the original images are extracted to represent the concept.

In this work, an interpretable network architecture is designed by introducing a plug-in transparent embedding space (**TesNet**) which is spanned by transparent basis concepts which are constructed on the Grassmann manifold. These basis concepts are enforced to be category-aware and within-category concepts are orthogonal to each other, which makes sure the embedding space is disentangled. In order to demonstrate the model efficiency, we evaluate our model on two case studies, i.e bird species identification and car model identification. Extensive experiments demonstrate the broad applicability of our model on different CNN architectures. To investigate the proposed basis concept construction strategy, a series of ablation studies

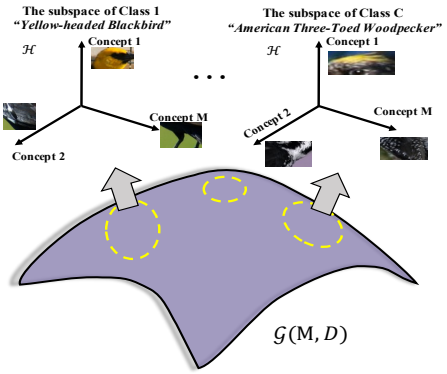

Figure 2: The illustration of constructing transparent embedding space on Grassmann Manifold. The space is spanned by category-aware transparent basis concepts. Each basis concept is demonstrated by its most related prototypical high-level patches.

have been conducted. As expected, our model achieves the state-of-the-art performance in terms of accuracy compared with interpretable methods providing the same level of interpretability.

The contribution of our work can be summarized as: 1) An interpretable neural network with transparent basis concepts (**TesNet**) is proposed to explain the prediction results semantically and quantitatively. 2) The basis concepts are category-aware and disentangled, which leverages discriminability and interpretability. 3) The latent space spanned by the basis concepts has the ability to represent the input and preserve the essential information. 4) The proposed transparent basis concepts construction layer is a generic solution and can be taken as a plug-in component for various CNN architectures.

2. Related work

We briefly survey relevant literature on post-hoc interpretability, interpretable representations learning.

Post-hoc interpretability has been extensively studied with building the mapping of the abstract patterns encoded by a pre-trained model into the human-understandable domains such as images (arrays of pixels) or texts (sequences of words) [25]. A great deal of previous work has focused on exploring the pattern hidden inside the neural units by assigning neurons importance scores and visualizing on images. These methods can be broadly divided into perturbation-based forward propagation approaches [34, 44, 45], which make perturbations on the input and observe the importance impact on later neurons, and backpropagation-based approaches [29, 30, 2, 35, 24, 31, 28] which propagate the importance signal from the output neuron to the input through each layer in the network. Some work disen-

tangle the representations of the pre-trained convolutional neural network into an explanatory graph [37, 36] or a decision tree [39, 3] or concept activation vectors [16, 9, 40] obtaining clear visual patterns Post-hoc interpretation provides a tool to understand a tangle of patterns in trained network, but these methods do not fundamentally address the problem of pattern entanglement, while our work is dedicated to learning clear visual patterns during the training process.

Interpretable representations learning aims to learn clear semantic representations rather than a black box during the training process of the network. Bau et al. [4] defined six types of semantics for CNNs, i.e. objects, parts, scenes, textures, materials, and colors. Many interpretable deep models handle with high-level convolutional layers to learn the disentangled concepts of objects and parts.Some work encourages each filter respond to a specific concept by adding regularization terms. For instance, Zhang et al. [38] designed a regularization loss to obtain disentangled representations by restricting each filter response to a specific object part in high-level convolutional layers. Liang et al. [21] designed a learnable sparse Class-Specific Gate structure to encourage each filter to respond to only one (or few) class. Chen et al. [6] introduced a module named *concept whitening (CW)* which axes of the latent space are aligned with pre-defined concepts. However, the space of possible concepts can be unclear or even unlimited, which severely limits the discovery power of CW. There are also some new network architectures being designed to encode a specific semantic concept. For example, Sabour et al. [27] designed a novel network architecture named Capsule Networks where each capsule outputs an activity vector encoding an object or an object part, instead of a scalar.

The proposed TesNet relates closely to methods [1, 5, 15] that aim to learn basis concepts and make a final decision based on these concepts. Alvarez-Melis and Jaakkola [1] proposed a self-explaining model by taking advantage of the auto-encoding technique to learn basis concepts and assigning relevance scores to basis concepts for prediction. Huang et al. [15] assumed a simple prior (Beta distribution) about the occurrence of object part concepts and explicitly encoded the generic concepts of object parts across categories by region grouping, i.e., all categories share the same set of concepts. To make the concepts discriminative, ProtoPNet [5] is proposed to learn the prototypical parts for each class and trace the evidence from prototypes to image patches. However, it is limited by the prototype learning in L^2-distance. Specifically, the prototypes are implicitly assumed following a Gaussian distribution, which is not proper for the complex data structure. Meanwhile, it can not explicitly ensure the learned prototypes are disentangled, which is an important property for interpretable learning.

Our work also relates to the methods that build attention-based interpretability into CNNs. These methods [17, 33, 41, 42] only expose which parts of an input the network focused on when making decision, but they don't point out that which parts they focus on are similar to the learned concepts. In other words, it is controversial whether these parts under attention are semantic concepts. In contrast, our proposed model provides a bridge between the high-level input parts and the learned basis concepts, thus, it is friendly to detect which parts are important for prediction and explicit to determine the meaning of the basis concepts with the aid of their most related image parts.

3. Method

In this section, we will describe the proposed interpretable image recognition deep model. Its main goal is to learn basis concepts that provide a bridge between the input high-level image features and output categories. These concepts span a transparent embedding space to re-represent the images and improve the concept-based classification performance for various CNN architectures.

3.1. The overview of TesNet architecture

Let $\mathbf{X}$ denote a set of training images, where $\mathbf{X}^{(c)} \subset \mathbf{X}$ represents the subset belonging to class c ($c = 1, 2, ..., C$). Image recognition aims to train a classifier on $\mathbf{X}$ and use it to predict the label information for any new coming image. To implement interpretable image recognition (label prediction), our proposed TesNet consists of three core factors: Convolutional layers $f(\cdot)$ with parameters ω_{conv}, a Transparent Subspace layer $s_{\mathbf{B}}(\cdot)$ with basis concepts $\mathbf{B}$, and a Classifier $h(\cdot)$ with the weight matrix $\mathbf{G}$, as shown in Figure 3.

Among them, the convolutional layers $f(\cdot)$ are borrowed from the traditional network (e.g. VGG-16, ResNet-34, DenseNet-121). The main difference is that extra 1×1 convolutional layers are added to adjust the the number of channels for top-level feature map. Given an input image $x_i \in \mathbf{X}^{(c)}$, the feature map $\mathbf{Z}_i \in \mathbb{R}^{W \times H \times D}$ is extracted by the convolutional layer $f(\cdot)$ with spatial resolution $W \times H$ and D channels.

Once having feature map $\mathbf{Z}_i$, the subspace layer will project it on the transparent embedding space which is spanned by the basis concepts. Specifically, the subspace layer convers C subspaces (one subspace for each class) and each subspace is spaned by M basis concepts $\mathbf{B}^{(c)} = \{\mathbf{b}_j^{(c)}\}_{j=1}^{M}$. These M within-class concepts are assumed orthogonal to each other and each concept $\mathbf{b}_j^{(c)} \in \mathbb{R}^D$ can be traced back to the high-level patches of the feature map. For all classes, there are total $M \times C$ basis concepts $\mathbf{B} = \{\{\mathbf{b}_j^{(1)}\}_{j=1}^{M}, \cdots, \{\mathbf{b}_j^{(C)}\}_{j=1}^{M}\}$. For convenient, we denote concept set as $\mathbf{B} = \{\mathbf{b}_j\}_{j=1}^{M \times C}$. These class-aware concepts are constrained by maximizing the projection met-

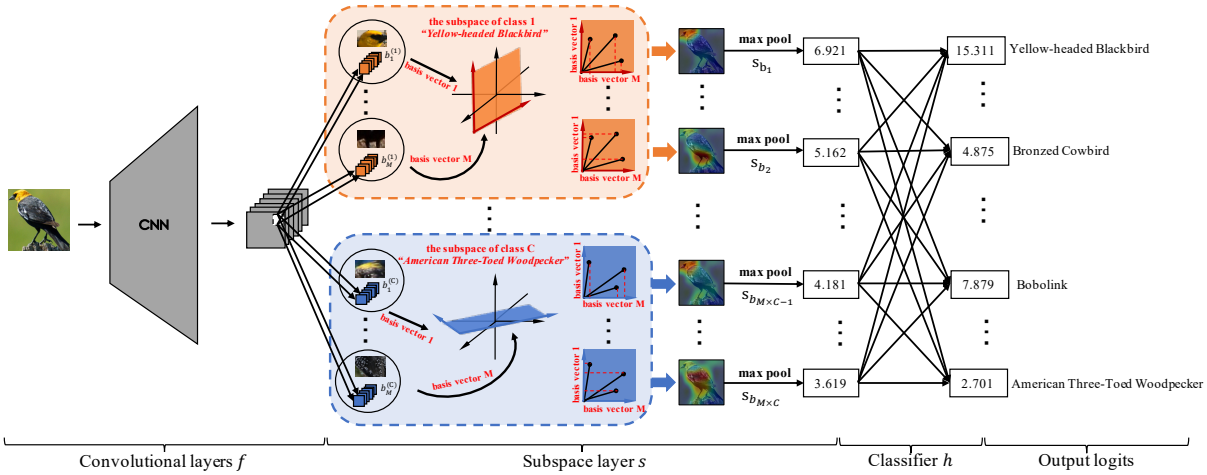

Figure 3: TesNet architecture. To facilitate the illustration, the original embedding space is drawn as 3-D black coordinates and the subspaces are drawn as 2-D red and blue coordinates.

ric so that the concepts in different classes are far away from each other. These settings make sure the learned basis concepts are transparent and disentangled. Each basis unit $s_{\mathbf{b}_j}(\cdot)$ in the subspace layer $s_{\mathbf{b}}$ will compute the projection distance from all 1×1 patches of feature map $\mathbf{Z}_i$ to the j-th basis concept, where this projection distance map will keep the size of feature map and retain the spatial relation well. Then compact the projection distance map to one value by global max pooling, i.e., $s_{\mathbf{b}_j}(\mathbf{Z_i}) = \max_{\mathbf{p} \in \text{patches}(Z_i)} \mathbf{p}^\top \mathbf{b}_j$.

Finally, the projection distances on all basis units $s_{\mathbf{b}_j}(\cdot)$ are taken as the input of Classifier which is composed of a full connection layer h with weight matrix $\mathbf{G}$. The final label is predicted according to the logistic regression value.

Next, we will focus on how to construct the basis concepts $\mathbf{B} = \{\mathbf{b}_j\}_{j=1}^{M \times C}$ and build the decision function $\phi(x_i; \omega_{\text{conv}}, \mathbf{B}, \mathbf{G})$ for label prediction. It will be implemented via three stages: embedding space learning, embedding space transparency and concept based classification.

3.2. Embedding space learning

This subsection aims to construct the basis concepts and use them to span the embedding space. For convenient computing, each basis concept is represented via a basis vector. These basis vectors should satisfy the following requirements: 1) there is no semantic overlapping between different basis vectors; 2) the categories can be separated well in the embedding space; 3) the basis vectors are helpful to cluster the similar high-level patches and separate the dissimilar patches.

To achieve these goals, a joint optimization problem is designed for the convolutional layers' parameters w_{conv} and the basis vectors $\mathbf{B}$, while keeping the last layer weight matrix $\mathbf{G} \in \mathbb{R}^{C \times (M \times C)}$ fixed. For class c, we set $G_{(c,j)} = 1$ if $\mathbf{b}_j \in \mathbf{B}^{(c)}$, otherwise, $\mathbf{G}_{(c,j)} = -0.5$.

Orthonormality for Within-class Concepts: From the cognitive point of view, the concepts should be diversity, i.e., different concepts should focus on different aspects even though they are from the same class. For example, a "bird" class can be recognized via "head" concept,"leg" concept, "tail" concept, "color" concept and etc. Obviously, these concepts have few or even no overlapping semantics. Meanwhile, according to Occam's Razor, humans usually detect a class via a few concepts rather than amounts of concepts. Mathematically, if each concept is represented via a vector, each class can be built via a few basis vectors, and these basis vectors have nothing to do with each other.

Therefore, we introduce an orthonormality loss to add an orthonormal constraint on the basis vectors $\mathbf{B}^{(c)}$ for each class. Its goal is to push the within-class basis vectors apart from each other. Orthonormality loss is defined as:

$$\mathcal{L}_{orth} = \sum_{c=1}^{C} \left\| \mathbf{B}^{(c)} \mathbf{B}^{(c)\top} - \mathbb{I}_M \right\|_F^2 \tag{1}$$

where $\| \cdot \|_F^2$ is Frobenius norm and $\mathbb{I}_M$ is an $M \times M$ identity matrix. By minimizing the correlation between basis vectors, we can get diverse concepts and guarantee there is no overlapping between concepts. These orthogonal basis vectors will span a subspace for the corresponding class. .

Separation for Class-aware Subspaces: In most classification models, the data is represented in the same embedding space. Actually, the data points from different classes usually fall into different subspaces, as demonstrated by subspace learning [10]. To make sure different classes distinctive, their corresponding subspaces should be far away from each other.

In TesNet, each subspace (for each class) is spanned by M orthogonal basis vectors ($\in \mathbb{R}^D$). According to the basic Riemannian geometry of Grassmann manifold [7], the

Grassmann manifold $\mathcal{G}(M, D)$ is the set of M-dimensional linear subspaces of the $\mathbb{R}^D$ and it is a $M(D-M)$ compact Riemannian manifold. An element of $\mathcal{G}(M, D)$ is a linear subspace, which is spanned by the orthonormal $M \times D$ basis matrix Q such that $QQ^\top = \mathbb{I}_M$, where $\mathbb{I}_M$ is the $M \times M$ identity matrix. Therefore, each class-aware subspace (spanned by $\mathbf{B}^{(c)}$) can be taken as a point on the Grassmann manifold. Helmke et al. have proved that there exists one unique project matrix corresponding to each point on the Grassmann mainifold [13]. In this case, the distance between subspaces can be quantified with the aid of projection mapping $\Phi(\mathbf{B}^{(c)}) = \mathbf{B}^{(c)^\top}\mathbf{B}^{(c)}$. One popular subspace distance metric on Grassmann manifold is Projection Metric [11] which can be formulated as:

$$d(\mathbf{B}^{(c_1)}, \mathbf{B}^{(c_2)}) = \frac{1}{\sqrt{2}}\|\mathbf{B}^{(c_1)^\top}\mathbf{B}^{(c_1)} - \mathbf{B}^{(c_2)^\top}\mathbf{B}^{(c_2)}\|_F \quad (2)$$

where $\|\cdot\|$ denotes the matrix Frobenius norm. $\mathbf{B}^{(c_1)}$ and $\mathbf{B}^{(c_2)}$ respectively denote the orthonormal basis matrix of class c_1 and class c_2.

To separate the class-aware subspaces, a new loss is proposed to maximize the projection metric among each pair of subspaces as follows:

$$\mathcal{L}_{ss} = \frac{-1}{\sqrt{2}}\sum_{c_1=1}^{C-1}\sum_{c_2=c_1+1}^{C}\|\mathbf{B}^{(c_1)^\top}\mathbf{B}^{(c_1)} - \mathbf{B}^{(c_2)^\top}\mathbf{B}^{(c_2)}\|_F \quad (3)$$

High-level Patches Grouping: The subspace layer is introduced to project the high-level image patches to the embedding subspaces and preserve the essential information. Therefore, the operation should encourage the patches to be close to at least one semantically similar basis vector of the ground truth class and stay away from the basis vectors of other classes. The former requirement is implemented via a compactness loss, which minimizes the distance between image patches and the basis vectors of the corresponding class in the cosine distance. The later one is obtained by a separation loss, which enforces the patches stay away from the basis vectors that are not of the ground truth class.

Given the basis vectors $\mathbf{B} = \{\mathbf{b}_j\}_{j=1}^{M\times C}$ with $||\mathbf{b}_j|| = 1$ and patches of the feature map $\{\mathbf{Z}_i = f(x_i; \omega_{\text{conv}})\}_{i=1}^{n}$, the compactness-separation loss can be defined as:

$$\mathcal{L}_{cs} = \mathcal{L}_{compactness} + \mu\mathcal{L}_{separation} \quad (4)$$

$$\mathcal{L}_{compactness} = \frac{1}{n}\sum_{i=1}^{n}\min_{j:\mathbf{b}_j\in\mathbf{B}^{(y_i)}}\min_{\mathbf{p}\in\text{patches}(\mathbf{Z}_i)} -\frac{\mathbf{p}^\top\mathbf{b}_j}{\|\mathbf{p}\|}$$

$$\mathcal{L}_{separation} = \frac{1}{n}\sum_{i=1}^{n}\min_{j:b_j\notin\mathbf{B}^{(y_i)}}\min_{\mathbf{p}\in\text{patches}(\mathbf{Z}_i)} \frac{\mathbf{p}^\top\mathbf{b}_j}{\|\mathbf{p}\|}$$

where μ is a hyper-parameter to balance two terms.

Identification: Given the training dataset $\{(x_i, y_i)\}_{i=1}^{n}$, the identification can be done via a cross entropy loss:

$$\mathcal{L}_{id} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{ic}\log\phi_c(x_i; \omega_{\text{conv}}, \mathbf{B}) \quad (5)$$

where n is the number of training images, y_{ic} corresponds to the c-th element of one-hot encoded label of the sample x_i, $\mathbf{y_i} = \mathbf{e}_{y_i} \in \{0,1\}^C$ such that $\mathbf{1}^\top\mathbf{y}_i = 1\,\forall i$, and ϕ_c denotes the c-th element of ϕ. Note the output layer adopts *softmax* function so that $\sum_{c=1}^{C}\phi_c(x_i; \omega_{\text{conv}}, \mathbf{B}) = 1$ and $\phi_c(x_i; \omega_{\text{conv}}, \mathbf{B}) \geq 0, \forall c, i, \omega_{\text{conv}}, \mathbf{B}$.

Finally, the embedding subspace can be obtained by solving the joint optimization problem in an end-to-end manner:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{id} + \lambda_1\mathcal{L}_{orth} + \lambda_2\mathcal{L}_{ss} + \lambda_3\mathcal{L}_{cs} \quad (6)$$

where λ_1, λ_2, and λ_3 are hyper-parameters to balance the corresponding terms.

3.3. Embedding space transparency

In order to make the embedding space transparent, i.e., the users can explicitly know the meaning of each basis concept, each basis vector is traced back its nearest image patches from the same class. Mathematically, the basis vector $\mathbf{b}_j$ of class c, i.e. $\mathbf{b}_j \in \mathbf{B}^{(c)}$ can be assigned via

$$\mathbf{b}_j \leftarrow \arg\max_{\mathbf{p}\in\mathcal{P}_c}\mathbf{p}^\top\mathbf{b}_j \quad (7)$$

where $\mathcal{P}_c = \{\tilde{\mathbf{p}} : \tilde{\mathbf{p}} \in \text{patches}(Z_i)\ \forall i \text{ s.t. } y_i = c\}$. In this way, each basis vector will be related to the particular image patches, which will provide a transparent bridge between the embedding domain and the human-understandable domain.

3.4. Concept based classification

Once having the transparent embedding space (i.e., fixing the parameter ω_{conv} and $\mathbf{B}$), we can effectively build classifier $h(\cdot)$ by optimizing the concept-class weight matrix $\mathbf{G}$, which connects the basis vector units and the logits of class. It is expected that $\mathbf{G}_{(c,j)} \approx 0$ (initially fixed at -0.5) if the j-th unit does not belong to c-th class, and each class is related to only a few units. Thus, the sparse constraint on the weight matrix is enforced to the identification loss as follows.

$$\mathcal{L}_h = \mathcal{L}_{id} + \lambda_4\sum_{c=1}^{C}\sum_{j:b_j\notin B^{(c)}}|\mathbf{G}_{(c,j)}| \quad (8)$$

This setting will guarantee the discriminative evidence comes from the concept of the ground truth class as much as possible and rely less on the concept of negative classes. A good by-product of this concept based classification is that the contribution of concepts to the final prediction can be quantified.

3.5. Implementation

Algorithm 1 describes the implementation for the decision function $\phi(x_i; \omega_{\text{conv}}, \mathbf{B}, \mathbf{G})$. Note that the three stages

(embedding space learning, embedding space transparency, and concept based classification) may be looped several times. ω_{base} and ω_{add} denote the parameters of the base and additional convolutional layers; N_{esl} and N_h denote the number of training epochs in embedding space learning and concept based classification stage; $\eta^{(t)}_{\text{base}}, \eta^{(t)}_{\text{add}}, \eta^{(t)}_{B}, \eta_h$ are learning rates (t denotes epoch index).

Algorithm 1: Overview of training algorithm

initialize: $\omega_{\text{base}} \leftarrow$ pre-trained on ImageNet ;
$\omega_{\text{add}} \leftarrow$ Kaiming uniform initialization;
$\forall j$: basis vectors $\mathbf{b}_j \leftarrow \text{Uniform}([0,1]^{1\times 1\times D})$;
$\forall c, j : \mathbf{G}_{(c,j)} \leftarrow 1$ if $\mathbf{b}_j \in \mathbf{B}_j$ and $\mathbf{G}_{(c,j)} \leftarrow 0$ if $\mathbf{b}_j \notin \mathbf{B}_j$;
while *Not(converge AND* $\mathcal{L}_{compactness} < -\mathcal{L}_{separation}$*)* **do**
  /* Embedding space learning */
  for $t \leftarrow 1$ **to** N_{esl} **do**
    foreach *batch* $[\overline{X}, \overline{Y}]$ *from* $[X, Y]$ **do**
      if $t > 5$ **then**
        $\omega_{\text{base}} \leftarrow \omega_{\text{base}} - \eta^{(t)}_{\text{base}} \nabla_{\omega_{\text{base}}} \mathcal{L}_{total}(\overline{X}, \overline{Y})$;
      $\omega_{\text{add}} \leftarrow \omega_{\text{add}} - \eta^{(t)}_{\text{add}} \nabla_{\omega_{\text{add}}} \mathcal{L}_{total}(\overline{X}, \overline{Y})$;
      $\mathbf{B} \leftarrow \mathbf{B} - \eta^{(t)}_{\mathbf{B}} \nabla_{\mathbf{B}} \mathcal{L}_{total}(\overline{X}, \overline{Y})$;
    $\mathbf{B} \leftarrow \text{Unit}(\mathbf{B}, 0, 1)$ // Constraint basis vectors as unit vectors
  /* Embedding space transparency */
  foreach *basis vector* $\mathbf{b}_j$ **do**
    $c \leftarrow$ class of $\mathbf{b}_j$;
    $\mathbf{b}_j \leftarrow \arg\max_{\mathbf{p}\in\mathcal{P}_c} \mathbf{p}^\top \mathbf{b}_j$ where $\mathcal{P}_c \in \{\tilde{\mathbf{p}} : \tilde{\mathbf{p}} \in \text{patches}(f(x_i))\ \forall (x_i, y_i) \in [X, Y]\ s.t.\ y_i = c\}$
  /* Concept based classification */
  for $t' \leftarrow 1$ **to** N_h **do**
    foreach *batch* $[\overline{X}, \overline{Y}]$ *from* $[X, Y]$ **do**
      $G \leftarrow \mathbf{G} - \eta_h \nabla_{\mathbf{G}} \mathcal{L}_h(\overline{X}, \overline{Y})$

4. Experiments

In experiments, two case studies are conducted to investigate our model and demonstrate its broad applicability on various CNN architectures. The first study is for bird species identification on CUB-200-2011 dataset [32] covering 200 bird species. In the second case study, the Stanford Cars dataset [18] with 196 car models is used to evaluate the proposed TesNet. In each study, a series of ablation tests are conducted to verify the corresponding performance by comparing with the state-of-the-art baselines.

Network Architecture. The proposed model is tested on several convolutional architectures: VGG-16, VGG-19, ResNet-34, ResNet-152, DenseNet-121, and DenseNet-161 (initialized with filters pretrained on ImageNet), following two additional 1×1 convolutional layers. The number of output channels in each additional layer is the same as the number of channels in the basis vectors. For VGG-16 and VGG-19, the number of channels in basis vectors is 128; for ResNet-34, ResNet-152, DenseNet-121 and DenseNet-161, the number of channels is 64. In all cases, 10 basis vectors per class are sufficient. Note only image-level labels are used for training.

4.1. Case study 1: bird species identification

Dataset. Caltecg-USCD Birds-200-2011 [32] (CUB-200-2011) is a dataset of 200 bird species for bird species recognition, which contains 5,994/5,794 images for training/test from 200 different bird species. Since the dataset has only about 30 images per class, we augmented the training set by offline data augmentation which used random rotation, skew, shear, and left-right flip, so that each class has 1200 training images.

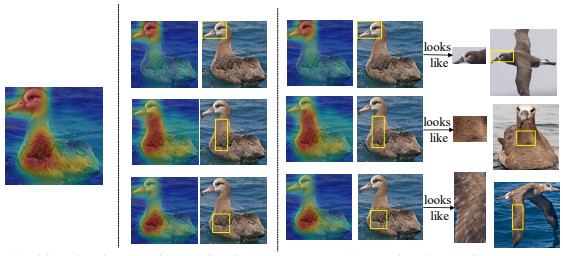

Figure 4: Visual comparison of different level of interpretability: (a) object-level attention map; (b) part-level attention; and (c) part-level attention with learned concepts

Recognition results. Our results of recognition are presented by the accuracy with different base CNN architectures on cropped bird images and compare them to baseline and ProtoPNet at the top of Table 1. We also compared with other deep methods providing the different levels of interpretability in Table 2, "full" means that the model was trained and tested on full images, "bb" means that the model was trained and tested on images cropped using bounding boxes. A visual comparison of different levels of interpretability is provided in Figure 4. To ensure the fairness of comparison, we trained the baseline models, ProtoPNet, and our model on the same augmented dataset of cropped bird images. As shown in Table 1, we switch from the non-interpretable baseline model to our interpretable TesNet with improving precision at most 8% and without the loss of accuracy. Since each TesNet can be understood as a "scoring sheet" (as in Figure 5), we can further improve the accuracy by adding the logits of several trained TesNet models together which is equivalent to combine these "scoring sheet" to compute the total points for each class with no loss of interpretability. Therefore, we combined a VGG19-, ResNet34-, DenseNet121-based TesNet on cropped bird images and the test accuracy can reach 86.2% while the same setting of ProtoPNet can obtain 84.8%. We also combined a VGG19-, Dense121-, and DenseNet161-based Tes-

Table 1: Top: Accuracy comparison on cropped bird images of CUB-200-2011 with different CNN architectures. Bottom: Ablation study on cropped bird images of CUB-200-2011. Recognition accuracy are reported.

Method	VGG16	VGG19	ResNet34	ResNet152	Dense121	Dense161
Baseline	73.3 ± 0.2	74.7 ± 0.4	82.2 ± 0.3	80.8 ± 0.4	81.8 ± 0.1	82.1 ± 0.2
ProtoPNet [5]	77.2 ± 0.2	77.6 ± 0.2	78.6 ± 0.1	79.2 ± 0.3	79.0 ± 0.2	80.8 ± 0.3
TesNet(Ours)	**81.3 ± 0.2**	**81.4 ± 0.1**	**82.8 ± 0.1**	**82.7 ± 0.2**	**84.8 ± 0.2**	**84.6 ± 0.3**

Method	VGG16	VGG19	ResNet34	ResNet152	Dense121	Dense161
TesNet ($\mathcal{L}_{id}$+$\mathcal{L}_{cs}$)	79.2 ± 0.2	79.8 ± 0.2	80.7 ± 0.2	81.6 ± 0.2	83.4 ± 0.1	82.6 ± 0.2
TesNet($\mathcal{L}_{id}$+$\mathcal{L}_{cs}$+$\mathcal{L}_{orth}$)	79.4 ± 0.2	80.3 ± 0.2	81.1 ± 0.2	80.3 ± 0.3	84.1 ± 0.2	83.0 ± 0.2
TesNet($\mathcal{L}_{id}$+$\mathcal{L}_{cs}$+$\mathcal{L}_{orth}$+$\mathcal{L}_{ss}$)	**81.3 ± 0.2**	**81.4 ± 0.1**	**82.8 ± 0.1**	**82.7 ± 0.2**	**84.8 ± 0.2**	**84.6 ± 0.3**

Table 2: Comparison of our model with other deep models in terms of accuracy on the CUB-200-2011 dataset.

Interpretability	Method	Accuracy
None	B-CNN [22]	85.1(bb), 84.1(full)
Object-level attention	CAM [43]	70.5 (bb), 63.0 (full)
	CSG [21]	**82.6 (bb), 78.5(full)**
Part-level attention	PA-CNN [17]	82.8 (bb)
	MG-CNN [33]	83.0 (bb) 81.7 (full)
	MA-CNN [41]	86.5 (full)
	TASN [42]	**87.0 (full)**
Part-level attention + learned concepts	Region [15]	81.5 (bb), 80.2 (full)
	ProtoPNet [5]	84.8 (bb), 80.8 (full)
	Ours	**86.2 (bb), 83.5 (full)**

Net on full images, even though the test accuracy of each individual network is 77.5%, 80.2%, 79.6% respectively.

Reasoning process. Figure 5 shows the transparent reasoning process of our TesNet how to make a decision on a test image of a European goldfinch. Given this test image x_i, our TesNet re-represent its feature map Z_i under the learned basis vectors. Specifically, for each class c, our model tries to find the evidence for x_i to be of class c by re-representing its patches under every learned basis vector $\mathbf{b}_j$ of class c. For example, in Figure 5, our model has found evidence for the European goldfinch class by learned basis vectors (visualized in the "Basis vector" column) of that class. As shown in the "Activation map" column, the first basis vector of the European goldfinch activates most strongly on the typical black and yellow wings of this class, the second basis vector activates on the head, and the third basis vector activates on the brown fur. The most activated image patch of the test image corresponding to the basis vector is marked by a bounding box in the "original image" column. In this case, our model finds a high similarity score between the wing of the test image and the typical wing of a European goldfinch with a similarity score of 5.792, as well as the other parts. Finally, these similarity scores are weighted and summed together to produce a final score for this class c. The reasoning process is similar to other classes. In the supplement, we provide more examples of how our TesNet classifies the images.

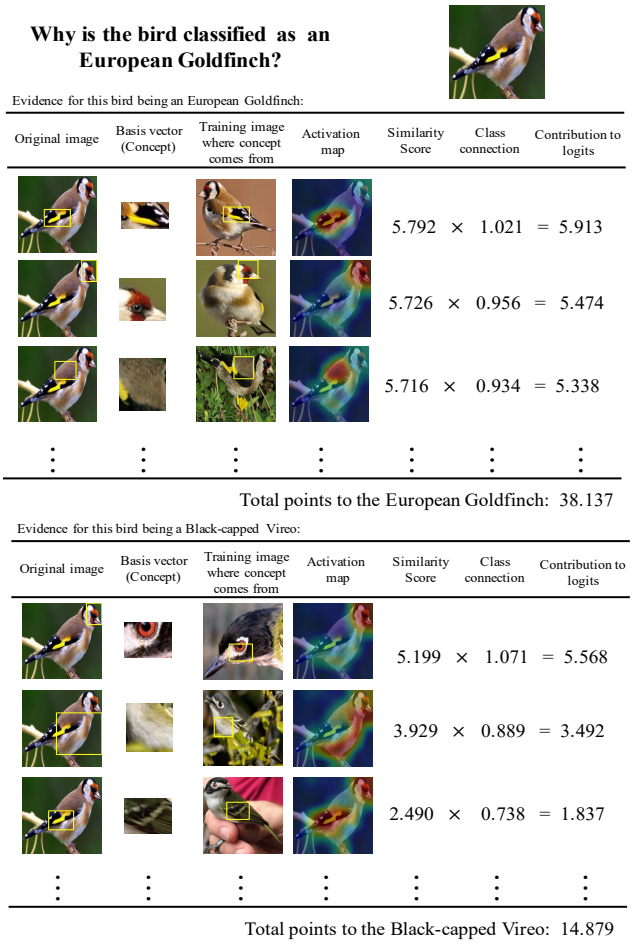

Figure 5: The interpretable reasoning process of our model in identifying the species of a bird.

Ablation study. An ablation study on the CUB-200-2011 is conducted to evaluate the components in embedding space learning. As shown at the bottom of Table 1, our study considers the effect of the orthonormality loss and subspace separation loss. The regularization of orthonormality can slightly improve recognition accuracy performance at most 0.7% to ensure that different basis vectors have different concepts in one class. The regularization of subspace sepa-

Table 3: Accuracy comparison on cropped car images of Stanford Cars dataset with different CNN architectures

Method	VGG16	VGG19	ResNet34	ResNet152	Dense121	Dense161
Baseline	87.3 ± 0.4	88.5 ± 0.3	$\mathbf{92.6 \pm 0.3}$	$\mathbf{92.8 \pm 0.4}$	$\mathbf{92.0 \pm 0.3}$	92.5 ± 0.3
ProtoPNet [5]	88.3 ± 0.2	89.4 ± 0.2	88.8 ± 0.1	88.5 ± 0.3	87.7 ± 0.1	89.5 ± 0.2
TesNet(Ours)	$\mathbf{90.3 \pm 0.2}$	$\mathbf{90.6 \pm 0.2}$	90.9 ± 0.2	92.0 ± 0.2	91.9 ± 0.3	$\mathbf{92.6 \pm 0.3}$

Table 4: Comparison of our model with other deep models in terms of accuracy on the Stanford Cars dataset.

Interpretability	Method	Accuracy
None	B-CNN [22]	91.3
Object-level attention	CSG [21]	91.6
Part-level attention	RA-CNN [8]	92.5
	MA-CNN [41]	92.8
	TASN [42]	**93.8**
Part-level attention + learned concepts	Region [15]	90.9
	ProtoPNet [5]	91.4
	Ours	**93.1**

ration loss largely improves the accuracy at most 2.4% for the subspace of each class well-separation.

4.2. Case study 2: car model identification

Dataset. The Stanford Cars dataset [18] contains 16,185 images of 196 classes of cars, which is split into 8,144 training images and 8,041 testing images. Since each class has only about 40 images in this dataset, we augmented the training set by offline data augmentation which used random rotation, skew, shear, and left-right flip, so that each class has 1300 training images.

Recognition results. The accuracy of TesNet is reported with the corresponding baseline models and ProtoPNet models on the cropped dataset, as shown in Table 3. When we switch from the non-interpretable baseline model to our interpretable TesNet, the loss of accuracy is at most 1.7%. The accuracy slightly decreases because the skip connections hurt learning part features. And we tested the combined network of a VGG19-, ResNet34-, and DenseNet121-based TesNet on full images which can reach 93.1%, even though the test accuracy of each individual network is 89.5%, 90.0%, 89.8% respectively. The test accuracy is on par with some state-of-the-art models on full images, such as RA-CNN (92.5%), MA-CNN (92.8%), and TASN (93.8%).

Reasoning process. Figure 6 shows the transparent reasoning process of our TesNet how to make a decision on a test image of a Tesla Model S Sedan 2012. The model accurately learned the significant concepts of the Tesla logo, front wheels, side door, etc.

The ablation study and more reasoning process of the Stanford Cars dataset are reported in the supplement.

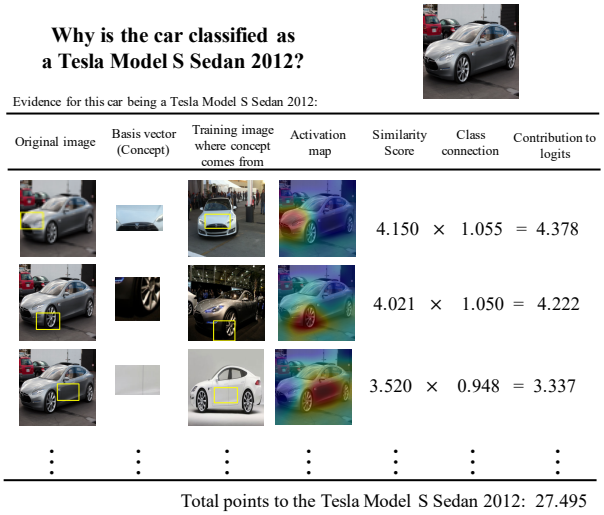

Figure 6: How our TesNet correctly classifies an image of the Tesla Model S Sedan 2012.

5. Conclusions and Future Work

An interpretable plug-in network architecture - TesNet, in this work, is proposed with the aid of transparent basis concepts learning. TesNet constructs class-aware basis concepts and within-class concepts are disentangled, which effectively improves the prediction performance. Empirically, TesNet has ability to explain what concepts a CNN deep network can learn and how the network combines the evidences to make a decision. TesNet assumes that the basis concepts are flat, which is contrast to humans categorize object. The main reason is that, in real world, concepts usually have taxonornical organization. Thus, it will be an interesting topic to consider hierarchical basis concepts learning for interpretable image recognition.

6. Acknowledgement

This work was partly supported by the Beijing Natural Science Foundation (Z180006); The National Key Research and Development Program (2020AAA0106800); The National Natural Science Foundation of China (61822601, 61773050, and 61632004); The Fundamental Research Funds for the Central Universities (2019JBZ110); Science and Technology Innovation Planning Foundation of Universities from Ministry of Education; the Open Project Program Foundation of the Key Laboratory of Opto-Electronics Information Processing, Chinese Academy of Sciences (OEIP-O-202004)

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 1, 2, 3

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 2

[3] Jiawang Bai, Yiming Li, Jiawei Li, Yong Jiang, and Shutao Xia. Rectified decision trees: Towards interpretability, compression and empirical soundness. *arXiv preprint arXiv:1903.05965*, 2019. 3

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 3

[5] 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*, 32:8930–8941, 2019. 2, 3, 7, 8

[6] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 3

[7] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 4

[8] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017. 8

[9] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019. 3

[10] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383, 2008. 4

[11] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian C Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *Proceedings of the IEEE international conference on computer vision*, pages 3120–3127, 2013. 5

[12] 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. 1

[13] Uwe Helmke, Knut Hüper, and Jochen Trumpf. Newton's method on graßmann manifolds, 2007. 5

[14] 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. 1

[15] Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8662–8672, 2020. 2, 3, 7, 8

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

[17] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5546–5555, 2015. 3, 7

[18] 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. 6, 8

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[21] Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *European Conference on Computer Vision*, pages 622–638. Springer, 2020. 3, 7, 8

[22] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015. 7, 8

[23] Huafeng Liu, Jiaqi Wang, and Liping Jing. Cluster-wise hierarchical generative model for deep amortized clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15109–15118, June 2021. 2

[24] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. 2

[25] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. 2

[26] Eleanor Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233, 1975. 1

[27] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R.

Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 3856–3866. Curran Associates, Inc., 2017. 3

[28] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017. 2

[29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1, 2

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 2

[31] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 2

[32] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[33] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 2399–2406, 2015. 3, 7

[34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[35] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 2

[36] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[37] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Growing interpretable part graphs on convnets via multi-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 3

[38] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018. 3

[39] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6261–6270, 2019. 3

[40] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11682–11690, 2021. 3

[41] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017. 3, 7, 8

[42] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5012–5021, 2019. 3, 7, 8

[43] 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. 7

[44] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015. 2

[45] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017. 2