EMBEDDING DEEP NETWORKS INTO VISUAL EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

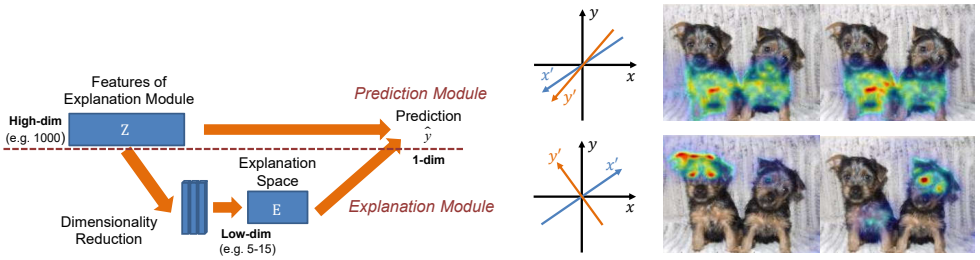
In this paper, we propose a novel explanation module to explain the predictions made by a deep network. Explanation module works by embedding a high-dimensional deep network layer nonlinearly into a low-dimensional explanation space while retaining faithfulness, so that the original deep learning predictions can be constructed from the few concepts extracted by the explanation module. We then visualize such concepts for human to learn about the high-level concepts that deep learning is using to make decisions. We propose an algorithm called Sparse Reconstruction Autoencoder (SRAE) for learning the embedding to the explanation space. SRAE aims to reconstruct part of the original feature space while retaining faithfulness. A visualization system is then introduced for human understanding of features in the explanation space. The proposed method is applied to explain CNN models in image classification tasks, and several novel metrics are introduced to evaluate the performance of explanations quantitatively without human involvement. Experiments show that the proposed approach could generate better explanations of the mechanisms CNN use for making predictions.

1 INTRODUCTION

Deep learning has made significant strides in recent years. It has surpassed human performance in many tasks, such as image classification (Krizhevsky et al., 2012; He et al., 2016), go-playing (Silver et al., 2016), and classification of medical images (Esteva et al., 2017). However, the usage of deep learning in real applications still must overcome a trust barrier. Imagine scenarios with a doctor facing a deep learning prediction: this CT image indicates malignant cancer, or a pilot facing a prediction: make an emergency landing immediately. These predictions may be backed up with a claimed high accuracy on benchmarks, but it is human nature not to trust them unless we are *convinced* that they are reasonable for each individual case. The lack of trust is worsened because of known cases where adversarial examples can fool deep learning to output wrong answers (Szegedy et al., 2013; Goodfellow et al., 2014). In order to establish trust, human needs to understand how deep learning makes decisions. Such understanding could also help the human to gain additional insights into new problems, potentially improve deep learning algorithms, and improve human-machine collaboration.

People prefer explanations of the form “A is something because of B, C, and D", e.g. this is a bird because it has feathers, wings and a beak. This type of explanation has two properties. Firstly, it is concise – there are not a hundred different reasons that add up to explain that A is something. Secondly, it relies on B, C, and D, which are high-level concepts as well. Both are often at odds with deep learning predictions, which are combinations of outputs from thousands of neurons in dozens of layers. Approaches have been proposed to visualize each of the filters (Zeiler & Fergus, 2014) and for humans to name them (Bau et al., 2017), but it is difficult for these approaches to obtain a concise representation. On the other hand, many other approaches generate attention maps that backtrack a decision to specific important areas in the original image (Simonyan et al., 2014; Cao et al., 2015; Zhou et al., 2016; Zhang et al., 2016b; Selvaraju et al., 2016b). These are often nice and quite informative, but they work on individual images and do not provide any high-level concept that can be broadly applicable to many images simultaneously, nor can we believe they are complete explanations so that we can trust them.

In this paper, we make an attempt to reconcile these explanation approaches by extracting several high-level concepts from deep networks to aid human understanding. Our model attaches a separate explanation module to a certain layer in the deep network to reduce the network to a few human-



(a) (b)

Figure 1: (a) The explanation module is a dimensionality reduction mechanism so that the original prediction $\hat{y}$ can be reproduced from this low-dimensional space. An explanation module can be attached to any layer in the prediction deep network (DNN). The output of the DNN can be faithfully recovered from this low-dimensional explanation space, which represents high-level features that are interpretable to humans. (b) Two non-localized and highly correlated heat map explanations in the first row; two localized and largely orthogonal heat map explanations in the second row.

understandable concepts, from where one can generate predictions similar to the original deep network (Fig. 1(a)). We focus on making those concepts to have several properties: *faithfulness*, that the deep learning predictions can be faithfully approximated from those few concepts; *locality*, that the concepts are relatively spatially localized in images so that human can understand them; as well as *orthogonality*, that the concepts themselves are as independent from each other as possible.

Our model does not train from ground truth concepts defined by human, but directly infers concepts from the learning network, hence it is difficult to evaluate the explanations quantitatively. We evaluate our approach on a fine-grained bird classification dataset where rich ground truth annotations allow us to define quantitative metrics for the aforementioned properties without active human involvement.

Although the experiments in the paper focus on convolutional neural networks (CNN) applied to images, the explanation framework we develop is general and applicable to other types of deep networks as well. We believe this is one of the first steps towards general explainable deep learning that can advance human knowledge and enhance future collaboration between humans and machines.

Our contributions in this paper are as follows:

- We propose a novel explanation module to form a low-dimensional explainable concept space from deep networks. A sparse reconstruction autoencoder approach is proposed to make the explanation module faithful and orthogonal as defined previously.
- We present a visualization paradigm for human understanding of the concept space.
- We propose automatic quantitative metrics to evaluate the performance of an explanation algorithm for faithfulness, locality and orthogonality. Experimental results show that the proposed explanation methods provide insights to how the deep network models work.

2 Model Formulation

2.1 The Explanation Module

Given a deep learning network (DNN) as a prediction module, we propose to learn an extra explanation module (Fig. 1(a)), which can be attached to any intermediate layer of the DNN. The explanation module attempts to learn an embedding that lowers the dimensionality of the intermediate layer of the DNN, and then directly learn a mapping from the embedding space to mimic the output of the original DNN model. We denote the input feature space of the explanation module as $\mathbf{Z}(\mathbf{x}; \mathbf{W})$, where $\mathbf{x}$ and $\mathbf{W}$ are the input features and parameters (from multiple layers) of the original DNN model, respectively, and $\mathbf{Z}$ represents the output of a particular intermediate layer of the DNN. The explanation module is used to embed $\mathbf{Z}$ to an explanation space, denoted as $\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z})$, where $\boldsymbol{\theta}$ represents parameters of the embedding that need to be learned. As a shorthand, we will also refer to the explanation space as an *x-layer*, and each dimension in the x-layer an *x-feature*. Note that in the explanation, we do not attempt to change the parameters $\mathbf{W}$ of the original DNN model. The explanation module can in principle be attached to any intermediate layer of the DNN, although the closer to the prediction, the higher level the concepts are and it becomes easier to mimic the prediction of DNN with a low-dimensional embedding.

We believe that for the explanation module to be understandable, it needs to generate a small amount of concepts that preserve the original prediction results $\hat{\mathbf{y}}$. In other words, we would need a low-dimensional feature embedding to be faithful to the DNN. This is generally difficult if the DNN is predicting many concepts simultaneously, such as a multi-class classification. In this paper we propose to obtain faithfulness by explaining 1-dimensional outputs, such as binary classification or one-against-all classifiers. A multi-class explanation can in principle be built up from separate explanations of one-against-all classifiers. For a 1-dimensional prediction $\hat{y}$, we can definitely assume that the explanation module could remain faithful to the prediction, since a naive case would be to use the 1-dim $\hat{y}$ as the explanation, which is perfectly faithful but not interpretable. Hence, the low-dimensional embedding $\mathbf{E}$ can also be thought of as expanding $\hat{y}$ to several dimensions, therefore enriching the explanations for a single prediction.

In this paper, we focus on attaching explanation modules to fully-connected layers. The concepts generated in these layers are rather high-level, and our conceptual goal is to visualize those concepts and to make humans learn them: human has a quite deep neural network for learning and generalizing perceptual concepts very well. Therefore we would like to show humans examples from a small number of perceptual concepts from the explanation space, so that they can utilize their own perceptual neural network for learning and naming those. Our primary tool for this display is heatmaps (e.g. Fig. 1(b), Fig. 3(a)) highlighting a specific region in the image, similar as those used in attention models in prior work. Our work will provide several different and largely orthogonal concepts, visualized by heatmaps, for improving the understanding of the predictions from a DNN. The two main topics in the explanation module are the embedding algorithm and the visualization of the explanations, which will be discussed in the next three subsections.

2.2 Embedding to the Explanation Space

Explanation space optimization attempts to be faithful to the prediction of the original DNN:

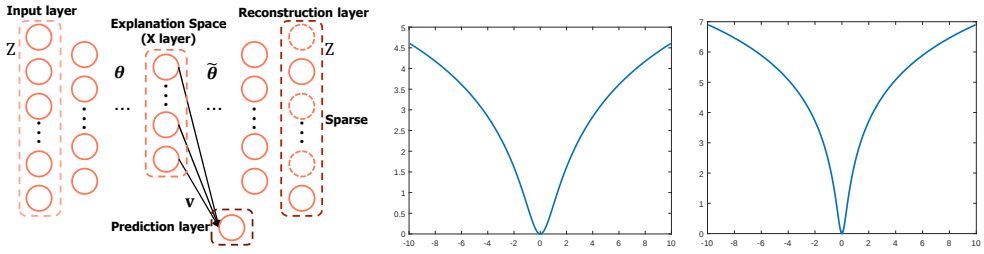
$$\min_{\boldsymbol{\theta},\mathbf{v}} \frac{1}{M}\sum_{i=1}^{M} L\Big(f\big(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)});\mathbf{v}\big),\hat{y}_j^{(i)}\Big) \tag{1}$$

where $\mathbf{Z}^{(i)}=\mathbf{Z}(\mathbf{x}^{(i)};\mathbf{W})$ is the output of an intermediate layer in DNN for instance $\mathbf{x}^{(i)}$; parameter $\boldsymbol{\theta}$ is used to form the explanation space $\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})$; parameter $\mathbf{v}$ is used to build a predictor $f(\mathbf{E};\mathbf{v})$ from the x-features to mimic $\hat{y}_j^{(i)}$, e.g. $f(\mathbf{E};\mathbf{v})=\mathbf{v}^\top\mathbf{E}$ would be a simple linear predictor from the explanation space and the one we use in this work; $\hat{y}_j^{(i)}$ is j-th output of the original DNN model and the explanation target for instance $\mathbf{x}^{(i)}$, we usually use the DNN output before the softmax layer to prevent interactions with other predictions; M is the number of the training examples; L is a loss function, usually a regression loss such as squared loss or log loss. However, as we argued in Sec. 2.1, this formulation might be almost degenerate if $\hat{y}_j^{(i)}$ can be used as the explanation variable. Hence, additional terms need to be added to prevent degeneracy and improve interpretability.

We claim that low-dimensional embeddings are more effective when they reconstruct the original high-dimensional feature space better. If the original DNN features are localized, then one can hope the low-dimensional explanation features are also localized, since aggregating different localities in the same dimension is also not helpful for reconstructing each of them. Thus, adding reconstruction loss $L\Big(\mathbf{E}_{\widetilde{\boldsymbol{\theta}}}^{-1}\big(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})\big),\mathbf{Z}^{(i)}\Big)$ to optimization (1) will prevent degeneracy and improve locality. Here $\mathbf{E}_{\widetilde{\boldsymbol{\theta}}}^{-1}$ is a mapping that maps from the explanation space $\mathbf{E}$ back to $\mathbf{Z}$, $\widetilde{\boldsymbol{\theta}}$ is the parameter for this mapping. However, when the weight of the reconstruction loss is large in the optimization, features irrelevant to the predict target may also be reconstructed.

To avoid this, we propose to enhance the objective by adding a sparsity term which reconstructs *some dimensions* of the original features $\mathbf{Z}$, but not all of them. By attempting to reconstruct some dimensions of $\mathbf{Z}$ with only a few embeddings, and to mimic the original predictions $\hat{y}$ with the same embeddings, the maximal amount of diverse information that are relevant to $\hat{y}$ in $\mathbf{Z}$ needs to be packed in the low-dimensional space. Packing redundant information in correlated dimensions would be harmful for reconstruction, and reconstructing irrelevant features would harm the ability to recover $\hat{y}$. By introducing a sparse penalty in the reconstruction loss, we obtain:

$$\min_{\boldsymbol{\theta},\widetilde{\boldsymbol{\theta}},\mathbf{v}} \frac{1}{M}\sum_{i=1}^{M} L\Big(f\big(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)});\mathbf{v}\big),\hat{y}_j^{(i)}\Big)+\beta\text{Sparsity}(\mathbf{Q});\quad Q_k=\frac{1}{M}\sum_{i=1}^{M} L\Big(\mathbf{E}_{\widetilde{\boldsymbol{\theta}}}^{-1}\big(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})\big)_k,Z_k^{(i)}\Big) \tag{2}$$



(a) (b) (c)

Figure 2: Illustration of the SRAE used for the explanation module. Both the prediction and a sparse reconstruction are generated from the explanation space; (b) The log penalty function $\log(1 + q \cdot r^2)$ when $q = 1$; (c) The log penalty function $\log(1 + q \cdot r^2)$ when $q = 10$.

where Q_k, $Z_k^{(i)}$, and $\mathbf{E}_{\tilde{\boldsymbol{\theta}}}^{-1}\left(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})\right)_k$ are the k-th dimension of $\mathbf{Q}$, $\mathbf{Z}^{(i)}$, and $\mathbf{E}_{\tilde{\boldsymbol{\theta}}}^{-1}\left(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})\right)$, respectively and β is the parameter for the sparsity term. In the optimization, Q_k measures the capability of reconstructing the k-th dimension in the space of $\mathbf{Z}$. The sparsity term will be detailed in Sec. 2.3. With this term, optimization (2) achieves faithfulness, locality, orthogonality, and little irrelevant information for the explanation space.

2.3 Dimensionality Reduction Method

In general, any dimensionality reduction method can be used to obtain the explanation space $\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z})$ Here we propose a novel network called Sparse Reconstruction Autoencoder (SRAE), which handles the objective as defined in (2). SRAE is also a neural network, hence can seamlessly combine with the prediction DNN, making the following visualization process (introduced in Sec. 2.4) simple. Our aim is at reconstructing some specific features which focus on the prediction target instead of reconstructing the whole feature space. We utilize the log penalty $\log(1 + q \cdot r^2)$ (Lee et al., 2007) (Figure 2(b-c)) to achieve the sparsity of the reconstruction errors for different features. Here r^2 is the average squared reconstruction loss on each dimension over the whole training set, which equals to Q_k using a square loss. Hence, we obtain:

$$\text{Sparsity}(\mathbf{Q}) = \frac{1}{S_z}\sum_{k=1}^{S_z}\log(1 + q \cdot Q_k) = \frac{1}{S_z}\sum_{k=1}^{S_z}\log\left(1 + \frac{q}{M}\sum_{i=1}^{M}\left\|\mathbf{E}_{\tilde{\boldsymbol{\theta}}}^{-1}\left(\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})\right)_k - Z_k^{(i)}\right\|^2\right) \quad (3)$$

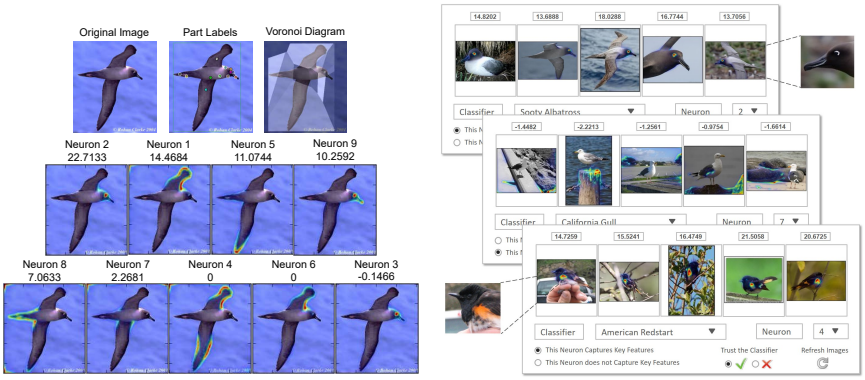
where $q > 0$ is a sparsity parameter (as shown in Figure 2(b-c)), S_z is the dimensionality of the feature space $\mathbf{Z}$. Note that SRAE is different from conventional sparse autoencoders in which the autoencoder activations in the hidden layers are constrained to be sparse. In SRAE, the sparsity constraint is on the amount of input dimensions to be reconstructed. In general, various sparsity functions can be used here such as the L_1 penalty function, epsilon-L_1 penalty function (Lee et al., 2007), the Kullback-Leibler divergence (Ng, 2011), etc. Here we choose the log penalty $\log(1+q\cdot r^2)$ in our proposed model. The log penalty (Figure 2(b-c)) is a robust loss function, in the sense that large r increases the loss function sublinearly (less than an L_1 penalty $|r|$ where the increase is linear). Some dimensions of $\mathbf{Z}$ can afford to have no reconstruction at all (large r) without suffering too much loss. Hence this loss function achieves the goal that only some of the input dimensions are *selectively reconstructed*, instead of all of them. The exact dimensions that are reconstructed are chosen automatically by the learning procedure itself.

The illustration of the proposed SRAE used for explanation module is shown in Figure 2(a). The encoding layers in SRAE forms the explanation space $\mathbf{E}$ (Figure 2(a)). Using the least squares loss again for faithfulness, the optimization of the SRAE is shown as follows:

$$\min_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}},\mathbf{v}} \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{v}^\top\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)}) - \hat{y}_j^{(i)}\right\|^2 + \beta\cdot\frac{1}{S_z}\sum_{k=1}^{S_z}\log(1 + q \cdot Q_k) + \lambda_1\|\boldsymbol{\theta}\|^2 + \lambda_2\|\tilde{\boldsymbol{\theta}}\|^2 + \lambda_3\|\mathbf{v}\|^2 \quad (4)$$

where the first 2 terms are faithfulness and sparse reconstruction, and the last 3 terms are L_2 regularizations for the weights of SRAE; λ_1, λ_2, λ_3 are the parameters to the three regularizations; and the prediction result $\mathbf{v}^\top\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})$ of SRAE is denoted as $\bar{y}_j^{(i)}$.

Compared with traditional autoencoders, the proposed SRAE method reconstructs only part of the inputs. SRAE can be applied as a general method to the domains where input feature selection and



(a) (b)

Figure 3: (a) an example generated by our SRAE. The first line shows the original image, the part labels of the image in the ground truth, and the Voronoi diagram of the image; the second and third lines show the visualization results for the 9 neurons in the x-layer sorted by the weights ($v_i E_i, i = 1, 2, \ldots, 9$) for the final prediction; (b) Examples of the interactive visualization system.

feature coding are both needed. The optimization in (4) can be solved effectively by backpropagation with the regularization terms handled by weight decay, with no weight decay on the bias terms in the network. Finally, we obtain an explanation embedding $\mathbf{E}_{\boldsymbol{\theta}}(\mathbf{Z})$ and a linear predictor $\mathbf{v}^\top \mathbf{E}$ which explains the prediction of a single-output deep network as a linear combination of explanation features. In conjunction with the visualization paradigm in the next subsection, this facilitates better understanding of black-box DNN models to both experts and non-specialists.

2.4 Visualizing the Explanation Space

The goal in the visualization of low-dimensional explanation features is to bridge the communication gap between human and machine, and enable human to name concepts learned by the explanation module and be able to construct sentences with those named concepts. For this paper though, we only focus on visualizing the concepts. We utilize ExcitationBP (Zhang et al., 2016b) to compute the contrastive marginal winning probability (c-MWP) from each neuron in x-layer to the pixels in the original image, then generate the heat maps using c-MWP normalized on each neuron for each image. A prototype interactive visualization system is introduced for human understanding of neurons in the explanation space, which contains two types of visualizations (Fig. 3(a) and Fig. 3(b)). The first type shows the heatmap for different neurons and their prediction weights in a single image (Fig. 3(a)), and the second type shows a single neuron across many images for human to name this particular neuron (Figure 3(b)) and vote for whether to trust this neuron in the final classifier.

3 Related Work

The explanation for high accuracy but black-box models has become a significant need in many real applications. In the medical domain, several approaches were proposed to utilize interpretable models to explain the predictions for individual patients in a concise way (Caruana et al., 2015; Letham et al., 2015; Ustun & Rudin, 2016). In Natural Language Processing, Kulesza et al. (2015) propose an interactive system which builds a cycle of explanations from the learning system to the user, and then back to the system. In computer vision, methods have been introduced to explain the predictions either by associating the images with captions/descriptions (Kiros et al., 2014; Kong et al., 2014; Lin et al., 2014; Karpathy & Fei-Fei, 2015; Hendricks et al., 2016), visualizing individual convolutional filters in the network (Zeiler & Fergus, 2014; Bau et al., 2017) or heatmaps that indicate important regions in the original images (Simonyan et al., 2014; Cao et al., 2015; Zhou et al., 2016; Zhang et al., 2016b; Selvaraju et al., 2016a). Park et al. (2016) and Selvaraju et al. (2016a) propose to explain via visual question answering which utilized both natural language descriptions and heatmaps. Ribeiro et al. (2016) propose an explanation technique which tries to explain single prediction of general models, and select several representative predictions to provide a global view of the model.

Image captioning approaches (Kiros et al., 2014; Kong et al., 2014; Lin et al., 2014; Karpathy & Fei-Fei, 2015; Hendricks et al., 2016) need to be trained on human-generated sentences, hence they would not work in any domain where human is not an expert in. Our approach does not require

any natural language descriptions. Visualizing individual neurons/filters were important for human intuition about CNNs (Zhou et al., 2015; Zeiler & Fergus, 2014; Jain et al., 2016). Recently, Bau et al. (2017) went to great lengths in visualizing thousands of neurons and asking human to name each of them. However, it is difficult for such efforts to provide a concise yet complete representation. Agrawal et al. (2014) analyzed the number of filters required to generate good performance on the PASCAL VOC dataset and the conclusion is that each class would need at least dozens of filters. We adopt the heatmap approach in (Zhang et al., 2016b), but visualize explanation features instead of directly visualizing classification results. With this approach we can generate high-level concepts that are broadly applicable to multiple images in the same category.

Recently, there has been a focus of detecting parts using deep neural networks without part annotations, usually in fine-grained classification. Simon & Rodner (2015b) and Xiao et al. (2015) use combinations of convolutional filters to generate part proposals that improves prediction performance. Gkioxari et al. (2015) and Zhang et al. (2016a;d;c) use various approaches to detect parts. Our focus is different in that we focus on explaining a trained deep model instead of trying to enhance it, and the explanation may not necessarily be parts that can be expressed in terms of bounding boxes as in those approaches. Gonzalez-Garcia et al. (2016) conducted comprehensive experiments on whether semantic parts naturally emerge from convolutional filters. They explored combinations of filters using a genetic algorithm but only combine an average of 5 filters, hence did not have the dramatic dimensionality reduction effect as in our work. Independent from our work, recently Zhao et al. (2017) train a hybrid CNN-LSTM model featuring diversified attention models jointly and generate diverse attention maps similar to ours in the middle of the network, but it cannot be utilized to explain an already-trained DNN because of the joint training that is needed, and there was no attempt in quantitatively evaluating the explanations.

Model compression for deep learning was proposed in (Ba & Caruana, 2014), where a shallow model is used to mimic the output of a deep network. Most model compression work since then were used for speeding up testing (Chen et al., 2015; Rastegari et al., 2016). Che et al. (2016) learn a decision tree on top of deep network results in an attempt for an interpretable model, however their framework cannot discover new features as they were only utilizing categorical predictions with known categories (that were trained on) as the basis for interpretation.

4 Experiments

4.1 Evaluation Metrics

The most challenging part in the experiments is to find objective metrics to evaluate the performance of the explanation module, since the explanation of images is a relatively subjective matter. Evaluating explanations objectively without a human study is important, because simple parameter variations can easily generate thousands of different explanations, vastly outpacing the speed of human studies. In this paper we make an attempt to define some quantitative metrics. We utilize the CUB-200-2011 dataset (Wah et al., 2011) in the experiments. This is a task for fine-grained bird classification into 200 categories. This dataset is chosen because in addition to category labels and bounding boxes surrounding each object, it also has part labels denoted as one pixel per part (Fig. 3(a)) for each object as additional ground truth. One can argue that the majority of bird classifications are based on specific, discriminative parts of the bird, which can be confirmed from encyclopedias and expert annotations (Reed et al., 2016). In order to measure locality, we propose a metric which associates neurons in the x-layer with various parts of one category in the image, and measures how well they associate with these parts. Note that this metric is by no means perfect and would struggle at features that do not represent a single part, it merely reflects our current best efforts in quantitatively measuring different explanations.

Given image I_m, for each neuron n in the x-layer and each pixel (i,j) in I_m, we denote $S^{n,m}_{i,j} \triangleq P(\text{Pixel}^m_{i,j}|\text{Neuron}_n) = \frac{C^{n,m}_{i,j}}{\sum_{(i,j)\in I} C^{n,m}_{i,j}}$, where $C^{n,m}_{i,j}$ is the c-MWP generated by ExcitationBP for pixel (i,j) in I_m with neuron n in x-layer, (i,j) is the coordinate of the pixel. For the CUB dataset, since the given part label ($p = 1, ..., 15$) of each image is just one pixel in the middle of the part, and there is no extra information about the shape and the size of the part regions, we utilize the Voronoi diagram to partition the bounding box into 15 regions where the nearest neighbor part annotation in each region would be the same (Fig. 3(a)). Then we compute the probability $S^{n,m}_{p} \triangleq P(\text{Part}^m_p|\text{Neuron}_n) = \sum_{(i,j)\in I_m} P(\text{Part}^m_p|\text{Pixel}^m_{i,j}) P(\text{Pixel}^m_{i,j}|\text{Neuron}_n)$ using Algorithm

1 in the Appendix. The Voronoi diagram is used instead of a segmentation, because firstly we do not have segmentation ground truth and do not wish to include additional errors from an arbitrary segmentation algorithm, and secondly because some of the heatmap activations fall slightly outside the object and we still want to capture those. For all the c-MWP outside of the ground truth bounding box, we introduce a 16-th part called *context*, which indicates that the x-feature is using the context to classify rather than the object features.

Next, we propose several metrics to evaluate the performance of the explanation module. For each x-feature n we have a histogram $\mathbf{S}_n$ whose element is $\bar{S}_p^n = \frac{1}{M}\sum_m S_p^{n,m}$. The **Locality** for each x-feature is defined as the entropy: $H_n = -\sum_p \left(\frac{\bar{S}_p^n}{\sum_p \bar{S}_p^n} \cdot \log(\frac{\bar{S}_p^n}{\sum_p \bar{S}_p^n})\right)$. Locality is roughly measuring the log of the number of parts captured by each x-feature. If the x-feature falls perfectly in one part, locality will be 0. Note that there are many small parts hence often x-features will fall on more than one of them just because the blur in the attention map. For the whole explanation module, we have: (1) **Faithfulness**: We introduce a regression metric and a classification metric for faithfulness. (a) $F_{reg} = \frac{1}{M}\sum_m L(\bar{y}^{(m)} - \hat{y}^{(m)}) = \frac{1}{M}\sum_m |\bar{y}^{(m)} - \hat{y}^{(m)}|$, the mean absolute loss between $\hat{y}^{(m)}$ and its approximation $\bar{y}^{(m)}$; (b) We replace $\hat{y}^{(m)}$ with $\bar{y}^{(m)}$ in the original multi-class prediction vector $\hat{\mathbf{y}}^{(m)}$ before softmax and check whether the classification result changes. We denote c_r as the number of examples whose classification results remain the same, then $F_{cls} = \frac{c_r}{M}$. (2) **Orthogonality**: In order to measure whether different attention maps fall on the same region, we directly treat attention maps of different x-features as different vectors and compute their covariance matrix. We denote $\mathbf{C}$ as the covariance matrix among x-features aggregated over the dataset. Then $\mathbf{P} = \text{diag}(\mathbf{C})^{-1/2}\mathbf{C}\text{diag}(\mathbf{C})^{-1/2}$ is the matrix of correlation coefficients. The orthogonality between neurons in the x-layer is defined as: (a) $O_1 = ||\mathbf{P}||_F - \sqrt{n}$, where $||\cdot||_F$ is the Frobenius norm for matrix; (b) $O_2 = -\text{logdet}(\mathbf{P})$, where logdet is the logarithm of determinant of a matrix. Both O_1 and O_2 obtain the optimum at 0, when $\mathbf{P}$ is a unit matrix.

4.2 Experiment Settings and Results

The fine-tuned VGG19 model (Simon & Rodner, 2015a) for CUB-200-2011 birds is used as the prediction DNN to be explained. The explanation module is a 3-middle-layer SRAE with $800-100-n$ hidden units in each layer, where n represents the number of x-features. We trained an explanation module on a random 30 of the 200 bird categories. For each category, we utilized 50 positive examples and $8,000$ negative examples as the training data; the remaining positive examples $(8-10)$ and $2,000$ negative examples as the testing data. In the training process, we enhance the weights of the positive examples to avoid imbalance. n is set to 5, as our experiments showed that more x-features do not improve performance and create x-features which have 0 weight in $v_i E_i$, indicating that one one-against-all classifier of one bird does not depend on many high-level visual features. We compared the proposed SRAE with a fully-connected neural network (NN), a conventional stacked autoencoder with faithfulness loss and traditional reconstruction loss (SAE), a classic autoencoder with only traditional reconstruction loss and without faithfulness loss (CAE), a feature selection model (Lasso) on $\mathbf{Z}$, as well as directly performing ExcitationBP on the classification output $\hat{y}$ (ExcitationBP). The baseline neural network methods (NN and SAE) can also perform a faithful dimensionality reduction, and are the most closely related to our approach. Lasso represents a feature selection approach which selects several most useful dimensions directly from $\mathbf{Z}$ and tries to mimic the network decision as a linear combination of these features. All the learning-based approaches (SRAE, NN, SAE, CAE, and Lasso) were tuned to the optimal parameters by cross-validation on the training set.

In Table 1(a), we summarize the results for different explanation embedding approaches with different parameters. Results show that we can achieve excellent faithfulness to the predictions when using SRAE, NN, and SAE. The F_{reg} in both training and testing are less than 0.2. Since $\hat{y}$ before softmax usually has a range in $[0, 50]$ and especially large in the positive examples, we consider the regression loss to be small. The classification faithfulness F_{cls} is even better, as only $1-2$ examples out of all the categories we tested have switched labels after replacing the original $\hat{y}$ with the approximation from the x-features. We also summarize the faithfulness for Lasso using different parameters in Table 2(a), where α is the parameter that multiplies the L_1 term in Lasso, Num_x is the average number of the selected features for 30 categories. From Table 1(a) and Table 2(a) we observe that the faithfulness for Lasso are all very bad with different parameters, indicating that it is almost impossible for the feature selection method to select few X-features form $\mathbf{Z}$ directly to make the prediction faithful.

Table 1: (a) The average faithfulness, orthogonality, and locality for different approaches in 30 categories selected randomly. The column $\mathbf{Z}$ represents the average locality computed over all the dimensions of $\mathbf{Z}$, the 4096-dimensional first fully-connected layer of the deep network. This is obtained by separately running ExcitationBP on each dimension of $\mathbf{Z}$ and evaluating the resulting heatmaps. (b) A preliminary human study comparing SRAE with Lasso.

(a)

Method		SRAE	NN	SAE	Lasso	CAE	$\mathbf{Z}$	ExcitationBP
F_{reg}	Training	0.0831	0.0657	0.0987	3.8039	3.9202	—	—
	Testing	0.1539	0.1170	0.1928	3.7028	3.8216	—	—
F_{cls}	Training	99.99%	99.99%	99.99%	71.82%	62.38%	—	—
	Testing	99.99%	99.99%	99.98%	68.33%	67.21%	—	—
O1	Positive	**0.6141**	0.8851	0.7161	1.1407	**0.5755**	—	—
O2	Positive	**2.0790**	3.6483	2.7710	2.8632	**1.7585**	—	—
Locality	Positive	**1.9694**	2.3078	2.1492	2.0941	1.9989	**1.9623**	2.4934

(b)

Method	SRAE	Lasso
O_1	1.39	1.88
O_2	8.17	9.21
Locality	1.83	1.92

Table 2: (a) The average faithfulness for Lasso with different α; (b) The average classification accuracy for images masked by our method and ExcitationBP so that only highlighted areas are shown to the classifier.

(a)

Lasso	α	2.5	1.5	0.5	0.1
Num_x		8	21	68	232
F_{reg}	Training	3.80	3.06	1.86	1.00
	Testing	3.70	2.99	1.84	1.03

(b)

Method	Original Image	Mask by X features	Mask by ExcitationBP
Classification Accuracy	0.8798	0.8428	0.6742

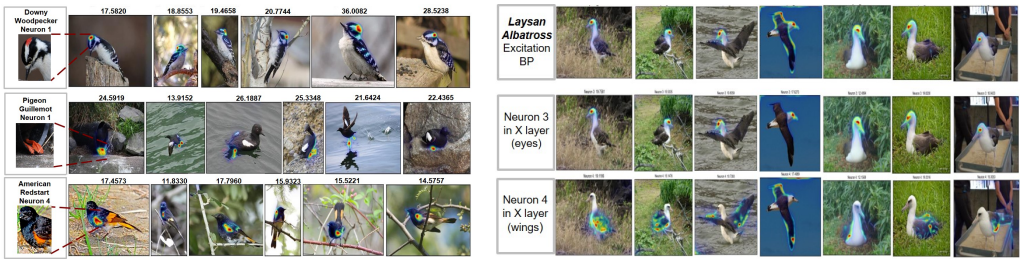
In terms of orthogonality and locality, our algorithm showed significant improvements over NN, SAE, and Lasso ($\alpha = 2.5$ in Table 1(a)). The orthogonality of CAE is better than that of the proposed SRAE, which is reasonable because the features in $\mathbf{E}(\mathbf{Z})$ are definite more orthogonal when there is only reconstruction loss in the optimization. The locality of CAE is slightly worse than SRAE, but the most important problem is that it is very difficult for CAE to achieve faithfulness to the original predictions because of the lack of the faithfulness loss in the optimization. Besides, the locality of SRAE improves significantly over the ones from ExcitationBP, indicating that we are capable of separating information that come from different parts. The average locality of the x-features generated by SRAE are almost matching the average locality of features in $\mathbf{Z}$. This means we are close to the limit of part separation on this layer: many of the features on the $\mathbf{Z}$ layer already represent multiple parts. In future work we plan to conduct more experiments explaining earlier convolutional layers to see whether the locality could be further lowered while preserving faithfulness.

In Table 1(b), we show the results from a preliminary human study to compare the performance of SRAE and Lasso. Given the visualization heatmaps for each x-feature, each participant needs to choose which parts (including 15 parts of birds and the context) each heatmap represents. For each neuron in $\mathbf{E}(\mathbf{Z})$, we summarize how many times it represents each part over the whole training set. Then we obtain a $n \cdot p$ matrix, where n is the number of neurons in $\mathbf{E}(\mathbf{Z})$, p is the number of parts, and the element is the times each neuron associates with each part. We use this matrix to compute the locality and orthogonality to validate whether the automatic ones have been computed properly. 8 persons participate in the human study, including 4 experts and 4 non-experts in computer science. We compared SRAE with Lasso in 5 randomly selected categories with 2,810 heatmaps in total. The number of the x-features is 5. We observe that the orthogonality and locality of SRAE are smaller than those of Lasso (the approach with the best locality among competitors), indicating that our method performs better in explaining the deep networks from human subjective judgments. From Table 1(a) and 1(b) we observe that the Locality, O_1, and O_2 of the most top important features of $\mathbf{Z}$ are all larger than those of $\mathbf{E}(\mathbf{Z})$ both from quantitative evaluations and human study, indicating that the explanation space $\mathbf{E}(\mathbf{Z})$ is better than the given latent space $\mathbf{Z}$.

We also show some qualitative examples from different categories in Fig. 4. Fig. 4(a) shows the most important x-feature in several categories, where we can see that they fit our intuitions on the discriminative features of the birds. Fig. 4(b) compares x-features with directly running ExcitationBP on $\hat{y}$. One can see x-features nicely separate different discriminative aspects of the bird while ExcitationBP sometimes focuses only on one part and miss others, and sometimes produces a heatmap that incorporates many parts simultaneously. Also, each x-feature seems distinct enough as a concept. Hence we believe they indeed provide more explanation on the decisions made by CNN algorithms. More qualitative results are shown in the Appendix.

To further examine whether the proposed algorithm offers a complete explanation of the decision made by the CNN, we attempted to try to classify just using the regions that are presented in the heatmaps, similar to (Gonzalez-Garcia et al., 2016). First, images are masked so that pixels that have

$< 5\%$ of the highest response in the heatmap are painted as black; then an inpainting algorithm is applied to recover the masked images; finally we utilize the prediction CNN to classify the recovered images and test the classification accuracy. In our work, we keep the highlighted regions while mask the background, which is different from (Gonzalez-Garcia et al., 2016) where the highlighted regions are removed. In Table 2(b), one can see that ExcitationBP fails in more cases whereas the 5 heatmaps from x-features result in substantially increased classification accuracy. More experimental results are shown in the Appendix.



(a) (b)

Figure 4: (a) The most important x-feature for several categories. The weight above the feature is $v_i E_i$, the product of the weight of the x-feature in the approximation of $\hat{y}$ timed by the activation of the x-feature; (b) ExcitationBP on the predictions and on the x-features.

5 Conclusion

In this paper we propose an explanation module, that can be attached to any layer in a deep network to compress the layer into several concepts that can approximate a 1-dimensional prediction output from the network. A sparse reconstruction autoencoder (SRAE) is proposed to avoid degeneracy and improve orthogonality. We also proposed automatic evaluation metrics to evaluate the explanations on a fine-grained bird classification dataset. Quantitative and qualitative results show that the network can indeed extract high-level concepts from a CNN that make sense to human. We view this work as one of the first steps toward understanding deep learning and have many future plans to it, including performing more experiments on different kinds of data, including those without ground truth, and extending it to explain other types of neural networks, such as recurrent networks and convolutional-recurrent ones.

References

Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European Conference on Computer Vision*, pp. 329–344. Springer, 2014.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, 2014.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2956–2964, 2015.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 1721–1730, 2015.

Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Interpretable deep models for icu outcome prediction. In *American Medical Informatics Association Annual Symposium*, 2016.

Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294, 2015.

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542:115–118, 2017.

Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Actions and attributes from wholes and parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2470–2478, 2015.

Abel Gonzalez-Garcia, Davide Modolo, and Vittorio Ferrari. Do semantic parts emerge in convolutional neural networks? *arXiv preprint arXiv:1607.03738*, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, 2016.

Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5308–5317, 2016.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schutt, Sven Dahne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods. *arXiv preprint arXiv:1711.00867v1*, 2017.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Multimodal neural language models. In *Icml*, volume 14, pp. 595–603, 2014.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3558–3565, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pp. 126–137. ACM, 2015.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9: 1350–1371, 2015.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2657–2664, 2014.

Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv Preprint:1612.04757*, 2016.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pp. 525–542. Springer, 2016.

Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

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. *arXiv Preprint:1610.02391*, 2016a.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016b.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe,

John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach adn Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015a.

Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1143–1151, 2015b.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 842–850, 2015.

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

Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1143–1152, 2016a.

Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pp. 543–559. Springer, 2016b.

Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1134–1142, 2016c.

Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing*, 25(4):1713–1725, 2016d.

Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia, to appear*, 2017.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

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*, pp. 2921–2929, 2016.

APPENDIX

For the CUB-200-2011 dataset, the given part label of each image is just one pixel in the middle of the part. For the p-th part label of image I_m, we denote (i_p, j_p) as its pixel location. The **pixel level probability** is defined as $S^{n,m}_{i_p,j_p} = P(\text{Pixel}^m_{i_p,j_p}|\text{Neuron}_n)$. Figure 5(a) shows the probability $S^{n,m}_{i_p,j_p}$ for each neuron $(n = 1, \ldots, 9)$ at the pixel locations of the part labels $(p = 1, \ldots, 15)$ for the example image shown in Figure 3(a). From Figure 5(a) we observe that the probability $S^{n,m}_{i_p,j_p}$ is reasonable when capturing small parts like *eye* and *beak*, but is not on larger parts like *wing* and *tail*, for the part label is just one pixel in the middle of the *wing* or *tail*, while the x-features mainly focus on the edges (Fig. 3(a) shows an example). Thus, we utilize the Voronoi diagram to partition the bounding box into 15 regions in which the nearest neighbor part annotation in each region would be the same. However, the larger parts such as *wing* and *tail* always obtain a much higher scores than the smaller parts such as *eye* and *beak* do; and there are also many background pixels far from the center contained in the Voronoi diagram. To solve these issues, we introduce the inverse distance as a factor when computing the **Voronoi-based probability** $S^{n,m}_p$ in Algorithm 1, trying to keep the balance between the large part region and the small part region. Figure 5(b) shows the probability $S^{n,m}_p$ for each neuron and each part label for the same example image in Figure 3(a). From Figure 5(b) one can also see evidence that the probabilities on *wing*, *tail*, and *belly* of some neurons are higher, indicating the metric based on the Voronoi diagram enhances the evaluation on these larger parts.

Algorithm 1: The metric based on Voronoi diagram

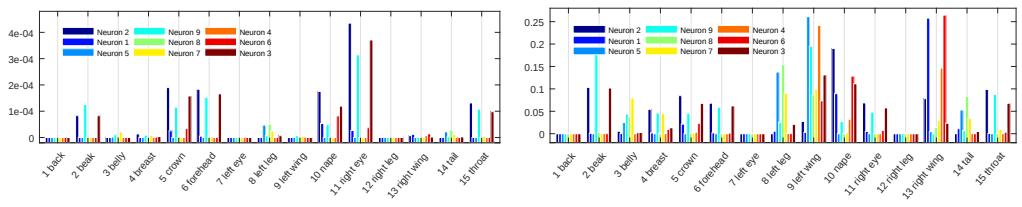
foreach *Neuron n of X layer in image I_m* **do**
  foreach *Part p with its Voronoi graph G_p* **do**
    foreach *Pixel $(i, j) \in G_p$* **do**
      Compute the distance between (i, j) and part label (i_p, j_p): $d_{ijp} = \left((i - i_p)^2 + (j - j_p)^2\right)^{\frac{1}{2}}$
    Normalize the distance d_{ijp} into $[0, 1]$, obtain the normalized distance $\bar{d}_{ijp}$
    foreach *Pixel $(i, j) \notin G_p$* **do**
      $\bar{d}_{ijp} = 1$
    $P(\text{Part}^m_p|\text{Pixel}^m_{i,j}) = 1 - \bar{d}_{ijp}$
    Compute the probability $S^{n,m}_p \triangleq P(\text{Part}^m_p|\text{Neuron}_n) =$
    $\sum_{(i,j)\in I_m} P(\text{Part}^m_p|\text{Pixel}^m_{i,j})P(\text{Pixel}^m_{i,j}|\text{Neuron}_n) = \sum_{(i,j)\in I_m}(1 - \bar{d}_{ijp})S^{n,m}_{i,j}$.

Table 3 shows more results for the mask, inpainting, and classification task. In Table 3, images are masked so that pixels that have $< \gamma$ of the highest response in the heatmap are painted as black; then an inpainting algorithm is applied to recover the masked images; finally we utilize the prediction CNN to classify the recovered images and test the classification accuracy.

Figure 7 shows the most important x-feature for several categories. The weight above the feature is $v_i E_i$, the product of the weight of the x-feature in the approximation of $\hat{y}$ timed by the activation of the x-feature.

Figure 8 shows some examples to illustrate the degeneration issue. Our propose method SRAE can avoid degeneration, and make the prediction model explainable.

Figure 9 compares x-features with directly running ExcitationBP on $\hat{y}$. One can see x-features nicely separate different discriminative aspects of the bird while ExcitationBP sometimes focus only on one part and miss others, and sometimes produces a heatmap that incorporates many parts simultaneously. Also, each x-feature seems like a distinct visual feature that makes sense at least to the authors.

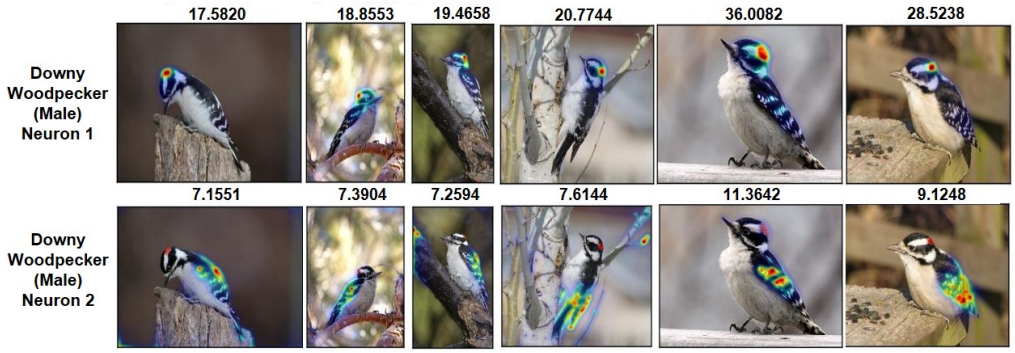


(a) (b)

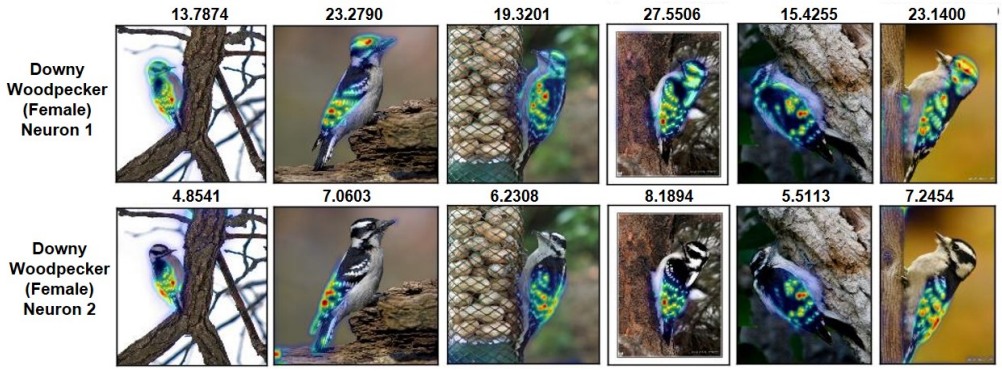
Figure 5: (a) Pixel-level probability $S^{n,m}_{i_p,j_p}$; (b) Voronoi-based probability $S^{n,m}_p$ for the example image in Figure 3(a).

Table 3: The average classification accuracy on images masked by our method and ExcitationBP, respectively. Here results with different thresholds γ are reported. With larger γ, less image region is shown to the classifier hence classification becomes more difficult.

	Method	Original Image	Mask by x-features	Mask by ExcitationBP
Classification Accuracy	$\gamma = 1\%$	0.8798	0.8762	0.7921
	$\gamma = 5\%$	0.8798	0.8428	0.6742
	$\gamma = 10\%$	0.8798	0.7771	0.5481
	$\gamma = 30\%$	0.8798	0.4097	0.1832



(a) Male downy woodpeckers



(b) Female downy woodpeckers

Figure 6: The x-features for male and female downy woodpeckers.

Figure 6 shows the x-features for male and female birds of downy woodpecker, respectively. The difference between the male and female birds of downy woodpecker is that the male birds have a red spot on the head while the female birds do not. Hence, for male birds Neuron 1 in the explanation space captures the red spot; while for female birds Neuron 1 captures the stripes on the ahead and the body. Neuron 2 in the explanation space captures the strips on the body for both male and female birds of downy woodpecker. The results indicate that the x-features in the explanation space truly justify the classification decisions by capturing the key features of the birds, and the proposed model generates visualizations which are explainable to human. However, the orthogonality and locality on the female birds suffered, probably because the most indicative feature (Neuron 1) was only available in the males, hence the algorithm went on to pick some other features into Neuron 1 as well. Neuron 2 was, however, consistent in both the male and the female birds.

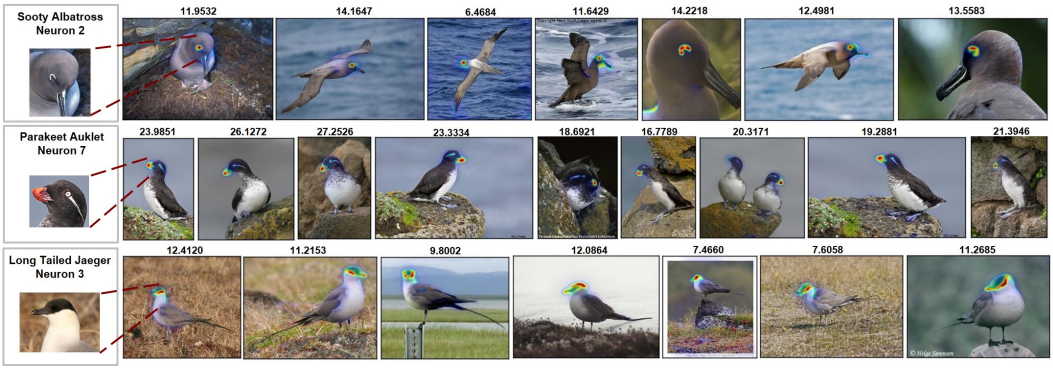
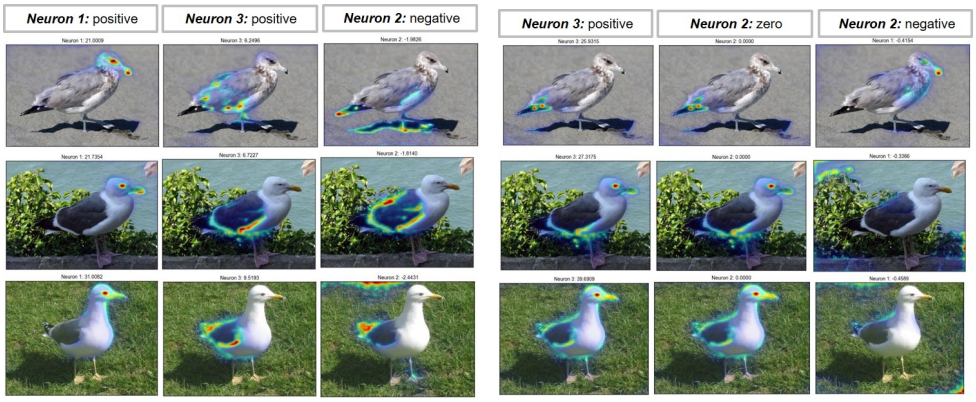
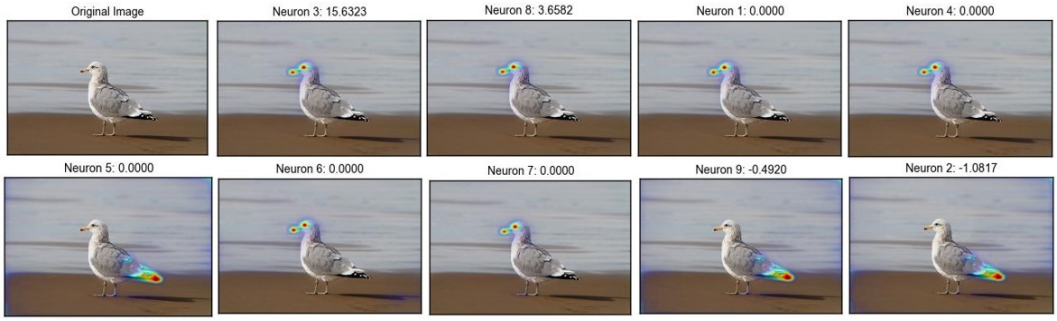


Figure 7: The most important x-feature for several categories. The weight above the feature is $v_i E_i$, the product of the weight of the x-feature in the approximation of $\hat{y}$ timed by the activation of the x-feature.

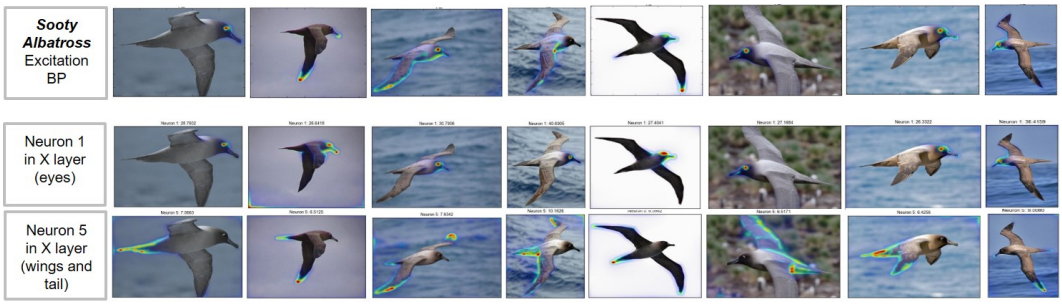


(a) (b)



(c)

Figure 8: (a) Good examples learned by SRAE, the number of the x-feature is 3, where the 3 neurons are orthogonal to each other; (b) Degenerated examples learned by NN, the number of the x-feature is 3, where the first two neurons are very similar, and there is only one positive neuron; (c) Another degenerated example learned by NN, the number of the x-feature is 9. Most of the neurons are very similar, and there are only two positive neurons.



(a)

(b)

(c)

(d)

Figure 9: ExcitationBP on the predictions and on the x-features.

Figure 10: The first row shows the heatmaps generated by ExcitationBP using the original images; the second row shows the heatmaps generated by ExcitationBP using the images with a constant vector shift similar to (Kindermans et al., 2017). The results show that ExcitationBP doesn't suffer from the issue that most of the existing saliency methods are sensitive to the transformation of the input.