Feature-flow Interpretation of Deep Convolutional Neural Networks

Xinrui Cui, Dan Wang, and Z. Jane Wang, *Fellow, IEEE*

***Abstract*—Despite the great success of deep convolutional neural networks (DCNNs) in computer vision tasks, their black-box aspect remains a critical concern. The interpretability of DCNN models has been attracting increasing attention. In this work, we propose a novel model, Feature-fLOW INterpretation (FLOWIN) model, to interpret a DCNN by its feature-flow. The FLOWIN can express deep-layer features as a sparse representation of shallow-layer features. Based on that, it distills the optimal feature-flow for the prediction of a given instance, starting from deep layers to shallow layers. Therefore, the FLOWIN can provide an instance-specific interpretation, which presents its feature-flow units and their interpretable meanings for its network decision. The FLOWIN can also give the quantitative interpretation in which the contribution of each flow unit in different layers is used to interpret the net decision. From the class-level view, we can further understand networks by studying feature-flows within and between classes. The FLOWIN not only provides the visualization of the feature-flow but also studies feature-flow quantitatively by investigating its density and similarity metrics. In our experiments, the FLOWIN is evaluated on different datasets and networks by quantitative and qualitative ways to show its interpretability.**

***Index Terms*—Model interpretability, feature-flow, sparse representation.**

I. Introduction

Deep convolutional neural networks (DCNNs) have had many successful applications, such as image classification [1], object detection [2], and image captioning [3]. Nevertheless, their complex internal mechanism is a double-edged sword, which enhances their representation ability while sacrifices their interpretability (due to the black-box nature of DCNNs). When DCNNs are deployed, it is crucial to give explanations to network decisions, especially for specific applications such as medical diagnosis and criminal justice. Therefore, the interpretation of DCNNs has attracted increasing attention in both academia and industry.

Existing visual interpretation approaches mainly focus on two directions: interpretation of a network prediction and interpretation of internal features in a neural network. Methods in the first direction explain a specific decision of neural network by providing its class-discriminative visual interpretation [4], [5], [6], [7], [8]. Among them, perturbation-based methods cannot interpret internal network features, and their performances are limited by the perturbation patch. Gradient-based methods mainly focus on the interpretation of the last convolutional layer owing to the dependence of gradient. For the latter direction, corresponding methods visualize internal features to interpret the internal mechanism of neural networks [9], [10], [11]. Nonetheless, they cannot provide an instance-specific interpretation for a given image. Previous studies do not explore the relationship between features at different layers of a DCNN. These approaches also do not interpret the hierarchical feature learning of a DCNN.

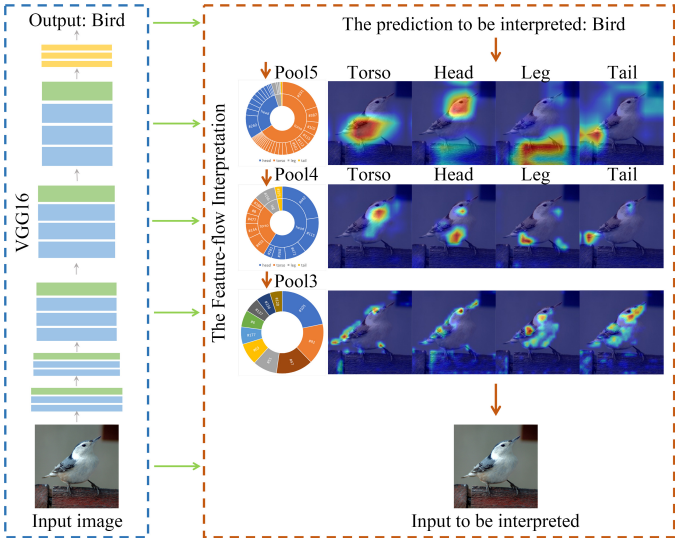


Fig. 1. The feature-flow interpretation for the VGG16 network decision. For the bird class prediction of the input image, the feature-flow provides the interpretation from deep to shallow layers. For the Pool5 and Pool4 layers, the sunburst charts present feature-flow units (the outer circle) and their corresponding interpretable body parts (the inner circle). Meantime, we also show the most critical feature-flow unit for each object part on the right side of the corresponding sunburst chart. For the Pool3 layer, the feature-flow units and their contributions to the network prediction are displayed in the doughnut chart. Besides, we show the top-4 feature-flow units in the Pool3 layer.

In neural networks, different from traditional machine learning methods, features are data-driven. Typically, there exist hundreds of units at each layer of a neural network. For brevity, we refer to feature maps in DCNNs as units. Each unit at a deep layer is computed by all units in its neighboring shallow layer. It means that the deep-layer unit is represented by a dense combination of the shallow-layer units. The internal learning structure of such a network is elusive to us.

To better understand DCNNs, we need to address the following:

- Does there exist an interpretable representation from deep

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council (NSERC), the Four Year Doctoral Fellowship and the International Doctoral Fellowship at the University of British Columbia.(Corresponding author: Dan Wang.)

X. Cui, D. Wang, and Z. J. Wang are with the Department of Electrical and Computer Engineering, University of British Columbia, BC, Canada. e-mail: (xinruic@ece.ubc.ca; danw@ece.ubc.ca; zjanew@ece.ubc.ca.)

layers to shallow layers to understand the network learning process for a network decision?

To tackle the above concern, here we first propose the feature-flow, which can distill critical features for the interpretation of the network decision. For a given instance, the proposed Feature-fLOW INterpretation (FLOWIN) model interprets the network decision by providing its feature-flow from deep to shallow layers. Specifically, it decomposes the features of a deep layer into important shallow-layer features by sparse representation. Therefore, it can interpret the network decision as well as the hierarchical network structure for feature learning.

Fig. 1 shows the feature-flow interpretation for the VGG16 network decision. For the bird class prediction of the input image, the feature-flow provides the interpretation from deep to shallow layers. In the sunburst charts of the Pool5 and Pool4 layers, the outer circle indicates feature-flow units, which are the network units selected by feature-flow, and the inner circle represents their corresponding body parts. The most critical feature-flow unit for each object part is visualized on the right side of the corresponding sunburst chart. For the instance being interpreted, the feature-flow interpretation can give not only feature-flow units but also their interpretable body-part meanings, which helps us understand its feature learning. Quantitatively, the FLOWIN model can provide the contributions of both feature-flow units and their corresponding body parts to net decisions. The doughnut chart of the Pool3 layer presents the feature-flow units and their contribution to the network prediction. Top-4 feature-flow units in the Pool3 layer are shown on the right.

The main contributions of this work are summarized as follows:

- For net decisions, the FLOWIN can give its feature-flow from deep layers to shallow layers. It distills key features to interpret the network learning process for a network decision.
- From the instance-level view, the FLOWIN presents more interpretable CNN units as its class-discriminative interpretation for a particular net decision.
- From the class-level view, the FLOWIN discovers feature-flow patterns and further analyzes intra-class and inter-class feature-flows for class predictions.
- In the feature-flow interpretation, flow units and their contributions to net predictions in different layers are shown to explain feature learning for net decisions. Based on the PASCAL VOC Part dataset, the FLOWIN further explains the interpretable meanings of feature-flow units, which also can be used to analyze the interpretability of flow units quantitatively.
- By quantitatively assessing density and similarity of feature-flow, the FLOWIN can give the general interpretation for net feature learning.
- In the FLOWIN model, we propose a novel approach to decompose deep-layer features by a sparse representation of shallow-layer features.
- In this paper, we evaluate the performance of the proposed FLOWIN on different convolutional networks and datasets by utilizing qualitative and quantitative analysis.

II. Related Work

In this section, we review the related work on the interpretation of DCNNs, categorized into the below two primary directions:

Interpretation of network prediction. The studies, which focus on providing the visual interpretation of a particular network decision, can be classified into two categories: perturbation-based and gradient-based methods.

Perturbation-based methods interpret a particular network decision by perturbing the input image [12], [13], [4], [14]. Firstly, it occludes different patches of the input image in regular grids [12] or super-pixels [13]. Then the change of the prediction is observed to find critical image regions for the prediction. The underlying principle is that the target output would drop by the maximum amount if regions, contributing maximally to the output, are occluded. Therefore, the design of perturbation patches affects the result of visual interpretation heavily. These approaches can explain a specific network prediction for the target class. It means that their results are class-discriminative visual interpretations for a given instance. However, the perturbation-based methods do not give interpretations of internal network features.

In the other category, researchers [15], [5] provide the visual interpretation of a prediction by using the gradient information of features in the last convolutional layer. These methods build a linear representation of features in the last convolutional layer to approximate the prediction. In Grad-CAM [5], weights in the linear combination are gradients of the prediction to corresponding features. By contrast, when obtaining the weights of features, CAM [15] needs to modify network architectures. Their explanations are also class-discriminative visual interpretations for an instance, which are similar to the perturbation-based approaches. However, the interpretations of these methods do not go beyond the last convolutional layer.

There exists a hybrid interpretation model, CHIP, which incorporates the merits of both categories [14]. This model is a net-perturbation approach that can distill the class-discriminative important features in different layers to interpret the network prediction. The interpretation of CHIP is not constrained to the last convolutional layer. Also, its interpretation result is not limited by the design of the perturbation patch. However, it mainly focuses on the interpretation of class-discriminative features, interpreting the relationship between internal features and the prediction. For studies on the interpretation of network prediction, they cannot explain the hierarchical learning structure of networks.

Interpretation of internal net-features. This direction also draws considerable research attention. Some methods interpret the network mechanism by visualizing internal features [16], [12]. Specifically, these approaches interpret patterns of an input image which can activate maximally internal net-features. Among them, Guided Backpropagation [16] and Deconvolution [12] can provide visual interpretation of internal features for a particular instance. Notwithstanding, such interpretation is not class-discriminative. Some studies in this direction are able to provide class-discriminative interpretation. However, they can not explain a specific network decision of an image. Methods

in this direction also cannot explain the hierarchical network feature learning.

To our best knowledge, our proposed FLOWIN model is the first attempt to understand DCNN from both directions based on the feature-flow. The feature-flow can not only give an instance-specific class-discriminative interpretation but also explain the hierarchical net-feature learning process.

Net pruning. The net pruning utilizes the sparsity regularization, which has some similarity with feature flow. However, the feature flow is different from net pruning from the following three aspects.

From the objective point-of-view, the net pruning is designed to cut the net size (or removing weights) without degrading its performance. Therefore, they retrain networks to compress networks, which can solve the problems of computationally expensive and memory intensive. In comparison, the objective of the FLOWIN model is to interpret the feature learning for net decisions by representing its feature flow from a shallow layer to a deep layer. Therefore, the FLOWIN model is a post-hoc interpretation without changing the net size.

From the optimization point-of-view, a branch of net pruning is the parameter pruning and sharing, which trains networks with sparsity constraints. At the network training process, the sparsity constraint in the net pruning is designed for net optimization by removing parameters that are not crucial to network performance. In contrast, to explain a specific output of a pre-trained network, the proposed post-hot interpretation method learns its critical net features in different layers without retraining net parameters. Consequently, in the post-hoc interpretation process, the sparsity constraint is applied to the net-unit importance for a particular net decision.

From the sparsity point-of-view, the net pruning assumes that net parameters for a multi-class classification have global sparsity for different categories. However, the FLOWIN model assumes that networks for a multi-class classification only activate a part of units for a specific class-prediction output without the assumption of global sparsity. Therefore, for a particular class-prediction of a given input, it assumes the activated net-units have local sparsity for this net-decision.

III. Methodology

A. The framework of FLOWIN model

The FLOWIN model interprets the network decision of a particular instance by presenting its feature flow. Fig. 2 illustrates the feature-flow from the deep layer to shallow layers to interpret network output for a particular instance. For image classification, the feature flow starts from the prediction in the output layer. In the first stage of the FLOWIN model, the target prediction in the output layer is decomposed as a combination of important features in the last convolutional layer.

Next, the feature flow from the deep layer to the shallow layer is built by learning a sparse representation of internal features. In a network, the feature extraction is the forward direction from the shallow to the deep layer. While the learning of feature-flow is the layer-by-layer backward direction in which features in a shallow layer are served as representation bases for the deep layer. Specifically, deep-layer features are expressed as sparse linear combinations of shallow-layer features. Along the backward direction, the obtained non-sparse representation bases in the shallow layer are the target features for the next decomposition, and shallower-layer features become new representation bases. Finally, the feature flow is formed by the layer-by-layer sparse representation.

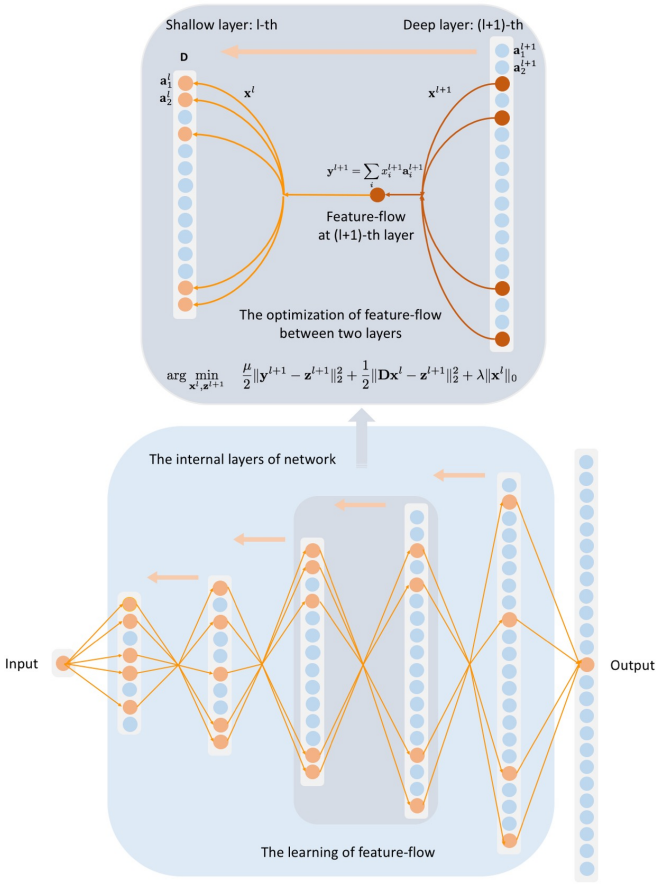


Fig. 2. The illustrative feature-flow from the deep layer to shallow layers for a network decision.

B. The sparse representation of feature-flow

In the FLOWIN model, shallow-layer features are the representation bases, and features in the deep layer are the target being decomposed, as illustrated in Fig. 2. The basic idea is that the feature-flow is sparse, which means that deep-layer features can be represented as a linear combination of a few features (bases) in the shallow layer. And the feature-flow units at a specific layer are the network units that have the non-sparse weights in its sparse representation.

The sparse-representation bases. In the feature-flow between the l-th and the $(l+1)$-th layer, the set of decomposition bases is denoted as

$$\mathbf{D} = [\mathbf{a}_1^l \ \mathbf{a}_2^l \cdots \mathbf{a}_i^l \cdots \mathbf{a}_m^l] \tag{1}$$

where the i-th feature in the l-th layer of size $\sqrt{n} \times \sqrt{n}$ pixels is ordered lexicographically as a column vector $\mathbf{a}_i^l$ of size $n \times 1$.

The corresponding weight vector for bases in the l-th layer is denoted as

$$\mathbf{x}^l = [x_1^l \; x_2^l \cdots x_i^l \cdots x_m^l]^T \tag{2}$$

where the x_i^l is the weight for i-th basis in the l-th layer.

Features from $(l+1)$-th layer converge into feature-flow at the $(l+1)$-th layer, as shown in Fig. 2. And the feature-flow at the $(l+1)$-th layer is decomposed as a sparse representation of bases in the l-th layer. Specifically, the feature-flow at the $(l+1)$-th layer is expressed as

$$\mathbf{y}^{l+1} = \sum_i x_i^{l+1} \mathbf{a}_i^{l+1} \tag{3}$$

where the x_i^{l+1} is the weight for $\mathbf{a}_i^{l+1}$ in the $(l+1)$-th layer.

Owing to the convolutional operation, the size of a deep-layer feature is smaller than that of a shallow-layer feature. In the sparse representation of feature-flow, features in the shallow layer need to be downscaled to the spatial size of features in the deep layer. The difference in feature size at different layers brings the noise to the learning process of feature-flow. Therefore, the FLOWIN model introduces an auxiliary denoised term that demands the proximity between the feature-flow being decomposed and its denoised (and unknown) version.

The feature-flow of the last convolutional layer is based on the target class prediction being explained. This feature-flow is the combination of class-discriminative features weighted by their importance values. We adopt the CHIP model [17] to distill class-discriminative features and their importance values.

C. FLOWIN model

The feature-flow is optimized by solving the sparse representation between the shallow layer and the deep layer. The optimization problem of feature-flow between the l-th and the $(l+1)$-th layer is formulated as

$$\begin{aligned} \arg\min_{\mathbf{x}^l, \mathbf{z}^{l+1}} \quad & \frac{\mu}{2}\|\mathbf{y}^{l+1} - \mathbf{z}^{l+1}\|_2^2 \\ & + \frac{1}{2}\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{x}^l\|_0 \end{aligned} \tag{4}$$

where λ and μ are regularization parameters.

In the FLOWIN algorithm, the problem in Eq. (4) is optimized iteratively. In each iteration, there are two stages: one for $\mathbf{x}^l$ and one for $\mathbf{z}^{l+1}$.

In Eq. (4), the first term is the log-likelihood force that demands the proximity between the feature-flow at the $(l+1)$-th layer $\mathbf{y}^{l+1}$ being decomposed and its denoised (and unknown) version $\mathbf{z}^{l+1}$. The second term is the loss function of the sparse representation. It makes sure that in the feature-flow, the deep-layer flow has a sparse representation with a bounded error. The third term is the sparsity regularization that constrains the feature-flow to be sparse. The notation $\|\mathbf{x}^l\|_0$ is the l_0 norm of $\mathbf{x}^l$, counting the nonzero entries of $\mathbf{x}^l$.

In the FLOWIN algorithm, the two variables $\mathbf{x}^l, \mathbf{z}^{l+1}$ are optimized separately.

The optimization for $\mathbf{x}^l$. In the first stage, we fix $\mathbf{z}^{l+1}$ and aim to find the best weight $\mathbf{x}^l$. In order to update $\mathbf{x}^l$, the optimization problem becomes

$$\arg\min_{\mathbf{x}^l} \quad \frac{1}{2}\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{x}^l\|_0 \tag{5}$$

For the optimization of $\mathbf{x}^l$, it starts with an initialization $\mathbf{z}^{l+1} = \mathbf{y}^{l+1}$.

The problem in (5) is NP hard owing to the discrete and nonconvex nature of l_0 norm. It is combinatorial and too complex to solve and cannot be solved in a straightforward way. To get an approximate solution to the problem, one approach is to replace l_0 norm with l_1 norm, since l_1 norm is naturally the best convex approximation of l_0 norm. Therefore the optimization problem in (5) can be converted to

$$\arg\min_{\mathbf{x}^l} \quad \frac{1}{2}\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{x}^l\|_1 \tag{6}$$

According to the objective function (6), it can be converted into the equivalent formulation

$$\begin{aligned} & \min_{\mathbf{x}^l, \mathbf{u}} \frac{1}{2}\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{u}\|_1 \\ & \text{subject to} \quad \mathbf{x}^l = \mathbf{u} \end{aligned} \tag{7}$$

The augmented Lagrangian for the problem is

$$\begin{aligned} \mathcal{L}(\mathbf{x}^l, \mathbf{u}, \mathbf{d}) = & \frac{1}{2}\left\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\right\|_2^2 + \lambda\|\mathbf{u}\|_1 \\ & + \mathbf{d}^T(\mathbf{x}^l - \mathbf{u}) + \frac{\rho}{2}\left\|\mathbf{x}^l - \mathbf{u}\right\|_2^2 \end{aligned} \tag{8}$$

This equation can be rewritten as

$$\begin{aligned} \mathcal{L}(\mathbf{x}^l, \mathbf{u}, \mathbf{w}) = & \frac{1}{2}\left\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\right\|_2^2 + \lambda\|\mathbf{u}\|_1 \\ & + \frac{\rho}{2}\left\|\mathbf{x}^l - \mathbf{u} - \mathbf{w}\right\|_2^2 \end{aligned} \tag{9}$$

where

$$\mathbf{w} \equiv -\frac{1}{\rho}\mathbf{d} \tag{10}$$

The problem in Eq. (9) is converted into a simple problem by a careful choice of the new variable. The optimization function over the variable $\mathbf{x}^l$ is

$$\mathbf{x}_{k+1}^l \leftarrow \arg\min_{\mathbf{x}^l} \frac{1}{2}\left\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\right\|_2^2 + \frac{\rho}{2}\left\|\mathbf{x}^l - \mathbf{u}_k - \mathbf{w}_k\right\|_2^2 \tag{11}$$

The solution of (11) is

$$\mathbf{x}_{k+1}^l \leftarrow (\mathbf{D}^T\mathbf{D} + \rho\mathbf{I})^{-1}\left[\mathbf{D}^T\mathbf{z}^{l+1} + \rho(\mathbf{u}_k + \mathbf{w}_k)\right] \tag{12}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{D}^T$ denotes the transpose of $\mathbf{D}$.

For calculating $\mathbf{u}$, the optimization problem to be solved is

$$\mathbf{u}_{k+1} \leftarrow \arg\min_{\mathbf{u}} \frac{\lambda}{\rho}\|\mathbf{u}\|_1 + \frac{1}{2}\left\|\mathbf{u} - (\mathbf{x}_{k+1}^l - \mathbf{w}_k)\right\|_2^2 \tag{13}$$

The solution of optimization problem in (13) is the soft threshold.

$$\mathbf{u}_{k+1} \leftarrow soft(\mathbf{x}_{k+1}^l - \mathbf{w}_k, \frac{\lambda}{\rho}) \tag{14}$$

The Lagrange multiplier can be updated to

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - (\mathbf{x}_{k+1}^l - \mathbf{u}_{k+1}) \tag{15}$$

The optimization for $\mathbf{z}^{l+1}$. In the second stage, given that the optimization is considered over the variable $\mathbf{z}^{l+1}$, we fix the

Algorithm 1: Pseudocode of the FLOWIN Algorithm

Input: the target-class prediction for the instance being explained;
its original network features from the deep layer to the shallow layer;
its feature-flow in the last (L-th) convolutional layer $\mathbf{y}^L$;
Output: the optimal feature-flow interpretation of a particular net decision;
Initialization: set $l = L-1$, $\lambda > 0$, $\mu \geq 0$;
repeat
  The optimization problem of the feature-flow between the l-th and the $(l+1)$-th layer:
  $\arg\min_{\mathbf{x}^l, \mathbf{z}^{l+1}} \quad (\mu/2)\|\mathbf{y}^{l+1} - \mathbf{z}^{l+1}\|_2^2 + (1/2)\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{x}^l\|_0$ (in Eq. (4));
  initialization: $\mathbf{z}^{l+1} = \mathbf{y}^{l+1}$
  repeat
    1: **update:** $\mathbf{x}^l \leftarrow \arg\min_{\mathbf{x}^l} \quad (1/2)\|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \lambda\|\mathbf{x}^l\|_1$ (in Eq. (6));
    2: **update:** $\mathbf{z}^{l+1} \leftarrow \arg\min_{\mathbf{z}^{l+1}} \quad \|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \mu\|\mathbf{y}^{l+1} - \mathbf{z}^{l+1}\|_2^2$ (in Eq. (16));
  until *stopping criterion is satisfied*;
  return the optimal weight vector $\mathbf{x}^l$ of sparse bases in the l-th layer;
  the feature-flow of the l-th layer $\mathbf{y}^l = \sum_i x_i^l \mathbf{a}_i^l$;
  update iteration: $l \leftarrow l-1$;
until *the feature-flow is optimized from the deep layer to the shallow layer*;
return optimal weight vectors $\mathbf{x}^*$ of sparse bases for the feature-flow;

updated weights of bases and turn to update $\mathbf{z}^{l+1}$. Therefore, the optimization problem to be solved becomes

$$\arg\min_{\mathbf{z}^{l+1}} \quad \|\mathbf{D}\mathbf{x}^l - \mathbf{z}^{l+1}\|_2^2 + \mu\|\mathbf{y}^{l+1} - \mathbf{z}^{l+1}\|_2^2 \tag{16}$$

The optimization in Eq. (16) is a simple quadratic problem that has a closed-form solution of the form

$$\mathbf{z}^{l+1} \leftarrow (1+\mu)^{-1}(\mathbf{D}\mathbf{x}^l + \mu\mathbf{y}^{l+1}) \tag{17}$$

It can also be regarded as the averaging of denoised feature-flow and the original noisy one at the $(l+1)$-th layer.

Given the updated $\mathbf{z}^{l+1}$, the optimization problem of $\mathbf{x}^l$ in Eq. (6) is repeated. Once this is done, a next optimization of $\mathbf{z}^{l+1}$ in Eq. (16) should be calculated, and so on and so forth. After the optimization of Eq. (4), we obtain the optimal sparse representation of feature-flow between the $(l+1)$-th and the l-th layer. Subsequently, the optimization is conducted backward for the feature-flow from the l-th to the $(l-1)$-th layer. Finally, the interpretation model can get the optimal feature-flow from the deepest layer to the shallowest layer.

The pseudocode of the FLOWIN algorithm is shown in Algorithm 1.

D. The Similarity of Feature-flow

FLOWIN is an interpretation method for a particular instance, which shows the feature-flow for its network decision. From the class-level perspective, feature-flow can also be used to explore the rule of internal feature learning in networks.

The similarity between feature-flows is introduced to analyze feature-flows of different images. For a certain layer, the similarity of feature-flows is denoted as

$$s_l = \frac{|U_A^l \bigcap U_B^l|}{|U_A^l \bigcup U_B^l|} \tag{18}$$

where U_A^l and U_B^l are the subsets of l-th layer's units belonging to feature-flows of the compared images A and B, respectively. And $|\cdot|$ denotes the number of elements in a set.

E. The Density of Feature-flow

Here, we define the density of feature-flow for a certain layer which is computed by the number of units in the feature-flow divided by the total number of units in the layer. The density indicates the sparsity of feature-flow. Feature-flow with low density is more sparse. The density of feature-flow in l-th layer is defined as

$$t^l = \frac{N_{flow}^l}{N^l} \tag{19}$$

where N_{flow}^l is the unit number of l-th layer in a feature-flow and N^l is the total number of units in this layer.

IV. Experiments

Experiments are conducted in three aspects where section IV-A and IV-B are the analysis of feature-flow for the PASCAL VOC Part dataset [18] and the ImageNet dataset [19] and section IV-C is the general interpretation for the net feature learning based on the feature-flow. The proposed method is to interpret two convolutional networks, VGG16 [20] and ResNet101 [21], for image classification on the PASCAL VOC Part dataset and the ImageNet dataset.

A. Feature-flow Interpretation on PASCAL VOC Part Dataset

In the experiment, we utilize qualitative and quantitative ways to evaluate the performance of the FLOWIN model on PASCAL VOC Part dataset. Qualitatively, the FLOWIN model provides visual interpretation results to explain network predictions at the instance level (IV-A1) and the class level (IV-A2). Quantitatively, we evaluate the interpretability of units in feature-flow on the dataset (IV-A3).

The PASCAL VOC Part Dataset: It is a subset of PASCAL VOC 2010 dataset [22] with object part annotations. It provides segmentation masks for each body part of the object. To be consistent with most part-localization studies, here we use animal categories for evaluation. This dataset contains six

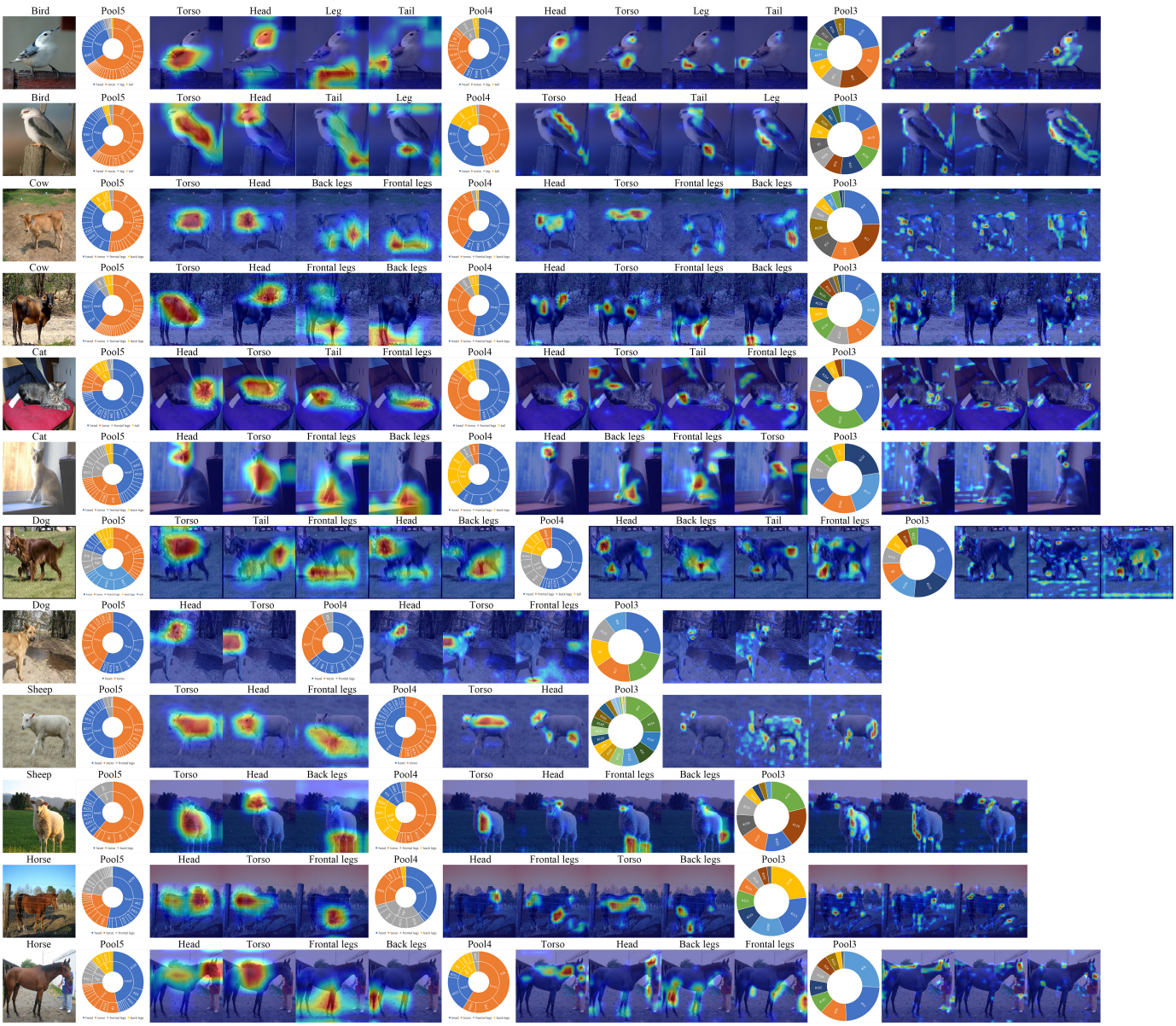


Fig. 3. The instance-level feature-flow interpretation for the VGG16 network decision on the PASCAL VOC Part dataset. The first column shows the randomly selected twelve images from six animal classes. Other columns are their feature-flow interpretations. For each instance, its interpretation gives the feature-flow from the Pool5 to the Pool3 layer. For Pool5 and Pool4 layers, the sunburst charts present the feature-flow units (the outer circle) and their corresponding interpretable body parts (the inner circle). The proportion of each feature-flow unit in the outer circle indicates its contribution to the network decision. Similarly, the contribution of a body part for the prediction is indicated by its proportion in the inner circle. Meantime, we also present the most critical feature-flow unit for each body part on the right side of the sunburst chart. For the Pool3 layer, feature-flow units distill the low-level features such as boundary. Thus, we only display feature-flow units and their contributions to network predictions in doughnut charts. Also, we show top-3 critical feature-flow units in the Pool3 layer.

animal categories (bird, cat, cow, dog, horse, and sheep). As in [23], we merge small object parts (such as the head, beak, left eye, and right eye) into several landmark parts (such as the head) for the six animal categories. For animal classes, we use landmark part annotations (head, torso, frontal legs, back legs, and tail) to qualitatively and quantitatively evaluate the proposed interpretation model.

1) ***Instance-level Feature-flow Interpretation****:* For a given instance, the feature-flow interprets its network decision by distilling critical features from deep to shallow layers. In this section, we provide the instance-level feature-flow interpretation for the VGG16 network decision.

From deep to shallow layers, to interpret the network decision for a specific input, the feature-flow interpretation provides its flow units and corresponding weights in each layer. Moreover, it also explains each feature-flow unit by assigning it with a specific interpretable object body part. Specifically, each unit in the feature-flow is aligned with the body part that achieves the highest Intersection over Union (IoU) than other parts. Based on that, we analyze the contributions of different body parts to the network prediction.

Fig. 3 presents the instance-level feature-flow interpretation for the VGG16 network decision on PASCAL VOC Part dataset. Specifically, we randomly select twelve images from

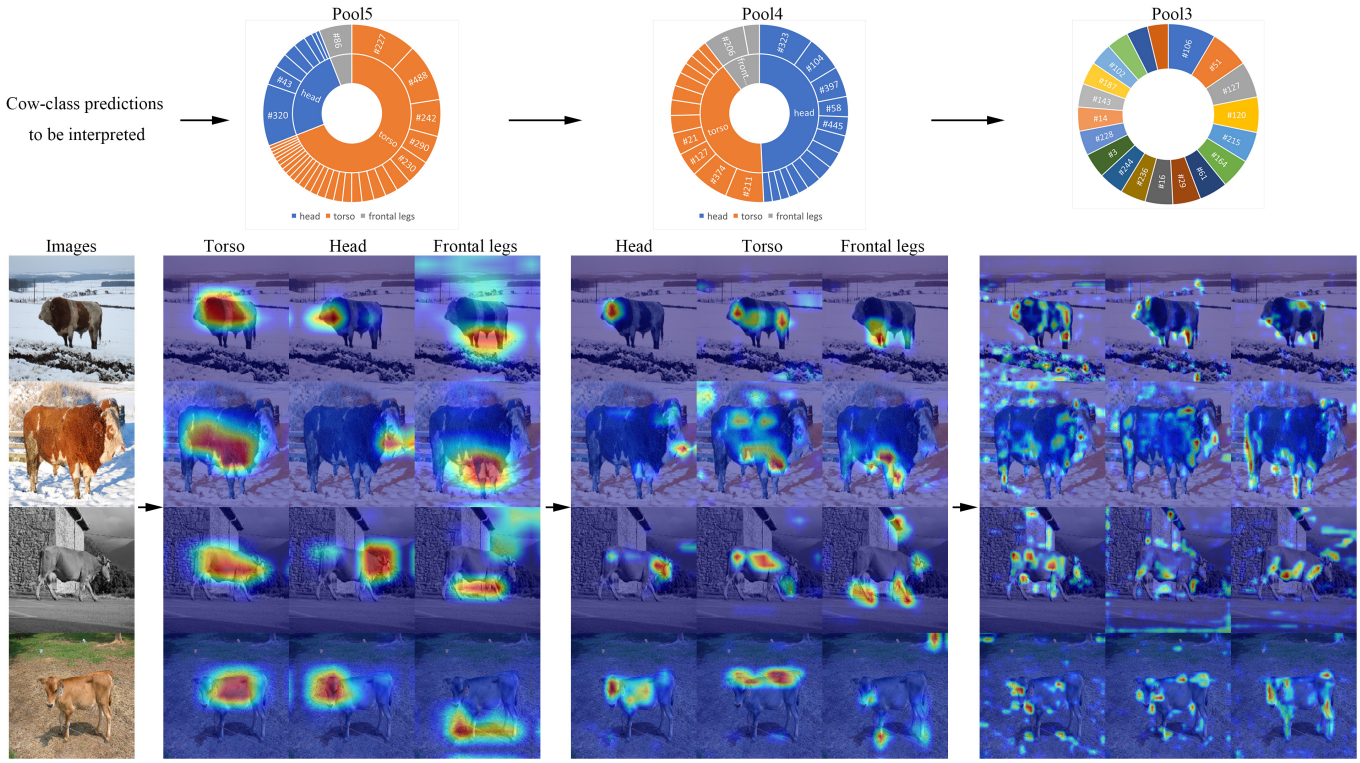


Fig. 4. The class-level feature-flow interpretation for the VGG16 network on the PASCAL VOC Part dataset. The first row shows feature-flow interpretations for the cow class in the last three pooling layers. Images in the first column are the randomly selected cow images from the PASCAL VOC Part validation dataset. Other columns are their feature-flow interpretations. For each instance, we display the most critical feature-flow unit for each object part in the above cow-class sunburst chart. For the Pool3 layer, we present top-3 feature-flow units for the cow-class prediction.

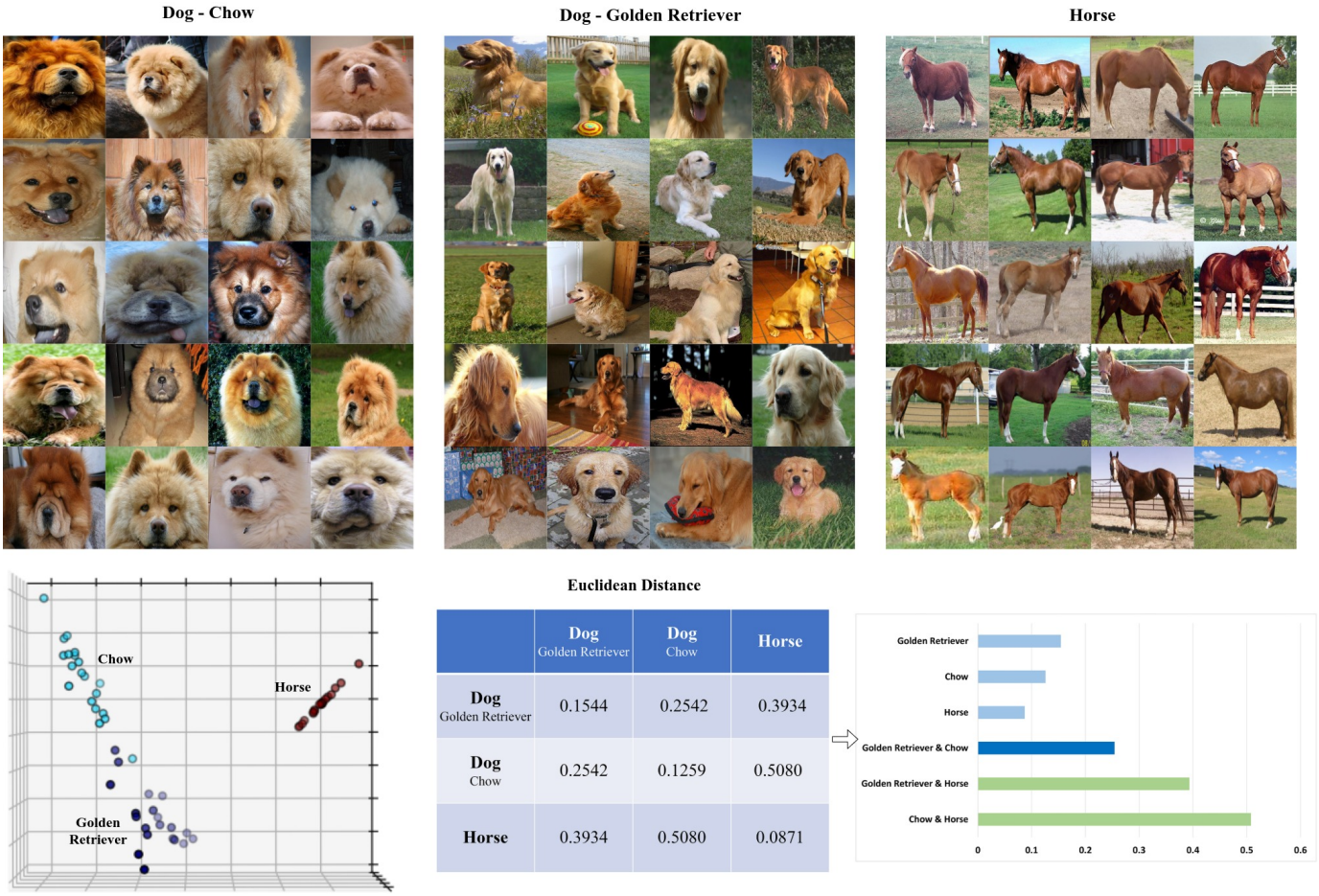
the six animal classes, as shown in the first column of Fig. 3. In other columns, we provide corresponding feature-flow interpretations for network decisions. The FLOWIN model can provide the feature-flow from the top to the bottom layers for the interpretation of the network decision. In Fig. 3, due to the space limit, we only present the feature-flow interpretation for the last three pooling layers of VGG16.

For Pool5 and Pool4 layers, the sunburst charts display feature-flow units (the outer circle) and their corresponding interpretable body parts (the inner circle). For example, in the feature-flow of the Pool5 layer for the first bird image (the first sunburst chart in the top row of Fig. 3), the feature-flow unit 331 and unit 260 are aligned with the torso, and head respectively. The proportion of each feature-flow unit in the outer circle means its contribution to the network decision. Similarly, the contribution of each body part for the prediction is indicated by its proportion in the inner circle. Meanwhile, in each sunburst chart, the body parts (torso, head, leg, and tail for the bird class) are sorted in descending order of the sum proportion of their corresponding feature-flow units. For bird images, the number and the proportion of feature-flow units for torso and head are larger than those for leg and tail. It is also consistent with human understanding for bird classification since torso and head are more discriminative than leg and tail. Because of the space limit, we only annotate the indices of units with large proportions.

Meanwhile, we also present the most critical feature-flow unit for each object part on the right side of the corresponding sunburst chart. For instance, for the Pool5 and Pool4 layers of the first bird image, the activation region of the most critical feature-flow unit for each object part is interpretable for the human visual understanding. For the Pool3 layer, we display the top-3 feature-flow units for network outputs. Compared with deep layers, the feature-flow units in the Pool3 layer distill low-level features such as object boundary, as shown in the top-2 feature-flow units for the first bird instance. Therefore, we only show the feature-flow units and their contributions to the prediction in the doughnut chart.

*2) **Class-level Feature-flow Interpretation**:* For a particular class, the FLOWIN model learns its class-level feature-flow interpretation from the target-class image dataset. In this section, the experiment is to illustrate the class-level feature-flow interpretation for the VGG16 network on the PASCAL VOC Part dataset.

Fig. 4 shows the class-level feature-flow interpretation for the VGG16 network on the PASCAL VOC Part dataset. The cow-class feature-flow interpretation is learned from all cow images (123 images) in the PASCAL VOC Part validation dataset. The class-level feature-flow interprets the target-class prediction from the deepest to the shallowest layer. In Fig. 4, due to the space limit, the class-level interpretation only presents its feature-flow in the Pool5, Pool4, and Pool3 layers of VGG16. Concretely, in Fig. 4, the first row presents feature-flow interpretations for the cow class in the last three pooling layers. Here, we randomly select four images from the cow dataset. Other columns in Fig. 4 are their feature-flow interpretations.



	Dog Golden Retriever	Dog Chow	Horse
Dog Golden Retriever	0.1544	0.2542	0.3934
Dog Chow	0.2542	0.1259	0.5080
Horse	0.3934	0.5080	0.0871

Fig. 5. Intra-class and inter-class feature-flow interpretation. The dog and horse are chosen to investigate the feature-flow between classes. Chow and golden retriever belonging to the dog class are selected to analyze the intra-class feature-flow. The top half displays twenty tested images for chow, golden retriever, and horse, respectively. The bottom half shows the comparative results between intra-class and inter-class feature-flow interpretations. The left bottom plot visualizes the 3D PCA for the feature-flow in the Pool5 layer. The right two charts represent the cluster distances.

For each instance, we display the most critical feature-flow unit for each object part in the above cow-class sunburst chart. For the Pool3 layer, we present top-3 feature-flow units for the cow-class prediction.

Consistent with the instance-level feature-flow interpretations for cow images (the third and fourth rows in Fig. 3), the contribution of the torso part to the cow class prediction and its number of corresponding feature-flow units are the largest in the class-level feature-flow for the Pool5 layer. Similarly, this observation also applies to the head part in the Pool4 layer for both the instance-level and the class-level interpretations. In Fig. 4, the selected feature-flow unit for each object part is understandable as an interpretation for the cow class prediction. According to Fig. 3 and Fig. 4, it is also observed that the feature-flow unit 488 in the Pool5 layer contributes significantly to the torso part in both the instance-level and the class-level interpretations.

Fig. 5 displays the analysis of the intra-class and inter-class feature-flow interpretation. The network outputs the predictions of six animal classes, including dog and horse. The dog and horse are chosen to investigate the feature-flow between classes. Chow and golden retriever belonging to the dog class are selected to analyze the intra-class feature-flow.

The top half displays twenty tested images for chow, golden retriever, and horse, respectively. From the visual aspect, chow and golden retriever share some similarities (golden fur). Meanwhile, they also have distinctive features, such as the shape of the ears. In comparison, the difference between horse and dog is more significant than that between the two dog species. The VGG16 can accurately predict the tested images as their corresponding ground-truth (dog or horse). Owing to the high semantic and the class-discriminative properties of the deep layer, we select the feature-flow in the Pool5 layer for analysis.

The left bottom plot visualizes the 3D PCA for the feature-flow in the Pool5 layer. This plot displays three clusters for chow, golden retriever, and horse. It means the feature-flow can be used to interpret the net output for the intra-class and inter-class images. Meantime, it is observed that the distance between chow and golden retriever is smaller than that between horse and dog, which is consistent with our visual perception.

In the table of Fig. 5, each diagonal value represents the average Euclidean distance for the corresponding cluster, which

TABLE I
THE QUANTITATIVE EVALUATION OF THE FEATURE-FLOW INTERPRETABILITY

Class \ Layer		Layer 5		Layer 4		Layer 3	
		Feature-flow Units	Top Activated Units	Feature-flow Units	Top Activated Units	Feature-flow Units	Top Activated Units
Bird	Head	**10.837**	5.478	**8.042**	0.647	**3.835**	0.873
	Torso	**15.817**	7.668	**7.115**	0.965	**3.350**	0.590
	Leg	**2.698**	1.483	**3.256**	0.515	**1.857**	0.315
	Tail	**3.802**	1.729	**3.872**	0.749	**2.827**	0.509
	All Parts	**33.154**	16.358	**22.285**	2.876	**11.869**	2.287
Sheep	Head	**9.641**	4.773	**6.326**	0.752	**3.052**	0.598
	Torso	**16.260**	8.431	**8.147**	0.868	**3.510**	1.100
	Frontal Leg	**2.013**	1.429	**2.591**	0.280	**0.901**	0.325
	Back Leg	**1.271**	0.715	**1.761**	0.233	**1.361**	0.221
	All Parts	**29.185**	15.348	**18.825**	2.133	**8.824**	2.244
Horse	Head	**8.202**	5.134	**5.354**	0.742	**2.148**	0.399
	Torso	**16.111**	5.313	**6.051**	0.558	**1.884**	0.553
	Frontal Leg	**4.135**	2.656	**4.747**	1.636	**1.788**	0.298
	Back Leg	**2.916**	1.834	**4.308**	0.719	**1.895**	0.323
	All Parts	**31.364**	14.937	**20.460**	3.655	**7.715**	1.573
Dog	Head	**10.357**	6.777	**6.689**	1.367	**2.413**	0.565
	Torso	**11.115**	3.449	**5.461**	0.586	**1.930**	0.543
	Frontal Leg	**3.391**	1.599	**5.065**	0.428	**1.883**	0.523
	Back Leg	**4.450**	0.883	**3.502**	0.573	**1.943**	0.540
	Tail	**2.272**	0.920	**1.846**	0.417	**1.608**	0.425
	All Parts	**31.585**	13.628	**22.563**	3.371	**9.777**	2.596
Cow	Head	**10.175**	4.508	**5.875**	0.682	**2.563**	0.791
	Torso	**13.344**	5.962	**6.729**	0.552	**2.530**	1.117
	Frontal Leg	**3.313**	2.175	**4.197**	0.923	**2.589**	0.361
	Back Leg	**3.079**	2.030	**4.417**	0.569	**2.371**	0.324
	All Parts	**29.911**	14.675	**21.218**	2.726	**10.053**	2.593
Cat	Head	**17.251**	13.215	**7.332**	1.333	**2.147**	0.634
	Torso	**10.839**	6.295	**4.116**	1.023	**1.509**	0.750
	Frontal Leg	**8.986**	2.584	**3.883**	0.476	**1.408**	0.438
	Back Leg	**4.511**	1.691	**2.892**	0.388	**1.368**	0.231
	Tail	**3.836**	0.935	**1.855**	0.170	**1.371**	0.604
	All Parts	**45.423**	24.720	**20.078**	3.390	**7.803**	2.657

is calculated as

$$dist(Cluster) = \frac{1}{N_C} \sum_{n} \sqrt{(\mathbf{x}_{C,n} - \bar{\mathbf{x}}_C)^T (\mathbf{x}_{C,n} - \bar{\mathbf{x}}_C)} \tag{20}$$

The average Euclidean distances for the golden retriever, chow, and horse are 0.1544, 0.1259, and 0.0871, respectively.

The non-diagonal values are the Euclidean distance between two clusters, which is obtained by

$$dist(Cluster_i, Cluster_j) = \sqrt{(\bar{\mathbf{x}}_{C_i} - \bar{\mathbf{x}}_{C_j})^T (\bar{\mathbf{x}}_{C_i} - \bar{\mathbf{x}}_{C_j})} \tag{21}$$

The intra-class distance between the golden retriever and chow is 0.2542, which is smaller than the inter-class distance between horse and dog (0.3934, and 0.508). The observation from the cluster distance is also aligned with the visual perception from images.

*3) **Quantitative Evaluation of the Feature-flow Interpretability**:* In this section, the experiment is conducted to quantitatively evaluate the interpretability of the feature-flow based on the PASCAL VOC Part dataset. Also, the proposed model is compared with the naive approach of selecting top-k activated units, to show that units in the feature-flow are more interpretable than the compared top activated units.

The interpretability of units obtained by different methods is assessed based on the interpretable body-part annotation in the PASCAL VOC Part dataset. In the experiment, we utilize the animal classes from the PASCAL VOC Part validation dataset for evaluation, which contains bird, sheep, horse, dog, cow, and cat. Images in each class have their body annotations, such as head, torso, leg, and tail for the bird class. For each class, we randomly select fifty images for the interpretability evaluation. For an instance being interpreted, the unit number of the top activated method in each layer is the same as that of the FLOWIN method.

In the quantitative comparison, the interpretability of units obtained by different methods is calculated by the Intersection over Union (IoU) metric, which is defined as follows:

$$\textbf{IoU} = \frac{\textit{Area of Overlap}}{\textit{Area of Union}} \tag{22}$$

where the numerator refers to the area of overlap between the activation region of a network unit and the ground-truth of a body part for the target object. The denominator denotes the area of the union, which includes both regions. A large IoU between the ground-truth of a body part and the activation region of a unit indicates a high interpretability of the unit.

For the feature-flow of an image, the interpretability for a particular body part in a layer is evaluated by the highest IoU between the target body part and feature-flow units in the layer. For the feature-flow of a specific class, the interpretability for a particular body part in a layer is the accumulated IoU for

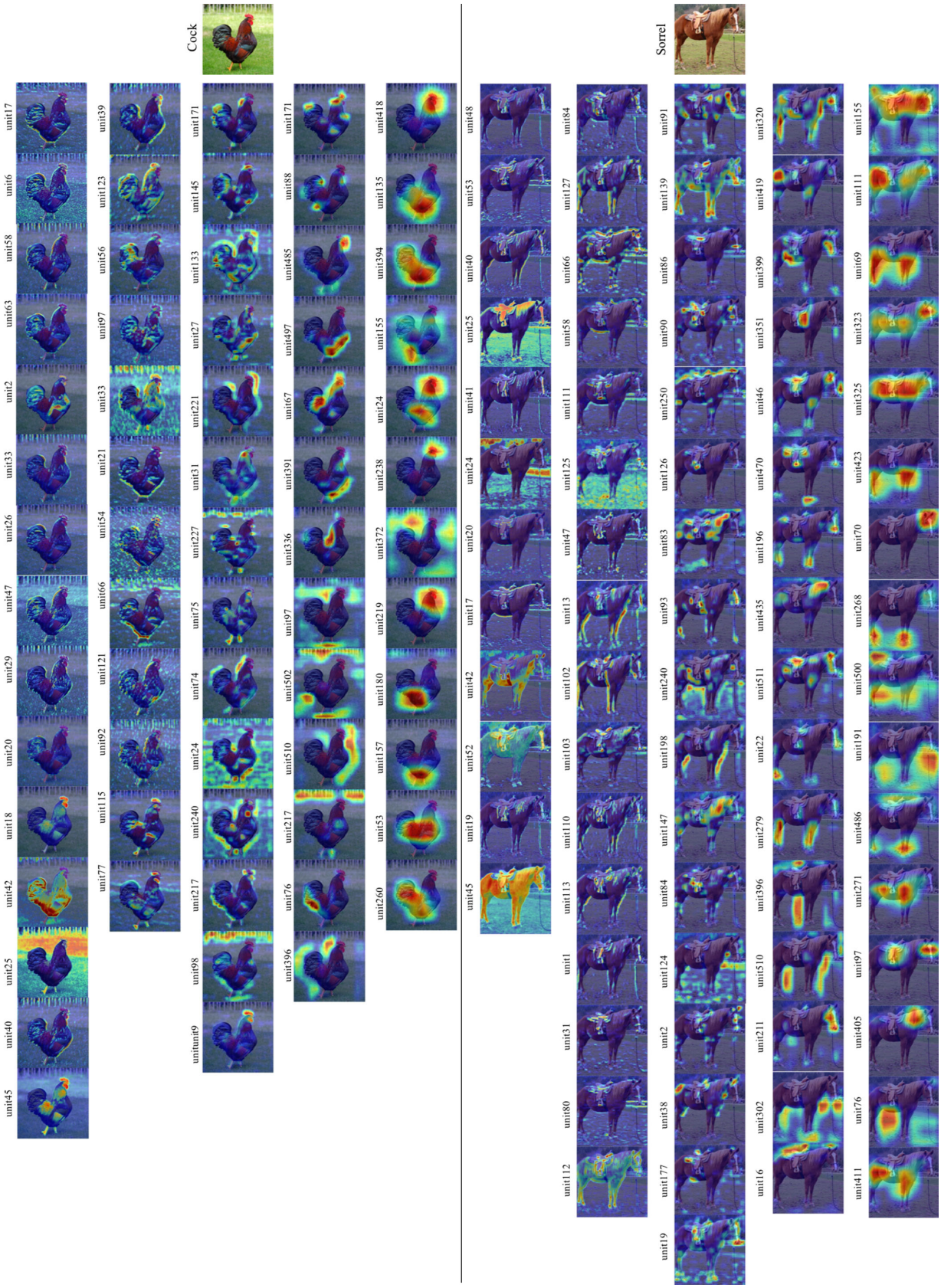


Fig. 6. The visualization of features in feature-flows of the VGG16-net decisions for cock and sorrel images. For a given instance, features in its flow from pool1 to pool5 layers are displayed from its first to its last column, respectively.

images in the target class. The same process is used to assess the interpretability of units in the top-k activated method.

Tab. I shows the quantitative results of the interpretability of two compared methods. It is observed that the interpretability of feature-flow units is better than that of the top activated units for all image classes and their body parts. For example, the interpretability of feature-flow units for bird head is nearly two times higher than that of top activated units (10.837 and 5.478, respectively). Therefore, compared with the top activated method, the feature-flow can distill network units that are more interpretable as the explanation for a network decision.

B. Feature-flow Interpretation on ImageNet dataset

In this section, we analyze the feature-flow interpretation for VGG16 and ResNet101 on the ImageNet dataset. Concretely, in section IV-B1, we visualize the feature-flow interpretation from deep to shallow layers for a particular net decision. In section IV-B2, the feature-flow interpretation for intra-class net decisions is analyzed to explain its convergence and divergence within the same class.

For illustration, in this experiment, we randomly choose two classes to show their results. To testify FLOWIN, we also have statistical results based on the ImageNet validation dataset, which contains 1000 classes and 50 images for each class.

1) ***Feature-flow Interpretation for a Particular Net Decision****:* In this section, experiments are conducted to visualize the feature-flow. The feature-flow interpretation is instance-specific as well as class-discriminative. We visualize the feature-flow to confirm that it distills key features for the network prediction being explained.

Here, VGG16 and ResNet101 pre-trained on ImageNet are selected as network architectures for the feature-flow visualization. It should be noted that ResNet101 is much deeper than VGG16. Concretely, VGG16 has 16 layers, while ResNet101 is 101 layers deep. Images from the validation dataset are used as instances to be interpreted by the FLOWIN model. For VGG16, we select 5 pooling layers to analyze its feature-flow. For ResNet101, 18 layers from 4 blocks are chosen to visualize its feature-flow.

The Visualization of Feature-flow Interpretation: Fig. 6 visualizes features in their flows of VGG16-net decisions for a cock image and a sorrel image. It can be observed that the feature-flow can extract critical features in the network to explain the network decision.

The Interpretation based on the Feature-Flow in Deep Layers: In deep layers, feature-flow mainly focuses on high semantic features such as the target object parts. For example, in Fig. 6, unit 418 and unit 157 in the Pool5 layer of the left feature-flow concentrate on cock head and feet, respectively. And unit 325 and unit 69 in the Pool5 layer of the right feature-flow focus on sorrel upper body and legs, respectively.

The Interpretation based on the Feature-Flow in Shallow Layers: In shallow layers, feature-flow contains more detailed features such as color and boundary. In the feature-flow of the cock image in Fig. 6, unit 42 in the pool1 layer distills black color. And in the feature-flow of the sorrel image, unit 45 in the pool1 layer highlights sorrel color.

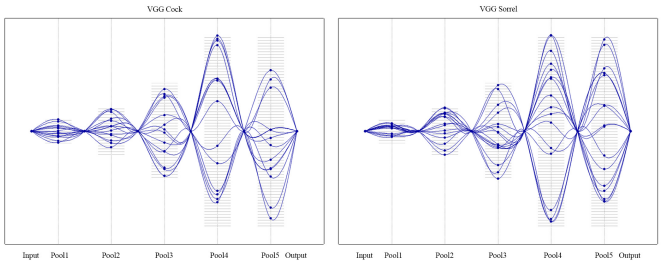


Fig. 7. The flow visualization of the decision of VGG16. The left and right plots are for a cock image and a sorrel image, respectively.

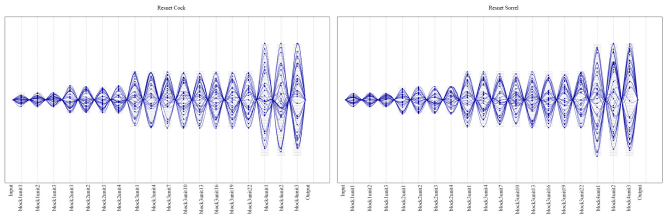


Fig. 8. The flow visualization of the decision of ResNet101. The left and right plots are for a cock image and a sorrel image, respectively.

Fig.7 and 8 show the visualization of feature-flow for the decision of VGG16 and ResNet101, respectively. In the left plot of Fig. 7, for a cock image, we plot its feature-flow to interpret the target prediction (cock) of VGG16. Specifically, the feature-flow starts from the image in the input space, and then flows into the internal layers, and finally converges into the target-class unit in the output layer. It can be observed that, for a specific prediction, its feature-flow is sparse in the different layers, which means only a few units in each layer are distilled by the feature-flow. This phenomenon exists in predictions of different inputs using different networks.

Therefore, compared with the cumbersome network, feature-flow provides an interpretable sparse representation, which distills the key features to interpret the network decision.

2) ***The Feature-flow Interpretation for Intra-Class Net Decisions****:* In this section, we further explore the convergence and divergence of feature-flows within the same class in pool5 and pool1 layers of VGG16.

Interpretation of Intra-Class Convergence of Feature-flows: Images in the same class share some common features in the shallowest and the deepest layers of their feature-flows. For example, for the deepest layer, feature-flow distills leg-feature unit 423 and unit 69 to interpret the network prediction of sorrel images in Fig. 9. Similarly, unit 418 and unit 238 in the deepest layer are obtained for the interpretation of cock instances. For the shallowest layer, similar color information for sorrel and cock is extracted by unit 45 in feature-flows. The result indicates that feature-flows pass the same neural unit to extract the common feature for different images in the same class.

The Interpretation of Intra-Class Divergence of Feature-flows: For images in the same class, key features exist in some of them, but not all. For instance, from the input space, cock images all contain side-view neck part except for the fourth image in Fig. 10.

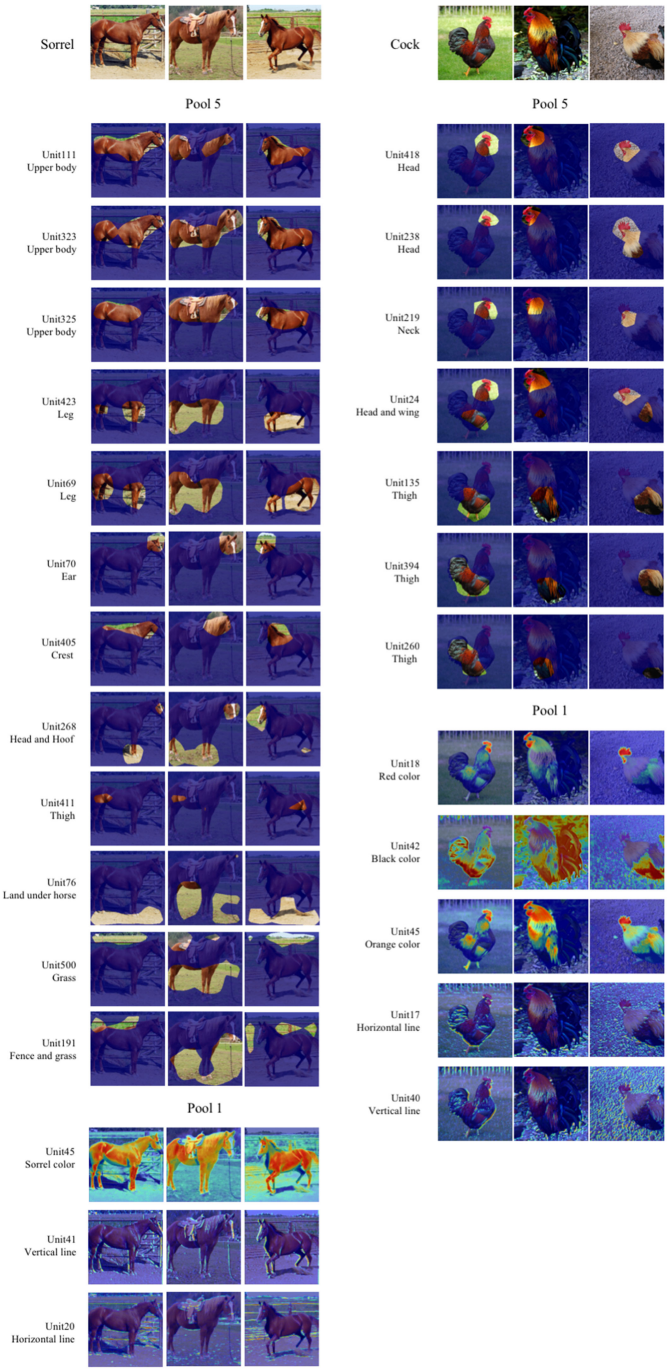


Fig. 9. The interpretation of intra-class convergence of feature-flows: representative common features in pool5 and pool1 layers of feature-flows for images within the same class based on VGG16. (Left: Sorrel, Right: Cock)

For a given input, its feature-flow distills key features to explain the network prediction. In Fig. 10, we show the same unit in the pool5 layer of VGG16 for different cock images. For the left three cock images, unit 219 is the common unit in their feature-flows. While unit 423 is the shared one in feature-flows for all cock images except the third one.

In Fig. 10, feature-flows of the left three cock images all contain unit 219 to extract the side-view neck information. Since the fourth image does not contain the side-view neck part, it is reasonable that its feature-flow does not include unit

Fig. 10. The interpretation of intra-class divergence of feature-flows: the same unit in the pool5 layer of VGG16 for different cock images. Features framed by the orange squares do not belong to the feature-flows of their network decisions.

219. Here, we also show unit 219 for the fourth image and observe that it is not related to the neck feature. It also explains why unit 219 cannot be regarded as one key feature of the feature-flow to explain the network prediction for the fourth image.

C. General Interpretation for the Net Feature Learning

In this section, experiments are conducted to quantitatively analyze the general interpretation for the net feature learning. In section IV-C1 and section IV-C2, we study the feature-flow similarity within and between classes to interpret the feature-flow from the class-level view. In section IV-C3, we provide the feature-flow interpretation based on the correlation matrix. In section IV-C4, the density of feature-flow in different layers is evaluated to interpret networks.

*1) **The Interpretation of Intra-Class Feature-flow Similarity in Different Layers**:* In this experiment, the similarity metric is utilized to quantitatively depict the feature-flow interpretation for the net feature learning. We compare the feature-flow similarity from shallow to deep layers to understand the hierarchical network structure. Meanwhile, it can also examine whether the feature-flow similarity in different layers of networks is consistent with the visual similarity in different semantic levels of human logic.

Fig. 11 and 12 show the feature-flow similarity of images within the same class using VGG16 and ResNet101. The left and right plots are for cock and sorrel, separately. In Fig. 11, the feature-flow similarity in pool1 and pool5 layers of VGG16 are relatively higher than that in middle layers. This observation is also applicable to the feature-flow similarity in the shallowest and the deepest layers of ResNet101, as shown in Fig. 12.

The reason behind that is the feature-flow in the shallowest layer distills general features that share among different images. Meanwhile, even for images in the same class, they contain different information. Therefore, for middle layers, the feature-flows of different images from the same class do not have the similarity as high as that for the shallowest layer. The feature-flow in the deepest layer extracts the class-discriminative features, which enhances similarity in the deepest layer for images within the same class.

*2) **The Interpretation of Inter-Class Feature-flow Similarity in Different Layers**:* As shown in Fig. 13, the feature-

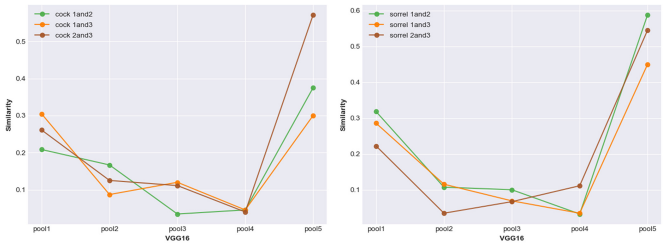


Fig. 11. The feature-flow similarity of images within the same class based on VGG16. The left and right plots are for cock and sorrel, separately.

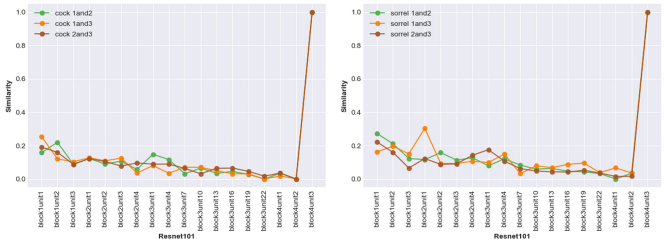


Fig. 12. The feature-flow similarity of images within the same class based on ResNet101. The left and right plots are for cock and sorrel, separately.

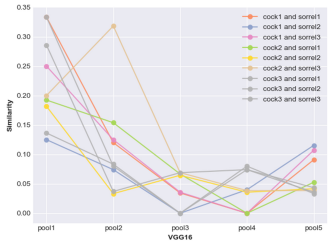


Fig. 13. The feature-flow similarity of images between classes based on VGG16.

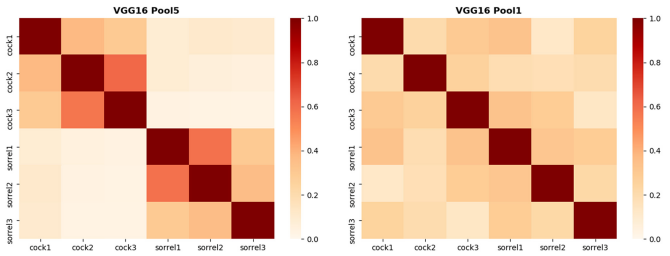


Fig. 14. The correlation matrices of feature-flow within and between classes. (Left: pool5 layer in VGG16, Right: pool1 layer in VGG16)

flow similarity of images between classes (cock and sorrel) in the pool1 layer of VGG16 is larger than that in other layers. The reason is that the general features, such as color, in the shallowest layer of feature-flow have much in common between images from different classes.

*3) **The Interpretation of the Correlation Matrix of Feature-flows**:* Fig. 14 displays the correlation matrices of feature-flows within and between classes for the shallowest and the deepest layers in VGG16. The above observation shows that the feature-flow similarity is comparatively higher in the shallowest and deepest layers. Here, we utilize the correlation matrix to understand this phenomenon further.

The Interpretation based on the Correlation Matrix of Feature-flows in the Deep Layer: In the left chart of Fig. 14, the feature-flow similarity within the same class is larger compared with that between different classes for the deepest layer. It indicates that the feature-flow similarity for the deep layer is class-discriminative.

The Interpretation based on the Correlation Matrix of Feature-flows in the Shallow Layer: For the shallowest layer, the feature-flow similarity within and between classes is nearly similar owing to the general property of features in the shallowest layer, as shown in the right chart in Fig. 14. These results are also consistent with the observation in Fig. 11, 12 and 13.

*4) **The Interpretation of the Feature-Flow Density in Different Layers**:* The density of feature-flow is a measurement of its sparsity. A sparse feature-flow has a small value of its density. We can analyze the feature-flow from shallow to deep layers by utilizing this metric. Moreover, by comparing the feature-flow density for different networks, we can examine the influence of network architectures on the sparsity of feature-flow.

The Density of Feature-flow in VGG16: Fig. 15 shows the density of feature-flow in different layers for VGG16. The left two plots are for cock and sorrel classes, respectively. The rightmost plot shows the average density of images from cock and sorrel classes.

For a given network decision being interpreted, the density of feature-flow witnesses a decrease with the deepening of layers, as in Fig. 15. Intuitively it is because of the hierarchical learning structure of networks. Features in deep layers are higher-level and more abstract than those in shallow layers.

The Density of Feature-flow in ResNet101: Fig. 16 shows the density of feature-flow in different layers for ResNet101. Specifically, the three plots are for cock, sorrel, and the average density, respectively. For ResNet101, its network structure is divided into 4 blocks. In each block, different layers have the same number of channels (i.e., the number of channels from the 1st to the 4th blocks are 256, 512, 1024, and 2048).

Fig. 16 reveals that the density of feature-flow declines with the deepening of blocks. Within the same block, the density of feature-flow fluctuates slightly. It suggests that different layers in the same block have similar sparsity of feature-flow. This can also help understand the network structure.

V. Conclusion

In this paper, the FLOWIN model, an interpretation model for DCNNs, is proposed to distill the feature-flow based on a sparse representation of network features. The FLOWIN can interpret the decisions of a DCNN by showing its feature-flow from deep to shallow layers, and it is understandable owing to its flow's similarity to the human understanding way. Through experiment evaluations, we have demonstrated that the proposed FLOWIN model can provide a reasonable interpretation for net decisions.

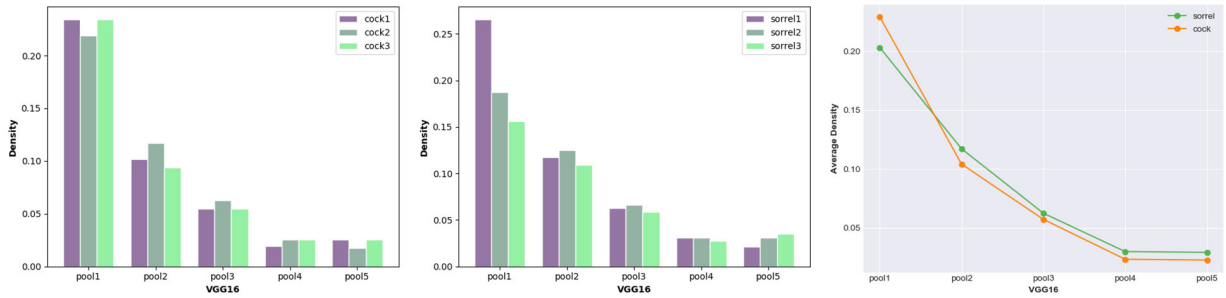


Fig. 15. The density of feature-flow in different layers for VGG16. (Left: cock, middle: sorrel, right: average density)

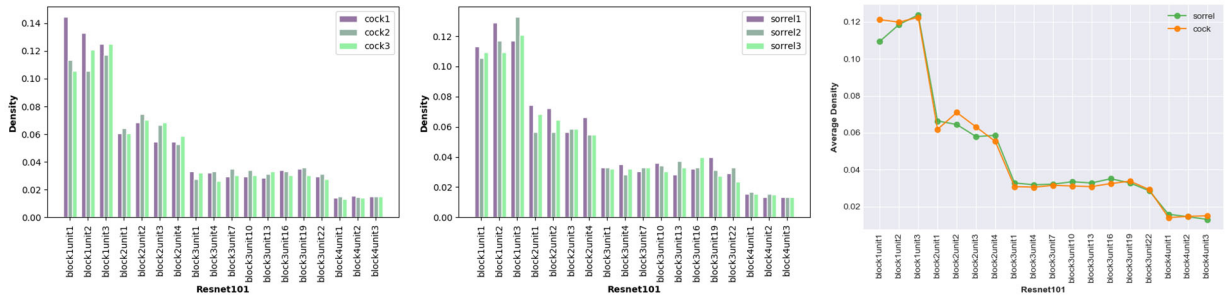


Fig. 16. The density of feature-flow in different layers for ResNet101. (Left: cock, middle: sorrel, right: average density)

REFERENCES

[1] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 6077–6086.

[4] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3449–3457.

[5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 618–626.

[6] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 119–134.

[7] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, July 2018, pp. 2668–2677.

[8] D. Wang, X. Cui, and Z. J. Wang, "Chain: Concept-harmonized hierarchical inference interpretation of deep convolutional neural networks," *arXiv preprint arXiv:2002.01660*, 2020.

[9] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari, "Do semantic parts emerge in convolutional neural networks?" *International Journal of Computer Vision*, vol. 126, no. 5, pp. 476–494, 2018.

[10] Q. Zhang, Y. Nian Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8827–8836.

[11] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding intra-class knowledge inside cnn," *arXiv preprint arXiv:1507.02379*, 2015.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[13] M. T. Ribeiro, S. Singh, and C. 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*. ACM, 2016, pp. 1135–1144.

[14] X. Cui, D. Wang, and Z. J. Wang, "Multi-scale interpretation model for convolutional neural networks: Building trust based on hierarchical interpretation," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2263–2276, Sep. 2019.

[15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2921–2929.

[16] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Workshop in International Conference on Learning Representations*, 2015.

[17] X. Cui, D. Wang, and Z. J. Wang, "Chip: Channel-wise disentangled interpretation of deep convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2019.

[18] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1971–1978.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

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

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[23] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting cnns via decision trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 6261–6270.