Interpretable Compositional Convolutional Neural Networks

Wen Shen[1,*] , **Zhihua Wei**[1,*] , **Shikun Huang**[1] , **Binbin Zhang**[1] , **Jiaqi Fan**[1] , **Ping Zhao**[1] , **Quanshi Zhang**[2,†]

[1]Tongji University, Shanghai, China
[2]Shanghai Jiao Tong University, Shanghai, China
{wen_shen,zhihua_wei,hsk,0206zbb,1930795,zhaoping}@tongji.edu.cn,zqs1022@sjtu.edu.cn

Abstract

The reasonable definition of semantic interpretability presents the core challenge in explainable AI. This paper proposes a method to modify a traditional convolutional neural network (CNN) into an interpretable compositional CNN, in order to learn filters that encode meaningful visual patterns in intermediate convolutional layers. In a compositional CNN, each filter is supposed to consistently represent a specific compositional object part or image region with a clear meaning. The compositional CNN learns from image labels for classification without any annotations of parts or regions for supervision. Our method can be broadly applied to different types of CNNs. Experiments have demonstrated the effectiveness of our method. *The code will be released when the paper is accepted.*

1 Introduction

Convolutional neural networks (CNNs) have exhibited superior performance in many visual tasks. Besides, the interpretability of CNNs has received increasing attention in recent years. Studies of network interpretability usually focus on the visualization of network features or the extraction of pixel-level correlations between network inputs and outputs. Training a CNN with interpretable features in intermediate layers is still a challenge to state-of-the-art algorithms, which helps people obtain more trustworthy and verifiable features.

In this paper, we aim to propose a method to modify a CNN, which makes filters in an intermediate layer encode interpretable and compositional features. Specifically, as Fig. 1 shows, each filter in the intermediate layer is supposed to be consistently activated by the same object part with specific shapes (*e.g.* the head part of a bird) or the same image region without specific structures (*e.g.* the sky in the background) through different images. Besides, different filters in the layer are supposed to be activated by different parts or regions, which ensures the diversity of visual patterns.

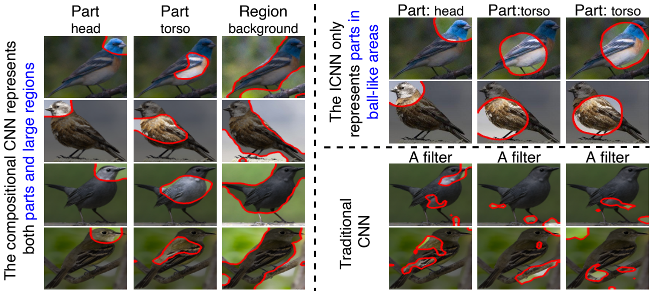

Part
head
Part
torso
Region
background
The compositional CNN represents both parts and large regions
The ICNN only represents parts in ball-like areas
Part: head
Part:torso
Part: torso
A filter
A filter
A filter
Traditional CNN

Figure 1: Compared with ICNN [Zhang *et al.*, 2018], the interpretable compositional CNN defines the filter interpretability in a more generic manner, thereby modeling more diverse visual patterns. In a compositional CNN, each filter consistently represents a specific object part or image region through different images. Different filters represent different object parts and image regions. In comparison, the ICNN can only represent object parts in ball-like areas.

Given different images, we learn the interpretable compositional CNN in an end-to-end manner **without** any annotations of object parts or image regions for supervision. To this end, we add a specific loss to the intermediate layer in a CNN. This loss encourages each filter to be consistently activated by the same object part or the same image region. The loss also pushes different filters to be activated by different object parts or image regions.

We notice that a CNN usually uses a set of filters to jointly represent a specific object part or image region, instead of using a single filter, which has been discussed in [Fong and Vedaldi, 2018]. Therefore, we divide filters in a convolutional layer into different groups. The loss is designed to force filters in the same group to be activated by the same object part or the same image region, and force filters in different groups to be activated by different parts or regions. Note that each filter in the group is required to represent almost the entire part/region, instead of a random sub-part/sub-region fragment inside, which ensures the clarity of the meaning of each filter. The mutual verification of visual patterns between filters in the same group ensures the stability of the visual patterns represented by each filter in this group. The slight difference of feature maps between filters in the same group encodes the fine-grained variety of the same type of parts/regions. To this

[*] Wen Shen and Zhihua Wei have equal contributions.

[†] Quanshi Zhang is the corresponding author. He is with the John Hopcroft Center and the MoE Key Lab of Artificial Intelligence, AI Institute, at the Shanghai Jiao Tong University, China.

end, we design a metric to measure the similarity between filters, which enables the loss to group filters. Besides, for multi-category classification, we design a loss to force different groups of filters to be activated by object parts or image regions of different categories.

In this study, we evaluate the interpretability of filters in the convolutional layer qualitatively and quantitatively. We visualize the feature map of a filter to qualitatively show the consistency of a filter's visual patterns through different images, in order to examine the fitness between the visual patterns automatically learned by a compositional CNN and the visual concepts in human's cognition. For the quantitative evaluation, previous metrics in [Zhang *et al.*, 2018] can only evaluate semantic consistency of object parts in ball-like areas and strong priors of object structures. Therefore, we design two metrics to evaluate both the consistency of a filter's visual patterns and the diversity of visual patterns represented by different filters, respectively.

Previous studies also developed CNNs, where filters in an intermediate layer represented meaningful features. Capsule nets [Sabour *et al.*, 2017] encoded different meaningful features, but these features usually did not represent parts or regions. Zhang *et al.* [2018] proposed interpretable CNNs (ICNNs), which learned filters in intermediate layers to represent object parts. They designed the information-theoretic interpretability loss to force filters to represent specific object parts. Filters in the ICNN could only represent object parts in ball-like areas. In comparison, we extend the filter interpretability to both object parts with specific shapes and image regions without clear structures, which proposes significant challenges to state-of-the-art algorithms. Thus, the compositional CNN can encode more types of features than the ICNN. Please see Fig. 1 for details.

Contributions of this study can be summarized as follows. We propose a method to modify traditional CNNs into compositional CNNs without any annotations of object parts or image regions for supervision. Each filter in a compositional CNN consistently represents the same object part or image region with a clear meaning. Experiments show that our method can be broadly applied to CNNs with different architectures.

2 Related Work

Learning interpretable features. Some studies directly trained networks to increase the interpretability of intermediate-layer features. Capsule nets [Sabour *et al.*, 2017] learned capsules to encode meaningful features via a dynamic routing mechanism. InfoGAN [Chen *et al.*, 2016] and β-VAE [Higgins *et al.*, 2017] learned interpretable representations for generative networks. These studies did not make each filter in the CNN encode a specific visual pattern. To this end, some studies [Li *et al.*, 2020; Liang *et al.*, 2020] learned class-specific filters, *i.e.* each filter only represented a specific category. However, such class-specific filters could not represent fine-grained meaningful visual patterns, such as object parts and image regions. Chen *et al.* [2019] proposed the ProtoPNet to extract similar object-part regions that were shared for fine-grained classification. The ProtoPNet did not ensure each filter to represent a clear meaning. Zhang *et al.* [2018] proposed interpretable CNNs to make each filter in a high convolutional layer represent a specific object part. In comparison, we extend the filter interpretability to both object parts and image regions, which presents significant challenges to state-of-the-art algorithms.

Compositional models. Previous studies in compositional models focused on learning hierarchical feature representations [Fidler and Leonardis, 2007; Zhu *et al.*, 2010a; Ommer and Buhmann, 2009], such as graph-based models [Si and Zhu, 2013; Wang and Yuille, 2015] and part-based models [Ott and Everingham, 2011; Zhu *et al.*, 2010b]. These models did not use neural networks to learn features. Other studies learned discriminative compositional parts directly through network training. Stone *et al.* [2017] manually designed a graphical model to organize CNN modules and represent object structures. Kortylewski *et al.* [2020] designed a specific compositional layer to enable the network to localize partial occlusion. Huang and Li [2020] learned discriminative object parts for fine-grained recognition based on manually labeled part priors. However, in all above studies, the compositional information of features was not automatically learned from data. In comparison, the proposed compositional CNN automatically learns compositional features without any annotations of parts or regions. *I.e.* the compositional CNN automatically regularizes its features into meaningful parts and regions without letting people supervise its semantic representations.

3 Algorithm

In this section, we aim to modify a convolutional layer of a CNN into an interpretable compositional layer. In this layer, each filter is supposed to consistently represent the same object part or the same image region through different images. To ensure the consistency of the visual patterns represented by each filter, we use a group of filters to represent the same object part or the same image region. The set of filters $\Omega = \{1, 2, \cdots, d\}$ in the target layer are divided into different groups $A_1, A_2, \cdots, A_K$, where $A_1 \cup A_2 \cup \cdots \cup A_K = \Omega$; $A_i \cap A_j = \emptyset$. $\mathbf{A} = \{A_1, A_2, \cdots, A_K\}$ denotes the partition of filters. Let θ denote parameters of the CNN. Given a set of training images, we aim to simultaneously optimize parameters θ and the partition $\mathbf{A}$ to ensure that filters in the same group consistently represent the same visual patterns through different images, and filters in different groups represent different visual patterns.

To measure the similarity of visual patterns represented by different filters, we propose a metric to measure the similarity between filters. Given an image I, let $x_i^I \in \mathbb{R}^m$ denote the feature map of the i-th filter in the target convolutional layer after the ReLU operation. Given a set of n training images $\mathbf{I}$, let $X_i = \{x_i^I\}_{I \in \mathbf{I}}$ denote the set of feature maps of the i-th filter. Then, we compute the similarity between the i-th and the j-th filters, which represents whether these two filters consistently represent the same visual patterns through different images. This similarity is formulated as $s_{ij} = \mathcal{K}(X_i, X_j) \in \mathbb{R}$, where $\mathcal{K}$ is a kernel function. Based on the similarity metric, we design the following loss to learn filters.

$$\text{Loss}(\theta, \mathbf{A}) = -\sum_{k=1}^{K} \frac{S_k^{\text{within}}}{S_k^{\text{all}}} = -\sum_{k=1}^{K} \frac{\sum_{i,j \in A_k} s_{ij}}{\sum_{i \in A_k, j \in \Omega} s_{ij}}, \quad (1)$$

where $S_k^{\text{within}} = \sum_{i,j \in A_k} s_{ij} = \sum_{i,j \in A_k} \mathcal{K}(X_i, X_j)$ measures the similarity between filters *within* the same group A_k; $S_k^{\text{all}} = \sum_{i \in A_k, j \in \Omega} s_{ij} = \sum_{i \in A_k, j \in \Omega} \mathcal{K}(X_i, X_j)$ measures the similarity between filters in A_k and *all* filters in Ω. This loss increases S_k^{within} to ensure that filters in the same group have high similartiy, and decreases S_k^{all} to ensure that filters in different groups have low similartiy. Specifically, the similarity metric is implemented as a kernel function.

$$s_{ij} = \mathcal{K}(X_i, X_j) = \rho_{ij} + 1 = \frac{\text{cov}(X_i, X_j)}{\sigma_i \sigma_j} + 1 \geq 0, \quad (2)$$

where $\rho_{ij} \in [-1, 1]$ denotes the Pearson's correlation coefficient between variables x_i^I and x_j^I through different images; Constant 1 is added to ensure the non-negativity of the similarity. $\text{cov}(X_i, X_j) \in \mathbb{R}$ denotes the covariance between variables x_i^I and x_j^I through different images, $\text{cov}(X_i, X_j) = \frac{1}{n-1} \sum_{I \in \mathbf{I}} (x_i^I - \mu_i)^\top (x_j^I - \mu_j) \in \mathbb{R}$; $\mu_i = \frac{1}{n} \sum_{I \in \mathbf{I}} x_i^I \in \mathbb{R}^m$; $\sigma_i^2 = \frac{1}{n-1} \sum_{I \in \mathbf{I}} (x_i^I - \mu_i)^2 \in \mathbb{R}$. The similarity metric can be understood as the sum of similarity between feature maps of the i-th and the j-th filters through all training images as follows, $s_{ij} = \mathcal{K}(X_i, X_j) = \sum_{I \in \mathbf{I}} \phi(x_i^I)^\top \phi(x_j^I)$, where $\phi(x_i^I)^\top = [(x_i^I - \mu_i)^\top, \sqrt{1 - 1/n\sigma_i^\top}]/\sqrt{n-1}\sigma_i$.

The proposed loss makes filters in the same group have similar feature maps, which ensures the clarity and the stability of the visual patterns represented by each filter in this group. Meanwhile, the slight difference of these feature maps encodes fine-grained variety of the same type of parts/regions. Besides, the loss also makes filters in different groups have different feature maps, which ensures the diversity of the visual patterns represented by different groups of filters.

Binary classification of a single category. We train a compositional CNN in an end-to-end manner by minimizing the following objective function.

$$\mathbf{L}(\theta, \mathbf{A}) = \lambda \text{Loss}(\theta, \mathbf{A}) + \frac{1}{n} \sum\nolimits_{I \in \mathbf{I}} \text{L}^{\text{cls}}(\hat{y}_I, y_I^*; \theta), \quad (3)$$

where $\text{L}^{\text{cls}}(\hat{y}_I, y_I^*; \theta)$ denotes the classification loss on image I; $\hat{y}_I, y_I^* \in \{-1, +1\}$ denote the output of the CNN and the ground-truth label, respectively; λ is a positive weight.

Multi-category classification. For the multi-category classification, besides $\text{Loss}(\theta, \mathbf{A})$, we design another loss to make different groups of filters to be activated by parts or regions of different categories. Given a set of n training images $\mathbf{I}$, let $\mathbf{I}_c \subset \mathbf{I}$ represent the subset of images of the category c, $(c = 1, 2, \cdots, C)$. Filters in a certain group are supposed to be mainly activated by a specific object part or image region of very few categories, and keep silent on other categories. To this end, for each filter, we quantify the distribution of its neural activations over different categories. We propose a metric to measure the similarity between such distributions of different filters. Given the p-th image I, let $z_k^{(p)} \in \mathbb{R}$ denote the average activation score of filters in group A_k, $z_k^{(p)} = \frac{1}{|A_k| \cdot m} \sum_{i \in A_k} \sum_{u=1}^{m} x_{i,u}^I$, where $|A_k|$ denotes the number of filters in group A_k; $x_{i,u}^I$ denotes the u-th element in $x_i^I \in \mathbb{R}^m$. The similarity between distributions of neural activations of different groups on the p-th and the q-th images is computed using the following kernel function. $s_{pq} = \mathcal{K}(\mathbf{z}^{(p)}, \mathbf{z}^{(q)}) = (\mathbf{z}^{(p)})^\top (\mathbf{z}^{(q)}) \in \mathbb{R}$, where $\mathbf{z}^{(p)} = [z_1^{(p)}, \cdots, z_K^{(p)}]^\top \in \mathbb{R}^K$; $x_{i,u}^I \geq 0$, thereby $s_{pq} \geq 0$. We propose the following loss to learn filters.

$$\text{L}^{\text{multi}}(\theta) = -\sum_{c=1}^{C} \frac{\sum\limits_{p,q \in \mathbf{I}_c} s_{pq}}{\sum\limits_{p \in \mathbf{I}_c, q \in \mathbf{I}} s_{pq}} = -\sum_{c=1}^{C} \frac{\sum\limits_{p,q \in \mathbf{I}_c} \mathcal{K}(\mathbf{z}^{(p)}, \mathbf{z}^{(q)})}{\sum\limits_{p \in \mathbf{I}_c, q \in \mathbf{I}} \mathcal{K}(\mathbf{z}^{(p)}, \mathbf{z}^{(q)})}. \quad (4)$$

The final objective function for multi-category classification is given as follows.

$$\mathbf{L}(\theta, \mathbf{A}) = \lambda \text{Loss}(\theta, \mathbf{A}) + \beta \text{L}^{\text{multi}}(\theta) + \frac{1}{n} \sum_{I \in \mathbf{I}} \text{L}^{\text{cls}}(\hat{y}_I, y_I^*; \theta), \quad (5)$$

where λ and β are positive weights.

Learning. During the learning process, we need to simultaneously optimize network parameters θ and the filter partition $\mathbf{A}$. Fortunately, we find that, when we fix θ, the minimization of $\text{Loss}(\theta, \mathbf{A})$ *w.r.t.* $\mathbf{A}$ is essentially equivalent to the problem of the spectral clustering in [Shi and Malik, 2000]. *I.e.* we can rewrite $\text{Loss}(\theta, \mathbf{A})$ as the following equation, which is exactly the same objective function in [Shi and Malik, 2000].

$$\frac{1}{2}(\text{Loss}(\theta, \mathbf{A}) + K) = \frac{1}{2} \sum_{k=1}^{K} \frac{\sum_{i \in A_k, j \notin A_k} s_{ij}}{\sum_{i \in A_k, j \in \Omega} s_{ij}}. \quad (6)$$

Here, we regard the set of filters Ω as data points in the spectral clustering that need to be clustered into different groups $A_1, \cdots, A_K$. s_{ij} corresponds to the similarity between two data points. In this way, $\mathbf{A}$ can be optimized by applying the clustering technique in [Shi and Malik, 2000]. Therefore, we alternately optimize θ and $\mathbf{A}$ to minimize $\text{Loss}(\theta, \mathbf{A})$.

4 Experiments

We applied our method to CNNs with six types of architectures to demonstrate the broad applicability of our method. We used object images in four different benchmark datasets to learn compositional CNNs for both the binary classification of a single category and the multi-category classification. We designed two metrics to measure the inconsistency of a filter's visual patterns and the diversity of visual patterns represented by different filters. We also visualized feature maps of a filter to qualitatively show the consistency of a filter's visual patterns. We compared the performance of learning interpretable filters in different convolutional layers of a compositional CNN. We also discussed the effects of the group number on the performance of learning interpretable filters. For binary classification of a single category, we set $\lambda = 1.0$ for most DNNs except for VGG-16 with $\lambda = 0.1$. It was because the learning of a residual network could be considered as the optimization of massive parallel shallow networks. From this perspective, the VGG-16 was the most difficult to optimize. For multi-category classification, we set $\lambda = 0.1$ and $\beta = 0.1$, because L^{multi} has partially taken the work of $\text{Loss}(\theta, \mathbf{A})$. During the training procedure, for each time we optimized θ through all training samples, we optimized $\mathbf{A}$ once.

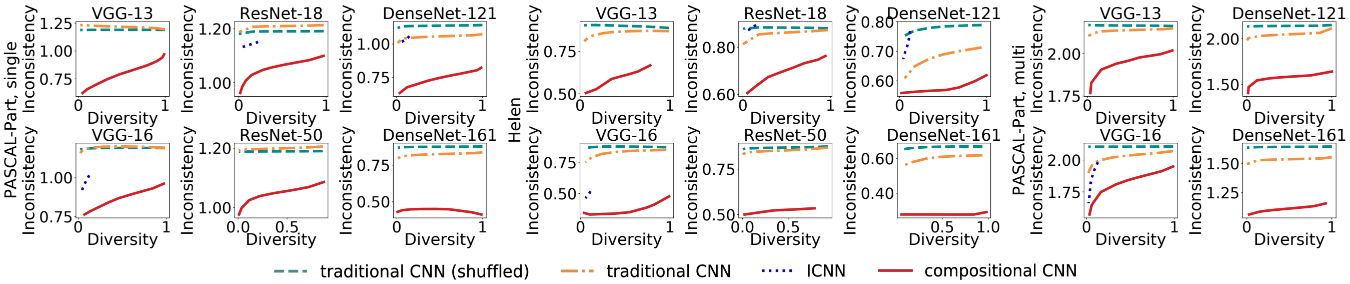


Figure 2: Comparisons of inconsistency of visual patterns and diversity of visual patterns between CNNs. For the binary classification of a single category, we showed curves of the average inconsistency of visual patterns and the average diversity of visual patterns over each CNN learned for each individual category. Results for each single category are shown in Fig. 7. Note that each value of inconsistency in this figure indicates the average inconsistency of all filters of a DNN.

4.1 Learning Compositional CNNs

Binary Classification of A Single Category

We learned six types of compositional CNNs based on the VGG-13[1], VGG-16[1] [Simonyan and Zisserman, 2015], ResNet-18, ResNet-50 [He *et al.*, 2016], DenseNet-121, and DenseNet-161 [Huang *et al.*, 2017] architectures. Just like in [Zhang *et al.*, 2018], we added the loss $\mathrm{Loss}(\theta, \mathbf{A})$ to the high convolutional layer in a CNN. It was because the previous study [Bau *et al.*, 2017] had revealed that filters in high convolutional layers were more likely to represent object parts or image regions, instead of detailed patterns (*e.g.* colors or textures). For the VGG-13, VGG-16, DenseNet-121, and DenseNet-161, we added the proposed loss to the top convolutional layer. For the ResNet-18, we added the loss to layer *conv4_4*. For the ResNet-50, we added the loss to layer *conv4_18*. All these compositional CNNs were learned based on the CUB200-2011 dataset [Wah *et al.*, 2011], the Large-scale CelebFaces Attributes (CelebA) dataset [Liu *et al.*, 2015], the Helen Facial Feature dataset [Smith *et al.*, 2013], and animal categories in the PASCAL-Part dataset [Chen *et al.*, 2014]. In the field of learning interpretable deep features, animal categories were widely used to evaluate the automatically learned interpretable features [Zhang *et al.*, 2018]. It was because animals usually contained deformable parts, which presented great challenges for part or region localization. Note that, the Helen Facial Feature dataset was usually used for the facial landmark localization. However, in this study, we used this dataset for the classification of faces and non-faces. It was because this dataset provided segmentation masks for face parts to evaluate the inconsistency and the diversity of visual patterns. We randomly selected the same number of samples from the PASCAL-Part dataset as negative samples for training and testing.

We followed experimental settings in [Zhang *et al.*, 2018] to learn compositional CNNs for binary classification of a single category on the CUB200-2011 dataset and the PASCAL-Part dataset. For compositional CNNs learned from the CUB200-2011 dataset, the PASCAL-Part dataset, and the Helen Facial Feature dataset, we set $K = 5$. For the CelebA dataset, we set $K = 16$, because these compositional CNNs usually learned detailed visual patterns from face images.

To compare the performance of learning interpretable filters in different convolutional layers, we learned two compositional CNNs based on the VGG-16 architecture by adding the proposed loss to layer *conv4_3* and layer *conv5_3*, respectively. These two compositional CNNs were learned on the PASCAL-Part dataset. To explore the effects of different values of K, we learned two compositional CNNs based on the ResNet-50 architecture using the CelebA dataset by setting $K = 8$ and $K = 16$, respectively.

Multi-Category Classification

We learned four types of compositional CNNs based on the VGG-13, VGG-16, DenseNet-121, and DenseNet-161 architectures for the classification on the PASCAL-Part dataset following experimental settings in [Zhang *et al.*, 2018]. We set $K = 16$.

For all compositional CNNs, we learned traditional CNNs based on the same architectures and datasets as baselines. We replaced the zero padding with the replication padding for all compositional CNNs. For traditional CNNs based on the DenseNet architectures, we initialized parameters of the fully-connected layers, and loaded parameters of other layers from the same architectures that were pre-trained using the ImageNet dataset [Deng *et al.*, 2009]. For traditional CNNs based on other architectures, we initialized parameters of the target layer (*i.e.* the convolutional layer would be modified to an interpretable compositional layer) and its following layers, and loaded parameters of other layers from the same architectures that were pre-trained using the ImageNet dataset. For all compositional CNNs, we loaded parameters of all layers from the above well-trained traditional CNNs.

4.2 Quantitative Evaluation of Filter Interpretability

Some previous studies also focused on learning interpretable filters, but their metrics usually have strong limitations and can not be used in our experiments. Metrics in [Zhang *et al.*, 2018] can only be used to evaluate semantic consistency of object parts in ball-like areas and strong priors of object structures. Bau *et al.* [2017] annotated six types of visual semantics for evaluation (including colors and materials), but filters in the compositional CNN were not designed towards

[1] The VGG-13 and VGG-16 used in this paper were slightly revised by adding the batch-normalization [Ioffe and Szegedy, 2015] operation after each convolution layer.

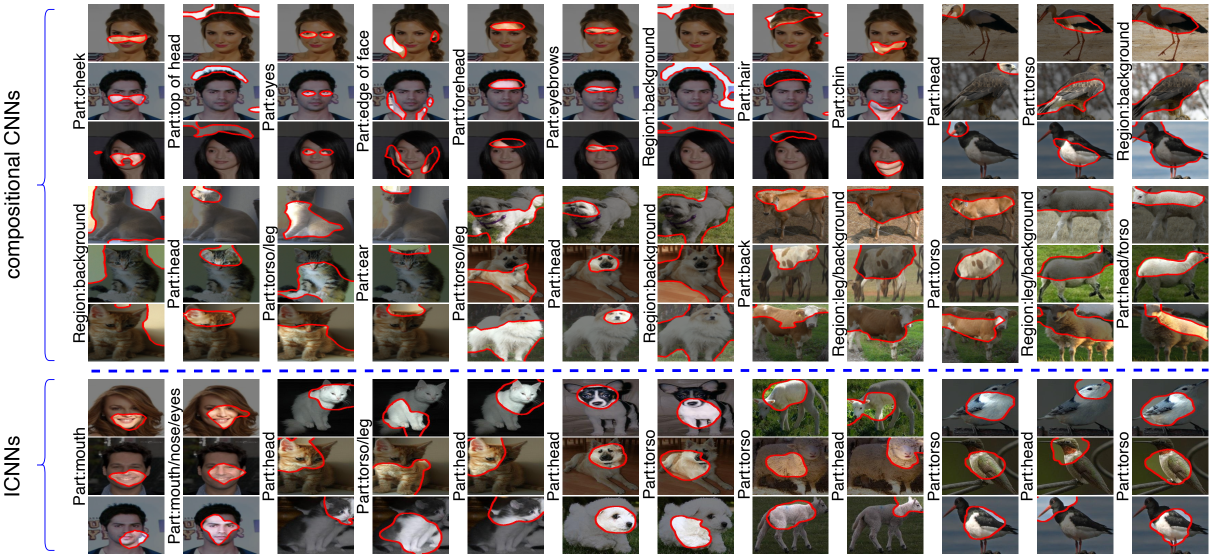


Figure 3: Visualization of feature maps of compositional CNNs and ICNNs [Zhang *et al.*, 2018]. Each column in the figure corresponds to a certain filter. Visualization results indicate that each filter in a compositional CNN consistently represented the same object part or the same image region, while different filters represented different parts and regions. In comparison, filters in an ICNN could only represent object parts. Note that we manually classified filters into part filters and region filters to help understand the visual patterns represented by the filter. In addition, part filters in the compositional CNN usually encoded more complex shapes than those in the ICNN.

such semantics. Therefore, we extended the metric in [Bau *et al.*, 2017] and proposed the inconsistency of visual patterns to evaluate the interpretability of filters. Besides, we evaluated the diversity of visual patterns represented by filters, which was an significant factor neglected in previous studies.

Evaluation Metric 1: Inconsistency of Visual Patterns

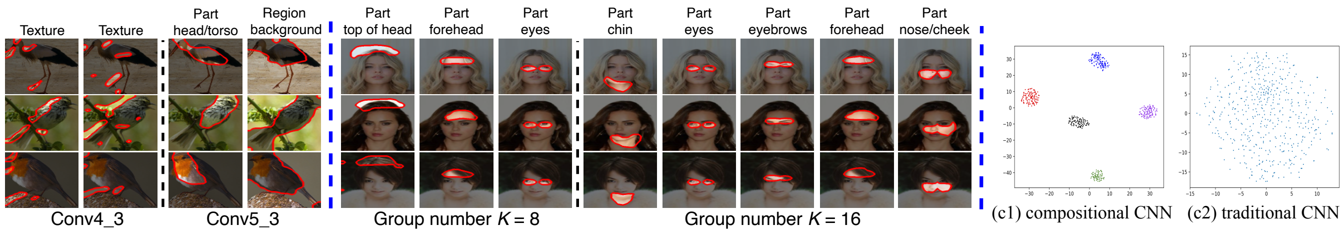
This metric was proposed to measure the consistency of visual patterns represented by a filter through different images. Ideally, an interpretable filter was supposed to have high consistency. We computed the probability of a filter being associated with a ground-truth semantic concept in a specific image (*e.g.* bird head, bird torso). Then, we defined the inconsistency of visual patterns as the entropy of such probabilities over different semantic concepts.

For simplicity, here, we only discussed the metric for a single filter below. We first computed the pixel-wise receptive field (RF) of neural activations of the filter on testing images [Zhang *et al.*, 2018]. Let $Q(I) \in \mathbb{R}^M$ denote activation scores of the target filter projected onto the test image I, where M denoted the number of pixels in the image I. We only considered activation scores in the feature map greater than a threshold τ as valid ones to represent the filter (the setting of τ would be explained later). Then, $\tilde{Q}(I) \in \{0,1\}^M$ s.t. $\tilde{Q}_u(I) = \mathbb{1}(Q_u(I) \geq \tau)$ denoted the RF. Let $G^j(I) \in \{0,1\}^M$ denote the ground-truth segmentation mask of the j-th concept ($j = 1, \cdots, T$) on the test image I. The probability of the target filter being associated with the j-th concept was computed as $P_j = \frac{\sum_{I \in \mathbf{I}^{\text{test}}} \sum_{u=1}^{M} \min\{\tilde{Q}_u(I), G_u^j(I)\}}{\sum_{I \in \mathbf{I}^{\text{test}}} \sum_{u=1}^{M} \tilde{Q}_u(I)}$, where $\mathbf{I}^{\text{test}}$ denoted the set of testing images. Then, the inconsistency of the target filter's visual patterns was defined as the entropy $H = -\sum_{j=1}^{T} P_j \log P_j$.

Binary classification of a single category. We followed [Zhang *et al.*, 2018] to merge certain areas of each animal category in the PASCAL-Part dataset to obtain stable landmark locations as stable concepts for evaluation. We used five concepts for the *bird* category, including (head, l/r-eyes, beak, neck), (torso, l/r-wings), (l/r-legs/feet), (tail), and (background). Here, all parenthesized areas were merged as a new concept. We used four concepts for the *cat* category, including (head, l/r-eyes, l/r-ears, nose, neck), (torso, tail), (lf/rf/lb/rb-legs, lf/rf/lb/rb-paws), and (background). We used four concepts for the *dog* category, areas of which were merged in the same way as the cat category, except for merging the additional muzzle area to the head concept. We used four concepts for the *cow* category, which were defined in a similar way as the dog category. We added l/r-horn to the head concept. We used four concepts for *sheep* and *horse* categories, which were defined in the same way as the cow category. Note that, in actual calculations, we only used images with relatively complete areas of each animal category[2]. In the Helen Facial Feature dataset[2], we used three concepts for the face category. We merged areas of face skin, l/r-eyebrow, l/r-eye, nose, u/l-lip, and inner mouth as the face concept. We used the areas of hair and background as the 2-nd and the 3-rd concepts, respectively.

Multi-category classification. In the PASCAL-Part dataset, for each category, we used the foreground object as a single concept, and used the background as another concept. We considered visual concepts of all categories equally, *i.e.* we would get $T = 2C$ concepts for the classification of C categories. Then, we used the aforementioned entropy

[2] The dataset for the computation of metrics in this paper will be released in https://github.com/ada-shen/icCNN.



(c1) compositional CNN (c2) traditional CNN

(a) Filters in different convolutional layers (b) Filters learned with different values of K (c) t-SNE visualization of feature maps

Figure 4: (a) Comparisons of interpretable filters in different convolutional layers. Results indicate that filters in a high convolutional layer tended to represent parts or regions, while filters in a middle convolutional layer tended to represent textures. (b) Filters learned with different values of K. Filters in the compositional CNN with $K = 16$ represented more detailed visual patterns than the CNN learned with $K = 8$. (c) t-SNE visualization of feature maps of a compositional CNN (c1) and a traditional CNN (c2). Each point represents a feature map. Different colors of points represent feature maps of filters in different groups.

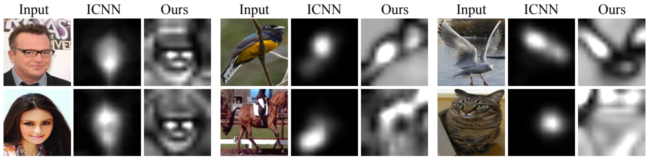


Figure 5: Visualizing distributions of visual patterns that are encoded in interpretable filters via the method in [Zhang *et al.*, 2018]. Results show that interpretable filters of a compositional CNN explained much more regions in an image than those of an ICNN.

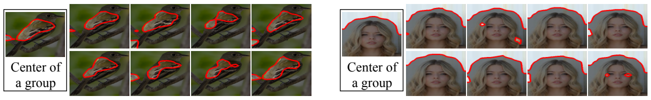


Figure 6: Comparisons of RFs between the center of the group and each filter in the group.

	single-category			multi-category
	PASCAL-Part	CUB200	CelebA	PASCAL-Part
VGG-13	**97.07**	**99.76**	–	**87.51**
compositional CNN	96.29	99.41	–	86.37
VGG-16	**98.66**	**99.86**	90.47	89.71
ICNN	95.39	96.51	89.11	**91.60**
compositional CNN	97.12	99.27	**90.70**	87.51
ResNet-18	**97.77**	**99.81**	89.60	–
ICNN	93.30	97.12	–	–
compositional CNN	96.90	98.49	**89.76**	–
ResNet-50	**97.88**	**99.88**	**90.21**	–
compositional CNN	97.30	99.27	89.63	–
DenseNet-121	**98.29**	**99.92**	–	91.28
ICNN	96.55	99.22	–	–
compositional CNN	97.52	98.83	–	**91.75**
DenseNet-161	**98.70**	**99.96**	–	**93.48**
compositional CNN	98.14	99.61	–	92.66

Table 1: Comparisons of classification accuracy between ICNNs and compositional CNNs revised from different classic CNNs.

H over the $2C$ concepts for evaluation. Note that, for the classification of a large number of categories, theoretically, each category only obtained very few filters, which decreased the filter interpretability. Therefore, we only learned compositional CNNs for multi-category classification based on all animal categories in the PASCAL-Part dataset.

Randomly shuffled feature maps as baselines. We constructed feature maps that totally had no consistency of visual patterns as a baseline. In implementation, we randomly shuffled different images' feature maps of a traditional CNN to approximately construct random feature maps.

Evaluation Metric 2: Diversity of Visual Patterns

This metric was proposed to evaluate whether a CNN learned various visual patterns. In this study, the diversity of visual patterns was approximately quantified as the number of pixels which had been explained by a CNN. We determined that a pixel was explained by a CNN, if this pixel was explained by some filters. Recall that, we had computed the pixel-wise RF of neural activations of a filter on the test image I based on [Zhang *et al.*, 2018]. Here, we used $\tilde{Q}^i(I) \in \{0,1\}^M$ to denote the RF of neural activations of the i-th filter. Then, we determined that the u-th pixel was explained by a CNN, if $(\frac{1}{d}\sum_{i=1}^{d}\tilde{Q}^i_u(I)) \geq \gamma$. We set $\gamma = 0.2$. The higher diversity meant that RFs of filters covered more pixels, *i.e.* more diverse concepts were encoded by the CNN. Therefore, the diversity of visual patterns was computed as $Diversity = \frac{1}{M}\mathbb{E}_I[\sum_{u=1}^{M}\mathbb{1}((\frac{1}{d}\sum_{i=1}^{d}\tilde{Q}^i_u(I)) \geq \gamma)]$.

Curves of Inconsistency and Diversity

Note that, the two metrics of inconsistency and diversity were closely related. Generally speaking, the greater the diversity was, the lower the consistency was. Therefore, in order to fairly compare different CNNs' inconsistency of visual patterns under different diversity, we reported *inconsistency-diversity* curves in this paper, as shown in Fig. 2. To this end, we sampled different values of τ to obtain different pairs of *inconsistency-diversity*, thereby obtaining *inconsistency-diversity* curves. Given n sampled values of τ, $[\tau_1, \tau_2, \cdots, \tau_n]$, we could calculate n pairs of *inconsistency-diversity* values, (p_1, q_1), (p_2, q_2), $\cdots$, (p_n, q_n). The sampling of τ was under the constraint that $q_1, q_2, \cdots, q_n$ were evenly distributed between $(0, 1]$.

4.3 Experimental Results and Analysis

***Inconsistency-diversity* curves and classification accuracy.** Fig. 2 shows the *inconsistency-diversity* curves of different CNNs. Each inconsistency value was the average inconsistency over all filters. Under the same diversity of visual patterns, compositional CNNs exhibited higher consistency of

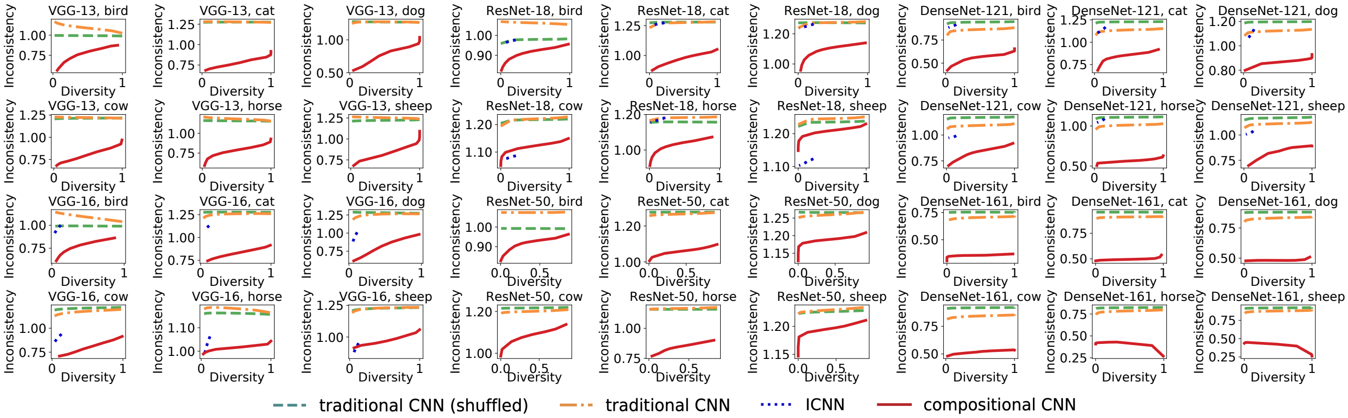


Figure 7: The *inconsistency-diversity* curves of CNNs based on different categories of the PASCAL-Part dataset.

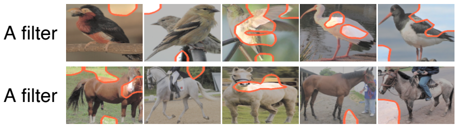


Figure 8: Very few cases when filters in compositional CNNs did not represent meaningful patterns.

	VGG-13	VGG-16	Res-18	Res-50	Dense-121	Dense-161
classic CNN	1.0	1.0	1.0	1.0	1.0	1.0
ICNN	–	99.70	1.0	–	1.0	–
compositional CNN	1.0	1.0	99.85	1.0	99.85	1.0

Table 2: Classification accuracy of CNNs based on the Helen Facial Feature dataset. Res indicates ResNet; Dense indicates DenseNet.

visual patterns than traditional CNNs and ICNNs. Besides, compositional CNNs always showed higher diversity than ICNNs. As shown in Fig. 3 and Fig. 5, interpretable filters of compositional CNNs could explain almost the entire region of the image, while filters of ICNNs could only represent small parts in ball-like areas. Note that sometimes we could not obtain large values of diversity for an ICNN, because RFs of all filters in the ICNN were small, as shown in Fig. 2. Traditional CNNs showed low consistency of visual patterns, which were close to that of randomly synthesized feature maps. This indicated that in terms of filter interpretability, features of filters in traditional CNNs did not show significantly better consistency than synthesized random features. As Table 1 and Table 2 shows, compositional CNNs exhibited comparable classification performance with traditional CNNs. Besides, compositional CNNs achieved higher accuracy than ICNNs in most comparisons.

Visualization of filters. We followed [Zhang *et al.*, 2018] to visualize RFs corresponding to a filter's feature maps. Fig. 3 shows RFs of features of compositional CNNs and ICNNs learned for the binary classification of a single category. In compositional CNNs, given different images, each filter consistently represented the same object part or the same image region. Different filters represented different object parts or image regions. In ICNNs, filters only represented small parts in ball-like areas. In addition, filters in the compositional CNN usually represented more complex shapes than filters in the ICNN. We specifically found out failure cases of interpretable filters in compositional CNNs, as shown in Fig. 8. We also compared RFs between the center of the group and each filter in the group in Fig. 6.

Comparison of interpretable filters in different convolutional layers. As shown in Fig. 4 (a), filters of a high convolutional layer usually represented object parts or image regions, while filters of a middle convolutional layer usually represented local textures or local shapes.

Comparison of interpretable filters learned with different values of K. As shown in Fig. 4 (b), filters in the compositional CNN with $K = 16$ represented more visual patterns than filters in the compositional CNN with $K = 8$.

t-SNE visualization. We visualized filters in a compositional CNN and a traditional CNN using t-SNE [van der Maaten and Hinton, 2008]. These two CNNs were learned based on the VGG-16 using the bird category in the PASCAL-Part dataset. As Fig. 4 (c) shows, feature maps of a compositional CNN seem more clustered than those of a traditional CNN.

5 Conclusion

In this paper, we have proposed a method to modify a traditional CNN into a compositional CNN, in order to make filters in a high convolutional layer encode meaningful visual patterns without any part or region annotations for supervision. Specifically, we design a loss to encourage each filter in the layer consistently represents the same object part or the same image region through different images, and encourage different filters in the layer to represent different object parts and image regions. Experiments have demonstrated the effectiveness of our method.

Acknowledgments

This work is partially supported by the National Nature Science Foundation of China (No. 61976160, 61906120, U19B2043), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

[Bau *et al.*, 2017] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

[Chen *et al.*, 2014] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[Chen *et al.*, 2019] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Fidler and Leonardis, 2007] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007.

[Fong and Vedaldi, 2018] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *CVPR*, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Higgins *et al.*, 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[Huang and Li, 2020] Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *CVPR*, 2020.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[Kortylewski *et al.*, 2020] Adam Kortylewski, Ju He, Qing Liu, and Alan Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *CVPR*, 2020.

[Li *et al.*, 2020] Yuchao Li, Rongrong Ji, Shaohui Lin, Baochang Zhang, Chenqian Yan, Yongjian Wu, Feiyue Huang, and Ling Shao. Interpretable neural network decoupling. *arXiv:1906.01166*, 2020.

[Liang *et al.*, 2020] Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *ECCV*, 2020.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[Ommer and Buhmann, 2009] Bjorn Ommer and Joachim M. Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE T-PAMI*, 32(3):501–516, 2009.

[Ott and Everingham, 2011] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *NeurIPS*, 2017.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE T-PAMI*, 22(8):888–905, 2000.

[Si and Zhu, 2013] Zhangzhang Si and Song-Chun Zhu. Learning and-or templates for object recognition and detection. *IEEE T-PAMI*, 35(9):2189–2205, 2013.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Smith *et al.*, 2013] Brandon M. Smith, Li Zhang, Jonathan Brandt, Zhe Lin, and Jianchao Yang. Exemplar-based face parsing. In *CVPR*, 2013.

[Stone *et al.*, 2017] Austin Stone, Huayan Wang, Michael Stark, Yi Liu, D. Scott Phoenix, and Dileep George. Teaching compositionality to cnns. In *CVPR*, 2017.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[Wang and Yuille, 2015] Jianyu Wang and Alan Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015.

[Zhang *et al.*, 2018] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *CVPR*, 2018.

[Zhu *et al.*, 2010a] Long (Leo) Zhu, Yuanhao Chen, Antonio Torralba, William Freeman, and Alan Yuille. Part and appearance sharing: Recursive compositional models for multi-view. In *CVPR*, 2010.

[Zhu *et al.*, 2010b] Long (Leo) Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.