Training Interpretable Convolutional Neural Networks by Differentiating Class-specific Filters

Haoyu Liang*[1], Zhihao Ouyang*[2,4], Yuyuan Zeng[2,3], Hang Su†[1], Zihao He[5], Shu-Tao Xia[2,3], Jun Zhu†[1], and Bo Zhang[1]

[1]Dept. of Comp. Sci. and Tech., BNRist Center, Inst. for AI, THBI Lab, Tsinghua University, Beijing 100084, China [2]Tsinghua SIGS, Shenzhen 518055, China [3]Peng Cheng Lab [4]ByteDance AI Lab [5]Dept. of CS, University of Southern California

{lianghy18@mails, oyzh18@mails, zengyy19@mails, suhangss@mail, xiast@sz, dcszj@mail, dcszb@mail}.tsinghua.edu.cn zihaoh@usc.edu

Abstract. Convolutional neural networks (CNNs) have been successfully used in a range of tasks. However, CNNs are often viewed as "black-box" and lack of interpretability. One main reason is due to the *filter-class entanglement* – an intricate many-to-many correspondence between filters and classes. Most existing works attempt post-hoc interpretation on a pre-trained model, while neglecting to reduce the entanglement underlying the model. In contrast, we focus on alleviating filter-class entanglement during training. Inspired by cellular differentiation, we propose a novel strategy to train interpretable CNNs by encouraging *class-specific filters*, among which each filter responds to only one (or few) class. Concretely, we design a learnable sparse Class-Specific Gate (CSG) structure to assign each filter with one (or few) class in a flexible way. The gate allows a filter's activation to pass only when the input samples come from the specific class. Extensive experiments demonstrate the fabulous performance of our method in generating a sparse and highly class-related representation of the input, which leads to stronger interpretability. Moreover, comparing with the standard training strategy, our model displays benefits in applications like object localization and adversarial sample detection. Code link: https://github.com/hyliang96/CSGCNN.

Keywords: class-specific filters, interpretability, disentangled representation, filter-class entanglement, gate

1 Introduction

Convolutional Neural Networks (CNNs) demonstrate extraordinary performance in various visual tasks [23, 17, 13, 16]. However, the strong expressive power of CNNs is still far from being interpretable, which significantly limits their applications that require humans' trust or interaction, e.g., self-driving cars and medical image analysis [8, 3].

*Haoyu Liang and Zhihao Ouyang contributed equally.

†Hang Su and Jun Zhu are corresponding authors.

In this paper, we argue that filter-class entanglement is one of the most critical reasons that hamper the interpretability of CNNs. The intricate *many-to-many correspondence* relationship between filters and classes is so-called *filter-class entanglement* as shown on the left of Fig.1. As a matter of fact, previous studies have shown that filters in CNNs generally extract features of a mixture of various semantic concepts, including the classes of objects, parts, scenes, textures, materials, and colors [45, 2]. Therefore alleviating the entanglement is crucial for humans from interpreting the concepts of a filter [45], which has been shown as an essential role in the visualization and analysis of networks [30] in human-machine collaborative systems [41, 44]. To alleviate the entanglement, this paper aims to learn *class-specific filter* which responds to only one (or few) class.

Usually, it is non-trivial to deal with the entanglement, as many existing works show. (1) Most interpretability-related research simply focuses on post-hoc interpretation of filters [2, 36], which manages to interpret the main semantic concepts captured by a filter. However, post-hoc interpretation fails to alleviate the filter-class entanglement prevalent in pre-trained models. (2) Many VAEs' variants [18, 6, 20, 9, 24] and InfoGAN [10] try to learn disentangled data representation with better interpretability in an unsupervised way. However, they are challenged by [25], which proves that it's impossible unsupervised to learn disentangled features without proper inductive bias.

Despite the challenges above, it's reasonable and feasible to learning class-specific filters in *high convolutional layers* in image classification tasks. (1) It has been demonstrated that high-layer convolutional filters extract high-level semantic features which might relate to certain classes to some extent [40]; (2) the redundant overlap between the features extracted by different filters makes it possible to learn specialized filters [31]; (3) specialized filters demonstrate higher interpretability [45] and better performance [31] in computer vision tasks; (4) [19, 38, 28] successfully learn class-specific filters in high convolution layers, though, under an inflexible predefined filter-class correspondence.

Therefore, we propose to learn class-specific filters in the last convolutional layer during training, which is inspired by cellular differentiation [35]. Through differentiation, the stem cells evolve to functional cells with specialized instincts, so as to support sophisticated functions of the multi-cellular organism effectively. For example, neural stem cells will differentiate into different categories like neurons, astrocytes, and oligodendrocytes through particular simulations from transmitting amplifying cells. Similarly, we expect the filters in CNN to "differentiate" to disparate groups that have specialized responsibilities for specific tasks. For specifically, we encourage the CNN to build a one-filter to one-class correspondence (differentiation) during training.

Specifically, we propose a novel training method to learn class-specific filters. Different from existing works on class-specific filters that predefine filer-class correspondence, our model learns a *flexible* correspondence that assigns only a necessary portion of filters to a class and allows classes to share filters. Specifically, we design a learnable Class-Specific Gate (CSG) structure after the last

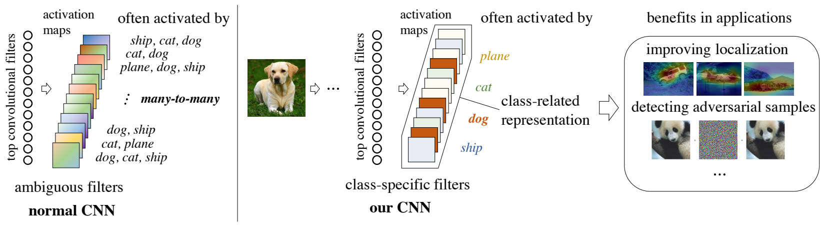

Fig. 1. The motivation of learning class-specific filters. In a normal CNN, each filter responds to multiple classes, since it extracts a mixture of features from many classes [45], which is a symptom of filter-class entanglement. In contrast, we enforce each filter to respond to one (or few) class, namely to be class-specific. It brings better interpretability and class-related feature representation. Such features not only facilitate understanding the inner logic of CNNs, but also benefits applications like object localization and adversarial sample detection.

convolutional filters, which assigns filters to classes and limits each filter's activation to pass only when its specific class(es) is input. In our training process, we periodically insert CSG into the CNN and jointly minimize the classification cross-entropy and the sparsity of CSG, so as to keep the model's performance on classification meanwhile encourage class-specific filters. Experimental results in Sec 4 demonstrate that our training method makes data representation sparse and highly correlated with the labeled class, which not only illustrates the alleviation of filter-class entanglement but also enhances the interpretability from many aspects like filter orthogonality and filter redundancy. Besides, in Sec 5 our method shows benefits in applications including improving objects localization and adversarial sample detection.

Contributions The contributions of this work can be summarized as: (1) we propose a novel training strategy for CNNs to learn a flexible class-filter correspondence where each filter extract features mainly from only one or few classes; (2) we propose to evaluate filter-class correspondence with the mutual information between filter activation and prediction on classes, and moreover, we design a metric based on it to evaluate the overall filter-class entanglement in a network layer; (3) we quantitively demonstrate the benefits of the class-specific filter in alleviating filter redundancy, enhancing interpretability and applications like object localization and adversarial sample detection.

2 Related Works

Existing works related to our work include post-hoc filter interpretation, learning disentangled representation.

Post-hoc Interpretation for Filters is widely studied, which aims to interpret the patterns captured by filters in pre-trained CNNs. Plenty of works visualize the pattern of a neuron as an image, which is the gradient [40, 27, 34] or accumulated gradient [29, 30] of a certain score about the activation of the neuron. Some works determine the main visual patterns extracted by a convolutional filter by treating it as a pattern detector [2] or appending an auxiliary detection module [14]. Some other works transfer the representation in CNN into an explanatory graph [43, 42] or a decision tree [46, 1], which aims to figure out the visual patterns of filters and the relationship between co-activated patterns. Post-hoc filter interpretation helps to understand the main patterns of a filter but makes no change to the existing filter-class entanglement of the pre-trained models, while our work aims to train interpretable models.

Learning Disentangled Representation refers to learning data representation that encodes different semantic information into different dimensions. As a principle, it's proved impossible to learn disentangled representation without inductive bias [25]. Unsupervised methods such as variants of VAEs [21] and InfoGAN [10] rely on regularization. VAEs [21] are modified into many variants [18, 6, 20, 9, 24], while their disentangling performance is sensitive to hyperparameters and random seeds. Some other unsupervised methods rely on special network architectures including interpretable CNNs [45] and CapsNet [33]. As for supervised methods, [37] propose to disentangle with interaction with the environment; [4] apply weak supervision from grouping information, while our work applies weak supervision from classification labels.

Class-specific Filters has been applied in image and video classification task. The existing works focus on improve accuracy, including label consistent neuron [19] and filter bank [38, 28]. However, those works predefine an unlearnable correspondence between filters and classes where principally each filter responds to only one class and all classes occupy the same number of filters. In contrast, this paper focuses on the interpretability of class-specific filters, and we propose a more flexible correspondence where similar classes can share filters and a class can occupy a learnable number of filters. Therefore, our learnable correspondence helps reveal inter-class similarity and intra-class variability.

3 Method

Learning disentangled filters in CNNs alleviates filter-class entanglement and meanwhile narrows the gap between human concept and CNN's representations. In this section, we first present an ideal case of class-specific filters, which is a direction for our disentanglement training, and then we elaborate on our method about how to induce filter differentiation towards it in training an interpretable network.

3.1 Ideally Class-Specific Filters

This subsection introduces an ideal case for the target of our filter disentanglement training. As shown in Fig.2, each filter mainly responds to (i.e. relates to) only one class. We call such filters *ideally class-specific* and call disentangling filter towards it in training as *class-specific differentiation* of filters.

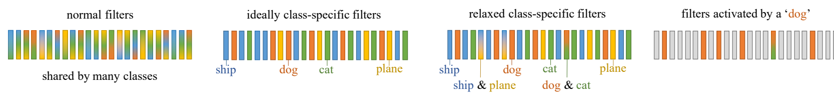

Fig. 2. The intuition of learning class-specific filters. In a standard CNN, a filter extracts a mixture of features from many classes [45], which is a symptom of filter-class entanglement. In contrast, an ideally class-specific filter extracts features mostly from only one class and a relaxed filter can be shared by few classes, For flexibility, we actually apply the relaxed class-specific filter which is allowed to be shared by few classes. Its activation to other classes is weak and has little effect on prediction.

To give a rigorous definition of "ideally class-specific" for a convolutional layers, we use a matrix $G \in [0,1]^{C\times K}$ to measure the relevance between filters and classes, where K is the number of filters, C is the number of classes. Each element $G_c^k \in [0,1]$ represents the relevance between the k-th filter and c-th class, (a larger G_c^k indicates a closer correlation). As shown in Fig.3.2, the k-th filter extracts features mainly in the c-th class iif $G_c^k = 1$. Denote a sample in dataset D as $(x, y) \in D$ where x is an image and $y \in \{1, 2, ..., C\}$ is the label. Given (x, y) as an input, we can index a row $G_y \in [0,1]^K$ from the matrix G, which can be used as a gate multiplied to the activation maps to shut down those irrelevant channels. Let $\tilde{y}$ be the probability vector predicted by the STD path, and $\tilde{y}^G$ be the probability vector predicted by the LSG path where the gate G_y is multiplied on the activation maps from the penultimate layer. Thus, we call convolutional filters as *ideally class-specific* filters, if there exists a G (all columns G_k are *one-hot*) that raises little difference between the classification performance of $\tilde{y}^G$ and $\tilde{y}$.

3.2 Problem formulation

In order to train a CNN towards differentiating filters to class-specific meanwhile keep classification accuracy, we introduce a Class-Specific Gate (CSG) path in addition to the standard (STD) path of forward propagation. In the CSG path, channels are selectively blocked with the learnable gates. This path's classification performance is regarded as a regularization for filter differentiation training.

To derive the formulation of training a CNN with class-specific filters, we start from an original problem that learns ideally class-specific filters and then relax the problem for the convenience of a practical solution.

The Original Problem is to train a CNN with ideally class-specific filters. See Fig.3.2 for the network structure. The network with parameters θ forward propagates in two paths: (1) the standard (STD) path predicting $\tilde{y}_\theta$, and (2) the CSG path with gate matrix G predicting $\tilde{y}_\theta^G$ where activations of the penultimate layer are multiplied by learnable gates G_y for inputs with label y.

In order to find the gate matrix G that precisely describes the relevance between filters and classes, we search in the binary space for a G that yields the best classification performance through the CSG path, i.e., to solve the opti-

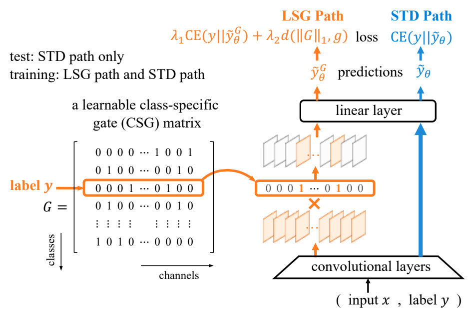

Fig. 3. In Class-Specific Gate (CSG) training, we *alternately* train a CNN through the CSG path and the standard (STD) path. In the CSG path, activations after the penultimate layer (i.e. the last convolution) pass through learnable gates indexed by the label. In training, network parameters and the gates are optimized to minimize the cross-entropy joint with a sparsity regularization for the CSG matrix. In testing, we just run the STD path.

mization problem $\mathit{\Phi}_0(\theta) = \min_G \mathrm{CE}(y||\tilde{y}_\theta^G)$[1] s.t. $\forall k \in \{1, 2, ..., K\}, G_k$ is one-hot. $\mathit{\Phi}_0$ evaluates the performance of the CNN with differentiated filters. Therefore, it is natural to add $\mathit{\Phi}_0$ into training loss as a regularization that forces filters to be class-specific. Thus, we get the following formulation of the *original problem* to train a CNN towards ideally class-specific filters as

$$\min_\theta L_0(\theta) = \mathrm{CE}(y||\tilde{y}_\theta) + \lambda_1 \mathit{\Phi}_0(\theta). \tag{1}$$

where the $CE(y||\tilde{y}_\theta)$ ensures the accuracy and $\lambda_1 \mathit{\Phi}_0(\theta)$ encourage sparsity of G.

However, the original problem is difficult to solve in practice. On the one hand, the assumption that each filter is complete one-filter/one-class assumption hardly holds, since it is usual for several classes to share one high-level feature in CNNs; on the other hand, binary vectors in a non-continuous space are difficult to optimize with gradient descent.

Relaxation To overcome the two difficulties in the original problem above, we relax relax the one-hot vector G^k to a sparse continuous vector $G^k \in [0,1]^C$ where at least one element equals to 1, i.e., $\left\|G^k\right\|_\infty = 1$. To encourage the sparsity of G, we introduce a regularization $d(\|G\|_1, g)$ that encourages the L1 *vector norm* $\|G\|_1$ not to exceed the upper bound g when $\|G\|_1 \geq g$, and has no effect when $\|G\|_1 < g$. A general form for d is $d(a, b) = \psi(\mathrm{ReLU}(a - b))$, where ψ can be any norm, including L1, L2 and smooth-L1 norm. Besides, we should set $g \geq K$ because $\|G\|_1 \geq K$ which is ensured by $\left\|G^k\right\|_\infty = 1$. Using the aforementioned relaxation, $\mathit{\Phi}_0$ is reformulated as

$$\mathit{\Phi}(\theta) = \min_G \{\mathrm{CE}(y||\tilde{y}_\theta^G) + \mu d(\|G\|_1, g)\} \quad s.t.\ G \in V_G, \tag{2}$$

where the set $V_G = \{G \in [0,1]^{C \times K} : \left\|G^k\right\|_\infty = 1\}$ and μ is a coefficient to balance classification and sparsity. $\mathit{\Phi}$ can be regarded as a loss function for *filter-class entanglement*, i.e., a CNN with higher class-specificity has a lower $\mathit{\Phi}$.

Replacing $\mathit{\Phi}_0$ in Eq. (1) with $\mathit{\Phi}$, we get $\min_\theta \mathrm{CE}(y||\tilde{y}_\theta) + \lambda_1 \mathit{\Phi}(\theta)$ as an intermediate problem. It is mathematically equivalent if we move $\min_G$ within $\mathit{\Phi}$ to

[1] $\mathrm{CE}(y||\tilde{y}_\theta^G) = -\frac{1}{|D|}\sum_{(x,y)\in D} \log((\tilde{y}_\theta^G)_y)$, where $\tilde{y}_\theta^G$ is a predicted probability vector.

Algorithm 1 CSG Training

for e in epochs **do**
  for n in batches **do**
    if $e \,\%\, period \leq epoch\_num\_for\_CSG$ **then**
      $\tilde{y}_\theta^G \leftarrow$ prediction through the CSG path with G
      $\mathcal{L} \leftarrow \lambda_1 \mathrm{CE}(y||\tilde{y}_\theta^G) + \lambda_2 d(\|G\|_1, g)$
      $G \leftarrow G - \epsilon \frac{\partial \mathcal{L}}{\partial G}$ ▷ update G using the gradient decent
      $G^k \leftarrow G^k / \left\|G^k\right\|_\infty$ ▷ normalize each column of G
      $G \leftarrow \mathrm{clip}(G, 0, 1)$
    else
      $\tilde{y}_\theta \leftarrow$ prediction through the STD path
      $\mathcal{L} \leftarrow \mathrm{CE}(y||\tilde{y}_\theta)$
    end if
    $\theta \leftarrow \theta - \epsilon \frac{\partial \mathcal{L}}{\partial \theta}$
  end for
end for

the leftmost and replace $\lambda_1 \mu$ with λ_2. Thus, combining Eq. (1) (2), we formulate a *relaxed problem* as

$$\min_{\theta, G} L(\theta, G) = \mathrm{CE}(y||\tilde{y}_\theta) + \lambda_1 \mathrm{CE}(y||\tilde{y}_\theta^G) + \lambda_2 d(\|G\|_1, g) \quad s.t.\ G \in V_G. \tag{3}$$

The relaxed problem is easier to solve by jointly optimizing θ and G with gradient, compared to either the discrete optimization in the original problem or the nested optimization in the intermediate problem. Solving the relaxed problem, we can obtain a CNN for classification with class-specific filters, where G precisely describes the correlation between filters and classes.

3.3 Optimization

To solve the optimization problem formulated in Eq. (3) we apply an approximate projected gradient descent (PGD): when G is updated with gradient, G^k will be normalized by $\left\|G^k\right\|_\infty$ to ensure $\left\|G^k\right\|_\infty = 1$, and then clipped into the range $[0, 1]$.

However, it is probably difficult for the normal training scheme due to poor convergence. In the normal scheme, we predict through both CSG and STD paths to directly calculate $L(\theta, G)$ and update θ and G with gradients of it. Due to that most channels are blocked in the CSG path, the gradient through the CSG path will be much weaker than that of STD path, which hinders converging to class-specific filters.

To address this issue, we propose an *alternate training scheme* that the STD/CSG path works alternately in different epochs. As shown in Algorithm 1, in the epoch for CSG path, we update G, θ with the gradient of $\lambda_1 \mathrm{CE}(y||\tilde{y}_\theta^G) + \lambda_2 d(\|G\|_1, g)$, and in the STD path we update θ with the gradient of $\mathrm{CE}(y||\tilde{y}_\theta)$ In this scheme, the classification performance fluctuates periodically at the beginning but the converged performance is slightly better than the normal scheme in our test. Meanwhile, the filters gradually differentiate into class-specific filters.

Table 1. Metrics of the STD CNN (baseline) and the CSG CNN (Ours).

Dataset	Model	C	K	Training	Accuracy	MIS	L1-density	L1-interval [2]
CIFAR-10	ResNet20	10	64	CSG	**0.9192**	**0.1603**	0.1000	$[0.1, 0.1]$
				STD	0.9169	0.1119	-	-
ImageNet	ResNet18	1000	512	CSG	0.6784	**0.6259**	0.0016	$[0.001, 0.1]$
				STD	**0.6976**	0.5514	-	-
PASCAL VOC 2010	ResNet152	6	2048	CSG	**0.8506**	**0.1998**	0.1996	$[0.1667, 0.2]$
				STD	0.8429	0.1427	-	-

4 Experiment

In this section, we conduct five experiments. We first delve into CSG training from three aspects, so as to respectively study the effectiveness of CSG training, the class-specificity of filters and the correlation among class-specific filters. Especially, to measure filter-class correspondences we apply the mutual information (MI) between each filter's activation and the prediction on each class. In the following parts, we denote our training method Class-Specific Gate as *CSG*, the standard training as *STD*, and CNNs trained with them as *CSG CNNs* and *STD CNNs*, respectively.

Training We use CSG/STD to train ResNet-18/20/152s [17] for classification task on CIFAR-10 [22]/ImageNet [11]/PASCAL VOC 2010 [12] respectively. We select six animals from PASCAL VOC and preprocess it to be a classification dataset. The ResNet-18/20s are trained from scratch and the ResNet-152s are finetuned from ImaageNet. See Appendix B for detailed training settings.

4.1 Effectiveness of CSG Training

First of all, we conduct experiments to verify the effectiveness of our CSG training in learning a sparse gate matrix and achieve high class-specificity of filters.

Quantitative Evaluation Metrics To evaluate the effectiveness of CSG, we calculate 3 metrics: L1-density, mutual information score, and classification accuracy. (1) Accuracy measures the classification performance. (2) To measure the correspondence between filters and classes, we propose the mutual information (MI) matrix $M \in \mathbb{R}^{K\times C}$ where $M_{kc} = \mathrm{MI}(a_k||\mathbf{1}_{y=c})$ is the MI between a_k – the activation of filter-k and class-c. To calculate the MI, we sample (x, y) across the dataset, a_k (the globally avg-pooled activation map of filter k over all the sampled x) is a continuous variable, and $\mathbf{1}_{y=c}$ is a categorical variable. The estimation method [32] for the MI between them is implemented in the API 'sklearn.feature_selection.mutual_info_classif'. Base on this, we propose a

[2] They are $[\frac{1}{C}, \frac{g}{CK}]$ – the theoretical convergence interval for L1-density of CSG CNNs, where g is the upper bound for $\|G\|_1$ in Eq. (2). See Appendix A for the derivation. When the dataset has numerous classes like ImageNet, the L1-density can drop much lower than $\frac{g}{CK}$ due to the projection in our approximate PGD.

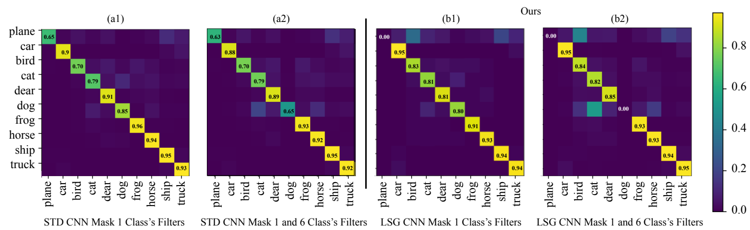

Fig. 5. Classification confusion matrix for STD / CSG models when masking filters highly related to the first class (a1, b1), and the first and sixth classes (a2, b2).

mutual information score $MIS = \text{mean}_k \max_c M_{kc}$ as an overall metric of class-specificity of all filters. Higher MIS indicates higher class-specificity, aka, lower filter-class entanglement. (3) The L1-density $= \frac{\|G\|_1}{KC}$ is the L1-norm of CSG normalized by the number of elements, which measures the sparsity of CSG.

Table 4.1 shows that CSG CNNs are comparable to or even slightly outperforms STD CNNs in test accuracy, while the CSG CNNs have MIS much higher than STD CNNs and the L1-density of G is limited in its theoretical convergence interval. These metrics quantitatively demonstrate CSG's effectiveness on learning a sparse gate matrix and class-specific filters without sacrifice on classification accuracy.

Visualizing the Gate/MI Matrices To demonstrate that the relevance between the learned filters and classes is exactly described by the gate matrix, we visualize gate matrix and MI matrices in Fig.4. (a) demonstrates that CSG training yields a sparse CSG matrix where each filter is only related to one or few classes. (b1,b2) shows that CSG CNN has sparser MI matrices and larger MIS compared to STD CNN. (c) shows the strongest elements in the two matrices almost overlaps, which indicates that CSG effectively learns filters following the guidance of the CSG matrix. These together verify that CSG training effectively learns a sparse gate matrix and filters focusing on the one or few classes described by the gates.

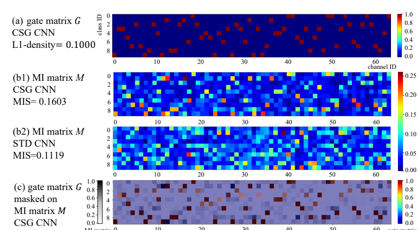

Fig. 4. (a) visualizes the CSG matrix of CSG CNN to verify its sparsity. (b1,b2) compare the MI matrices of CSG/STD CNN and their MIS. (c) is got by overlapping (a) and (b1).

4.2 Study on Class-Specificity

In this subsection, we study the property of class-specific filters and the mechanism of filter differentiation through experiments on ResNet20 trained in 4.1.

Indispensability for Related Class It is already shown by the MI matrix that filters are class-specific, while we further reveal that the filters related to a class are also *indispensable* in recognizing the class. We remove the filters highly related to certain class(es) referencing the gate matrix (i.e., $G_c^k > 0.5$), and then visualize the classification confusion matrices. For STD CNN, we simply remove 10% filters that have the largest average activation to certain classes across the dataset. As shown in Fig.5, when filters highly related to "plane" are removed, the CSG CNN fails to recognize the first class "plane"; nevertheless, the STD CNN still manages to recognize "plane". Analogously, when removed the filters highly related to the first and sixth classes, we observe a similar phenomenon. This demonstrates that the filters specific to a class are indispensable in recognizing this class. Such a phenomenon is because the filters beyond the group mainly respond to other classes and hence can't substitute for those filters.

Mechanism of Class-Specific Differentiation In this part we reveal that the mechanism of filter differentiation by studying the directional similarity between features and gates, and by the way, explain misclassification with the CSG. We use cosine to measure the directional similarity S_y^c between $a_y \in \mathbb{R}^K$ the mean feature (average-pooled feature from class-specific filters) for class y, and G_c the c-th row of the CSG matrix (see Appendix C for details). We calculate S_y^c over all true positive (TP) and false negative (FN) images and obtain similarity matrices $S_{TP}, S_{FN} \in [0,1]^{C\times C}$ respectively, as shown in Fig.6.

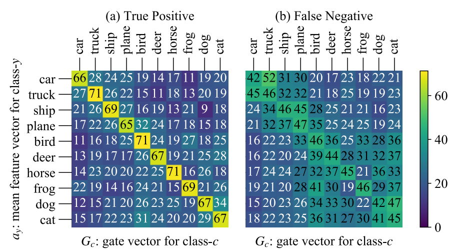

Fig. 6. The cosine similarity between mean feature vectors for a class (y-axis) and a row in the CSG matrix (x-axis), calculated on all TP/FN samples. We reorder classes in CIFAR10 for better visualization.

From the figure, we observe two phenomena and provide the following analysis. (1) TP similarity matrix is diagonally dominant. This reveals mechanism of learning class-specifics: CSG forces filters to yield feature vectors whose direction approaches that of the gate vector for its related class. (2) FN similarity matrix is far from diagonally dominant and two classes with many shared features, such as car & truck and ship & plane, have high similarity in the FN similarity matrix. CSG enlightens us that hard samples with feature across classes tend to be misclassified. Thus, the mechanism of misclassification in the CSG CNN is probably that the features across classes are extracted by the shared filters. To some extent, it proves differentiation is beneficial for accuracy.

4.3 Correlation Between Filters

We further designed several experiments to explore what happen to filters (why they are class-specified). Our analysis shows inter-class filters are approximately orthogonal and less redundant, and the class-specific filters yield highly class-related representation.

Fixing the Gates To make it convenient to study inter-class filters, we group filters in a tidier way – each class monopolizes m_1 filters and m_2 extra fil-

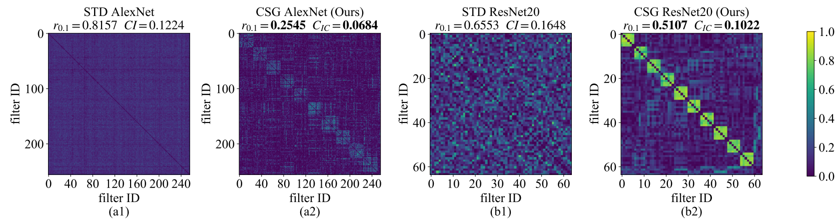

Fig. 7. The correlation matrix of filters (cosine between filters' weights) in AlexNet and ResNet20 trained with STD/CSG. $r_{0.1}$: the ratio of elements≥ 0.1, measures filter redundancy. C_{IC}: the inter-class filter correlation, measures inter-class filter similarity.

ters are shared by all classes. This setting can be regarded as tightening the constraint on the gate matrix to $G \in \{0,1\}^{C \times K}$, according to 3.3 (see Appendix E for illustration). The corresponding CSG matrix is fixed during training. We train an AlexNet [23] ($m_1 = 25, m_2 = 6$) and a ResNet20 ($m_1 = 6, m_2 = 4$) by STD and by CSG in this way. Models trained with this setting naturally inherits all features of previous CSG models and has tidier filter groups.

Filter Orthogonality Analysis

To study the orthogonality between filters, we evaluate filter correlation with the cosine of filters' weights. The correlations between all filters are visualized as correlations matrices $\mathcal{C}$ in Fig.7. In subfigures (a1, b1) for STD models, the filters are randomly correlated with each other. In contrast, in subfigures (a2, b2) for CSG models, the matrices are approximately block-diagonal, which means the correlation between the filters is limited to several class-specific filter groups. This indicates that filters for the same class are highly correlated (non-orthogonal) due to the co-occurrence of features extracted by them, while filters for different classes are almost uncorrelated (orthogonal) for the lack of co-occurrence. See Appendix D for a detailed explanation for orthogonality.

Filter Redundancy To verify that the redundancy of filters are reduced by CSG, we research the inter-class filter correlation. Let the filter-i,j are correlated if their correlation $\mathcal{C}_{i,j} \geq s$ (s is a threshold). We calculate r_s the ratio of such elements in a correlation matrix and plot the results in Fig.8. It shows that CSG significantly reduces the redundant correlation ($\mathcal{C}_{i,j} < s = 0.1$) between most filter pairs from different groups. We explain the reduction of filter redundancy as a natural consequence of encouraging inter-class filters to be orthogonal. For a set of filter groups orthogonal to each other, a filter in any group can not be a linear representation with the filers from other groups. This directly avoids redundant filter across groups. Besides, experiments in [31] also verify the opinion that filter orthogonality reduces filter redundancy.

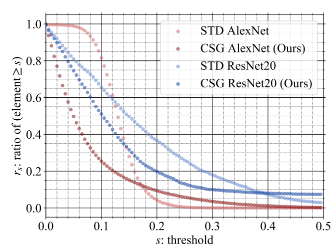

Fig. 8. the ratio of elements larger than a varying threshold in correlation matrices.

Highly Class-Related Representation

Based on filter correlation, we further find that filters trained with CSG yield highly class-related representation, namely the representation of an image tends to exactly correspond to its labeled class rather than other classes. Because the implied class is mainly decided on which filters are most activated, meanwhile those filters are less activated by other classes and less correlated to the filters for other classes.

To verify this reasoning, we analyze the correlation between the filters highly activated by each class. First, we pick out m_1 filters that have the strongest activation to class c, denoted as group A_c. We define inter-class filter correlation as the average correlation between filters in different classes' group A_c: $C_{\mathrm{IC}} = \frac{1}{C(C-1)m^2} \sum_c \sum_{c \neq c'} \sum_{k \in A_c} \sum_{k' \in A_{c'}} \mathcal{C}_{k,k'}$. The results C_{IC} in Fig.7 show that inter-class filter correlation in CSG is about half of that in STD, both on AlexNet and ResNet20. This demonstrates that different classes tend to activate uncorrelated filters in CSG CNNs, aka, CSG CNNs yield highly class-related representations where the representations for different classes have less overlap.

5 Application

Using the class-specificity of filters, we can improve filters' interpretability on object localization. Moreover, the highly class-related representation makes it easier to distinguish abnormal behavior of adversarial samples.

5.1 Localization

In this subsection, we conduct experiments to demonstrate that our class-specific filters can localize a class better, for CSG training is demonstrated to encourage each filter in the penultimate layer to focus on fewer classes.

Localization method

Gradient maps [34] and activation maps (resized to input size) is a widely used method to determine the area of objects or visual concepts, which not only works in localization task without bounding box labels [2], but also take an important role in network visualization and understanding the function of filters [47]. We study CSG CNNs' performs on localizing object classes with three localization techniques based on filters, including gradient-based saliency map (GradMap) and activation map (ActivMap) for a single filter and classification activation map (CAM) [2] for all filters. See Appendix G for the localization techniques.

Quantitative evaluation We train ResNet152s to do classification on preprocessed PASCAL VOC and use Avg-IoU (average intersection over union) and AP20/AP30 (average precision 20%/30%) to evaluate their localization. Higher metrics indicate better localization. See Appendix H for a detailed definition of the metrics. The results for localization with one or all filters are shown in Table 2. For localization with one filter, most classes are localized better with CSG CNN. That's because CSG encourages filters to be activated by the labeled class

Table 2. The performance of localization with resized activation maps in the CSG/STD CNN. For almost all classes, CSG CNN significantly outperforms STD CNN both on Avg-IoU and AP20/AP30.

localization	metric	training	bird	cat	dog	cow	horse	sheep	total
GradMap for one filter	Avg-IoU	CSG	**0.2765**	**0.2876**	**0.3313**	**0.3261**	**0.3159**	**0.2857**	**0.3035**
		STD	0.2056	0.2568	0.2786	0.2921	0.2779	0.2698	0.2606
	AP20	CSG	**0.6624**	**0.8006**	**0.9170**	**0.8828**	**0.8204**	**0.8089**	**0.8165**
		STD	0.4759	0.7081	0.7556	0.8069	0.7621	0.7764	0.7029
ActivMap for one filter	Avg-IoU	CSG	**0.3270**	**0.4666**	**0.4005**	**0.4228**	**0.3358**	**0.4344**	**0.4290**
		STD	0.2865	0.3815	0.3443	0.3674	0.3020	0.3653	0.3602
	AP30	CSG	**0.5876**	**0.8609**	**0.7502**	**0.8075**	**0.5811**	**0.8606**	**0.8254**
		STD	0.4751	0.7003	0.6365	0.7216	0.5154	0.6972	0.6759
CAMs for all filters	Avg-IoU	CSG	**0.3489**	**0.4027**	**0.3640**	**0.3972**	**0.3524**	**0.3562**	**0.3694**
		STD	0.3458	0.3677	0.3492	0.3516	0.3170	0.3470	0.3483
	AP30	CSG	0.6399	**0.8382**	**0.7197**	**0.7517**	**0.7136**	**0.7073**	**0.7294**
		STD	**0.6495**	0.7832	0.7085	0.6621	0.5825	0.6504	0.6853

rather than many other classes, which alleviates other classes' interference on GradMaps and ActivMaps for each filters. Furthermore, as a weighted sum of better one-filter activation maps, CSG also outperform STD on CAMs.

Discussion It is widely recognized [2] that localization reveals what semantics or classes a filter focuses on. Compared with a vanilla filter, a class-specific filter responds more intensively to the region of relevant semantics and less to the region of irrelevant semantics like background. Thus localization is improved. For STD training, confusion on the penultimate layer is caused by all convolutional layers, however, in our CSG training they are jointly trained with backpropogation to disentangle the penultimate layer.

Visualiziation Besides the quantitative evaluation above, in Fig.9, we also visualize some images and their CAMs from the STD/CSG CNN on ImageNet [11]. We observe that the CAMs of STD CNN often activate extra area beyond the labeled class. However, CSG training successfully helps the CNN find a more precise area of the labeled objects. Such a phenomenon vividly demonstrates that CSG training improves the performance in localization.

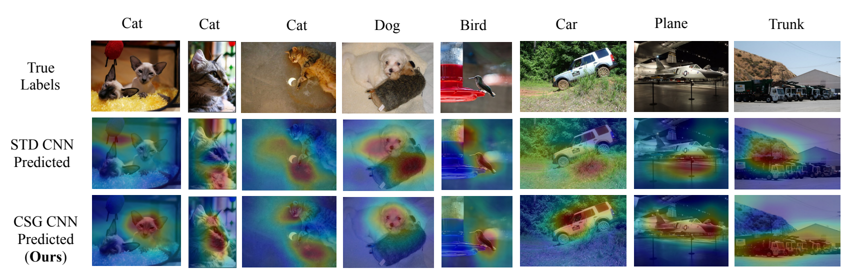

Fig. 9. Visualizing the localization in STD CNN and CSG CNN with CAM [47].

5.2 Adversarial Sample Detection

This subsection shows that the highly class-related representation of our CSG training can promote adversarial samples detection. It is studied [39] that adversarial samples can be detected based on the abnormal representation across layers: in low layers of a network, the representation of an adversarial sample is similar to the original class, while on the high layers it is similar to the target class. Such mismatch is easier to be detected with the class-related representation from CSG, where the implied class are more exposed.

To verify this judgment, we train a random forest [5] with the features of normal samples and adversarial samples extracted by global average pooling after each convolution layers of ResNet20 trained in 4.1. We generate adversarial samples for random targeted classes by commonly used white-box attacks such as FGSM [15] , PGD [26], and CW [7]. See Appendix I for detailed attack settings. We repeat each experiment five times and report the mean error rates in Table 3. The experimental results demonstrate that the class-related representation can better distinguish the abnormal behavior of adversarial samples and hence improve the robustness of the model. Further experiments in Appendix J show that CSG training can improve robustness in defending adversarial attack.

Num. of training samples		500	1000	2000
FGSM	STD	4.76	4.60	4.15
	CSG (Ours)	**4.50**	**3.94**	**3.84**
PGD	STD	5.03	6.20	4.08
	CSG (Ours)	**4.52**	**3.95**	**3.25**
CW	STD	7.33	6.76	6.46
	CSG (Ours)	**7.03**	**6.18**	**5.87**

Table 3. The mean error rate(%) for random forests on adversarial samples detection with the features of CNN.

6 Conclusion

In this work, we propose a simply yet effective structure – Class-Specific Gate (CSG) to induce filter differentiation in CNNs. With reasonable assumptions about the behaviors of filters, we derive regularization terms to constrain the form of CSG. As a result, the sparsity of the gate matrix encourages class-specific filters, and therefore yields sparse and highly class-related representations, which endows model with better interpretability and robustness. We believe CSG is a promising technique to differentiate filters in CNNs. Referring to CSG's successful utility and feasibility in the classification problem, as one of our future works, we expect that CSG also has the potential to interpret other tasks like detection, segmentation, etc, and networks more than CNNs.

Acknowledgement This work was supported by the National Key R&D Program of China (2017YFA0700904), NSFC Projects (61620106010, U19B2034, U1811461, U19A2081, 61673241, 61771273), Beijing NSF Project (L172037), PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019), Beijing Academy of Artificial Intelligence (BAAI), Tsinghua-Huawei Joint Research Program, a grant from Tsinghua Institute for Guo Qiang, Tiangong Institute for Intelligent Computing, the JP Morgan Faculty Research Program, Microsoft Research Asia, Rejoice Sport Tech. co., LTD and the NVIDIA NVAIL Program with GPU / DGX Acceleration.

References

1. Bai, J., Li, Y., Li, J., Jiang, Y., Xia, S.: Rectified decision trees: Towards interpretability, compression and empirical soundness. arXiv preprint arXiv:1903.05965 (2019)
2. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6541–6549 (2017)
3. Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U.: Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911 (2017)
4. Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
5. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
6. Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β-vae. arXiv preprint arXiv:1804.03599 (2018)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: S&P (2017)
8. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1721–1730. ACM (2015)
9. Chen, T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. In: Advances in Neural Information Processing Systems. pp. 2610–2620 (2018)
10. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2172–2180 (2016)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html
13. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
14. Gonzalez-Garcia, A., Modolo, D., Ferrari, V.: Do semantic parts emerge in convolutional neural networks? International Journal of Computer Vision **126**(5), 476–494 (2018)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2014)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969 (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

18. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. International Conference on Learning Representations **2**(5), 6 (2017)
19. Jiang, Z., Wang, Y., Davis, L., Andrews, W., Rozgic, V.: Learning discriminative features via label consistent neural network. In: 2017 IEEE Winter Conference on Applications of Computer Vision. pp. 207–216. IEEE (2017)
20. Kim, H., Mnih, A.: Disentangling by factorising. arXiv preprint arXiv:1802.05983 (2018)
21. Kingma, D.P., Welling, M.: Stochastic gradient vb and the variational autoencoder. In: International Conference on Learning Representations (2014)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical Report TR-2009 (2009)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
24. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848 (2017)
25. Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359 (2018)
26. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
27. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5188–5196 (2015)
28. Martinez, B., Modolo, D., Xiong, Y., Tighe, J.: Action recognition with spatial-temporal discriminative filter banks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5482–5491 (2019)
29. Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going deeper into neural networks (2015), https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html
30. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill (2018). https://doi.org/10.23915/distill.00010, https://distill.pub/2018/building-blocks
31. Prakash, A., Storer, J., Florencio, D., Zhang, C.: Repr: Improved training of convolutional filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10666–10675 (2019)
32. Ross, B.C.: Mutual information between discrete and continuous data sets. PloS one **9**(2), e87357 (2014)
33. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. pp. 3856–3866 (2017)
34. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
35. Smith, A.G., Heath, J.K., Donaldson, D.D., Wong, G.G., Moreau, J., Stahl, M., Rogers, D.: Inhibition of pluripotential embryonic stem cell differentiation by purified polypeptides. Nature **336**(6200), 688–690 (1988)
36. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)

37. Thomas, V., Bengio, E., Fedus, W., Pondard, J., Beaudoin, P., Larochelle, H., Pineau, J., Precup, D., Bengio, Y.: Disentangling the independently controllable factors of variation by interacting with the world. arXiv preprint arXiv:1802.09484 (2018)
38. Wang, Y., Morariu, V.I., Davis, L.S.: Learning a discriminative filter bank within a cnn for fine-grained recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4148–4157 (2018)
39. Wang, Y., Su, H., Zhang, B., Hu, X.: Interpret neural networks by identifying critical data routing paths. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8906–8914 (2018)
40. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. pp. 818–833. Springer (2014)
41. Zhang, Q., Cao, R., Nian Wu, Y., Zhu, S.C.: Mining object parts from cnns via active question-answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 346–355 (2017)
42. Zhang, Q., Cao, R., Shi, F., Wu, Y.N., Zhu, S.C.: Interpreting cnn knowledge via an explanatory graph. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
43. Zhang, Q., Cao, R., Wu, Y.N., Zhu, S.C.: Growing interpretable part graphs on convnets via multi-shot learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
44. Zhang, Q., Cao, R., Zhang, S., Redmonds, M., Wu, Y.N., Zhu, S.C.: Interactively transferring cnn patterns for part localization. arXiv preprint arXiv:1708.01783 (2017)
45. Zhang, Q., Nian Wu, Y., Zhu, S.C.: Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8827–8836 (2018)
46. Zhang, Q., Yang, Y., Ma, H., Wu, Y.N.: Interpreting cnns via decision trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6261–6270 (2019)
47. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2921–2929 (2016)