Research Article

SAL-Net: Self-Supervised Attribute Learning for Object Recognition and Segmentation

Shu Yang ¹, JingWang ¹, Sheeraz Arif ², Minli Jia ³, and Shunan Zhong ¹

¹School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China

²Department of Computer Science, Faculty of Information Technology, Salim Habib University, Karachi 75400, Pakistan

³Research Institute of China Mobile Communications Co., Ltd., Beijing 100053, China

Correspondence should be addressed to JingWang; wangjing@bit.edu.cn

Received 30 June 2021; Accepted 7 September 2021; Published 30 September 2021

Academic Editor: Ming Yan

Copyright © 2021 Shu Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Existing attribute learning methods rely on predefined attributes, which require manual annotations. Due to the limitation of human experience, the predefined attributes are not capable enough of providing enough description. This paper proposes a self-supervised attribute learning (SAL) method, which automatically generates attribute descriptions by differentially occluding the object region to deal with the above problems. The relationship between attributes is formulated with triplet loss functions and is utilized to supervise the CNN. Attribute learning is used as an auxiliary task of a multitask image classification and segmentation network, in which self-supervision of attributes motivates the CNN to learn more discriminative features for the main semantic tasks. Experimental results on public benchmarks CUB-2011 and Pascal VOC show that the proposed SAL-Net can obtain more accurate classification and segmentation results without additional annotations. Moreover, the SAL-Net is embedded into a multiobject recognition and segmentation system, which realizes instance-aware semantic segmentation with the help of a region proposal algorithm and a fusion nonmaximum suppression algorithm.

1. Introduction

Visual attributes are designed as midlevel semantic features to describe objects. However, most of the existing attribute learning methods [1–7] rely on manually annotated datasets [8–11]. In these datasets, domain experts predefined semantic attributes, and then, annotators give their judgments on whether a certain attribute appears in each object. On the one hand, the higher the dimensionality of the predefined attributes, the longer it takes to label each object, making attributes with limited dimensions more effective in obtaining large-scale datasets with limited resources. On the other hand, the predefined attributes with limited dimensions cannot fully describe the object, reducing the attribute dataset's usability. Since there are always some potential attributes essential for object representation, the predefined attribute is not capable enough to describe the real world, even with high dimensionality.

With the increasing demand for large-scale training sets for deep learning models [12–22], some researchers propose

self-supervised learning methods [23–26], which train CNNs by automatic generation of labels based on the structure or characteristics of the image itself. We observe that each attribute of the object corresponds to a particular part of the object region. For example, the attribute “blue wings” depends on the visual features inside the wings. Certain attributes are hidden by occluding a part of the object region, making the occluded object have a slightly different attribute distribution than the original object. Inspired by the aforementioned fact, this research article proposes an automatic attribute generation method by occluding the object twice to obtain augmented objects with different attributes from the original object. The relationship between attributes can be formulated and utilized as the self-supervision to learn an attribute CNN. Based on the proposed method, this paper extends our previous work [27] by replacing the manually annotated supervision with the formulated attribute relationship to improve object recognition and segmentation.

The contributions of this paper can be summarized as follows:

- (1) A self-supervised attribute learning (SAL) method is proposed by automatically generating attribute relationships to learn the CNN
- (2) We design a multitask deep neural network SAL-Net, in which attribute learning is an auxiliary self-supervised task to alleviate intraclass variations in recognition tasks and refine the segmentation results
- (3) We embed the SAL-Net into an instance-aware semantic segmentation system, in which a Fusion NMS algorithm is proposed to deal with repeated extraction of the same object

The remainder of this paper is organized as follows. The following section presents some relevant studies, including attribute learning methods and self-supervised learning methods. Then, the methodology of the proposed self-supervised attribute learning approach is described, and extensive experiments have been carried out to compare with baseline and state-of-the-art methods on two public datasets. Finally, the article is concluded.

2. Related Work

2.1. Attribute Learning. Visual attributes denote some semantic characteristics, which can be predefined to describe objects, including color, shape, and material. Lampert et al. [5] developed an AWA dataset consisting of 37,322 images of 50 animal classes and provided an 85-dimensional attribute distribution for each class. Deng et al. [9] composed a PETA dataset containing 19,000 images of 8705 pedestrians and provided 61 binary and four multiclass attributes for each pedestrian. Farhadi et al. [8] annotated 12,679 samples of Pascal VOC [28] to develop an attribute dataset aPY, where each sample is labeled with a 64-dimensional attribute description. Patterson and Hays [11] provided 204-dimensional attribute vectors on 29 categories of samples in COCO [29], while only 18,073 samples are labeled with over ten positive attributes.

Attribute learning has been widely used in computer vision fields, such as fine-grained recognition [2–4], zero-shot learning [5–7], face analysis [30–33], person reidentification [9, 34–38], and semantic segmentation [39–41]. Akata et al. [6] presented a model that embedded each category label in the space of attribute vectors for unseen object recognition. Han et al. [4] proposed an attribute-aware attention model to learn local attribute representation and global category representation for fine-grained image classification. In [37], pedestrian attributes and ID labels have been jointly learned with a shared backbone CNN, where these attributes are combined with low-level visual features to improve reidentification accuracy. Zhang et al. [38] jointly learned the center position, scale, offset, and semantic attributes in a multitask CNN for pedestrian detection tasks. In [41], an end-to-end attribute-aware semantic segmentation method was proposed by replacing the single category class *person* with nine orientation attribute classes.

However, most of the existing attribute learning methods are limited by the representative ability of human-defined

attributes. Therefore, some researchers have proposed to learn latent attributes to obtain more discriminative features. Fu et al. [42] introduced a concept of semilatin attribute space, expressing user-defined and latent attributes in a unified framework, and proposed a novel scalable probabilistic topic model for learning multimodal semilatin attributes. Peng et al. [43] proposed a novel dictionary learning model which decomposes the dictionary space into three parts corresponding to semantic, latent discriminative, and latent background attributes. Li et al. [44] proposed an end-to-end network that is capable of learning discriminative semantic representations in an augmented space introduced for both user-defined and latent attributes. Xie et al. [45] proposed a semantic dictionary learning approach to exploit the latent visual attributes and align the visual-semantic spaces at the same time. Unlike the aforementioned methods, which learn latent attributes complementary to human-defined attributes for cross-class transfer learning, this paper proposes a self-supervised attribute learning method for object recognition and segmentation, which does not require additional attribute annotations.

2.2. Self-Supervised Learning. Convolutional neural networks (CNNs) [12–15] are always end-to-end learned in a supervised style, obtaining more discriminative features than traditional hand-craft methods [46–48]. Large-scale training samples are easy to obtain, while their human-annotated labels are costly. Thus, some researchers have proposed self-supervised learning models, which automatically generate labels for specific tasks by processing the samples. Doersch et al. [23] trained a neural network that can predict relative position, where the position labels are generated by splitting the entire image into blocks. Zhang et al. [24] transformed colorful images into gray-scale images and proposed to learn latent semantic information of pixels from the colorization task. Gidaris et al. [25] trained a network to predict the rotation angle of the image, where training samples are obtained by rotating the original image by different angles.

Different from the above methods, which pretrain the CNN on a set of unlabeled data to build representations and utilize them for downstream tasks, some researchers leverage self-supervised learning with multitask learning. Lee et al. [26] embedded the rotation prediction task into the traditional classification task to attain a significant improvement in classification accuracy. In the proposed SAL-Net, self-supervised attribute prediction, image-level classification, and pixel-level segmentation are jointly learned to more generalized features for different semantic analysis tasks.

3. Methodology

In this section, the proposed self-supervised attribute learning method is described, including an attribute generation approach and the definition of attribute loss. Then, a multitask deep neural network, SAL-Net, is designed based on the proposed method. In the network, attribute learning performs as an auxiliary task to constrain the CNN to learn more discriminative representations for object recognition

and segmentation. Moreover, the proposed self-supervised attribute learning method is extended to instance-aware semantic segmentation by a multiobject recognition and segmentation system.

3.1. Attribute Generation. There is a correspondence between attributes and each local region of the object, e.g., the *yellow forehead* attribute of the bird depends on the features in the *forehead* region. Therefore, an automatic attribute generation method is proposed, generating two additional training samples and the attribute relationship between them by differentially occluding an original training sample. As shown in Figure 1, the original sample $A = \{\mathbf{I}^A, \mathbf{c}^A, \mathbf{s}^A\}$ includes an image $\mathbf{I}^A$, a category label $\mathbf{c}^A$, and a mask $\mathbf{s}^A$. According to the contour information provided by the mask, the object's bounding box can be obtained. The area inside the bounding box is divided into 3×3 blocks, labeled 1–9. Two blocks are randomly selected, and the first block is occluded to obtain a new sample $B = \{\mathbf{I}^B, \mathbf{c}^B, \mathbf{s}^B\}$. The pixels in the occluded block are set to a constant, as given in

$$\begin{cases} \mathbf{I}(i, j)[0] = 0.406 \times 256, \\ \mathbf{I}(i, j)[1] = 0.456 \times 256, \\ \mathbf{I}(i, j)[2] = 0.485 \times 256, \end{cases} \quad (1)$$

where $\text{mean} = [0.406, 0.456, 0.485]$ denotes the normalized mean BGR value among all pixels of the ImageNet dataset [49] and (i, j) represents the position of each occluded pixel. In the mask $\mathbf{s}^B$, the occluded pixel is set to 0. Based on sample B , the second block is occluded to obtain a new sample $C = \{\mathbf{I}^C, \mathbf{c}^C, \mathbf{s}^C\}$. In this way, the original single training sample A is expanded into a group $\{A, B, C\}$. The positive attributes of the occluded sample are more likely to be less than the positive attributes of the sample before the occlusion. In particular, sample C is more occluded than sample B , which results in the attribute difference between C and A greater than the difference between B and A . The attribute relationship is described in

$$0 \leq |\mathbf{a}^B - \mathbf{a}^A| \leq |\mathbf{a}^C - \mathbf{a}^A|, \quad (2)$$

where $\mathbf{a}^A$, $\mathbf{a}^B$, and $\mathbf{a}^C$ represent the attribute distribution of samples A , B , and C . $|\cdot|$ is a measure of the difference between two attribute distributions. The equal sign $\leq$ means that when the occluded block does not contain any attributes, the occluded sample maintains the original attribute distribution. In the classification task, the category label of sample A is directly applied to B and C , i.e., $\mathbf{c}^A = \mathbf{c}^B = \mathbf{c}^C$, which makes the learned features invariant to occlusion for categorization.

3.2. Triplet Loss. A triplet loss function is defined to learn the attribute prediction model, which consists of two parts.

The one constrains the attribute difference between C and A greater than the attribute difference between B and A :

$$\mathcal{L}_1(\Theta) = \left[\|P_a(\mathbf{I}^B, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 - \|P_a(\mathbf{I}^C, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 + \alpha \right]_+, \quad (3)$$

where $P_a(\mathbf{I}, \Theta) \in \mathbb{R}^{K_a}$ represents the predicted attribute distribution of the input image $\mathbf{I}$ based on the parameter set Θ . $\|\cdot\|_2$ denotes the Euclidean distance between two distributions. α is the minimum margin between two metrics:

$$[x]_+ = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (4)$$

It means that when the distance between $P_a(\mathbf{I}^A, \Theta)$ and $P_a(\mathbf{I}^C, \Theta)$ is larger than the distance between $P_a(\mathbf{I}^B, \Theta)$ and $P_a(\mathbf{I}^A, \Theta)$ plus α , the cost will be 0. Thus, by minimizing $\mathcal{L}_1(\Theta)$, we can keep A away from C and bring A close to B in the attribute space.

However, $\mathcal{L}_1(\Theta)$ can be minimized via $\|P_a(\mathbf{I}^B, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 \rightarrow 0$. It results in the same attribute description for A and B , which is not our expectation. Thus, an additional triple loss function is defined:

$$\mathcal{L}_2(\Theta) = \left[\alpha - \|P_a(\mathbf{I}^B, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 \right]_+. \quad (5)$$

It means that when the distance between $P_a(\mathbf{I}^B, \Theta)$ and $P_a(\mathbf{I}^A, \Theta)$ is smaller than α , there will be a cost.

By simultaneously minimizing $\mathcal{L}_1$ and $\mathcal{L}_2$, we can learn appropriate parameters to realize the attribute relationship represented in Equation (2). The randomly selected block probably contains attributes for the entire training set, so it is reasonable to set α to a positive constant rather than 0.

3.3. SAL-Net. In this section, an object recognition and segmentation network SAL-Net is designed based on the proposed self-supervised attribute learning method. As shown in Figure 2, the end-to-end network consists of three main components highlighted in different colors:

- (1) A shared feature coding module (the green block). It consists of a backbone CNN to extract image features and an attribute feature coding model to obtain attribute feature maps for segmentation and recognition tasks
- (2) A segmentation module (the orange block). It projects the attribute feature maps into multiscale segmentation predictions
- (3) A classification module (the blue block). It projects semantic features to a smooth category space and combines the obtained category description with the attribute description to predict the category label

3.3.1. Attribute Feature Encoding. In CNN, features extracted by different layers contain different attribute information. Local features are extracted by low-level layers, e.g., yellow forehead and blue wing, while holistic features are

FIGURE 1: Illustration of the proposed automatic attribute generation approach.

FIGURE 2: The architecture of the proposed SAL-Net. It contains (1) a shared feature encoding module (the green block), in which features from multiple CNN layers are encoded as attribute features under the supervision of attribute relationships; (2) a segmentation module (the orange block), which maps the attribute features into coarse-to-fine segments; (3) a classification branch (the blue block), which predicts category labels with the combination of category and attribute description.

extracted by high-level layers, e.g., chicken-like shape. Training with different features can improve the robustness of the attribute model. Therefore, we encode the visual feature $\psi^k \in \mathbb{R}^{D^k \times h^k \times w^k}$ with a shared attribute coding module f_e and obtain the attribute-wise feature $\psi_e^k \in \mathbb{R}^{K_a \times h^k \times w^k}$:

$$\psi_e^k(\Theta) = f_e\left(\psi^k, \Theta_t^k, \Theta_e\right), \quad k \in [1, K]. \quad (6)$$

There are $K = 3$ layers selected in the backbone CNN, i.e., the Conv3_x, Conv4_x, and Conv5_x. Since the dimensions of the features extracted from different layers are

different, a convolutional layer Θ_t^k is applied to convert D^k feature maps into $C = 256$ feature maps. Then, a shared 3×3 convolutional layer Θ_e is utilized for encoding the features as K_a -dimensional attribute-wise feature maps.

The encoded K features are fused to obtain the attribute prediction $P_a \in [0, 1]^{K_a}$:

$$P_a = \frac{1}{1 + e^{-\sum_{k=1}^K V^k}}, \quad k_a \in [1, K_a], \quad (7)$$

where each item in $V^k \in \mathbb{R}^{K_a}$ denotes the maximum in each attribute feature map ψ_e^k .

According to Equations (3) and (5), the attribute loss function is defined as

$$\begin{aligned} \mathcal{L}_a(\Theta) = \mathcal{L}_1 + \mathcal{L}_2 = & \left[\|P_a(\mathbf{I}^B, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 \right. \\ & \left. - \|P_a(\mathbf{I}^C, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 + \alpha \right]_+ \\ & + \left[\alpha - \|P_a(\mathbf{I}^B, \Theta) - P_a(\mathbf{I}^A, \Theta)\|_2^2 \right]_+. \end{aligned} \quad (8)$$

3.3.2. Segmentation Module. It consists of K segmentation models $\{\Theta_m^k\}_{k=1}^K$. Each segmentation model maps an attribute-wise feature ψ_e^k into a segmentation map $\psi_m^k \in \mathbb{R}^{H \times W}$:

$$\psi_m^k(\Theta) = f_m(\psi_e^k, \Theta_m^k), \quad k \in [1, K]. \quad (9)$$

The segmentation model f_m consists of a 1×1 convolutional layer to localize the object and a deconvolutional layer to upsample the segmentation to the same size as the original image.

The final segmentation is obtained by element-wise adding K coarse-to-fine segmentation maps and then performing the sigmoid operation:

$$P_m = \frac{1}{1 + e^{-\sum_{k=1}^K \psi_m^k}}. \quad (10)$$

$P_m \in [0, 1]^{H \times W}$ represents the probability of each pixel belonging to the foreground object.

The segmentation loss function is defined as the cross-entropy between the prediction P_m and the mask s provided by the dataset. $\mathcal{H}(v, u)$ is defined as the cross-entropy of the ground-truth v and the prediction u :

$$\mathcal{H}(v, u) = -v \log u - (1 - v) \log (1 - u). \quad (11)$$

Three training samples are fed into the network during the training phase. Thus, the segmentation loss function is defined as

$$\begin{aligned} \mathcal{L}_m(\Theta) = & \frac{1}{W \times H} \sum_{w=1}^W \sum_{h=1}^H \left[\mathcal{H}(s^A, P_m(I^A, \Theta))_{w,h} \right. \\ & \left. + \mathcal{H}(s^B, P_m(I^B, \Theta))_{w,h} + \mathcal{H}(s^C, P_m(I^C, \Theta))_{w,h} \right]. \end{aligned} \quad (12)$$

3.3.3. Classification Module. It combines category descriptions and attribute descriptions to classify the image. The semantic feature ψ , which is the output of the global pooling layer in the backbone CNN, is encoded into a category description feature P_{cate} :

$$P_{\text{cate}}(\Theta) = f_{\text{cate}}(\psi, \Theta_c), \quad P_{\text{cate}} \in [0, 1]^{K_c}. \quad (13)$$

f_{cate} denotes a fully connected layer with softmax activation. The obtained description P_{cate} represents the location of the input sample in a category description space Q . The space Q is defined by applying the label smoothing [15] strategy on the typical one-hot encoding category labels:

$$Q(k_c) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{K_c}, & k_c = \mathbf{c}, \\ \frac{\varepsilon}{K_c}, & \text{otherwise.} \end{cases} \quad (14)$$

The loss function of the category feature is defined as

$$\begin{aligned} \mathcal{L}_{\text{cate}}(\Theta) = & - \sum_{k_c=1}^{K_c} Q(k_c) \log (P_{\text{cate}}(\mathbf{I}^A, \Theta, k_c)) \\ & - \sum_{k_c=1}^{K_c} Q(k_c) \log (P_{\text{cate}}(\mathbf{I}^B, \Theta, k_c)) \\ & - \sum_{k_c=1}^{K_c} Q(k_c) \log (P_{\text{cate}}(\mathbf{I}^C, \Theta, k_c)) \\ = & -(1 - \varepsilon) \log \left(\prod_{A,B,C} P_{\text{cate}}(I^i, \Theta, c) \right) \\ & - \varepsilon \sum_{k_c=1}^{K_c} \frac{1}{K_c} \log \left(\prod_{A,B,C} P_{\text{cate}}(I^i, \Theta, k_c) \right). \end{aligned} \quad (15)$$

$\prod_{A,B,C}$ represents the product of the category features of three samples. Then, the category feature P_{cate} and the attribute feature P_a are concatenated to obtain a $(K_c + K_a)$ -dimensional semantic feature vector, which is utilized to predict the category label with a classifier:

$$P_c(\Phi) = f_c(P_{\text{cate}}, P_a, \Phi), \quad P_c \in [0, 1]^{K_c}. \quad (16)$$

Φ denotes the parameter set of the classifier, e.g., a fully connected layer with softmax activation works well. An additional loss is defined to train the classifier:

$$\begin{aligned}\mathcal{L}_c(\Phi) = & -\log(P_c(P_{\text{cate}}^A, P_a^A, \Phi, c)) - \log(P_c(P_{\text{cate}}^B, P_a^B, \Phi, c)) \\ & - \log(P_c(P_{\text{cate}}^C, P_a^C, \Phi, c)) = -\log\left(\prod_{A,B,C} P_c(P_{\text{cate}}, P_a, \Phi, c)\right).\end{aligned}\quad (17)$$

In the training phase, we automatically generate N group samples $\{(A, B, C)\}_{i=1}^N$ based on N original samples and utilize these samples to optimize parameters of the above SAL-Net by minimizing the total cost of three tasks:

$$\Theta^*, \Phi^* = \underset{\Theta, \Phi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N [\mathcal{L}_a(\Theta) + \mathcal{L}_m(\Theta) + \mathcal{L}_{\text{cate}}(\Theta) + \mathcal{L}_c(\Phi)]. \quad (18)$$

3.4. Multiobject System. In this section, a multiobject recognition and segmentation system is designed based on the proposed SAL-Net to achieve instance-aware semantic segmentation. As shown in Figure 3, the system consists of four parts: an attribute generation module, a region proposal module, the SAL-Net, and a postprocessing module.

3.4.1. Candidate Region Proposal Module. This module automatically generates candidate regions from the input image based on the selective search [50] algorithm. In the training phase, this module generates negative samples. We calculate the IoU (intersection over union) score between each proposal and ground-truth regions. The proposal with a small maximum IoU, e.g., smaller than 0.1, is taken as a “background” sample. In the testing phase, this module generates testing samples which are fed into the SAL-Net.

3.4.2. Attribute Generation Module. It generates positive samples for training SAL-Net, as shown in Figure 1. If a training image contains N_+ objects, then N_+ groups of positive samples can be obtained, where each group contains three samples $\{A, B, C\}$.

3.4.3. SAL-Net Module. The structure is similar to Figure 2, except N_- negative samples being utilized for training along with N_+ groups of positive samples. We add a background class into the category description space; that is, the category feature P_{cate} becomes a $(K_c + 1)$ -dimensional vector. It is worth noticing that negative samples are not for training on attributes and segmentation tasks. Equation (18) is modified with the following given formula:

$$\begin{aligned}\Theta^*, \Phi^* = & \underset{\Theta, \Phi}{\operatorname{argmin}} \frac{1}{N_+} \sum_{i=1}^{N_+} [\mathcal{L}_a(\Theta) + \mathcal{L}_m(\Theta) + \mathcal{L}_{\text{cate}}(\Theta) + \mathcal{L}_c(\Phi)] \\ & + \frac{1}{N_-} \sum_{i=1}^{N_-} [\mathcal{L}_{\text{cate}}(\Theta) + \mathcal{L}_c(\Phi)].\end{aligned}\quad (19)$$

In the testing phase, we use the category description P_{cate} as the basis for distinguishing the “object/background.” If the maximum probability of P_{cate} appears on the background class, that means the testing sample belongs to the

background; otherwise, the maximum probability of P_c is taken as the confidence of the sample, and the corresponding category is regarded as the recognition result.

3.4.4. Postprocessing Module. In the testing phase, this module filters the recognition and segmentation results of multiple test samples to obtain the final instance-aware semantic segmentation results. A fusion nonmaximum suppression algorithm is proposed to deal with repeated extraction of the same object. As described in Algorithm 1, we calculate the segmentation IoU of each pair of predictions with the same category. When the IoU is larger than a threshold (set to 0.2), we remove the prediction with lower confidence and fuse its segmentation into the other prediction.

4. Experiments

4.1. Datasets and Implementation Details. This section conducted a series of experiments on two public benchmarks, including CUB-2011 [10] and Pascal VOC [28].

- (i) *CUB-2011* [10]. Wah et al. [10] established a dataset with 11,788 images from 200 bird categories. Each bird is annotated with a 312-dimensional attribute distribution. The standard training/testing split (i.e., 5994/5794) is adopted. In the training phase, each image is resized into 224×224 , and the batch size is set to 64 for both SAL-Net and baseline models. All networks are trained 50,000 iterations with a learning rate of 0.001, and then 30,000 iterations with a learning rate of 0.0001, and finally 20,000 iterations with a learning rate of 0.00001.
- (ii) *Pascal VOC* [28]. The Pascal VOC dataset is adopted to evaluate the effectiveness of both SAL-Net and the proposed multiobject system. We use 1464 images for training and 1449 images for testing. In the training phase, 3508 groups of positive samples and 1464 negative samples have been used. The batch size is set to the number of samples from 4 training images, and each cropped sample is resized into 160×160 . The training for both SAL-Net and baseline models is performed in three stages. First, it performs 20,000 iterations with a learning rate of 0.001, then performs 12,000 iterations with a learning rate of 0.0001, and finally performs 8000 iterations with a learning rate of 0.00001. In order to verify the performance of SAL-Net, a bounding box is provided to extract each object in the testing image and use it as input. In order to verify the performance of the proposed system, the entire testing image is fed into the system.

We compute the average intersection over union (aIoU) among testing samples to evaluate the segmentation performance and the accuracy (Acc) to evaluate the classification performance. For instance-aware semantic segmentation, we calculate the mean average precision and recall among categories (mask AP and AR defined in [29]).

FIGURE 3: The proposed multiobject recognition and segmentation system pipeline. In the training phase, the input image is fed into the attribute generation module to obtain positive samples and the region proposal module to obtain negative samples. Then, the positive and negative samples are utilized to train the parameters of SAL-Net. In the testing phase, the input image is fed into the region proposal module to obtain testing samples, which are sent to the SAL-Net for prediction. Finally, the recognition and segmentation results are sent to the postprocessing module for filtering.

Input: Mask set M , category set C , confidence set P
Output: updated M , C and P

- [1] Sort M , C and P according to the value in P from large to small;
- [2] Initialize the number of predictions k .
- [3] **for** $m_i \in M, c_i \in C, p_i \in P$ **do**
- [4] **for** $j \in (i, k]$ **do**
- [5] **if** $c_i == c_j$ **then**
- [6] Calculate $IoU(m_i, m_j)$;
- [7] **if** $IoU > T_{iou}$ **then**
- [8] Fuse the prediction: $m_i = m_i \cup m_j$;
- [9] Remove the prediction: $M = M - m_j, C = C - c_j, P = P - p_j$;
- [10] $k = k - 1$;
- [11] **end if**
- [12] **end if**
- [13] **end for**
- [14] **end for**
- [15] **return** Updated M , C and P

ALGORITHM 1: The proposed fusion nonmaximum suppression algorithm (Fusion NMS).

4.2. *Comparative Methods.* We compare the following degenerated models with the proposed SAL-Net. All comparative models are based on the ResNet50 [14], which is pretrained on the ImageNet dataset [49].

- (i) *ST-cls (A Single-Task Model for Classification).* The output of Conv5_x is fed into a global pooling layer to obtain a 2048-dimensional feature vector, which is projected onto the category space with a fully connected layer.
- (ii) *ST-seg (A Single-Task Model for Segmentation).* The output of Conv5_x is fed into a 1×1 convolutional layer and a deconvolutional layer to obtain the foreground confidence map.

- (iii) *MT-SF (A Multitask Model with Single-Layer Features).* This is a two-branch multitask network, where the classification task and the segmentation task share the CNN backbone. The output of Conv5_x is fed into a classification branch and a segmentation branch, which have the same structure as ST-cls and ST-seg, respectively.
- (iv) *MT-MF (A Multitask Model with Multilayer Features).* The network has a similar structure to SAL-Net except for the auxiliary attribute. For the classification branch, the Conv5_x feature is fed into a global pooling layer to obtain a 2048-dimensional vector. For the segmentation branch, features from Conv3_x, Conv4_x, and Conv5_x are, respectively,

TABLE 1: Effectiveness of the proposed strategies, including multitask learning, features from multiple CNN layers, and joint learning with attributes.

Model	Strategies			CUB-2011 [10]		Pascal VOC [28]	
	Multitask	Multifeature	Attribute learning	Acc (%)	aIoU (%)	Acc (%)	aIoU (%)
ST-cls				64.67	—	67.46	—
ST-seg				—	79.76	—	73.81
MT-SF	✓			70.37	74.22	69.11	74.02
MT-MF	✓	✓		71.45	82.61	71.42	77.59
AFE-Net [27]	✓	✓	✓	74.37	85.07	—	—
SAL-Net (ours)	✓	✓	✓	77.55	86.66	75.00	78.19

fed into three segmentation projection models; each model contains a 1×1 convolutional layer and a deconvolutional layer. The outputs of the three segmentation models are fused into the final prediction.

- (v) *AFE-Net* [27] (*Our Previous Attribute-Aware Feature Encoding Network*). It projects the visual features onto attribute-wise feature maps before sending the features to the segmentation branch. At the same time, the classification branch adopts the strategy of smoothing the category space and combining attributes and category descriptions. Compared with SAL-Net, the only difference is that such a baseline applies supervision from manually annotated attributes, while SAL-Net is based on the proposed self-supervised attribute learning method.

4.3. Performance of SAL-Net

4.3.1. Comparisons with Baseline Models. The proposed SAL-Net is compared with baseline models in Table 1. Compared with the single-task model ST-cls, the multitask model MT-SF obtains better classification results. It is thanks to the segmentation supervision signal transferring the object contour information to the classification task. The extracted object region contains fewer background features' adverse effects to improve the stability of the classification model. However, MT-SF has limited advantages over the single-task model ST-seg in the segmentation task. The additional classification supervision motivates the CNN to extract semantic features rather than location features. The above problem can be solved by applying the multilayer feature strategy. Compared with MT-SF, MT-MF significantly improves the segmentation accuracy, which obtains a gain of 8.39% aIoU on the CUB-2011 dataset and 3.57% on the Pascal VOC dataset.

In our previous work [27], attribute knowledge is applied to constrain the feature extraction, which has made significant improvements in both classification and segmentation tasks on the Pascal VOC dataset. However, it is not suitable to Pascal VOC since there are no attribute annotations provided by the dataset. This paper proposes a self-learning attribute learning (SAL) method. Compared with MT-MF, SAL-Net produces a 4.05% aIoU gain and a 6.1% accuracy gain on the CUB-2011 dataset. On the Pascal VOC dataset,

TABLE 2: Comparison with state-of-the-art fine-grained recognition methods on CUB-2011. Extra annotations are provided during the training phase, including attribute (Att), bounding box (BBox), and part location (Part).

Methods	Annotations			Acc (%)
	Att	BBox	Part	
ResNet50 [14]				82.0
Bilinear-CNN [53]				84.1
AutoBD [54]				81.6
SAL-Net (ours)				84.6
Liu et al. [51]	✓			82.5
VTRL [52]	✓			86.3
CAP-Net [55]	✓	✓		82.5
A3M [4]	✓	✓		86.2
PBC [56]		✓	✓	83.7
BBPL [57]		✓	✓	87.6
Liu et al. [51]	✓	✓	✓	85.5

it obtains 0.6% aIoU gain and 3.55% accuracy gain. The reason is that SAL-Net introduces attribute relationships to constrain the feature extraction network and smooth the category space, which makes the extracted features more discriminative and robust. Meanwhile, concatenating the attribute features to the category description can further improve classification accuracy. Compared with [27], SAL-Net produces a segmentation gain of 1.59% aIoU and a classification gain of 3.18% accuracy on the CUB-2011 dataset. It can be explained by the advantages of latent attributes, where the attribute knowledge automatically mined from images are more suitable than manually annotated attributes for identifying and segmenting objects.

4.3.2. Comparisons with State-of-the-Art Methods. In Table 2, SAL-Net is compared with state-of-the-art fine-grained recognition methods on the CUB-2011 benchmark. A widespread data augmentation approach is applied, where each input image is resized into 512×512 and rotated with a random angle. The randomly cropped patch is resized and flipped into a 448×448 input image. Based on the data augmentation, the standard ResNet50 achieves high accuracy on the CUB-2011 testing set, while our SAL-Net improves the accuracy from 82.0% to 84.6%. Existing attribute-based methods [4, 27, 51, 52] also learn the attribute attention to

FIGURE 4: Classification and segmentation performance w.r.t. the number of shared 3×3 convolutional layers of the attribute feature coding module. The models are trained by randomly selecting 90% samples from the CUB-2011 training set and using the remaining 10% samples for validation.

localize discriminative regions. These methods rely on the pre-defined attributes or textual descriptions provided by the dataset. In contrast, our SAL-Net automatically exploits latent attributes based on the self-supervised learning method, which does not need any extra annotations.

4.3.3. Effects of the Structure of Attribute Feature Coding Module. Figure 4 illustrates the classification and segmentation performance by using different attribute feature coding modules. The horizontal axis represents the number of shared 3×3 convolutional layers, which encode CNN features as K_a -dimensional attribute-wise feature maps. We can see that the performance of using one convolutional layer is relatively better. As the number of hidden layers increases, the supervision of the attributes imposes fewer constraints on the feature extraction. The result indirectly verifies the effectiveness of the proposed self-supervised attribute learning method for representation learning.

4.3.4. Effects of the Attribute Dimension. Figure 5 compares the classification and segmentation results by using different attribute dimensions. We can see that the change of the attribute dimension K_a has a limited effect on the segmentation task. The reason is that SAL-Net mainly refines the segmentation through the localization of latent attributes. When K_a is small, the area of each latent attribute is learned to be large, and when K_a is large, the area of each latent attribute is learned to be small. Thus, the fused segmentation result is stable. For the classification task, the network with a small K_a cannot learn enough attributes to describe the object. On the contrary, an excessively large K_a will weaken the category knowledge in the concatenated feature and decrease the recognition accuracy. In the Pascal VOC dataset, there

FIGURE 5: Classification and segmentation performance on CUB-2011 validation set w.r.t. the dimension of attributes.

FIGURE 6: Classification and segmentation performance on CUB-2011 validation set w.r.t. the minimum margin between two attribute distributions (α in Equations (3) and (5)).

are remarkable visual differences among the 20 categories. There are also noticeable visual differences between individuals from a common category, resulting in the entire dataset with diversified characteristics. Therefore, we select the value of $K_a = 1000$ to learn as many latent attributes as possible. In the fine-grained CUB-2011 dataset, 200 categories share attributes, and birds from a common category contain almost the same attribute distribution. Therefore, we choose $K_a = 400$, which is 88 higher than the predefined attributes provided by the dataset.

4.3.5. Effects of the Attribute Margin. Figure 6 illustrates the influence of the minimum margin α in Equation (3). We can see that when α is small, the occluded sample is closer to the original sample in the attribute space, which reduces the discrimination of the representation. When α is large, the attribute features of the occluded sample and the original sample

FIGURE 7: The visualization of each module in the proposed multitarget recognition and segmentation system: (a) region proposals generated by selective search algorithm; (b) SAL-Net predictions filtered by score threshold 0.6; (c) results processed by Fusion NMS.

are quite different, making it difficult for the concatenated features to be classified into the same category. Therefore, an appropriate $\alpha = 1$ is selected to learn discriminative attributes and ensure their occlusion invariance to category labels.

4.4. Performance of Instance-Aware Semantic Segmentation. Figure 7 shows the output of each module in the proposed multiobject recognition and segmentation system. It can be seen from Figure 7(a) that the region proposal module based on the selective search [50] can obtain multiple target candidates concentrated in the object region. Figure 7(b) illustrates the classification and segmentation results of SAL-Net. Each candidate region is fed into the network to obtain a binary segmentation mask and category confidence. Some background regions are filtered out with a threshold of 0.6 on the confidence score. However, there are still some

multiple overlapping results for the same object. The proposed Fusion NMS approach is applied to process the above predictions. In Figure 7(c), we can see that most overlapping areas are filtered out, and the segmentation result obtained after fusion is more accurate.

Table 3 further demonstrates the effectiveness of the proposed Fusion NMS method. We can see that the baseline result without postprocessing has a higher recall and a lower precision. It is because many overlapping SAL-Net predictions are retained (as shown in Figure 7(b)). Mask NMS [58] can filter out overlapping results with lower scores, while the performance is sensitive to the accuracy of each candidate box. Based on the coarse boxes extracted by the SS algorithm, Mask NMS cannot obtain effective improvement. Matrix NMS [59] updates the confidence of all predictions instead of directly filtering out those low-score predictions, while its gain of AR and AP is limited. The

TABLE 3: Effectiveness of the proposed Fusion NMS on Pascal VOC.

Module	AP	AP ₅₀	AP ₇₅	AP _L	AR _L	AR ₁	AR ₁₀	AR ₁₀₀
Without NMS (baseline)	2.1	6.0	0.8	2.9	20.0	6.6	14.0	14.7
Mask NMS [58]	2.5	8.2	0.7	3.5	11.1	6.6	8.4	8.4
Matrix NMS [59]	2.8	9.2	0.8	3.8	14.7	6.3	10.4	10.9
Fusion NMS (ours)	7.7	19.1	5.5	10.7	20.9	13.6	15.6	15.6

proposed Fusion NMS method is an advancement of Mask NMS [58]. We filter out the overlapping prediction with a lower score while fusing its segmentation into the retained prediction, making the final segmentation more accurate. As shown in Figure 7(c), even if the retained box does not contain the entire object, the segmentation after fusion is still satisfactory.

5. Conclusion

This paper proposes a self-supervised attribute learning method, which automatically generates latent attribute descriptions for training. Based on the proposed method, we design a multitask network SAL-Net, in which attributes constrain the feature coding network to learn more discriminative representations for recognition and segmentation. Experimental results on CUB-2011 and Pascal VOC datasets illustrate that the proposed SAL-Net can significantly improve object recognition and segmentation accuracy without extra annotations. It achieves state-of-the-art fine-grained recognition performance on CUB-2011, superior to existing methods based on predefined attributes. In addition, we naturally embed SAL-Net into a multiobject system to achieve instance-aware semantic segmentation. Qualitative and quantitative experiments on Pascal VOC verify the effectiveness of each module in the system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant No.: 61620106002) and the Central Government Guides Local Science and Technology Development Fund Project of China (Grant No.: 2021ZY0004).

References

- [1] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Advances in Neural Information Processing Systems*, pp. 433–440, MIT Press, 2007.
- [2] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3474–3481, Providence, RI, USA, June 2012.
- [3] Y. Lu, L. Yang, K. Yang, and Y. Rui, "Mining latent attributes from click-through logs for image recognition," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1213–1224, 2015.
- [4] K. Han, J. Guo, C. Zhang, and M. Zhu, "Attribute-aware attention model for fine-grained representation learning," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 2040–2048, October 2018.
- [5] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, Miami, FL, USA, June 2009.
- [6] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425–1438, 2015.
- [7] B. Demirel, R. G. Cinbis, and N. Ikizler-Cinbis, "Attribute-s2Classname: a discriminative model for attribute-based unsupervised zero-shot learning," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1241–1250, Venice, Italy, October 2017.
- [8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, Miami, FL, USA, June 2009.
- [9] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Pedestrian attribute recognition at far distance," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 789–792, Orlando, USA, November 2014.
- [10] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, *The Caltech-UCSD Birds-200-2011 Dataset*, Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [11] G. Patterson and J. Hays, "COCO attributes: Attributes for people, animals, and objects," in *European Conference on Computer Vision*, pp. 85–100, Springer, 2016.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [13] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, June 2015.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

- [16] Y. Xing, L. Zhong, and X. Zhong, "An encoder-decoder network based FCN architecture for semantic segmentation," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8861886, 9 pages, 2020.
- [17] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6689134, 16 pages, 2020.
- [18] M. Yan, Z. Li, X. Yu, and C. Jin, "An end-to-end deep learning network for 3D object detection from RGB-D data based on hough voting," *IEEE Access*, vol. 8, pp. 138810–138822, 2020.
- [19] C. Jin, T. Zhang, S. Liu, and T. Yun, "Cross-modal deep learning applications: audio-visual retrieval," in *International Conference on Pattern Recognition Workshops*, pp. 301–313, Paris, France, 2021.
- [20] C. Jin, Y. Tie, Y. Bai, X. Lv, and S. Liu, "A style-specific music composition neural network," *Neural Processing Letters*, vol. 52, no. 3, pp. 1893–1912, 2020.
- [21] C. Jin, T. Wang, S. Liu et al., "A transformer-based model for multi-track music generation," *International Journal of Multimedia Data Engineering and Management*, vol. 11, no. 3, pp. 36–54, 2020.
- [22] C. Jin, W. Zhao, and H. Wang, "Research on objective evaluation of recording audio restoration based on deep learning network," *Advances in Multimedia*, vol. 2018, Article ID 3748141, 13 pages, 2018.
- [23] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1422–1430, Santiago, Chile, December 2015.
- [24] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision*, pp. 649–666, Springer, 2016.
- [25] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, Vancouver, Canada, 2018.
- [26] H. Lee, S. J. Hwang, and J. Shin, "Rethinking data augmentation: self-supervision and self-distillation," in *International Conference on Learning Representations*, Vienna, Austria, 2020.
- [27] S. Yang, Y. Wang, K. Chen, W. Zeng, and Z. Fei, "Attribute-aware feature encoding for object recognition and segmentation," *IEEE Transactions on Multimedia*, p. 1, 2021.
- [28] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [29] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, Springer, 2014.
- [30] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962–1977, 2011.
- [31] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: a deep multi-task learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2597–2609, 2018.
- [32] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer, "Facial attributes classification using multi-task representation learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 752–760, Las Vegas, NV, USA, June 2016.
- [33] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2019.
- [34] S. Rui, H. Qiheng, F. Wei, and Z. Xudong, "Attributes-based person re-identification via CNNs with coupled clusters loss," *Journal of Systems Engineering and Electronics*, vol. 31, no. 1, pp. 45–55, 2020.
- [35] Z. Shi, T. M. Hospedales, and T. Xiang, "Transferring a semantic representation for person reidentification and search," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4184–4193, Boston, MA, USA, June 2015.
- [36] S. Chi, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *European Conference on Computer Vision*, pp. 475–491, Springer, 2016.
- [37] Y. Lin, L. Zheng, Z. Zheng et al., "Improving person re-identification by attribute and identity learning," *Pattern Recognition*, vol. 95, pp. 151–161, 2019.
- [38] J. Zhang, L. Lin, J. Zhu, Y. Hu, and S. C. H. Hoi, "CSID: center, scale, identity and density-aware pedestrian detection in a crowd," 2019, <https://arxiv.org/abs/1910.09188>.
- [39] S. Zheng, M.-M. Cheng, J. Warrell et al., "Dense semantic image segmentation with objects and attributes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3214–3221, Columbus, OH, USA, June 2014.
- [40] Z. Shi, Y. Yang, T. M. Hospedales, and T. Xiang, "Weakly-supervised image annotation and segmentation with objects and attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2525–2538, 2017.
- [41] M. D. Sulistiyono, Y. Kawanishi, D. Deguchi et al., "Attribute-aware semantic segmentation of road scenes for understanding pedestrian orientations," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2698–2703, Maui, HI, USA, November 2018.
- [42] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Learning multimodal latent attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 303–316, 2014.
- [43] P. Peng, Y. Tian, T. Xiang et al., "Joint semantic and latent attribute modelling for cross-class transfer learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1625–1638, 2017.
- [44] Y. Li, J. Zhang, J. Zhang, and K. Huang, "Discriminative learning of latent features for zero-shot recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7463–7471, Salt Lake City, UT, USA, June 2018.
- [45] Y. Xie, X. He, J. Zhang, and X. Luo, "Zero-shot recognition with latent visual attributes learning," *Multimedia Tools and Applications*, vol. 79, no. 37–38, pp. 27321–27335, 2020.
- [46] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, Kerkyra, Greece, September 1999.
- [47] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference*

- on *Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, June 2005.
- [48] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: speeded up robust features,” in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.
 - [49] O. Russakovsky, J. Deng, H. Su et al., “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
 - [50] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
 - [51] X. Liu, J. Wang, S. Wen, E. Ding, and Y. Lin, “Localizing by describing: attribute guided attention localization for fine-grained recognition,” in *AAAI Conference on Artificial Intelligence*, pp. 4190–4196, San Francisco, USA, 2017.
 - [52] X. He and Y. Peng, “Fine-grained visual-textual representation learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 520–531, 2020.
 - [53] T. Y. Lin, A. Roychowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1449–1457, Santiago, Chile, December 2015.
 - [54] H. Yao, S. Zhang, C. Yan, Y. Zhang, J. Li, and Q. Tian, “Autobd: automated bi-level description for scalable fine-grained visual categorization,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 10–23, 2018.
 - [55] S. Yang, Y. Wang, Y. Shi, and Z. Fei, “Can categories and attributes be learned in a multi-task way?,” *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3194–3204, 2019.
 - [56] C. Huang, H. Li, Y. Xie, Q. Wu, and B. Luo, “Pbc: polygon-based classifier for fine-grained categorization,” *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 673–684, 2017.
 - [57] J. Chen, J. Hu, and S. Li, “Learning to locate for fine-grained image recognition,” *Computer Vision and Image Understanding*, vol. 206, p. 103184, 2021.
 - [58] Y. Dai, H. Zheng, Y. Gao et al., “Fused text segmentation networks for multi-oriented scene text detection,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3604–3609, Beijing, China, August 2018.
 - [59] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: dynamic, faster and stronger,” 2020, <https://arxiv.org/abs/2003.10152>.