International Journal of Computer Vision (2021) 129:2731–2744
https://doi.org/10.1007/s11263-021-01499-z

Cascaded Split-and-Aggregate Learning with Feature Recombination for Pedestrian Attribute Recognition

Yang Yang[1,2] · Zichang Tan[4,5] · Prayag Tiwari[6] · Hari Mohan Pandey[7] · Jun Wan[1,2] · Zhen Lei[1,2,3] · Guodong Guo[4,5] · Stan Z. Li[1]

Received: 27 June 2020 / Accepted: 5 June 2021 / Published online: 18 July 2021

Abstract
Multi-label pedestrian attribute recognition in surveillance is inherently a challenging task due to poor imaging quality, large pose variations, and so on. In this paper, we improve its performance from the following two aspects: (1) We propose a cascaded Split-and-Aggregate Learning (SAL) to capture both the individuality and commonality for all attributes, with one at the feature map level and the other at the feature vector level. For the former, we split the features of each attribute by using a designed attribute-specific attention module (ASAM). For the later, the split features for each attribute are learned by using constrained losses. In both modules, the split features are aggregated by using several convolutional or fully connected layers. (2) We propose a Feature Recombination (FR) that conducts a random shuffle based on the split features over a batch of samples to synthesize more training samples, which spans the potential samples' variability. To the end, we formulate a unified framework, named CAScaded Split-and-Aggregate Learning with Feature Recombination (CAS-SAL-FR), to learn the above modules jointly and concurrently. Experiments on five popular benchmarks, including RAP, PA-100K, PETA, Market-1501 and Duke attribute datasets, show the proposed CAS-SAL-FR achieves new state-of-the-art performance.

Keywords Pedestrian attribute recognition · Attention · Split-and-aggregate learning · Feature recombination

1 Introduction

Visual analysis of pedestrian attributes (Wang et al. 2017; Sarfraz et al. 2017; Lin et al. 2019; Liu et al. 2017; Zhao et al. 2018; Xiang et al. 2019; Tan et al. 2019b; Han et al. 2019; Li et al. 2019a, b; Tang et al. 2019c), e.g., gender, age and

Communicated by Dima Damen.

Yang Yang and Zichang Tan contribute equally to this work.

✉ Jun Wan
jun.wan@nlpr.ia.ac.cn

Yang Yang
yang.yang@nlpr.ia.ac.cn

Zichang Tan
tanzichang@baidu.com

Prayag Tiwari
prayag.tiwari@aalto.fi

Hari Mohan Pandey
pandeyh@edgehill.ac.uk

Zhen Lei
zlei@nlpr.ia.ac.cn

Guodong Guo
guoguodong01@baidu.com

Stan Z. Li
szli@nlpr.ia.ac.cn

[1] Center for Biometrics and Security Research, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing, China

[3] Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hongkong, China

[4] Institute of Deep Learning, Baidu Research, Beijing, China

[5] National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China

[6] Department of Computer Science, Aalto University, Espoo, Finland

[7] Department of Computer Science, Edge Hill University, Ormskirk, UK

hair style, has recently received increasing attention due to its potential applications in surveillance and security applications. Although the performance has been greatly improved owing to the success of deep learning (Krizhevsky et al. 2012; Simonyan and Zisserman 2015; He et al. 2016, 2017; Huang et al. 2017; Hu et al. 2018a; Liu et al. 2020; Zhu et al. 2019; Shifeng et al. 2019) especially the Convolutional Neural Network (CNN) (LeCun et al. 1998), accurate recognition of pedestrian attributes remains a challenging task because of poor imaging quality (e.g., low resolution and motion blur), complex variations (e.g., arbitrary human poses, various camera viewing angles, and background), small training datasets and so on.

For multi-label pedestrian attribute classification, most previous works (Wang et al. 2017; Sarfraz et al. 2017; Lin et al. 2019; Liu et al. 2017; Li et al. 2019a, b) employ simple multi-task learning (MTL) framework to analyze all attributes together with a shared feature extractor. Such a shared strategy may prefer to capture the common and general features for all attribute (commonality), while the specific semantics of each attribute may be less involved (individuality). In other words, the commonality of all attributes may be overemphasized while the individuality of each attribute may be ignored. Actually, for multi-label pedestrian attribute recognition, different attributes are often related with different body regions and semantics. For example, we mainly look at the head region to recognize a pedestrian's hairstyle while the upper body region is often used to judge the upper clothing style. Moreover, color information often determines a pedestrian's shoe color, while the texture and shape features are essential to identify the type of shoes. Therefore, learning the individuality of each attribute is also very necessary, which ensures each attribute can learn their own semantics.

In this paper, we propose a *split-and-aggregate learning (SAL)* to learn both individuality and commonality among pedestrian attributes. The features of each attribute are firstly split out to capture the individualities for all attributes and then aggregated together by using several CNN layers to exploit their commonalities and relations. To fully capture the individuality and commonality, we formulate two SALs in a cascaded way, with one at the feature map level and the other at the feature vector level. For the former, an Attribute-Specific Attention Module (ASAM), which assigns each attribute with several attention maps, is designed to capture the features of an attribute from most relevant body regions. ASAM has been implemented at different feature levels to access abundant semantic information. For the later, we learn the attribute-specific features by using the constrained losses with each loss corresponding to several neurons.

Moreover, we further propose a new feature recombination operation to synthesize new representations. The key idea is to recombine the components of different attributes, which is different from the previous works of creating new samples via generative models (Goodfellow et al. 2014; Kingma and Welling 2014; Zheng et al. 2017; Fu et al. 2019). The extracted features of a sample can be regarded as a combination of the split features of all attributes, with each denoting the semantics of a specific attributes. By shuffling the split features over a batch of samples, new synthetic representations with different attribute semantics can be achieved. In recombination stage, our method keeps the integrity and semantics for each attribute-specific feature while spanning the potential samples' variability. To our best knowledge, it is the first attempt to synthesize new samples at feature level for pedestrian attribute recognition.

The main contributions of our work are as follows: (1) We propose a novel unified framework with a cascaded split-and-aggregate features learning to capture both individuality and commonality among pedestrian attributes. (2) We propose a new feature recombination operation to synthesize new representations. (3) We propose a novel attribute-specific attention module, which can capture the features from the most important regions/pixels for each attribute. (4) We conduct extensive experiments on five popular pedestrian attribute benchmarks including RAP, PA-100K, PETA, Market-1501 and Duke attribute datasets, which shows the proposed method achieves the new state-of-the-art performance.

2 Related Works

2.1 Pedestrian Attribute Recognition

Earlier methods of pedestrian attribute recognition (Deng et al. 2014; Zhu et al. 2013) typically model each attribute independently based on the hand-crafted features like color and texture histograms. Recently, owing to the great successes of deep learning (Simonyan and Zisserman 2015; He et al. 2016), many approaches based on deep networks also have been developed for pedestrian attribute recognition (Wang et al. 2017; Sarfraz et al. 2017; Liu et al. 2018b; Zhao et al. 2019; Li et al. 2019a, b; Tan et al. 2019b). Previous works mainly solve the task of pedestrian attribute recognition from following aspects: (1) constructing attention mechanisms to capture discriminative features (Liu et al. 2017; Sarafianos et al. 2018; Zhao et al. 2019; Tan et al. 2019b; Tang et al. 2019c); (2) formulating a part-based model by using human poses (Liu et al. 2018b; Li et al. 2018a; Zhao et al. 2018) or Spatial Transformer Networks (STN) (Tang et al. 2019c); (3) exploiting the relations among attributes or image regions (Wang et al. 2017; Zhao et al. 2019); and (4) dealing with the imbalance data problem (Sarafianos et al. 2018; Wang et al. 2019). Most of the previous works construct their models based on the multi-task learning (MTL) frame-

work, while the traditional MTL framework usually prefers to learn the commonality of all attributes while ignoring the individuality of each attribute. In this work, we aim to capture both the individuality and commonality of all attributes, where cascaded split-and-aggregate learning is proposed to achieve this. The proposed method is also different from the work (Tang et al. 2019c), which aims to select the most important regions for each attribute. Our work not only considers to capture the discriminative features for each attribute, but also considers how to aggregate those split features with capturing the commonalities and relations among those attributes. The split-and-aggregate learning is considered to be implemented at different levels, and then learns them jointly and concurrently.

2.2 Attention Mechanism

Attention models (Hu et al. 2018a; Fu et al. 2017; Li et al. 2018c; Woo et al. 2018; Liu et al. 2017; Tan et al. 2019b; Guo et al. 2019; Chen et al. 2019; Shuzhe et al. 2019; Xiangyu et al. 2020) have aroused great interests in recent years. Hu et al. (2018a) propose Squeeze-and-Excitation Networks with recalibrating the channel-wise responses by using a channels attention. Li et al. (2018c) jointly learn both soft pixel attention and hard regional attention for person re-identification. Woo et al. (2018) formulate an attention mechanism by sequentially extracting the discriminative features at channel and spatial dimensions. Moreover, Chen et al. (2019) propose a high-order attention to model and utilize the complex and high-order statistics information. Inspired by those works, we also propose an attribute-specific attention module to select important regions/pixels for each attribute.

2.3 Augmenting Training Samples

Previous methods of augmenting samples can be classified to following categories: (1) basic image manipulations, like flipping, translating, adding noises, random erasing (Zhong et al. 2020), mixup (Hongyi et al. 2018) and so on, (2) synthesizing new samples by using generative models (Goodfellow et al. 2014; Kingma and Welling 2014; Zheng et al. 2017; Fu et al. 2019). For example, Zheng et al. (2017) generate the synthetic samples with GAN in person re-identification, (3) borrowing the samples from relevant categories (Lim et al. 2011; Tan et al. 2018). For example, Lim et al. (2011) augment data of the classes with few samples by borrowing and transforming examples from other classes, and (4) feature space transfer (Dixit et al. 2017; Liu et al. 2018a). In the above methods, mixup (Hongyi et al. 2018) is somewhat related with our proposed FR, where both our FR and mixup augment training samples by using combinations of different samples and their labels. However, there are still some crucial differences between our proposed FR and mixup. For mixup, it employs a liner combination of a pair of images and their labels to generate new samples. For our FR, it first obtains the split features (separating the features of each attribute), and then just uses a random shuffle over a batch of samples to generate new samples, which keeps the semantic information of each attribute unchanged.

2.4 Split-and-Aggregate Learning

Some previous works (Zhang et al. 2020; Tan et al. 2019a; Szegedy et al. 2015, 2016, 2017) also adopt the idea of split-and-aggregate learning to capture more discriminative features. For example, Zhang et al. (2020) propose a Split-Attention Block, which splits the features into several groups and learns them individually. Then, the split features of all groups are aggregated together by using a concatenation. We further use a recurrent fusion to aggregate those branches together. Moreover, inception block also can be regarded as a special case of split-and-aggregate learning (Szegedy et al. 2015, 2016, 2017), where the input features are split by using several different CNN branches and each one learns different features from each other. Then, the block finally aggregates all features together to form more comprehensive features. Our work conducts the split-and-aggregate learning from a different aspect. In the split stage, we first split the attribute-specific features out and capture the individuality for each task/attribute and then aggregate those split features together to learn the commonality among all attributes.

3 Proposed Method

To construct a deep model $\mathcal{M}$ for pedestrian attribute recognition, we assume the available training set contains n images and is denoted as $\mathcal{D} = \{\mathbf{I}_i\}_{i=1}^n$, with corresponding labels $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^n$. Each pedestrian is annotated with m attributes. For the ith image $\mathbf{I}_i$, the corresponding image-level annotation is denoted as $\mathbf{y}_i = [y_{i1}, y_{i2}, \cdots, y_{im}]$, where y_{ij} represents the label of the jth attribute. In the following, we will introduce the proposed network, namely CAScaded Split-and-Aggregate Learning with Feature Recombination (CAS-SAL-FR).

3.1 Network Architecture Design

The network architecture of the proposed CAS-SAL-FR is illustrated in Fig. 1. It adopts ResNet-50 (He et al. 2016) as the backbone to extract the features of different levels [the backbone also can be replaced with any other CNN architecture, e.g., GoogleNet (Szegedy et al. 2015) and DenseNet (Huang et al. 2017)]. For convenience, we denote the features after `res3d`, `res4f` and `res5c` blocks as $\mathcal{F}_1(\mathbf{I}_i)$, $\mathcal{F}_2(\mathbf{I}_i)$ and $\mathcal{F}_3(\mathbf{I}_i)$, respectively. If we adopt a 256×128 image as

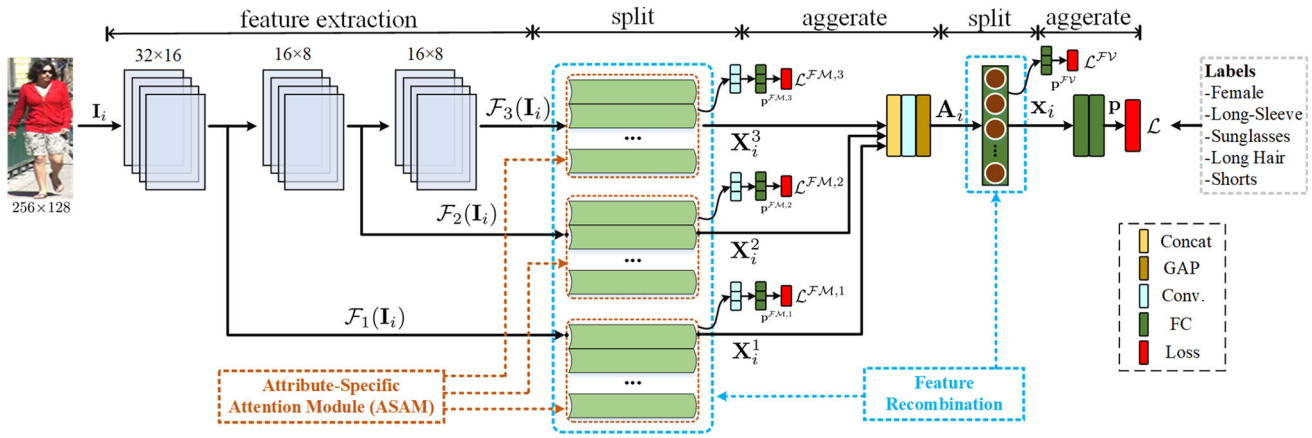

Fig. 1 An overview of the proposed CAS-SAL-FR. Two Split-and-Aggregate Learning (SAL) modules are sequentially applied on the feature maps level and feature vector level, where the split operation mainly learns the attribute-specific features for each attribute while the aggregate operation exploits the commonalities and relations among multiple attributes by learning how to aggregate them together. Specifically, the split operation at feature maps level is achieved by an Attribute-Specific Attention Module (ASAM). Moreover, a Feature Recombination (FR) strategy is adopted to synthesize new samples by shuffling the split features

the input, $\mathcal{F}_1(\mathbf{I}_i)$, $\mathcal{F}_2(\mathbf{I}_i)$ and $\mathcal{F}_3(\mathbf{I}_i)$ would have the size of 32×16, 16×8 and 8×4, respectively. However, the resolution of 8×4 is too low, which may hardly contain enough information for all attributes. Therefore, we change the stride of the final residual block from 2 to 1. In this way, the size of $\mathcal{F}_3(\mathbf{I}_i)$ would be 16×8. After extracting those features, two SALs are formulated in a cascaded way to learn both the individuality and commonality for all attributes, which will be introduced in the following.

(I) SAL at Feature Map Level After obtaining $\mathcal{F}_1(\mathbf{I}_i)$, $\mathcal{F}_2(\mathbf{I}_i)$ and $\mathcal{F}_3(\mathbf{I}_i)$, each of them is followed by an ASAM to select some important pixels or regions for each attribute and capture the attribute-specific features (also re-called as the split features). For the employed ASAM, its detailed structure is illustrated in Fig. 2. It contains multiple sub-networks, each of which extracts the most relevant features for a specific attribute. More specifically, each sub-network has two streams, with one generating attention masks and the other extracting high-level features. For clarity, we take the sub-network for the jth attribute at the κth level as an example ($\kappa \in \{1, 2, 3\}$). For the upper stream, the attention masks $\mathbf{M}_{ij}^{\kappa}$ are generated by using several convolutional layers and a softmax function (the softmax function is applied to the spatial dimension including height and width). Inspired by the works (Chen et al. 2017; Yu and Koltun 2016), we use the spatial pyramid convolutional layers with different receptive fields to capture abundant semantics. For the lower stream, it only contains two convolutional layers to extract high-level features $\mathbf{H}_{ij}^{\kappa}$. To reduce the number of parameters, the first convolutional layer in both two streams is shared for all attributes. Finally, the attentive features $\mathbf{X}_{ij}^{\kappa}$ for the jth attribute are generated by an element-by-element multiplication between the attention masks $\mathbf{M}_{ij}^{\kappa}$ and high-level features $\mathbf{H}_{ij}^{\kappa}$. The whole process can be denoted as:

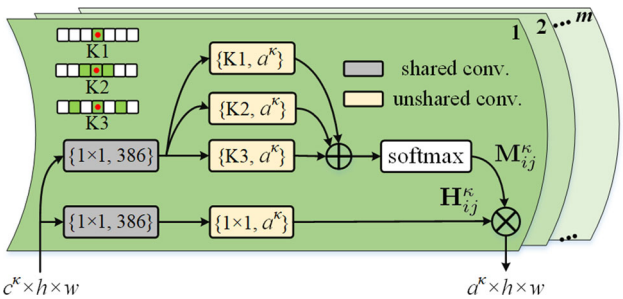

Fig. 2 Illustration of the proposed ASAM. For each convolutional layer, $\{K, C\}$ indicates its employed kernel size and output channels, respectively. In the spatial pyramid convolutional layers, K1, K2 and K3 represent the kernel size of 1×1, 3×3, and 3×3 with a dilatation rate of 2. We share the first convolutional layer in both two streams for all sub-networks, which aims to reduce the parameters of the network. The number of input channels c^1, c^2 and c^3 are 512, 1024 and 2048, respectively

$$\mathbf{X}_{ij}^{\kappa} = \mathcal{S}_j^{\mathcal{FM},\kappa}(\mathcal{F}_{\kappa}(\mathbf{I}_i)) \tag{1}$$

where $\mathcal{FM}$ indicates the feature map level and $\mathcal{S}_j^{\mathcal{FM},\kappa}$ represents the jth subnetwork in the ASAM at the κth level feature. In our implementations, we let $\mathbf{X}_{ij}^{\kappa}$ has the same shape of $a^{\kappa} \times h^{\kappa} \times w^{\kappa}$ (the number of channels, height and width, respectively), with each a^{κ} channels capturing the discriminative features for a specific attribute.

Till now, we have only allocated several attentive maps for each attribute, without forcing them to capture the information only from that attribute. To achieve this, additional constrained networks should be added. For each attribute-

specific feature map $\mathbf{X}_{ij}^{\kappa}$, it would be followed by a small constrained network to produce the corresponding predicted score for the jth attribute, which can be mathematically denoted as:

$$p_{ij}^{\mathcal{FM},\kappa} = \phi_j^{\mathcal{FM},\kappa}\left(\mathbf{X}_{ij}^{\kappa}\right) \tag{2}$$

where $\phi_j^{\mathcal{FM},\kappa}(\cdot)$ indicates a constrained network with a convolutional layer, a fully connected layer and a sigmoid function as shown in Fig. 1. In this way, the predicted score $p_{ij}^{\mathcal{FM},\kappa}$ is generated only from the features $\mathbf{X}_{ij}^{\kappa}$, which ensures its learning under the label supervision of the jth attribute. For our proposed ASAM, it is different from Squeeze-and-Excitation Networks (SE-Net) (Hu et al. 2018a) in following aspects: (1) SE-net can be considered as a channel attention which remodulates neurons' responses by a channel-wise mask, while our ASAM is constructed based on a spatial attention mechanism; (2) Our ASAM aims to capture attribute-specific attention features with considering each attribute individually.

After the splitting stage, we aggregate those split features by using a concatenation layer, a convolutional layer and a Global Average Pooling (GAP) as shown in Fig. 1. Mathematically, we denote such aggregation as $\mathcal{G}^{\mathcal{FM}}(\cdot)$, and thus the aggregated features are obtained by:

$$\mathbf{A}_i = \mathcal{G}^{\mathcal{FM}}(\mathbf{X}_i^1, \mathbf{X}_i^2, \mathbf{X}_i^3) \tag{3}$$

where $\mathbf{X}_i^{\kappa}$ means $\{\mathbf{X}_{ij}^{\kappa}\}_{j=1}^{m}$. In the aggregating process, the network learns how to aggregate the features from different attributes, it also learns the commonalities and relations among those attributes.

(II) SAL at Feature Vector Level At feature vector level, a SAL is further employed as shown in Fig. 1. After the GAP layer in the former aggregation module, m fully connected layers are employed to extract the split features with each layer corresponding to an attribute. We use $\mathcal{S}_j^{\mathcal{FV}}$ to denote a fully connected layer for the jth attribute, and then the split features can be produced by:

$$\mathbf{x}_{ij} = \mathcal{S}_j^{\mathcal{FV}}(\mathbf{A}_i)\,. \tag{4}$$

where $\mathcal{FV}$ indicates the feature vector level. Similarly, to ensure $\mathbf{x}_{ij}$ only captures the information for the jth attribute, those features are then followed by a fully connected layer to generate the predicted score $p_{ij}^{\mathcal{FV}}$ [the generated process is similar to Eq. (2)], which is optimized by the label supervision of the jth attribute.

Then, two fully connected layers are further employed to aggregate those split features together. It helps to exploit the commonalities and relations among those attributes. Different from the previous aggregating module that outputs the aggregated features, this module directly generates the predicted scores $\mathbf{p}_i$ for final attribute predictions, which is represented as:

$$\mathbf{p}_i = \mathcal{G}^{\mathcal{FV}}(\mathbf{x}_i) \tag{5}$$

where $\mathbf{x}_i = [\mathbf{x}_{i1}, \cdots, \mathbf{x}_{im}]$ denotes the feature vector for all attributes, and $\mathcal{G}^{\mathcal{FV}}$ indicates two fully connected layers.

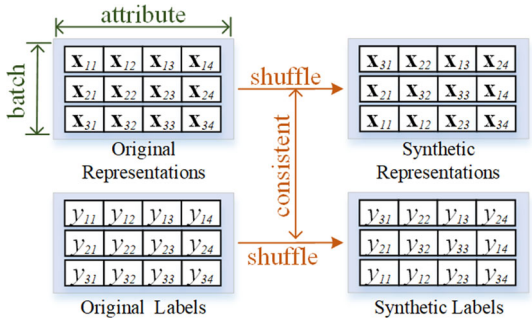

Fig. 3 An illustration of the feature recombination. Each row indicates a feature vector or label vector for a sample, and each column indicate the features or labels from a specific attribute over a batch of samples

Remark Our cascaded split-and-aggregated learning is different from previous works (Tan et al. 2019b; Tang et al. 2019c): (1) The work Tang et al. (2019c) extracts the attribute-specific features by using a hard regional attention while we our ASAM uses a soft scheme of assigning each attribute with several attention maps to explore more information. (2) Previous works (Tan et al. 2019b; Tang et al. 2019c) do not consider the attribute-specific features learning in feature vector level, while we achieve this by using a simple but elegant structure (constrained loss). (3) Two SALs are then formulated in a cascaded way to access more effective features.

3.2 Feature Recombination (FR)

The proposed FR aims to synthesize new samples by shuffling the split features over a batch of samples. Figure 3 illustrates a simple example of FR on a batch of 3 samples, and each sample is annotated with 4 attributes. In the proposed FR, a random shuffle is conducted at the batch level, where the split features from different samples will be recombined to be new samples. The labels of those new synthetic samples are obtained by a consistent shuffle on the original labels. For example, the new corresponding label vector for the synthetic sample $[\mathbf{x}_{31}, \mathbf{x}_{22}, \mathbf{x}_{13}, \mathbf{x}_{24}]$ is $[y_{31}, y_{22}, y_{13}, y_{24}]$. The attribute-specific features $\mathbf{x}_{ij}$ contains the semantics of indicating the absence or presence of the corresponding attribute. After feature recombination, the value of $\mathbf{x}_{ij}$ remains unchanged, where semantic information will be maintained. Thus, the corresponding label of the new

synthetic sample for the jth attribute also will be consistent with the original label y_{ij}.

Assume the batch size we used in the training stage is set as n^{bs}. For the convenience in the following, we remove the subscript j in the sign of features to denote the features over a batch of samples. For example, the split features over a batch can be denoted as $\mathbf{X}^{\kappa} = [\mathbf{X}_1^{\kappa}, \cdots, \mathbf{X}_{n^{bs}}^{\kappa}]$ and $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_{n^{bs}}]$. Their corresponding labels also can be denoted as $\mathbf{y} = [\mathbf{y}_1, \cdots, \mathbf{y}_{n^{bs}}]$. We denote the random shuffle operation as $\mathcal{R}(\cdot)$, and thus the recombined split features at feature map level and their corresponding labels can be generated as:

$$\mathbf{X}^{\kappa,\mathcal{R}_1} = \mathcal{R}(\mathbf{X}^{\kappa}), \quad \mathbf{y}^{\mathcal{R}_1} = \mathcal{R}(\mathbf{y}). \tag{6}$$

All of $\{\mathbf{X}^{\kappa,\mathcal{R}_1}\}_{\kappa=1}^{3}$ should employ a consistent shuffle to ensure the semantic consistency in the later aggregating stage, and thus we use the same superscript $\mathcal{R}_1$ to indicate the consistent shuffle among them, and such superscript would be used in a similar way in the following section. Later, $\mathbf{X}^{\kappa,\mathcal{R}_1}$ is further used for inference to generate the predicted scores $\mathbf{p}^{\mathcal{R}_1}$. The inference details can refer to Eqs. (3), (4) and (5). Similarly, the random shuffle also can be conducted on $\mathbf{x}$, and thus the corresponding recombined features $\mathbf{x}^{\mathcal{R}_2}$ and the corresponding new labels $\mathbf{y}^{\mathcal{R}_2}$ can be obtained by:

$$\mathbf{x}^{\mathcal{R}_2} = \mathcal{R}(\mathbf{x}), \quad \mathbf{y}^{\mathcal{R}_2} = \mathcal{R}(\mathbf{y}). \tag{7}$$

The predicted scores $\mathbf{p}^{\mathcal{R}_2}$ of $\mathbf{x}^{\mathcal{R}_2}$ can be generated by using Eq. (5).

For a specific attribute of a generated sample in the proposed FR, its values can be that attribute's of an arbitrary sample over a sample batch. Thus, we can generate a lot of new samples as long as we do more random shuffles. To generate more synthesized samples, we conduct the random shuffle 10 times, which generates feature representations $\{\mathbf{X}^{\kappa,\mathcal{R}_1}\}$ and $\{\mathbf{x}^{\mathcal{R}_1}\}$ with a number of $10 \times n^{bs}$ for training.

3.3 Model Training

In the training stage, the weighted binary cross-entropy loss (Li et al. 2015; Tan et al. 2019b; Tang et al. 2019c) is employed as the loss function on homogeneous binary datasets, including RAP, PA-100K, and PETA. While the multi-class softmax loss is employed on heterogeneous attribute datasets, including Matket-1501 and Duke. For clarity, we take the weighted binary cross-entropy loss as an example, and the loss form for the multi-class softmax loss can be produced similarly. All of the predicted scores are followed by loss functions. For the predicted score $p_{ij}^{\mathcal{FM},\kappa}$ of the jth attribute, its loss over a batch can be calculated as:

$$\begin{aligned}\mathcal{L}_j^{\mathcal{FM},\kappa} = -\frac{1}{n^{bs}} \sum_{i=1}^{n^{bs}} \rho_{ij} \Big(& y_{ij} \log(p_{ij}^{\mathcal{FM},\kappa}) \\ & + (1 - y_{ij}) \log(1 - p_{ij}^{\mathcal{FM},\kappa}) \Big)\end{aligned} \tag{8}$$

where ρ_{ij} is a penalty coefficient used to alleviate the imbalanced data problem in pedestrian attribute recognition, and set the same as the work (Tan et al. 2019b). We use r_j to denote the ratio of the images with the jth attribute, and then ρ_{ij} is calculated as follows: $\rho_{ij} = \sqrt{\frac{1}{2r_j}}$, if $y_{ij} = 1$; otherwise $\rho_{ij} = \sqrt{\frac{1}{2(1-r_j)}}$. The sum of the losses over all attributes can be denoted as: $\mathcal{L}^{\mathcal{FM},\kappa} = \sum_j \mathcal{L}_j^{\mathcal{FM},\kappa}$. The losses for the predicted scores $\mathbf{p}$ and $\mathbf{p}^{\mathcal{FV}}$ can be produced in a similar way, and we denote them as $\mathcal{L}$ and $\mathcal{L}^{\mathcal{FV}}$, respectively. For the generated samples, the losses $\mathcal{L}^{\mathcal{R}_1}$ and $\mathcal{L}^{\mathcal{R}_2}$ are calculated based on predicted scores $\mathbf{p}^{\mathcal{R}_1}$ and $\mathbf{p}^{\mathcal{R}_2}$ and corresponding labels $\mathbf{y}^{\mathcal{R}_1}$ and $\mathbf{y}^{\mathcal{R}_2}$, respectively. The calculations of $\mathcal{L}^{\mathcal{R}_1}$ and $\mathcal{L}^{\mathcal{R}_2}$ have a similar form to Eq. 8 but with minor changes. Here we take the $\mathcal{L}_j^{\mathcal{R}_1}$ as an example and it can be formulated as:

$$\begin{aligned}\mathcal{L}_j^{\mathcal{R}_1} = -\frac{1}{10 \times n^{bs}} \sum_{i=1}^{10 \times n^{bs}} \rho_{ij} \Big(& y_{ij}^{\mathcal{R}_1} \log(p_{ij}^{\mathcal{R}_1}) \\ & + (1 - y_{ij}{}^{\mathcal{R}_1}) \log(1 - p_{ij}^{\mathcal{R}_1}) \Big)\end{aligned} \tag{9}$$

Then, the losses for all attributes are summed together: $\mathcal{L}^{\mathcal{R}_1} = \sum_j \mathcal{L}_j^{\mathcal{R}_1}$. The loss $\mathcal{L}^{\mathcal{R}_2}$ is calculated in a similar way. The overall loss is the sum of all of those losses, and can be denoted as:

$$\begin{aligned}\mathcal{L}^{overall} = & \mathcal{L} + \mathcal{L}^{\mathcal{FM},1} + \mathcal{L}^{\mathcal{FM},2} + \mathcal{L}^{\mathcal{FM},3} \\ & + \mathcal{L}^{\mathcal{FV}} + \mathcal{L}^{\mathcal{R}_1} + \mathcal{L}^{\mathcal{R}_2}.\end{aligned} \tag{10}$$

In the above equation, $\mathcal{L}$ denotes the loss for the whole network training. $\mathcal{L}^{\mathcal{FM},1}$, $\mathcal{L}^{\mathcal{FM},2}$, $\mathcal{L}^{\mathcal{FM},3}$ and $\mathcal{L}^{\mathcal{FV}}$ are the constrained losses for extracting the attribute-specific features. Besides, $\mathcal{L}^{\mathcal{R}_1}$ and $\mathcal{L}^{\mathcal{R}_2}$, which are used to guide the learning of synthetic samples, can be regarded as the regularization terms to span the potential samples' variability. In the test stage, the predictions are obtained based on the predicted scores $\mathbf{p}$.

4 Experiments

We first introduce the datasets, settings, and evaluation metrics employed in our experiments. Then, the experimental results and analysis are presented to validate the effectiveness of our method.

4.1 Datasets and Metrics

Five popular datasets including PA-100K (Liu et al. 2017), RAP (Li et al. 2018b), PETA (Deng et al. 2014), Market-1501 (Lin et al. 2019) and Duke (Lin et al. 2019) are employed for experiments. For PA-100K, RAP and PETA three datasets, all of them contain homogeneous binary attributes, while for both Market-1501 and Duke datasets, they contain heterogeneous attributes where different attributes may have a different number of categories. We adopt both homogeneous and heterogeneous attribute datasets for experiments to thoroughly verify the effectiveness of the proposed method.

PA-100K contains 100,000 pedestrian images from various outdoor scenes and is the largest dataset for pedestrian attribute recognition. Each image is annotated with 26 commonly used attributes, e.g., gender, clothing types. According to the works (Liu et al. 2017; Tan et al. 2019b) the dataset is randomly split into three subsets with 80,000, 10,000 and 10,000 images for training, validation and test, respectively. **RAP** is the largest pedestrian attribute dataset of indoor scenes, with containing 41,585 images. 51 attributes with the positive ratio over 1% are selected for experiments. We evaluate the proposed method over 5 random splits, where 33,268 images are used for training and 8317 images for the test in each split. We then average the results overall splits to achieve the final result. **PETA** is a classical dataset for pedestrian attribute recognition. Following the works (Deng et al. 2014; Tan et al. 2019b), 35 binary attributes are selected for evaluation. The whole dataset is split into three sub-sets: 9500 images for training, 1900 images for validation and 7600 images for test. **Market-1501** attribute dataset contains 32,688 images of 1501 identities. This dataset is annotated in the identity level, and each image is annotated with 10 binary attributes and 2 multi-class attributes. Following to the works (Lin et al. 2019; Tan et al. 2019b), 751 identities are used for training, and 750 identities are used for the test. **Duke** attribute dataset is also labeled in the identity level. It contains 34,183 images from 1812 identities, and each image is annotated with 8 binary attributes, and 2 multi-class attributes. According to the works (Lin et al. 2019; Tan et al. 2019b), 16,522 images are used for training and 17,661 images are used for the test.

According to previous works (Liu et al. 2017; Li et al. 2018b; Tan et al. 2019b; Tang et al. 2019c), a label-based criterion mean accuracy (mA) and four instance-based criteria accuracy (Accu), precision (Prec), Recall, and F1 are employed for evaluation on PA-100K, RAP, and PETA datasets. When evaluating on Market-1501 and Duke datasets, we employ the accuracy on all attributes as the criterion used in Lin et al. (2019) and Tan et al. (2019b).

Table 1 The ablation studies of the SAL-FM

Method	RAP		PA-100K		PETA	
	mA	F1	mA	F1	mA	F1
w/o attention	81.87	79.50	80.76	86.92	85.52	86.78
w/o multi-level	82.40	79.78	82.18	87.15	85.48	86.36
w/o split	81.62	79.75	81.54	87.17	84.85	86.33
w/o aggregate	68.98	79.50	77.96	87.83	78.67	86.12
SAL-FM	82.96	79.92	82.46	87.22	85.62	86.56

4.2 Experimental Settings

The RGB image with a size of 256×128 is used as the input in our experiments. The input image is first normalized by subtracting a mean and dividing a standard deviation for each color channel before being fed to the network. We also employ the data augmentation to improve the performance of pedestrian attribute recognition, including random horizontal flipping, random scaling, rotation, translation, cropping, erasing and adding random gaussian blurs. Those augmentations also facilitate the model to handle the variations of pedestrian position, human poses, camera angle, and so on. For the attribute-specific features at the feature map level, the output shapes $a^\kappa \times h^\kappa \times w^\kappa$, $\kappa = 1, 2, 3$ are set to $1 \times 32 \times 16$, $3 \times 16 \times 8$ and $6 \times 16 \times 8$, respectively. The parameter selection of a^κ can be founded in Sect. 4.5. The feature dimension of $\mathbf{x}_{ij}$ is set as 32. Therefore, the shape of those generated features $\mathbf{X}^{1,\mathcal{R}_1}$, $\mathbf{X}^{2,\mathcal{R}_1}$, $\mathbf{X}^{3,\mathcal{R}_1}$ and $\mathbf{x}^{\mathcal{R}_2}$ are $10 \cdot n^{bs} \times 1 \cdot m \times 32 \times 16$, $10 \cdot n^{bs} \times 3 \cdot m \times 16 \times 8$, $10 \cdot n^{bs} \times 6 \cdot m \times 16 \times 8$ and $10 \cdot n^{bs} \times 32 \cdot m$, respectively. All networks are initialized with the pretrained weights of ImageNet (Deng et al. 2009), and then finetuned on pedestrian attribute datasets. We employ the Adam optimizer (Kingma and Ba 2015) for optimization, and set $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The batch size is set to 32 in the training stage. The learning rate is started with 0.0001 and reduced by a factor of 10 along with the increasing iterative times. All models are trained and tested with PyTorch on GTX 1080Ti GPU.

4.3 Ablation Studies

In this section, we conduct the ablation studies on the following modules: SAL at Feature Map level (SAL-FM), SAL at Feature Vector level (SAL-FV), Feature Recombination (FR) and cascaded learning (CAS). Following to the work (Tang et al. 2019c), two important criteria, namely mA and F1, are employed for evaluation. The experiments are conducted on RAP, PA-100K, and PETA datasets.

Analysis on SAL-FM Various experiments are conducted by removing attention masks, multi-level features, split losses and aggregating layers. From experimental results in Table 1 ('w/o' denotes 'without'), we find attention

Table 2 The ablation studies of the SAL-FV

Method	RAP		PA-100K		PETA	
	mA	F1	mA	F1	mA	F1
w/o split	82.25	79.94	81.85	86.98	85.29	86.44
w/o aggregate	74.48	80.03	80.33	87.18	84.37	86.41
SAL-FV	82.51	79.83	82.39	86.83	85.74	86.47

masks, split losses, and aggregating layers are extremely important to SAL-FM. When removing the attention masks, attribute-specific attention modules degenerate into plain CNN layers and lose their strong abilities to select the important regions/pixels. When removing the splitted losses, the module fails to capture the attribute-specific features, which hardly capture the individuality for each attribute. When removing the aggregating layers, the module makes the predictions based on the averaging scores of $\mathbf{p}_i^{\mathcal{FM},1}$, $\mathbf{p}_i^{\mathcal{FM},2}$ and $\mathbf{p}_i^{\mathcal{FM},3}$, which are generated by only using split features. We can observe that the performance dramatically drops by over 10% at mA. The drop may be due to the following reasons: (1) In split features, the features of each attribute are denoted by the feature maps with only several channels (1, 3, 6 channels are set in the 1th, 2th and 3th feature level, respectively), which may hardly contain enough semantics for each attribute. (2) The information exchanges are not allowed among multiple attributes at high-level features, which hardly capture the relations among them. (3) Although the split features contain the individual semantics for each attribute, they may still be less discriminative than the aggregated features.

Analysis on SAL-FV To analyze this module, the network with removed the split losses and aggregating layers are employed for comparisons. The experimental results are shown in the Table 2. Although the split losses can only improve the performance slightly, those losses are indispensable components of extracting attribute-specific features, which are very necessary for later feature recombination and cascaded learning modules. When removing the aggregating layers, the performance drops significantly on all three datasets. The poor performance may due to the same reasons as mentioned above. The results also show the aggregating layers are very necessary after splitting the features.

Analysis on Feature Recombination We further add the feature recombination on both SAL-FM and SAL-FV, which are denoted as SAL-FM-FR and SAL-FV-FR, respectively. The comparisons between methods with and without feature recombination are shown in Table 3. The feature recombination improves the performance on both SAL-FM and SAL-FV networks, which clearly demonstrates its effectiveness. More specially, for SAL-FM, it improves the mA by 0.95%, 0.77% and 0.36% on RAP, PA-100K and PETA datasets, respectively.

Table 3 The ablation studies of the feature recombination

Method	RAP		PA-100K		PETA	
	mA	F1	mA	F1	mA	F1
SAL-FM	82.96	79.92	82.46	87.22	85.62	86.56
SAL-FM-FR	83.91	80.16	83.23	87.42	85.98	86.88
SAL-FV	82.51	79.83	82.39	86.83	85.74	86.47
SAL-FV-FR	83.00	79.91	82.41	87.02	86.04	86.55

Table 4 The ablation studies of the cascaded learning

Method	RAP		PA-100K		PETA	
	mA	F1	mA	F1	mA	F1
SAL-FM-FR	83.91	80.16	83.23	87.42	85.98	86.88
SAL-FV-FR	83.00	79.91	82.41	87.02	86.04	86.55
CAS-SAL-FR	84.18	80.56	82.86	87.79	86.40	87.18

Analysis on Cascaded Learning We further combine SAL-FM-FR and SAL-FV-FR together, and formulate a cascaded Split-and-Aggregate Learning with Feature Recombination (CAS-SAL-FR). As shown in Table 4, the performance can be further improved on all three datasets, which shows the effectiveness of cascaded learning.

4.4 Comparisons with State-of-the-arts

In this subsection, we compare the proposed CAS-SAL-FR against previous state-of-the-art methods, including HP-net (Liu et al. 2017), VeSPA (Sarfraz et al. 2017), JRL (Wang et al. 2017), Fusion (Li et al. 2018a), LG-Net (Liu et al. 2018b), VAA (Sarafianos et al. 2018), GRL (Zhao et al. 2018), RA (Zhao et al. 2019), JLPLS-PAA (Tan et al. 2019b), CoCNN (Han et al. 2019), Da-HAR (Wu et al. 2020), MT-CAS (Zeng et al. 2020), (Jia et al. 2020), DTM+AWK (Zhang et al. 2020), Gao et al. (2019), Tang et al. (2019c), PedAttriNet (Lin et al. 2019) and APR (Lin et al. 2019). On RAP, PA-100K and PETA datasets, we further add the mean accuracy of mA, Accu, Prec, Recall and F1 five criteria for evaluation (denoted as mFive). This is because some models may perform very well on a specific criterion while obtaining low performance on other criteria. Take mFive into account, and the evaluation can be more comprehensive.

The experimental results of the homogeneous binary attributes on RAP, PA-100K and PETA datasets are shown in Tables 5, 6 and 7, respectively. The experimental results of the heterogeneous attributes on Market-1501 and Duke attribute datasets are shown in Table 8. The proposed CAS-SAL-FR achieves the highest performance on all five datasets (including three homogeneous binary attribute datasets and two heterogeneous attribute datasets), showing its superiority for pedestrian attribute recognition. The mean performance of our method on RAP, PA-100K, PETA, Market, and Duke

Table 5 The comparisons on RAP dataset

Method	References	Backbone	mA	Accu	Prec	Recall	F1	mFive
VeSPA (Sarfraz et al. 2017)	BMVC'17	GoogleNet	77.70	67.35	79.51	79.67	79.59	76.76
HP-net (Liu et al. 2017)	ICCV'17	Inception_v2	76.12	65.39	77.33	78.79	78.05	75.14
JRL (Wang et al. 2017)	ICCV'17	AlexNet	77.81	–	78.11	78.98	78.58	–
Fusion (Li et al. 2018a)	ICME'18	CaffeNet	74.31	64.57	78.86	75.90	77.35	74.20
LG-Net (Liu et al. 2018b)	BMVC'18	Inception-v2	78.68	68.00	80.36	79.82	80.09	77.39
GRL (Zhao et al. 2018)	IJCAI'18	Inception-v3	81.20	–	77.70	80.90	79.29	–
RA (Zhao et al. 2019)	AAAI'19	Inception-v3	81.16	–	79.45	79.23	79.34	–
(Gao et al. 2019)	ACM MM'19	ResNet-50	82.45	49.10	55.00	80.44	65.33	66.46
JLPLS-PAA (Tan et al. 2019b)	TIP'19	SE-Net	81.25	67.91	78.56	81.45	79.98	77.83
CoCNN (Han et al. 2019)	IJCAI'19	ResNet-50	81.42	68.37	81.04	80.27	80.65	78.35
(Tang et al. 2019c)	ICCV'19	Inception-v3	81.87	68.17	74.71	86.48	80.16	78.28
Da-HAR (Wu et al. 2020)	AAAI'20	ResNet-101	79.44	68.86	80.14	81.30	80.72	78.09
DTM+AWK (Zhang et al. 2020)	Arxiv'20	ResNet-50	82.04	67.42	75.87	84.16	79.80	77.86
Jia et al. (2020)	Arxiv'20	ResNet-50	76.48	67.17	82.84	76.25	78.94	76.33
CAS-SAL-FR	This work	ResNet-50	84.18	68.59	77.56	83.81	80.56	78.94

Table 6 The comparisons on PA-100K dataset

Method	References	Backbone	mA	Accu	Prec	Recall	F1	mFive
HP-net (Liu et al. 2017)	ICCV'17	Inception_v2	74.21	72.19	82.97	82.09	82.53	78.79
Fusion (Li et al. 2018a)	ICME'18	CaffeNet	74.95	73.08	84.36	82.24	83.29	79.58
LG-Net (Liu et al. 2018b)	BMVC'18	Inception_v3	76.96	75.55	86.99	83.17	85.04	81.54
JLPLS-PAA (Tan et al. 2019b)	TIP'18	SE-Net	81.61	78.89	86.83	87.73	87.27	84.47
CoCNN (Han et al. 2019)	IJCAI'19	ResNet-50	80.56	78.30	89.49	84.36	86.85	83.91
Tang et al. (2019c)	ICCV'19	Inception_v3	80.68	77.08	84.21	88.84	86.46	83.45
MT-CAS (Zeng et al. 2020)	ICME'20	ResNet-34	77.20	78.09	88.46	84.86	86.62	83.04
Jia et al. (2020)	Arxiv'20	ResNet-50	79.38	78.56	89.41	84.78	86.55	83.73
DTM+AWK (Zhang et al. 2020)	Arxiv'20	ResNet-50	81.67	77.57	84.27	89.02	86.58	83.82
CAS-SAL-FR	This work	ResNet-50	82.86	79.64	86.81	88.78	87.79	85.18

Table 7 The comparisons on PETA dataset

Method	References	Backbone	mA	Accu	Prec	Recall	F1	mFive
HP-net (Liu et al. 2017)	ICCV'17	Inception_v2	81.77	76.13	84.92	83.24	84.07	82.03
VeSPA (Sarfraz et al. 2017)	BMVC'17	GoogleNet	83.45	77.73	86.18	84.81	85.49	83.53
JRL (Wang et al. 2017)	ICCV'17	AlexNet	85.67	–	86.03	85.34	85.42	–
Fusion (Li et al. 2018a)	ICME'18	CaffeNet	82.97	78.08	86.86	84.68	85.76	83.67
VAA (Sarafianos et al. 2018)	ECCV'18	DenseNet-201	84.59	78.56	86.79	86.12	86.46	84.50
GRL (Zhao et al. 2018)	IJCAI'18	Inception_v3	86.70	–	84.34	88.82	86.51	–
Gao et al. (2019)	ACM MM'19	ResNet-50	86.23	77.21	84.52	87.22	85.85	84.20
RA (Zhao et al. 2019)	AAAI'19	Inception_v3	86.11	–	84.69	88.51	86.56	–
JLPLS-PAA (Tan et al. 2019b)	TIP'19	SE-Net	84.88	79.46	87.42	86.33	86.87	84.99
Tang et al. (2019c)	ICCV'19	Inception_v3	86.30	79.52	85.65	88.09	86.85	85.28
MT-CAS (Zeng et al. 2020)	ICME'20	ResNet-34	83.17	78.78	87.49	85.35	86.41	84.24
Jia et al. (2020)	Arxiv'20	ResNet-50	85.12	79.14	86.99	86.33	86.39	84.79
DTM+AWK (Zhang et al. 2020)	Arxiv'20	ResNet-50	85.79	78.63	85.65	87.17	86.11	84.67
CAS-SAL-FR	This work	ResNet-50	86.40	79.93	87.03	87.33	87.18	85.57

Table 8 The comparisons on Market-1501 and Duke datasets

Dataset	Market-1501	Duke
PedAttriNet (Lin et al. 2019)	84.64	80.07
APR (Lin et al. 2019)	85.33	80.12
JLPLS-PAA (Tan et al. 2019b)	87.88	85.24
CAS-SAL-FR	88.30	85.91

attribute datasets are 78.94%, 85.18%, 85.57%, 88.30% and 85.91%, respectively. Moreover, compared with the recent work, Tang et al. (2019c), the proposed method outperforms it by 0.66%, 1.73% and 0.29% on RAP, PA-100K, and PETA datasets, respectively. For the recent work JLPLS-PAA, our method outperforms it by 1.13%, 0.71%, 0.58%, 0.42% and 0.67% on RAP, PA-100K, PETA, Market-1501 and Duke attribute datasets, respectively. Those improvements are promising because the performance is averaging on dozens of attributes where the accuracies of some attributes are really hard to be improved due to low resolution, occlusions, unbalanced data, and so on.

Owing to the lack of a unified benchmark method in the field of pedestrian attribute recognition, different methods may adopt different backbones. The backbone of all models, e.g., AlexNet (Krizhevsky et al. 2012), CaffeNet (Jia et al. 2014), Inception_v2/v3 (Ioffe and Szegedy 2015), GoogleNet (Szegedy et al. 2015), DenseNet-201 (Huang et al. 2017), ResNet-50/101 (He et al. 2016) and SE-Net (Hu et al. 2018b), also have been clarified. Our method is constructed based on ResNet-50. Although some previous methods (Tan et al. 2019b; Sarafianos et al. 2018) construct their models based on more advanced backbones. For example, JLPLS-PAA (Tan et al. 2019b) and VAA (Sarafianos et al. 2018) build their models based on SE-Net (Hu et al. 2018a) and DenseNet-201 (Huang et al. 2017), respectively. However, our method can still achieve better performance. Moreover, in some previous methods (Tan et al. 2019b; Zhao et al. 2018; Gao et al. 2019), external information is employed to improve the performance further. For example, JLPLS-PAA (Tan et al. 2019b) captures the external semantics from human parsing, and GRL (Zhao et al. 2018) utilizes the human pose information for human body localization. Moreover, some researchers (Wang et al. 2017) employ an ensemble of multiple models to obtain higher performance. Our proposed CAS-SAL-FR still outperforms those models, which shows the effectiveness of the proposed cascaded split-and-aggregate learning and feature recombination.

4.5 Further Analysis

Performance on all Classifiers We visualize the results of all classifiers in Fig. 4a. FM(1), FM(2), FM(3), FV denote the

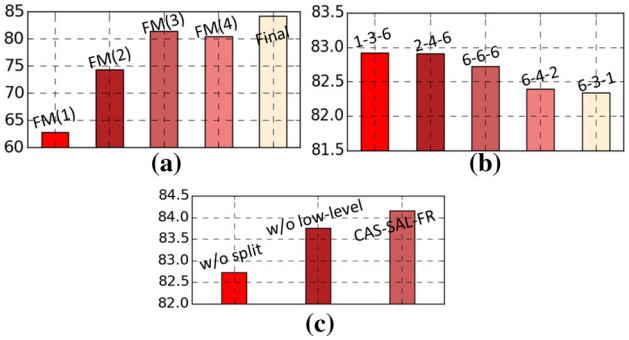

Fig. 4 The mA results on RAP of **a** all classifiers, **b** the network with using the attention module of different settings and **c** removing the splitting operation and low-level features

classifiers at 1st, 2nd, 3th feature map levels and feature vector level, respectively. The highest performance is obtained by the final classifier (denoted by Final). The poor performance obtained by FM(1), FM(2), FM(3) and FV may due to that the split features may be less discriminative than the aggregated features'.

Analysis on a^κ in ASAM The number of channels for each attribute of the ASAM at 1st, 2nd, 3th feature map levels is denoted as a^1, a^2, a^3 (denoted by $a^1 - a^2 - a^3$), respectively. The experimental results by varying their values are shown in Fig. 4b. The highest performance is achieved using $1-3-6$, where the small number of ASAM channels is used at low levels. $a^1 = 1$, $a^2 = 3$ and $a^3 = 6$ are adopted in other experiments.

Removing Split Operation To investigate how much the split operation can contribute to the final performance, we conduct an additional experiment with removing the split operation on both feature map and vector levels. As shown in Fig. 4c, the mA is dropped by 1.42% when removing the split operations, which shows their effectiveness. The split operation helps the network to capture the individuality for each attribute, which ensures each attribute can learn its own semantics.

Removing Low-level Features To further verify the effectiveness of the low-level features (including both 1th and 2th levels), we conduct the experiments by removing those features in our CAS-SAL-FR. The experimental results can be found in Fig. 4c. The mA is dropped by 0.41% when removing low-level features, which shows the low-level features also contribute a lot to the final performance.

Efficiency Analysis We compare our method with three popular methods in parameters, FLOPs and inference speed, including JRL (Wang et al. 2017), VAA (Sarafianos et al. 2018) and JLPLS-PAA (Tan et al. 2019b). As shown in Table 9, our method has certain advantages compared with three compared method. For example, JLPLS-PAA is constructed based on two large models, where one model (SE-BN-inception) is used for pedestrian attribute recogni-

Table 9 The comparisons on parameters, FLOPs and speed

Method	Param	FLOPs	Speed
JRL* (Wang et al. 2017)	58.3M×10	0.72G×10	2.06 ms×10
VAA† (Sarafianos et al. 2018)	20.2M	4.37G	39.45 ms
JLPLS-PAA (Tan et al. 2019b)	92.75M	48.07G	54.56 ms
CAS-SAL-FR	35.2M	5.58G	22 ms

JRL* is a combined model based on 10 AlexNets, so its complexity is estimated by using 10 AlexNets. VAA† is constructed based on the DenseNet-201, and its efficiency is estimated with a DenseNet-201

Table 10 Comparisons between FR and mixup

Method	RAP		PA-100K		PETA	
	mA	F1	mA	F1	mA	F1
SAL-FM + mixup	80.40	80.70	81.61	87.63	85.37	86.63
SAL-FM-FR	83.91	80.16	83.23	87.42	85.98	86.88

tion and the other model (PSPNet, ResNet-101) is used for generate pedestrian parsing maps. Thus, JLPLS-PAA contains lots of parameters and FLOPs.

GPU memory consumption of FR Although FR generates a large number of synthetic samples for training, all of them only need to be delivered in the last few layers of the network, which takes up very little GPU memory. We conduct experiments with two settings, namely CAS-SAL and CAS-SAL-FR to verify this. Experiments show that CAS-SAL takes up 5357M GPU memory when running with a batch size of 32, and the GPU memory occupation increases to 5697M when adding the proposed FR strategy. This indicates that only about 340M GPU memory are taken up for FR, which verifies that FR only takes up very little GPU memory.

FR versus mixup We conduct experiments with both FR and mixup to compare their performance, and the corresponding experimental results are listed in Table 10. Compared with using mixup, FR helps the model to obtain a higher accuracy on mA. For example, on RAP dataset, the mA accuracty of SAL-FM-FR is 3.5% higher than that of SAL-FM + mixup. In our FR, we just uses a random shuffle on the split features over a batch of samples to generate new samples, which keeps the semantic information of each attribute unchanged. For mixup, it achieves low mA accuracy may due to that the liner combination may destroy the integrity of the feature especially when the quality of images and features are not so high.

Visualizations of Attention We visualize the attention masks in ASAM after $\mathcal{F}_3(\mathbf{I}_i)$. We select 6 representative attributes, i.e., Gender, Glasses, HandBag, LongSleeve, UpperPlaid and Shorts, and visualize their mean attention masks. As shown in Fig. 5, different attributes may focus on different regions to extract the discriminative features for corresponding attributes. For example, the attention module focuses on the head region for Glasses attribute. More visualizations on RAP and PETA datasets can be founded in Figs. 6 and 7. The visualizations can qualitatively show the proposed ASAM can really capture discriminative features from some important regions for each attribute. ASAM is designed to extract discriminative attribute-specific features for each attribute with attention mechanisms. Similarly, Our ASAM also can be extended to the fields of multi-task learning and multi-label classification, which helps the network to learn the discriminative features of each attribute/task separately and capture specific semantics of each task/category.

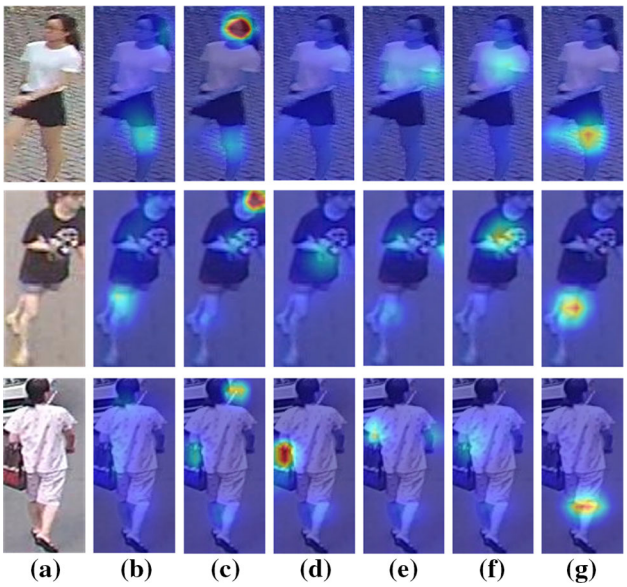

Fig. 5 Visualizations of the attention masks of ASAM on PA-100K dataset. **a** Indicates the raw image, and **b**–**g** represent the attention masks of Gender, Glasses, HandBag, LongSleeve, UpperPlaid and Shorts attributes, respectively

Qualitative Analysis Two predicted examples on the test set of the PA-100K dataset are shown in Fig. 8. The ground truth (GT) labels and the predictions of the baseline ResNet-50 and CAS-SAL-FR are denoted by red, green, and blue colors, respectively. Benefited from the cascaded split-and-aggregate learning and feature recombination, the proposed CAS-SAL-FR can achieve more reliable predictions than the baseline ResNet-50. For the first image, some attributes like

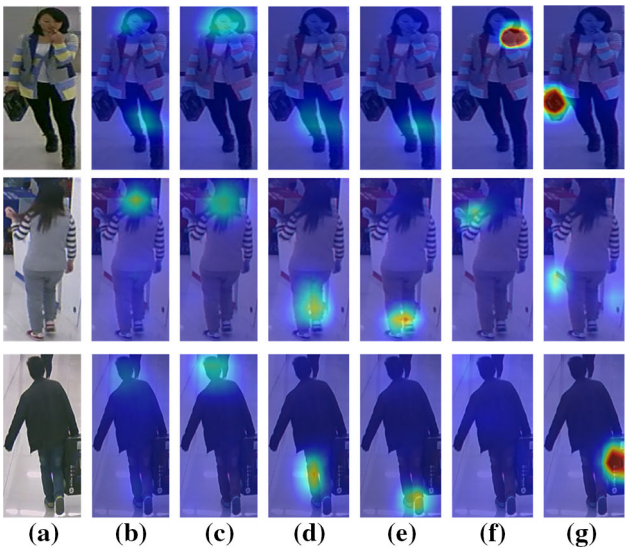

Fig. 6 Visualizations of attention masks of ASAM on RAP dataset. **a** Indicates the raw image, and **b–g** represent the attention masks of Female, BlackHair, LongTrousers, LeatherShoes, Calling and CarryingbyHand attributes, respectively

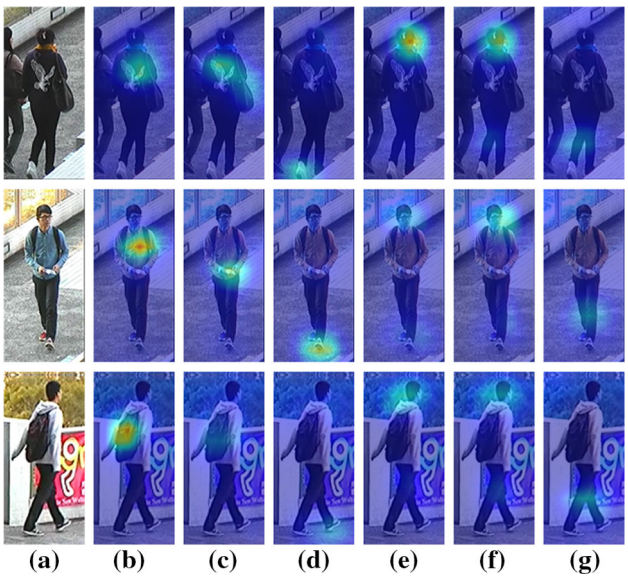

Fig. 7 Visualizations of attention masks of ASAM on PETA dataset. **a** Indicates the raw image, and **b–g** represent the attention masks of carryingBackpack, carryingOther, footwearLeatherShoes, hairLong, personalMale and lowerBodyShorts attributes, respectively

Gender and Skirt&Dress are wrongly predicted by ResNet-50, while our CAS-SAL-FR can well correct them.

5 Conclusions

In this work, we have proposed a new framework for pedestrian attribute recognition, named CAScaded Split-and-Aggregate Learning with Feature Recombination (CAS-SAL-FR). At first, a cascaded Split-and-Aggregate Learning (SAL) has been proposed to capture both the individuality and commonality for all pedestrian attributes. Besides, feature recombination has been further proposed to synthesize more training representations for achieving better performance. The experiments have been conducted on five popular datasets including RAP, PA-100K, PETA, Market-1501 and Duke attribute datasets, showing the proposed method achieves the new state-of-the-art performance. Finally, we also have presented feature visualizations and a comprehensive analysis on CAS-SAL-FR to qualitatively verify its effectiveness.

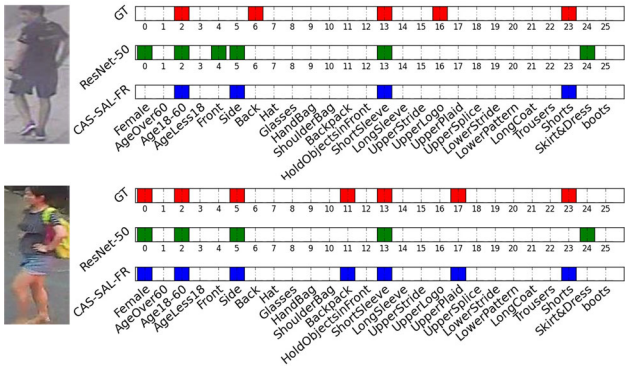

Fig. 8 Two prediction examples on PA-100K dataset

Acknowledgements This work was partially supported by the National Key Research and Development Program (No. 2020YFC2003901), the Chinese National Natural Science Foundation Projects #61806203, #61961160704, #61876179, the External cooperation key project of Chinese Academy Sciences #173211KYSB20200002, the Key Project of the General Logistics Department Grant No. AWS17J001, Science and Technology Development Fund of Macau (No. 0010/2019/AFJ, 0025/2019/AKP, 0019/2018/ASC), the Spanish project TIN2016-74946-P (MINECO/FEDER, UE) and CERCA Programme/Generalitat de Catalunya. This work was also supported in part by the Academy of Finland (Grants 336033, 315896), Business Finland (Grant 884/31/2018), and EU H2020 (Grant 101016775).

References

Chen, B., Deng, W., & Hu, J. (2019). Mixed high-order attention network for person re-identification. In *ICCV*.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009) . Imagenet: A large-scale hierarchical image database. In *CVPR*.

Deng, Y., Luo, P., Loy, C. C., & Tang, X. (2014). Pedestrian attribute recognition at far distance. In *ACM MM*.

Dixit, M., Kwitt, R., Niethammer, M., & Vasconcelos, N. (2017). Aga: Attribute-guided augmentation. In *CVPR*.

Fu, C., Wu, X., Hu, Y., Huang, H., & He, R. (2019). Dual variational generation for low-shot heterogeneous face recognition. In *NeurIPS*.

Fu, J., Zheng, H., & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*.

Gao, L., Huang, D., Guo, Y., & Wang, Y. (2019). Pedestrian attribute recognition via hierarchical multi-task learning and relationship attention. In *ACM MM*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *NeurIPS*.

Guo, H., Zheng, K., Fan, X., Yu, H., & Wang, S. (2019). Visual attention consistency under image transforms for multi-label image classification. In *CVPR*.

Han, K., Wang, Y., Shu, H., Liu, C., Xu, C., & Xu, C. (2019). Attribute aware pooling for pedestrian attribute recognition. In *IJCAI*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *ICCV*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Hongyi, Z., Moustapha, C., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International conference on learning representations*.

Hu, J., Shen, L., & Sun, G. (2018a). Squeeze-and-excitation networks. In *CVPR* (pp. 7132–7141).

Hu, J., Shen, L., & Sun, G. (2018b) . Squeeze-and-excitation networks. In *CVPR*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR* (pp. 4700–4708).

Ioffe, S., & Szegedy, C. (2015) . Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Jia, J., Huang, H., Yang, W., Chen, X., & Huang, K. (2020). Rethinking of pedestrian attribute recognition: Realistic datasets with efficient method. arXiv preprint arXiv:2005.11909

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Li, D., Chen, X., & Huang, K. (2015). Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In *ACPR*.

Li, D., Chen, X., Zhang, Z., & Huang, K. (2018a). Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. In *ICME*.

Li, D., Zhang, Z., Chen, X., & Huang, K. (2018b). A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. In *IEEE TIP*.

Li, Q., Zhao, X., He, R., & Huang, K. (2019a). Pedestrian attribute recognition by joint visual-semantic reasoning and knowledge distillation. In *IJCAI*.

Li, Q., Zhao, X., He, R., & Huang, K. (2019b). Visual-semantic graph reasoning for pedestrian attribute recognition. In *AAAI*.

Li, W., Zhu, X., & Gong, S. (2018c). Harmonious attention network for person re-identification. In *CVPR*.

Lim, J. J., Salakhutdinov, R. R., & Torralba, A. (2011). Transfer learning by borrowing examples for multiclass object detection. In *NeurIPS*.

Lin, Y., Zheng, L., Zheng, Z., Wu, Y., Hu, Z., Yan, C., et al. (2019). Improving person re-identification by attribute and identity learning. *Pattern Recognition*, *95*, 151–161.

Liu, B., Wang, X., Dixit, M., Kwitt, R., & Vasconcelos, N. (2018a). Feature space transfer for data augmentation. In *CVPR*.

Liu, P., Liu, X., Yan, J., & Shao, J.(2018b). Localization guided learning for pedestrian attribute recognition. In *BMVC*.

Liu, X., Zhao, H., Tian, M., Sheng, L., Shao, J., Yi, S., Yan, J., & Wang, X. (2017). Hydraplus-net: Attentive deep features for pedestrian analysis. In *ICCV*.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikainen, M. (2020). Deep learning for generic object detection: A survey. In *IJCV*.

Sarafianos, N., Xu, X., & Kakadiaris, I. A. (2018). Deep imbalanced attribute classification using visual attention aggregation. In *ECCV*.

Sarfraz, M. S., Schumann, A., Wang, Y., & Stiefelhagen, R. (2017). Deep view-sensitive pedestrian attribute inference in an end-to-end model. In *BMVC*.

Shifeng, Z., Longyin, W., Shi, H., Lei, Z., Lyu, S., & Li, S. Z. (2019). Single-shot scale-aware network for real-time face detection. In *IJCV*.

Shuzhe, W., Meina, K., Shan, S., & Chen, X. (2019). Hierarchical attention for part-aware face detection. In *IJCV*.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Tan, Z., Wan, J., Lei, Z., Zhi, R., Guo, G., & Li, S. Z. (2018). Efficient group-n encoding and decoding for facial age estimation. In *IEEE TPAMI*.

Tan, Z., Yang, Y., Wan, J., Guo, G., & Li, S. Z. (2019a). Deeply-learned hybrid representations for facial age estimation. In *IJCAI* (pp. 3548–3554).

Tan, Z., Yang, Y., Wan, J., Hang, H., Guo, G., & Li, S. Z. (2019b). Attention-based pedestrian attribute analysis. *IEEE TIP*, *28*(12), 6126–6140.

Tang, C., Sheng, L., Zhang, Z., & Hu, X. (2019c). Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization. In *ICCV*.

Wang, J., Zhu, X., Gong, S., & Li, W. (2017). Attribute recognition by joint recurrent learning of context and correlation. In *ICCV*.

Wang, Y., Gan, W., Wu, W., & Yan, J. (2019). Dynamic curriculum learning for imbalanced data classification. In *ICCV*.

Woo, S., Park, J., Lee, J. Y., & So Kweon, I. (2018). Cbam: Convolutional block attention module. In *ECCV*.

Wu, M., Huang, D., Guo, Y., & Wang, Y. (2020). Distraction-aware feature learning for human attribute recognition via coarse-to-fine attention mechanism. In *AAAI*.

Xiang, L., Jin, X., Ding, G., Han, J., & Li, L. (2019). Incremental few-shot learning for pedestrian attribute recognition. In *IJCAI*.

Xiangyu, Z., Hao, L., & Gong, S. (2020). Scalable person re-identification by harmonious attention. In *IJCV*.

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *ICLR*.

Zeng, H., Ai, H., Zhuang, Z., & Chen, L. (2020). Multi-task learning via co-attentive sharing for pedestrian attribute recognition. In *ICME*.
Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., & Li, M. (2020). Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955
Zhang, J., Ren, P., & Li, J. (2020) . Deep template matching for pedestrian attribute recognition with the auxiliary supervision of attribute-wise keypoints. arXiv preprint arXiv:2011.06798
Zhao, X., Sang, L., Ding, G., Guo, Y., & Jin, X. (2018). Grouping attribute recognition for pedestrian with joint recurrent learning. In *IJCAI*.
Zhao, X., Sang, L., Ding, G., Han, J., Di, N., & Yan, C. (2019). Recurrent attention model for pedestrian attribute recognition. In: *AAAI*.
Zheng, Z., Zheng, L., & Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*.
Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*.
Zhu, J., Liao, S., Lei, Z., Yi, D., & Li, S. Z. (2013). Pedestrian attribute classification in surveillance: Database and evaluation. In *ICCVW*.
Zhu, X., Liu, H., Lei, Z., Shi, H., Yang, F., Yi, D., Qi, G., & Li, S. Z. (2019). Large-scale bisample learning on id versus spot face recognition. In *IJCV*.