# Attribute Aware Pooling for Pedestrian Attribute Recognition

**Kai Han**[1], **Yunhe Wang**[1], **Han Shu**[1], **Chuanjian Liu**[1], **Chunjing Xu**[1], **Chang Xu**[2*]
[1]Huawei Noah's Ark Lab
[2]School of Computer Science, FEIT, University of Sydney, Australia
{kai.han, yunhe.wang, han.shu, liuchuanjian, xuchunjing}@huawei.com, c.xu@sydney.edu.au

## Abstract

This paper expands the strength of deep convolutional neural networks (CNNs) to the pedestrian attribute recognition problem by devising a novel attribute aware pooling algorithm. Existing vanilla CNNs cannot be straightforwardly applied to handle multi-attribute data because of the larger label space as well as the attribute entanglement and correlations. We tackle these challenges that hampers the development of CNNs for multi-attribute classification by fully exploiting the correlation between different attributes. The multi-branch architecture is adopted for fucusing on attributes at different regions. Besides the prediction based on each branch itself, context information of each branch are employed for decision as well. The attribute aware pooling is developed to integrate both kinds of information. Therefore, attributes which are indistinct or tangled with others can be accurately recognized by exploiting the context information. Experiments on benchmark datasets demonstrate that the proposed pooling method appropriately explores and exploits the correlations between attributes for the pedestrian attribute recognition.

## 1 Introduction

Pedestrian attribute recognition has appealed much research effort due to its continuing demands for intelligent video surveillance [Layne *et al.*, 2012; Peng *et al.*, 2016; Liu *et al.*, 2017b]. It is a challenging task because of the large variance in pedestrian images, such as the viewpoint, lighting or pose changes. Thanks to the development of deep learning, pedestrian attribute recognition based on convolutional neural networks (CNNs) has made tremendous progress on the benchmark datasets, *e.g.* PETA [Deng *et al.*, 2014], RAP [Li *et al.*, 2016], and PA-100K [Liu *et al.*, 2017b]. However, existing methods still have a long way to the practical application where the scenario is pretty complex.

*We thank anonymous reviewers for their helpful comments. Chang Xu was supported by the Australian Research Council under Project DE180101438.

Extraordinary capability of CNN to accomplish the single-label image classification task has been comprehensively validated [He *et al.*, 2016; Ren *et al.*, 2015; Wang *et al.*, 2016]. An example in the ImageNet Dataset [Russakovsky *et al.*, 2015] has only one label, such as "dog", "cat" or "car". However, pedestrian often contains multiple attributes, *e.g.* "age", "gender" and "clothing". Moreover, the fact that these attributes usually do not correspond to some certain objects will bring in hidden semantic recognition problem. Multi-label pedestrian attribute classification is thus a more practical and challenging problem to be thoroughly investigated. The superiority of CNNs in the standard single-label natural image classification cannot be straightforwardly extended to the attribute classification scenario due to the difference in input image domains and problem settings. Specifically, in multi-attribute recognition, candidate attributes may locate in different regions of the given image and the label space of a $k$-category classification task becomes $2^k$, which requires more training data and parameters for establishing a deep neural network to achieve a comparable accuracy. Current CNN solutions for the pedestrian attribute classification problem mainly focus on extracting high-level features from the entire image [Sudowe *et al.*, 2015; Li *et al.*, 2015; Xiao *et al.*, 2016], and multi-branch architecture for different attribute groups [Zhu *et al.*, 2015; Lu *et al.*, 2017; Zhao *et al.*, 2018; Han *et al.*, 2018a], to accomplish the subsequent classification.

The multi-branch manner for pedestrian attribute recognition achieves better performance via sharing the bottom $l$ layers then breakuping into tree-like architecture, resulting in task-specific sub-networks for similar attribute groups. This is inspired by the common observation that bottom layers in CNNs mainly extract low level visual features which can be shared across different tasks, while the top layers capture high level semantic features that are more task specific. [Hand and Chellappa, 2017] proposed a multi-task deep convolutional neural network (MCNN) for facial attribute classification. [Lu *et al.*, 2017] dynamically created a tree-like deep architecture where similar tasks reside in the same branch until the top layers. [Wang *et al.*, 2017; Zhao *et al.*, 2018] utilized a CNN-RNN model to take advantage of the intragroup mutual exclusion and inter-group correlation. These methods mine the correlations of attributes, but ignore the prior knowledge underlying the attribute data.

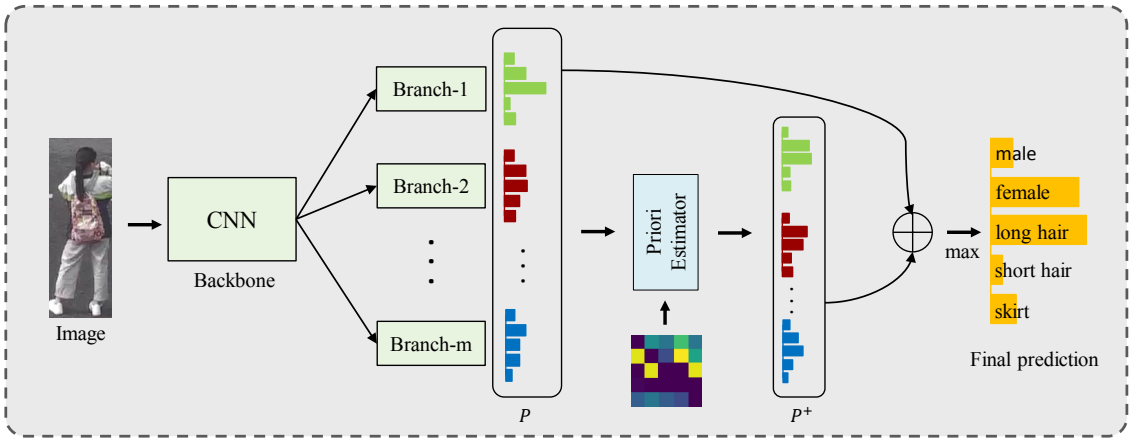


Figure 1: The diagram of the proposed attribute aware pooling approach. The input instance is fed into a shared CNN and produce multiple predictions with multi-branch architecutre (details in Fig. 3). Afterward, we exploit the co-occurrence priori to integrate these conditional probabilities, thereby producing a attribute aware pooling estimation.

In this paper, we propose to develop a new CNN architecture for pedestrian multi-attribute recognition by exploring the correlation between different attributes, *i.e.* attribute co-occurrence priori [Mensink *et al.*, 2014; Modiri Assari *et al.*, 2014; Guo and Gu, 2011; Boguraev and Ando, 2005]. In particular, there is a high probability that some attributes that often co-occur in the training set will also co-occur in the testing set. For example, given a pedestrian annotated with "long hair", "skirt" and "high-heeled shoes", we can easily deduce its gender attribute as "female". Based on this insightful observation, we develop a novel attribute aware pooling method (AAP) for integrating the information from different predictions, namely CoCNN. More specifically, the base network follows the multi-branch architecture, and then the context information in these branch is collected to estimate the attribute probabilities, which is then combined with the individual estimations of each branch for improving the resulting decision. The diagram of the proposed scheme is shown in Fig. 1. Experimental results on benchmarks demonstrate the superiority of the proposed algorithm over the state-of-the-art methods for pedestrian attribute classification.

## 2 Preliminaries

We first briefly introduce some related works on CNN and then build the relationship between the output of a conventional CNN and the proposed attribute aware pooling method.

Let $\mathcal{X}$ and $\mathcal{Y}$ denote instance space and $k$-label space, respectively. Given a labeled training set with $n$ instances, $\{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), ..., (\mathbf{x}^n, \mathbf{y}^n)\}$, where $\mathbf{x}^i$ is the $i$-th instance and $\mathbf{y}^i$ denotes its label vector which is a $k$-dimensional binary vector, *i.e.* $\mathbf{y}^i = [\mathbf{y}^i_1, ..., \mathbf{y}^i_k]$ and $\mathbf{y}^i_k \in \{0, 1\}$. Denote the feature of $\mathbf{x}_i$ calculated by a given CNN $\mathcal{N}$ as $x^i = \mathcal{N}(\mathbf{x}^i) \in \mathbb{R}^d$, the conventional probability hypothesis for the $j$-th attribute is

$$\hat{\mathbf{y}}^i_j = \Pr(\mathbf{a}^j | x^i) = \frac{1}{1 + e^{-\theta_j^T x^i}}, \tag{1}$$

where $\mathbf{a}^j$ is the $j$-th attribute in the dataset corresponding to the $j$-th dimensionality of the label space $\mathcal{Y}$, $\theta_j \in \mathbb{R}^d$ is the parameter of linear classification layer.

In practice, for a given image $\mathbf{x}$ (image index $i$ is omitted to simplify notation), it is difficult to accurately estimate its conditional probabilities w.r.t. all attributes labels simultaneously, due to the tangle between attributes and the deformation of objects, *e.g.* occlusions, illumination changes, rotations, and scale transforms. To address this problem, the multi-branch architecture is utilized in this work, *i.e.* we have $m$ branches $\{b_1, ..., b_m\}$. The lower layers are shared across branches and the top layers are separated to focus on the attributes at different positions, as shown in Fig. 1. The detailed multi-branch architecture is given in the experiments section. Formally for each branch, we can construct the following $m \times k$ matrix,

$$P = \begin{bmatrix} \Pr(\mathbf{a}^1|b_1) & \Pr(\mathbf{a}^2|b_1) & \dots & \Pr(\mathbf{a}^k|b_1) \\ \Pr(\mathbf{a}^1|b_2) & \Pr(\mathbf{a}^2|b_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \\ \Pr(\mathbf{a}^1|b_m) & \dots & & \Pr(\mathbf{a}^k|b_m) \end{bmatrix}, \tag{2}$$

where $\Pr(\mathbf{a}^j|b_l)$ stands for the conditional probability of $\mathbf{a}^j$ given the feature of $l$-th branch, and each row in $P$ corresponds to the softmax output of the branch $b_l$. Moreover, conditional probabilities of the whole image $\mathbf{x}$ w.r.t. different attributes can be estimated through a max-pooling operation:

$$\Pr(\mathbf{a}^j|\mathbf{x}) \approx \max_{l \in [1,m]} \Pr(\mathbf{a}^j|b_l) = \max_{l \in [1,m]} P_{l,j}. \tag{3}$$

It is instructive to note that the accuracy of Eq. 3 strongly depends on the prediction result of different branches. These branches are extracted from input images by exploiting empirical knowledge such as position information. However, attributes tangled with each other are difficult to be separated by branches accurately. Hence, the estimated conditional probability vector using Eq. 3 could be biased when the $j$-th attribute in $\mathbf{x}$ has not be well separated. Elements in the $j$-th column of $P$ will be extremely small, which leads to the $\Pr(\mathbf{a}^j|\mathbf{x})$ after the max-pooling being small as well.

A widely accepted fact in the filed of multi-label classification [Boguraev and Ando, 2005; Wang *et al.*, 2013; Guo and Gu, 2011] is that correlated labels have a strong probability to simultaneously occur in real-world images. In practice, for an attribute $\mathbf{a}^j$, we can use the other attributes to estimate the conditional probability w.r.t. the $j$-th attribute. The auxiliary estimation is generated as follow:

$$P^+_{l,j} = \frac{1}{s}\sum_{i=1}^{k} S^i_l \Pr(\mathbf{a}^j|\mathbf{a}^i), \tag{4}$$

where $S^i_l \in \{0,1\}$ is an indicator and $S^i_l = 1$ denotes $\mathbf{a}^i$ appears in the image, and vice versa, and $s = \sum_{i=1}^{k} S^i_l$ is the number of the positive attributes. $\Pr(\mathbf{a}^j|\mathbf{a}^i)$ is the conditional probability of two attributes or labels which can be precalculated by accounting the number of their co-occurrence in the dataset. $P^+$ is an $m \times k$ matrix, which can be applied for refining the prediction result of conventional CNNs for the multi-label pedestrian attribute classification problem. Since the co-occurance information $S^j_l$ of any branch $b_l$ is not given in the dataset, Eq. 4 is hard to calculate directly. We therefore propose a novel attribute aware pooling method which first integrates the co-occurance information of every branch $\{b_1, ..., b_m\}$ and the priori knowledge, and then generates the estimation of the whole image by exploiting the co-occurrence priori over different attributes.

## 3 Attribute Aware Pooling

To further explore the correlation between attributes using the priori knowledge in the real data, in this section, we first build conditional probability matrices w.r.t. different attributes using the co-occurrence priori and then develop the new attribute aware pooling method.

### 3.1 Co-occurrence Priori Embedding

Given a training set with $k$ labels and $n$ instances, let $N_i$ denote the number of the $i$-th label occur in the dataset, and $p_i = \Pr(\mathbf{a}^i) = N_i/n$ be the probability of the $i$-th label. By denoting the co-occurrence number of the $i$-th label and the $j$-th label in the dataset as $N_{i,j}$, we can construct a matrix $J \in \mathbb{R}^{k\times k}$, where $J_{i,j} = \Pr(\mathbf{a}^i, \mathbf{a}^j) = N_{i,j}/n$ is the joint probability of $\mathbf{a}^i$ and $\mathbf{a}^j$. Subsequently, we can obtain the conditional probability of an attribute given another attribute as

$$C_{i,j} = \Pr(\mathbf{a}^i|\mathbf{a}^j) = \frac{\Pr(\mathbf{a}^i, \mathbf{a}^j)}{\Pr(\mathbf{a}^j)} = \frac{J_{ij}}{p_j}. \tag{5}$$

After constructing this matrix, we can further estimate the auxiliary probability $P^+_{l,j}$ by exploiting the context information of the $l$-th branch $\mathbf{x}_l$. Fig. 2 shows the matrix $C$ discovered on PA-100K dataset [Liu *et al.*, 2017b], we can see that some attributes frequently co-occur in the dataset, *e.g.* "Female" and "Skirt&Dress", "LongSleeve" and "Trousers".

In the $k$-label classification task, given an instance with $m$ branches $\{b_1, ..., b_m\}$, the softmax classifier outputs $m$ $k$-dimensional vectors, which are stacked into an $m \times k$ matrix $P$ as shown in Eq. 2. For any branch $b_l$, there are $m-1$ branches in $\mathbf{x}$ that surround with $b_l$ and we denote these

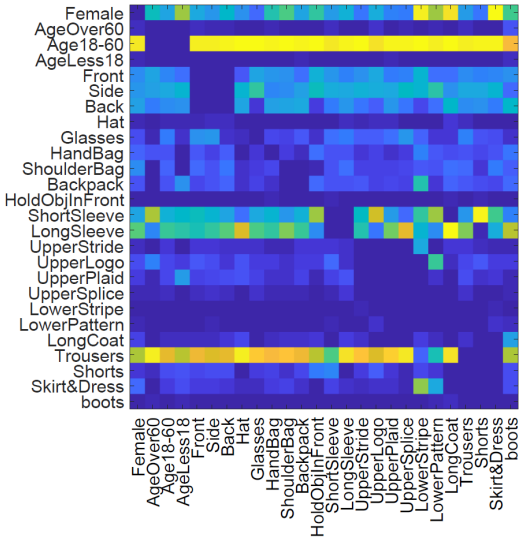


Figure 2: $C$ learned on PA-100K dataset. Darker color means smaller value.

branches as $\tilde{b}_l$. The probability of the occurrence of attribute $\mathbf{a}^j$ in $\tilde{b}_l$ can be calculated by

$$\Pr(\mathbf{a}^j_l) = \Pr(\mathbf{a}^j|b_1, ..., b_{l-1}, b_{l+1}, ..., b_m). \tag{6}$$

However, this high-order posterior probability cannot be accurately calculated. Alternatively, we use the following locally max-pooling as an approximation:

$$Q_{l,j} = \Pr(\mathbf{a}^j_l) \approx \max_{i\neq l} \Pr(\mathbf{a}^j|b_i). \tag{7}$$

Attributes identification in $\tilde{b}_l$ can be easily accomplished through the following threshold function with the parameter $\tau$:

$$S^j_l = \begin{cases} 1, & \text{if } Q_{l,j} > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

and we can use this indicator matrix for calculating Eq. 4. Compared with the output conditional probability $P_{l,j}$ of the $l$-th branch $b_l$ given the $j$-th attribute derived from CNNs, $P^+_{l,j}$ is calculated by exploiting outputs of other branches, and can be taken as another important auxiliary predictor.

However, given the biased recognition of the branches, this hard threshold strategy may bring in inaccurate classification result and the introduction of the hyper-parameter $\tau$ will make $s$ various for different branches. In addition, since attributes have some conflicts, absence of an attribute may lead to the appearance of another one. Therefore, we use the total probability theorem to modify $P^+_{l,j}$ as follow:

$$\begin{aligned} P^+_{l,j} = &\frac{1}{k}\sum_{i=1}^{k} \Pr(\mathbf{a}^i_l)\Pr(\mathbf{a}^j|\mathbf{a}^i) \\ &+ \frac{1}{k}\sum_{i=1}^{k} \Pr(\tilde{\mathbf{a}}^i_l)\Pr(\mathbf{a}^j|\tilde{\mathbf{a}}^i) \end{aligned} \tag{9}$$

where $\tilde{\mathbf{a}}^i$ indicates that $\mathbf{a}^i$ does not occur, and the second term

can be calculated as

$$\begin{aligned}\Pr(\tilde{\mathbf{a}}_l^i)\Pr(\mathbf{a}^j|\tilde{\mathbf{a}}^i) &= \Pr(\tilde{\mathbf{a}}_l^i)\frac{\Pr(\mathbf{a}^j,\tilde{\mathbf{a}}^i)}{\Pr(\tilde{\mathbf{a}}^i)}\\ &= \Pr(\tilde{\mathbf{a}}_l^i)\frac{\Pr(\mathbf{a}^j)-\Pr(\mathbf{a}^j,\mathbf{a}^i)}{1-\Pr(\mathbf{a}^i)}\\ &= (1-\Pr(\mathbf{a}_l^i))\frac{p_j-J_{i,j}}{1-p_i},\end{aligned} \tag{10}$$

and we can first calculate the matrix

$$\tilde{C}_{i,j} = \Pr(\mathbf{a}^i|\tilde{\mathbf{a}}^j) = \frac{p_i - J_{i,j}}{1-p_j} \tag{11}$$

based on the training set in advance for convenience. By exploiting the pre-defined two matrices $C$ and $\tilde{C}$, the additional estimated probability matrix $P^+$ can be simply calculated as

$$P^+ = \frac{1}{k}\left[QC + (\mathbf{1}-Q)\tilde{C}\right], \tag{12}$$

where $\mathbf{1}$ is a $k \times k$ full one matrix.

Eq. 12 is estimated by exploiting context information and co-occurrence tables, which provides another approach to estimate the classification result of the input instance $\mathbf{x}$. The attribute aware pooling method is thus developed to combine the two decision matrices of different properties. Formally, the output of the attribute aware pooling layer is

$$\hat{P} = P + \lambda P^+, \tag{13}$$

where $\lambda$ is the parameter for balancing the prediction results of the branch itself and context information from the other branches. We will further test its impact on the classification accuracy in the following section experimentally.

Finally, we assemble predictions of every branch to form a $k$-dimensional vector. Since every element in $\hat{P}$ is greater than zero, for a given image $\mathbf{x}^i$, the output of the proposed method is

$$\hat{\mathbf{p}}_j^i = \Pr(\mathbf{a}^j|\mathbf{x}^i) = \frac{\max_{l\in[1,m]}\hat{P}_{l,j}}{\sum_{j=1}^k \max_{l\in[1,m]}\hat{P}_{l,j}}, \tag{14}$$

where $M \in \{0,1\}^{m\times k}$ is the max-pooling projection which has only one 1 per column. Let $\mathcal{E}^i = \text{diag}(\hat{P}^T M)$, and we have

$$\hat{\mathbf{p}}^i = \frac{\mathcal{E}^i}{||\mathcal{E}^i||_1}, \tag{15}$$

where $||\cdot||_1$ is the conventional $\ell_1$ norm for vectors. The ground-truth conditional probabilities of $\mathbf{x}^i$ is $\mathbf{p}^i = \mathbf{y}^i/||\mathbf{y}^i||_1$. Therefore, the loss function of the network using the proposed method is

$$\begin{aligned}J =& \frac{1}{2n}\sum_{i=1}^n\sum_{j=1}^k\left(\hat{\mathbf{p}}_j^i - \mathbf{p}_j^i\right)^2\\ =& \frac{1}{2n}\sum_{i=1}^n||\hat{\mathbf{p}}^i - \mathbf{p}^i||_2^2.\end{aligned} \tag{16}$$

By optimizing the above loss function using the mini-batch SGD approach, we can fine-tune filters in the neural network $\mathcal{N}$ and obtain a new CNN with improvement in attribute recognition accuracy. The diagram of the proposed attribute aware pooling architecture is shown in Fig. 1.

### 3.2 Back Propagation

We proceed to introduce the optimization method for CNNs with the proposed attribute aware pooling method. Stochastic gradient descent (SGD) needs to calculate the gradient of the loss function and then utilizes the back propagation strategy for updating parameters. We next detail the gradient and the back propagation of the proposed method.

In fact, the proposed attribute aware pooling method is attached after the softmax layer of a general CNN on $m$ branches and followed by some pooling and probability calculations as shown in Fig. 1. Therefore, we do not have to modify any additional convolution layer and filters, and we only need to calculate gradient of the loss function in the last layer w.r.t. the individual branch estimation matrix $P$ in Eq. 2. Commonly, this gradient can be computed using the standard chair rule, *i.e.*

$$\frac{\partial J}{\partial P} = \frac{\partial J}{\partial \hat{\mathbf{p}}}\cdot\frac{\partial \hat{\mathbf{p}}}{\partial \mathcal{E}}\cdot\frac{\partial \mathcal{E}}{\partial \hat{P}}\cdot\frac{\partial \hat{P}}{\partial P}. \tag{17}$$

We first calculate the gradient of the first two terms for a given instance $\mathbf{x}$ as follow:

$$\frac{\partial J}{\partial \mathcal{E}} = \frac{\hat{\mathbf{p}}-\mathbf{p}}{||\mathcal{E}||_1} - \frac{(\hat{\mathbf{p}}-\mathbf{p})^T\mathcal{E}}{||\mathcal{E}||_1^2}, \tag{18}$$

where $||\mathcal{E}||_1 = \sum_{j=1}^k|\mathcal{E}_j|$, and $\mathcal{E}\in\mathbb{R}^{k\times 1}$. Subsequently, the gradient of $\hat{P}$ is calculated as follow:

$$\frac{\partial J}{\partial \hat{P}} = M\ \text{diag}\left(\frac{\hat{\mathbf{p}}-\mathbf{p}}{||\mathcal{E}||_1} - \frac{(\hat{\mathbf{p}}-\mathbf{p})^T\mathcal{E}}{||\mathcal{E}||_1^2}\right), \tag{19}$$

where the variables in $\text{diag}(\cdot)$ make up a $k$-dimensional vector and $M$ is the max-pooling projection which maps the gradient into a $m \times k$ sparse matrix.

As for the gradient of $P$, consider $Q_l = \text{diag}(P^T S_l)$ where $S_l \in \{0,1\}^{m\times k}$ is the $l$-th mask w.r.t. the locally max-pooling (Eq. 7) for generating the $l$-th row of $Q$. Elements in the $l$-th row of $S_l$ are zeros. The $l$-th row of $\hat{P}$ can be reformulated as

$$\begin{aligned}\hat{P}_l &= P_l + \lambda P_l^+ = P_l + \lambda(Q_l C + (1-Q_l)\tilde{C})\\ &= P_l + \frac{\lambda}{k}\left[\text{diag}(P^T S_l)C + (\mathbf{1}-\text{diag}(P^T S_l))\tilde{C}\right].\end{aligned} \tag{20}$$

Removing terms irrelevant to $P$, the $j$-th element in $P_l^+$ can be simplified as

$$\begin{aligned}P_{l,j}^+ &= \text{diag}(P^T S_l)C_j - \text{diag}(P^T S_l)\tilde{C}_j\\ &= \text{Tr}(P^T S_l\ \text{diag}(C_j)) - \text{Tr}(P^T S_l\ \text{diag}(\tilde{C}_j)),\end{aligned} \tag{21}$$

where $C_j$ and $\tilde{C}_j$ are the $j$-th columns in $C$ and $\tilde{C}$, respectively. The gradient of $P_{l,j}^+$ to $P$ is

$$\begin{aligned}\frac{\partial P_{l,j}^+}{\partial P} &= S_l\ \text{diag}(C_j) - S_l\ \text{diag}(\tilde{C}_j)\\ &= S_l\ \text{diag}(C_j - \tilde{C}_j).\end{aligned} \tag{22}$$

Therefore, the gradient of the classification loss w.r.t. $P$ is

$$\frac{\partial J}{\partial P} = \frac{\partial J}{\partial \hat{P}} + \frac{\lambda}{k}\sum_{l=1}^m\sum_{j=1}^k\frac{\partial J}{\partial \hat{P}_{l,j}}S_l\ \text{diag}(C_j - \tilde{C}_j), \tag{23}$$

**Algorithm 1** Attribute aware pooling method for pedestrain attribute classification with CNNs.

**Input:** The multi-branch CNN $\mathcal{N}$, a given image $\mathbf{x}$ and its label vector $\mathbf{y}$. Two co-occurrence matrices $C$ and $\tilde{C}$, weight parameter $\lambda$, and learning rate $\eta$.

**Feed Forward**
Calculate conditional probabilities from the $m$ branches and constitute $P$ (Eq. 2);
**for** $l = 1$ to $m$ **do**
  Locally max-pooling: $Q_l \leftarrow \max_{i \neq l} P_i$ (Eq. 7);
  Estimate $P_l^+$ using the co-occurrence priori: $P_l^+ \leftarrow (Q_l C + (1 - Q_l)\tilde{C})/k$ (Eq. 12);
  Form the estimation: $\hat{P}_l \leftarrow P_l + \lambda P_l^+$ (Eq. 13);
**end for**
Calculate the overall estimation $\hat{\mathbf{p}}$ of $\mathbf{x}$ (Eq. 14);
**Back Propogation**
$\nabla(\hat{\mathbf{p}}) \leftarrow \hat{\mathbf{p}} - \mathbf{p}$;
$\nabla(\mathcal{E}) \leftarrow \nabla(\hat{\mathbf{p}})/(\mathcal{E}^T\mathbf{1}) - (\nabla(\hat{\mathbf{p}})^T\mathcal{E})/(\mathcal{E}^T\mathbf{1})^2$;
$\nabla(\hat{P}) \leftarrow M\ \mathrm{diag}(\nabla(\mathcal{E}))$;
$\nabla(P) \leftarrow \nabla(\hat{P})$;
**for** $l = 1$ to $m$ **do**
  **for** $j = 1$ to $k$ **do**
    $\nabla(P) \leftarrow \nabla(P) + \frac{\lambda}{k}\nabla(\hat{P}_{l,j})S_l\ \mathrm{diag}(C_j - \tilde{C}_j)$ (Eq. 23);
  **end for**
**end for**
$P \leftarrow P - \eta\nabla(P)$;

**Output:** The classification result $\hat{\mathbf{p}}$ and $\nabla(P)$.

and $P$ can be updated as

$$P = P - \eta\frac{\partial J}{\partial P}, \tag{24}$$

where $\eta$ is the learning rate. Alg. 1 summarizes the feed forward and the back propagation of the proposed method.

# 4 Experiments

Here we will present the setting of experiments for evaluating the proposed method and compare it with the state-of-the-art methods on benchmark pedestrian attribute recognition datasets.

## 4.1 Datasets and Metrics

The evaluation is conducted on three largest publicly available pedestrian attribute datasets:

- PETA [Deng *et al.*, 2014]: The PEdesTrian Attribute dataset consists of 19,000 pedestrian images with 65 attributes (61 binary and 4 multi-class). Following [Li *et al.*, 2015; Zhao *et al.*, 2018], we randomly divide the whole dataset into three partitions: 9,500 images for training, 1,900 for validation, and 7,600 for testing, and focus on the 35 attributes whose positive proportions are bigger than 1/20.
- RAP [Li *et al.*, 2016]: The Richly Annotated Pedestrian attribute dataset contains 41,585 pedestrian images with 72 attributes (69 binary and 3 multi-class). It is split into 33,268 images for training and the remaining 8,317 for testing. The same 51 binary attributes are evaluated for fair comparison following [Li *et al.*, 2016].
- PA-100K [Liu *et al.*, 2017b]: The PA-100K dataset has 100,000 images with 26 commonly used binary attributes. The whole dataset is randomly split into training, validation and test sets with a ratio of $8:1:1$.

We use five evaluation metrics including a class-centric metric mean accuracy (mA), and four instance-centric metrics, *i.e.* accuracy, precision, recall and F1-score, following [Li *et al.*, 2016; Liu *et al.*, 2017b; Zhao *et al.*, 2018]

## 4.2 Implementation Details

In the proposed method, there are $m$ branches for attribute prediction. We use ResNet-50 [He *et al.*, 2016] as the backbone network for building our multi-branch architecture. The main body of ResNet-50 except the last residual block is shared and the the last block is adopted as the architecutre of each branch. For a $448 \times 224$ input image, the feature maps $F$ from the shared CNN is of size $2048 \times 14 \times 7$. Specially, we design 4 branches in our model and the whole feature $F$ are fed into the first branch. Inspired by [Yi *et al.*, 2014; Sun *et al.*, 2018], we spatially partition the feature maps $F$ into 3 parts with sizes of $4 \times 7$, $6 \times 7$ and $6 \times 7$, serving for the other three branches, as shown in Fig 3. Each part focuses on different human position, *i.e.* head, upper body and lower body, for position-aware attribute recognition.

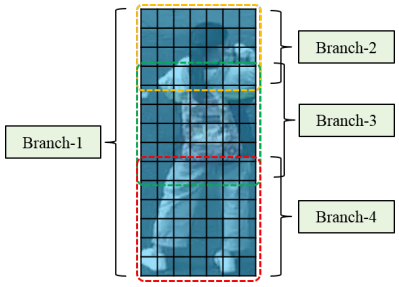


Figure 3: Feature maps parition for multi-branch architecture.

In our method, we first counted occurrence and co-occurrence numbers of different categories among train images and built two matrices $C$ and $\tilde{C}$ for each dataset. The backbone model parameters are initialized by directly using the pre-trained models on ILSVRC 2012 [Russakovsky *et al.*, 2015]. Adam optimizer [Kingma and Ba, 2014] with a batch size of 32 is used to fine-tune the entire model with an initial learning rate of 0.01. The images are resized to $512 \times 256$ and during training randomly cropped to the size of $448 \times 224$ with random horizontal flippling. All methods were implemented using PyTorch [Paszke *et al.*, 2017] and run on NVIDIA V100 graphics cards. We denote the multi-branch architecture with the proposed attribute aware pooling method as CoCNN.

The proposed CNN architecture with the attribute aware pooling method has only one hyper-parameter $\lambda$ whose functionality is to combine the original CNN output and the es-

| Method \ Dataset | PETA | | | | | RAP | | | | | PA-100K | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mA | acc | prec | recall | F1 | mA | acc | prec | recall | F1 | mA | acc | prec | recall | F1 |
| ACN [Sudowe *et al.*, 2015] | 81.15 | 73.66 | 84.06 | 81.26 | 82.64 | 69.66 | 62.61 | 80.12 | 72.26 | 75.98 | - | - | - | - | - |
| DeepMAR [Li *et al.*, 2015] | 82.60 | 75.07 | 83.68 | 83.14 | 83.41 | 73.79 | 62.02 | 74.92 | 76.21 | 75.56 | 72.7 | 70.39 | 82.24 | 80.42 | 81.32 |
| JRL [Wang *et al.*, 2017] | 82.13 | - | 82.55 | 82.12 | 82.02 | 74.74 | - | 75.08 | 74.96 | 74.62 | - | - | - | - | - |
| HP-net [Liu *et al.*, 2017b] | 81.77 | 76.13 | 84.92 | 83.24 | 84.07 | 76.12 | 65.39 | 77.33 | 78.79 | 78.05 | 74.21 | 72.19 | 82.97 | 82.09 | 82.53 |
| CTX C-RNN [Li *et al.*, 2017] | 80.13 | - | 79.68 | 80.24 | 79.68 | 70.13 | - | 71.03 | 71.20 | 70.23 | - | - | - | - | - |
| SR C-RNN [Liu *et al.*, 2017a] | 82.83 | - | 82.54 | 82.76 | 82.65 | 74.21 | - | 75.11 | 76.52 | 75.83 | - | - | - | - | - |
| LG-Net [Liu *et al.*, 2018] | - | - | - | - | - | 78.68 | 68.00 | 80.36 | 79.82 | 80.09 | 79.96 | 75.55 | 86.99 | 83.17 | 85.04 |
| VAA [Sarafianos *et al.*, 2018] | 84.59 | 78.56 | 86.79 | 86.12 | 86.46 | - | - | - | - | - | - | - | - | - | - |
| GRL [Zhao *et al.*, 2018] | 86.70 | - | 84.34 | **88.82** | 86.51 | 81.20 | - | 77.70 | **80.90** | 79.29 | - | - | - | - | - |
| Baseline | 84.68 | 78.89 | 85.38 | 86.41 | 85.83 | 79.67 | 66.01 | 77.93 | 79.25 | 78.58 | 78.12 | 74.11 | 84.42 | 84.09 | 84.25 |
| Multi-branch | 85.60 | 79.07 | 86.05 | 87.26 | 86.65 | 80.54 | 66.84 | 79.06 | 79.64 | 79.35 | 79.26 | 76.20 | 86.44 | 84.08 | 85.24 |
| CoCNN | **86.97** | **79.95** | **87.58** | 87.73 | **87.65** | **81.42** | **68.37** | **81.04** | 80.27 | **80.65** | **80.56** | **78.30** | **89.49** | **84.36** | **86.85** |

Table 1: Evaluation of CoCNN on PETA, RAP and PA-100K datasets with **bold** best result and underline the second best result, compared with previous benchmark methods. - represents no reported result available.

timation from the context information. We tested the impact of $\lambda$ by tuning it from 0 to 0.5 with the step of 0.05 on the mA metric using the PETA dataset. In fact, a larger $\lambda$ makes the result $\hat{\mathbf{p}}$ more inclined to the auxiliary estimation $P^+$ and vice versa. As a result, we obtained a 86.97% mA value at $\lambda = 0.2$ which is the best trade-off between estimations in Eq. 13. Moreover, for RAP and PA-100K datasets, we keep $\lambda = 0.2$ in the following experiments.

### 4.3 Results and Analysis

**Effect of Multi-branch Architecture and AAP**

The improvement of CoCNN mainly comes from two aspects, *i.e.* multi-branch architecture and attribute aware pooling. In this section, we will measure how much improvement these two aspects bring. The vanilla ResNet-50 with binary cross entropy loss is adopted as baseline model, and the multi-branch ResNet-50 without attribute aware pooling is denoted as multi-branch model. From the results in Tab. 1, we can see that multi-branch architecture improve the metrics generally in the three datasets by focusing on different parts of the human body. With attribute aware pooling, CoCNN further improves the accuracy to higher level.

Two examples from the PA-100K dataset are given in Fig. 4 for qualitative analysis. From the results, we found that multi-branch model made wrong predictions "Skirt&Dress" and "AgeLess18" for the two pedestrians, respectively. By exploiting their correlation with other attributes, especially the "Female" in the left person, and the "UpperPlaid" and "Backpack" in the right example, CoCNN well corrected the wrong predictions in multi-branch model.

**Comparison with State-of-the-art Methods**

After verifing the effect of the proposed method, we compared the proposed approach with several state-of-the-art approaches, *e.g.* DeepMAR [Li *et al.*, 2015], HP-net [Liu *et al.*, 2017b], SR C-RNN [Liu *et al.*, 2017a], VAA [Sarafianos *et al.*, 2018], and GRL [Zhao *et al.*, 2018]. Results of comparison models are mostly reported from their papers directly. The results are listed in Tab. 1.

From results on PETA dataset, we found that, CoCNN clearly outperformed other state-of-the-art methods and achieved the highest mA, acc, prec and F1 values (86.97, 79.95, 87.65). The recall of CoCNN is the second best one. This improvement mainly comes from that the proposed method takes multiple attributes into account when generating the classification annotation.

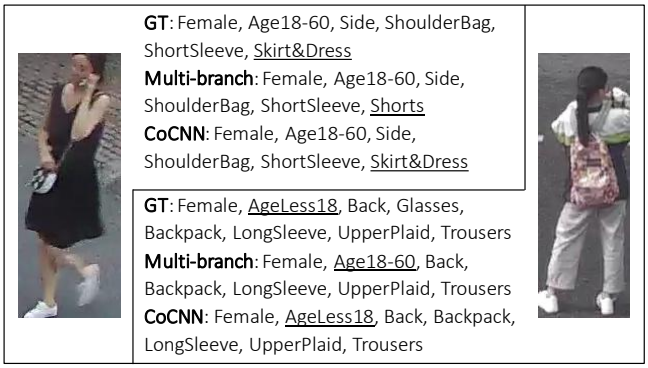


Figure 4: Qualitative results from PA-100K dataset of CoCNN and the multi-branch model. Wherein, GT means groundtruth, underlined tags are attributes which are needed to be noticed.

In addition, similar observation also can be seen on RAP and PA-100K datasets. This phenomenon further illustrates that the proposed attribute aware pooling method is a general auxiliary regularization for enhancing pedestrian attribute recognition and can be embedded into any similar task.

## 5 Conclusions

Existing vanilla CNNs cannot be effectively applied to handle the pedestrian attributes recognition task. Therefore, we propose using the co-occurrence priori for improving performance of CNNs, namely, attribute aware pooling. With a multi-branch CNN as base architecture, we first construct two co-occurrence tables through training sets and use context information of every branch to complete the decision derived from each branch itself, which not only excavates properties of different human parts but also investigates the relationship between different branch. Experiments conducted on several benchmark datasets show that the performance of the proposed method is better than those of state-of-the-art methods in terms of the commonly used metrics.

## References


[Boguraev and Ando, 2005] Branimir Boguraev and Rie Kubota Ando. Timeml-compliant text analysis for temporal reasoning. In *IJCAI*, 2005.

[Deng *et al.*, 2014] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *ACM MM*, 2014.

[Guo and Gu, 2011] Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In *IJCAI*, 2011.

[Han *et al.*, 2018a] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-aware attention model for fine-grained representation learning. In *ACM MM*, 2018.

[Han *et al.*, 2018b] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *ICASSP*. IEEE, 2018.

[Hand and Chellappa, 2017] Emily M Hand and Rama Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI*, pages 4068–4074, 2017.

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

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Layne *et al.*, 2012] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Q Mary. Person re-identification by attributes. In *Bmvc*, volume 2, page 8, 2012.

[Li *et al.*, 2015] Dangwei Li, Xiaotang Chen, and Kaiqi Huang. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In *ACPR*, 2015.

[Li *et al.*, 2016] Dangwei Li, Zhang Zhang, Xiaotang Chen, Haibin Ling, and Kaiqi Huang. A richly annotated dataset for pedestrian attribute recognition. *arXiv preprint arXiv:1603.07054*, 2016.

[Li *et al.*, 2017] Yao Li, Guosheng Lin, Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Sequential person recognition in photo albums with a recurrent network. In *CVPR*, 2017.

[Liu *et al.*, 2017a] Feng Liu, Tao Xiang, Timothy M. Hospedales, Wankou Yang, and Changyin Sun. Semantic regularisation for recurrent image annotation. In *CVPR*, July 2017.

[Liu *et al.*, 2017b] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *ICCV*, 2017.

[Liu *et al.*, 2018] Pengze Liu, Xihui Liu, Junjie Yan, and Jing Shao. Localization guided learning for pedestrian attribute recognition. In *BMVC*, 2018.

[Lu *et al.*, 2017] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, volume 1, page 6, 2017.

[Luo *et al.*, 2018] Yong Luo, Yonggang Wen, and Dacheng Tao. Heterogeneous multitask metric learning across multiple domains. *IEEE T-NNLS*, 29(9):4051–4064, 2018.

[Luo *et al.*, 2019] Yong Luo, Yonggang Wen, Tongliang Liu, and Dacheng Tao. Transferring knowledge fragments for learning distance metric from a heterogeneous domain. *IEEE T-PAMI*, 41(4):1013–1026, 2019.

[Mensink *et al.*, 2014] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014.

[Modiri Assari *et al.*, 2014] Shayan Modiri Assari, Amir Roshan Zamir, and Mubarak Shah. Video classification using semantic concept co-occurrences. In *CVPR*, 2014.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[Peng *et al.*, 2016] Peixi Peng, Yonghong Tian, Tao Xiang, Yaowei Wang, and Tiejun Huang. Joint learning of semantic and latent attributes. In *ECCV*, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[Sarafianos *et al.*, 2018] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Deep imbalanced attribute classification using visual attention aggregation. In *ECCV*, 2018.

[Sudowe *et al.*, 2015] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. Person attribute recognition with a jointly-trained holistic cnn model. In *ICCV Workshops*, 2015.

[Sun *et al.*, 2018] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, 2018.

[Wang *et al.*, 2013] Yunhe Wang, Miaojing Shi, Yuan Gao, and Chao Xu. Visual words refining exploiting spatial co-occurrence table. In *2013 IEEE Global High Tech Congress on Electronics*, pages 99–104. IEEE, 2013.

[Wang *et al.*, 2016] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. Cnnpack: Packing convolutional neural networks in the frequency domain. In *NeurIPS*, 2016.

[Wang *et al.*, 2017] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Attribute recognition by joint recurrent learning of context and correlation. In *ICCV*, Oct 2017.

[Wang *et al.*, 2018] Yunhe Wang, Chang Xu, XU Chunjing, Chao Xu, and Dacheng Tao. Learning versatile filters for efficient convolutional neural networks. In *NeurIPS*, 2018.

[Xiao *et al.*, 2016] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.

[Yi *et al.*, 2014] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *ICPR*, 2014.

[Zhao *et al.*, 2018] Xin Zhao, Liufang Sang, Guiguang Ding, Yuchen Guo, and Xiaoming Jin. Grouping attribute recognition for pedestrian with joint recurrent learning. In *IJCAI*, 2018.

[Zhu *et al.*, 2015] Jianqing Zhu, Shengcai Liao, Dong Yi, Zhen Lei, and Stan Z Li. Multi-label cnn based pedestrian attribute learning for soft biometrics. In *ICB*. IEEE, 2015.