Pattern Analysis and Applications
https://doi.org/10.1007/s10044-018-0725-0

THEORETICAL ADVANCES

Pedestrian gender classification using combined global and local parts-based convolutional neural networks

Choon-Boon Ng[1] · Yong-Haur Tay[1] · Bok-Min Goi[1]

Received: 12 November 2017 / Accepted: 9 July 2018

Abstract
The identification of a person's gender plays an important role in various visual surveillance and monitoring applications which are growing more ubiquitously. This paper proposes a method for gender classification of pedestrians based on whole body images which, unlike facial-based methods, allows for observation from different viewpoints. We use a parts-based model that combines global and local information to make inference. Convolutional neural network (CNN) is leveraged for its superior feature learning and classification capability. Our method requires that only the gender label is available for the training images, without the need for any other expensive annotation such as the anatomical parts, key points or other attributes. We trained a CNN on the bounding box containing the whole body (global CNN) or a defined portion of the body (local CNN). Experimental results show that the upper half region of the body is the most discriminative for gender, in comparison with the middle or lower half. The best model is a jointly trained combination of a global CNN and a local upper body CNN, which achieves higher accuracy than previous works on publicly available datasets.

Keywords Gender classification · Gender recognition · Sex identification · Convolutional neural network · Deep learning · Pedestrian

1 Introduction

Gender classification using computer vision-based methods is an active area of research. Gender is one of the significant attributes of people, which can play an important role in many applications such as surveillance, demographics data collection, intelligent billboard and human–computer interaction. Most research works have concentrated on investigating how to infer gender from the face of a person [17]. These methods have been able to achieve a high accuracy level, especially in constrained environments with frontal or near-frontal view of the head.

In recent years, there has been an increased interest in classifying the gender of pedestrians from whole body images. This is particularly so for unconstrained situations when the person's face may not be not visible, i.e. from the back view, which is common in surveillance and monitoring imagery. Even if visible, the face may not have sufficient resolution, especially when the system is designed to be non-intrusive. Thus, a gender classification system based on facial cues alone will fail under these circumstances.

Classifying gender (which, in our work, we consider to be only male or female, but not other classes such as transgender) is a binary classification problem. While it appears to be a straightforward task, one which humans seem to perform quite easily, there are a few problems associated with it. The visual cues to gender may not always be present or tend to be contradictory. For example, biological differences between men and women make the body shape a vital cue, but it may be concealed by loose fitting or thick layers of clothes. Sartorial tastes may be significantly different, but certain types of clothing or accessories such as cap, pants and T-shirt are worn by both genders. For back view, hairstyle can provide a useful cue, but it is not always dependable. These factors, in addition to different viewpoints and poses, lead to large intraclass variations which make it challenging from the machine learning aspect.

Previous works have approached the problem using various engineered feature descriptors that extract information, such as shape, texture, color and edges, from the image of a

✉ Choon-Boon Ng
ngcb@utar.edu.my

[1] Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Kajang 43000, Selangor, Malaysia

Published online: 31 July 2018

person [2, 6, 11]. The extracted features were used to train powerful supervised learning algorithms. Recently, with the rise of deep learning, feature learning, in particular via convolutional neural networks (CNN), has also achieved good results [1, 16]. The works based on CNN used the whole body image, i.e. global information, as the input to the network. On the other hand, gender information can be localized in body parts [2, 15], so we propose to also incorporate local information.

The technical contribution of our work is a novel parts-based framework for pedestrian gender classification that combines both global and local information to make inference. With the aid of spatial jittering, we used a simple method of extracting local information that does not require costly annotation of body parts, key points or other attributes. It also does not require the use of a body parts detector. We employed CNNs in our framework, which were jointly trained using images from publicly available datasets with only the gender labeled. These advantages make our method more scalable, while achieving better classification accuracy than previous methods.

The other contributions of our work are:

- We compared the accuracy of different regions corresponding to the upper, middle and lower half of the body. Our results show that the upper half body region is the most discriminative for gender when compared to the middle or lower half.
- We showed that by combining global and local information, classification accuracy can be increased compared to using only global or local information alone.
- We gained more understanding on what gender information is learnt by a trained CNN classifier by using feature visualization.

The paper is organized as follows. In the next section, a brief overview of the state of the art is presented. Section 3 describes the proposed parts-based classification framework. Section 4 reports on the experimental results and discussion, feature visualization, error analysis and the effect of image resolution. Finally, in Sect. 5, we present our conclusion.

2 Related work

Cao et al. [2] first studied the classification of gender using the image of a person's whole body, using a modified Histogram of Oriented Gradients (HOG) for feature representation. HOG is a technique which was popular for human detection, arising from the work of Dalal and Triggs [5]. HOG feature vectors were extracted from partitions in the image (using smaller and overlapping patches compared to the original technique) and acted as features to train weak classifiers. They considered each patch to correspond to some body part, so instead of concatenating the feature vectors, they used each part individually in a combined boosting-type classifier with weighted voting. This method obtained better results than using the original HOG features, raw pixels or Canny edge maps.

Collins et al. [4] investigated the suitability of feature representations such as Pyramid Histogram of Orientation Gradients and Pyramid Histogram of Words, which are, respectively, shape and appearance descriptors. Then, they proposed dense HOG features which were obtained from a custom edge map of the image. They combined this with a local Hue-Saturation-Value (HSV) color histogram feature which captures color information in the image. This was found to give the best performance. They used support vector machine (SVM) as the classifier, experimenting with different kernels.

Geelen et al. [6] made an in-depth comparison of several features, HOG, local binary patterns (LBP) and HSV, along with different classification methods, SVM and Random Forests (RF). Unlike HOG, which contains spatial shape information, LBP was developed as a texture descriptor [19]. The best accuracy was achieved using a combination of HSV and LBP features with a hybrid SVM-RF classifier.

Guo et al. [11] combined biologically inspired features with manifold learning for dimension reduction, using SVM as classifier. The features were derived by applying Gabor filters on the input image followed by an operation taking the maximum values in the local spatial neighborhoods. This feature extraction method, introduced by Riesenhuber and Poggio [21], is based on the hierarchical visual processing model in the mammalian cortex. To improve accuracy, they included a view classifier in their framework, which classifies the images into front, back or mixed view. This is followed by a separate gender classifier for each view.

Instead of the whole body image, Li et al. [15] proposed using only the head–shoulder region. HOG, LBP and Gabor filters were employed as feature descriptors to extract gradient, texture and orientation information. A discriminative subspace was learnt using Partial Least Squares (PLS) method. An SVM classifier was then trained on the reduced dimension features. Meanwhile, Khan [13] proposed combining information from the whole body, upper body and face regions. Their approach relied on using pre-trained upper body and face detectors to determine the bounding box of these regions, from which multiple features were extracted in a three-level spatial pyramid scheme.

As opposed to handcrafted features of previous works mentioned above, Ng et al. [16] trained a CNN for pedestrian gender classification. In recent years, CNN has gained tremendous popularity due to its superior achievement in many pattern recognition problems, starting with the breakthrough results in object recognition by Krizhevsky et al. [14]. A CNN learns the appropriate features for the problem from the training images, integrating feature extraction and classification in a single framework. Their gender classification network consisted of two convolutional weight layers (with ten and 20 feature maps per layer) followed by a fully connected layer of 25 neuron units. Despite the small network of about 65k weight parameters, the performance was comparable to previous works using hand-engineered feature extraction methods. Later, Antipov et al. [1] made a comparison between handcrafted features and features learnt by CNN, with the results showing that learnt features performed with better generalization for heterogeneous data.

To improve on their previous work, Ng et al. [18] used larger CNNs whose architecture parameters were found using a random search. To address the limitation caused by the small amount of labeled training data, they proposed a pre-training strategy. The weights in the first convolutional layer were initialized with filters learnt by *k*-means clustering on a separate pedestrian dataset and then trained in a related task, pedestrian classification. The pre-trained CNN was then fine-tuned for pedestrian gender classification. This strategy gave better results when compared to random initialization of the CNN weights.

As the previous works using CNN considered only the whole body image as the input, i.e. global information, we propose to combine with local information obtained from different regions of the body, which could be the upper half, middle or lower half. Unlike [13], we do not specifically attempt to locate the upper body and face of the person, hence relieving from the need of accurate body parts detectors. Other than the gender label, our method also does not require any other training data annotation such as parts, keypoints or other attributes which would be an expensive and laborious process to obtain for a large number of images.

3 Classification framework

The aim of the classifier is to determine the gender of the pedestrian in an image. It does so based on the bounding box of a person received from the pedestrian detector in the system.

In our parts-based classification framework, as shown in Fig. 1, we propose a CNN-based model for gender classification of pedestrians based on both global and local information, utilizing patches cropped from an image. We assume that the person in the image is upright and approximately centered (as obtained from the output of the pedestrian detector). Patches are extracted from positions that would correspond to different regions of the body.

Given the image of a pedestrian, a slightly smaller-sized central crop of the image (the numerical details will be provided in the next section) is extracted as the global patch to train the CNN to make inference from the whole body of the person. The reason for using a slightly smaller crop is to allow for spatial jittering during training of the CNN. In this technique, instead of extracting the patch from its nominal location, random horizontal and vertical shifts are applied, thus artificially increasing the amount of training data. This form of data augmentation has commonly been applied in practice to reduce overfitting. Furthermore, in our case, it acts naturally to mitigate the problem of not having the person exactly centered in the image. Thus, it allows the position of the person to be slightly off-center, which lightens the pedestrian detector's localization accuracy requirement.

Different patches are also extracted from the image to train the CNN for inference based on local information. We experimented with local patches extracted from up to three different fixed locations at the upper half, lower half and middle section of the image. In our work, the middle section patch overlaps with the upper half and lower half patches. The upper half of the image corresponds roughly to the head, shoulder and chest region of a person. The middle section contains the portion surrounding the waist and hips, while

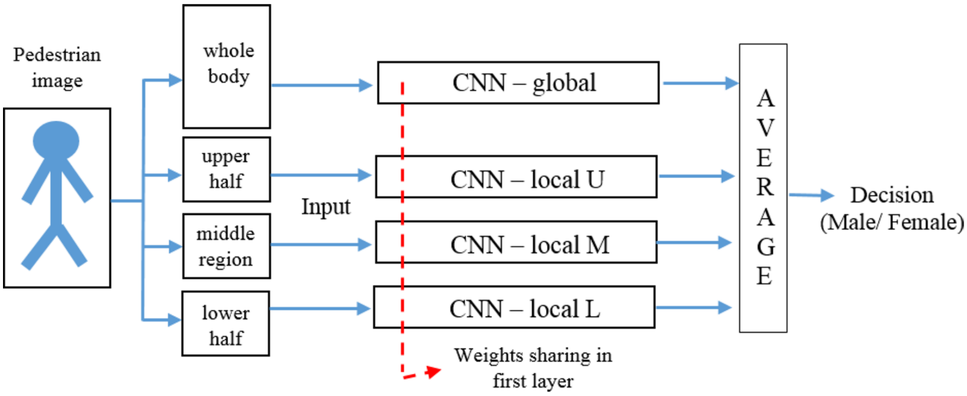

Fig. 1 Classification framework

the lower half includes the legs and feet. The local patches are also extracted such that it allows for spatial jittering during training.

We take advantage of the fact that a person who is walking or standing upright will have their body parts constrained in the local regions mentioned above. The simplicity of this approach means that time-consuming, laborious and expensive body parts or key points labeling of the training images is not required. The system also does not need to make any attempt in detecting, locating and pose normalizing the body parts of the person during the classification process.

As shown in Fig. 1, each global and local patch is used as the input to a CNN. In particular, there is one CNN for each patch. The networks are jointly trained to classify the gender of pedestrians based on each of its individual input patches. The information from each CNN is combined to make the final inference, by taking the average of the probabilities output from each CNN, with the decision taken based on the class with the higher probability.

In our framework, we used the same network architecture for each of the global and local CNNs. We experimented with several different architectures, which will be described in the next section.

3.1 CNN architecture

We experimented with various standard CNN architectures—VGG-19 [24], BAIR Reference CaffeNet [12] and the CNNs used for pedestrian gender classification from Ng et al. [18]. Our explanation of the CNN structure will be based on these.

In the work of [18], the CNN architecture was selected based on a random search for hyperparameters optimization. We chose the top three networks in that work for our experiment, which we refer to here as CNN-1, CNN-2 and CNN-3. Their architecture details are shown in Table 1. We briefly explain the CNN architecture, which consists of several layers of convolutional filtering and subsampling operation. Starting from the input feature maps, which in this case is an RGB image in the form of a three-channel array, the output from the convolution filtering operation will be passed through a nonlinear activation function to obtain a set of output feature maps C_j given by:

$$C_j = \sigma\left(\sum_{i \epsilon S} W_{i,j} \otimes I_i + b_j\right) \tag{1}$$

In the equation, $\otimes$ denotes the convolution operation of the j-th filter $W_{i,j}$ on the i-th input feature map I_i, with b_j being a trainable bias. The set of the input feature maps is denoted by S. The nonlinear activation function, σ, uses the popular rectified linear unit function (ReLU) [9], which is mathematically expressed by the operation $f(x) = \max(0, x)$. In Table 1, $a \times b \times c$ filters denote the number of filters × filter height × filter width. The number of output feature maps equals the number of filters.

The output feature maps will be reduced further in size by the subsampling layer operation. The max pooling operation [23] is used, which partitions each feature map into square pxp local neighborhoods and then takes the maximum value in each neighborhood to form the subsampled feature map.

Note that for the networks shown in Table 1, the filter strides are all equal to 1 while the pooling strides equal the pooling sizes (non-overlap pooling). For example, with CNN-2, the stride for the pooling operation in layers 2 and 4 is 2 and 3, respectively. Furthermore, no zero padding is applied on the feature maps at any layer.

Each fully connected (FC) layer is a single layer of perceptrons, where each of its neuron unit is connected to all the units in the preceding layer. Dropout [26] with a factor of 0.5 is applied in the fully connected layers to reduce overfitting. In this regularization scheme, a fraction of the activations, as given by the factor, is set to zero during training. In the forward pass during classification, all the weights in the layer will be scaled by the same factor.

The final output layer consists of soft-max activation units. In this case, which is a binary classification problem, there are two units to represent the male and female

Table 1 CNN architectures for pedestrian gender classification from [18]

Layer	CNN-1	CNN-2	CNN-3
1	Conv with 50 × 3 × 3 filters	Conv with 25 × 5 × 5 filters	Conv with 50 × 5 × 5 filters
2	Max pooling 2 × 2	Max pooling 2 × 2	Max pooling 2 × 2
3	Conv with 80 × 3 × 3 filters	Conv with 100 × 5 × 5 filters	Conv with 80 × 3 × 3 filters
4	Max pooling 2 × 2	Max pooling 3 × 3	Max pooling 3 × 3
5	Conv with 120 × 3 × 3 Filters	FC-50	Conv with 120 × 3 × 3 filters
6	FC-100	FC-2	Max pooling 2 × 2
7	FC-2	Soft-max	FC-100
8	Soft-max		FC-2
9			Soft-max

classes. The output value, which represents the probability of the class given an input, is calculated as follows:

$$f_i(x) = \frac{\exp(x_i)}{\sum_{i=1,2} \exp(x_i)} \tag{2}$$

BAIR Reference CaffeNet and VGG-19 are off-the-shelf deep CNNs which were chosen for the experiments because they achieved state-of-the-art results for general object recognition, in particular the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [22]. CaffeNet is a replication of the AlexNet architecture [14] that was trained to achieve similar accuracy for ILSVRC2012. Details of these architectures, which contain similar structural components already mentioned above, can be found in their respective papers. One architectural difference that we experimented with was the width of the fully connected layers, where we tried several different values, specifically 4096 (original), 2048, 1024 and 512. Smaller values reduce the number of free parameters during training, which may be beneficial since the CNNs were originally trained with a significantly larger number of images.

For comparison in terms of depth, CaffeNet and VGG-19 have 8 and 19 weight layers, respectively, while for CNN-1/2/3, the depth is only 4 or 5 weight layers. CaffeNet and VGG-19 also have more feature maps in each convolutional layer.

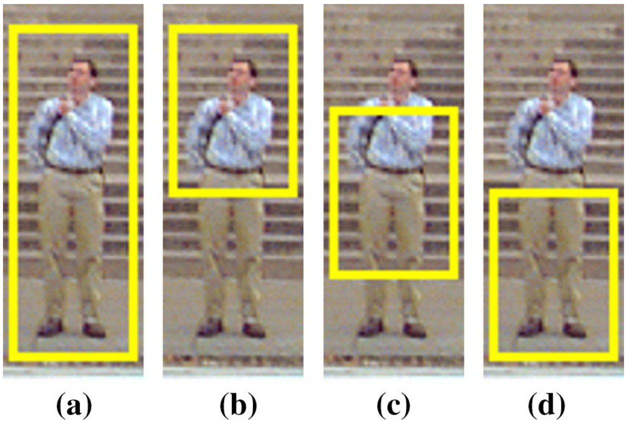

Fig. 2 An example of a pedestrian image from the dataset, with the bounding box of the extracted patch shown. **a** Global patch, **b** local patch U, **c** local patch M, **d** local patch L

Since the global and local CNNs in our framework use the same network architecture (i.e. one of the above), this allows the convolutional layer weights to be shared between the networks. In our work, the first-layer filter weights are shared. That is, each network uses the same set of filters when performing the convolution operation to obtain the first convolutional layer. It is known that the first convolutional layer has the most primitive filters, such as simple edges and spot detectors, which is expected to be common for both global and local features. Weights sharing helps to reduce the number of free parameters.

It should be noted that the size of the feature maps is different in the global and local CNNs due to the difference in the input patch sizes. That is, in order to avoid the possibility of distortion in the local information, we do not resize the local patches to the same size and aspect ratio as the global patch. This is possible because we do not share the weights of the FC layers between the networks. So, the size of the feature maps in the layer preceding the first FC layer of a network need not be the same as in the other networks.

3.2 Patches size and location

The size of the images in the dataset used in our experiments is 48 × 128 pixels (width × height). As mentioned previously and shown in Fig. 2, a slightly smaller central crop is taken as the global patch to enable spatial jittering during training. When using deep CNNs, CaffeNet and VGG-19, the default size of the training image, as used in ILSVRC, is 256 × 256 pixels. The default size of the input patch is 227 × 227 and 224 × 224 for CaffeNet and VGG-19, respectively. We thus resized the training images to 256 × 256, with the global patch having the default size of the input patch.

Given these default sizes, the amount of spatial jittering for the global patch that can be applied during training is calculated. For example, with VGG-19, the value is equal to (256 – 224)/2, which is 16 pixels in both horizontal and vertical directions. Similarly, for CaffeNet, the value is 14 pixels. From this, we obtained the location of the global patch in an image. Assuming the zero coordinates are at the top-left corner of the image, the origin coordinates of the global patch have the same values as the maximum amount of jittering allowed. This is summarized in Table 2.

Table 2 Size and origin coordinates of the global and local patches for the various CNNs

CNN	Image size	Patch size (width × height)		Patch origin coordinates[a] (x,y)			
	(width × height)	Global	Local	Global	U	M	L
VGG-19	256 × 256	224 × 224	224 × 112	(16,16)	(16,16)	(16,72)	(16,128)
CaffeNet	256 × 256	227 × 227	227 × 114	(14,14)	(14,14)	(14,71)	(14,128)
CNN-1/2/3	48 × 128	42 × 112	42 × 56	(3,8)	(3,8)	(3,36)	(3,64)

[a] With respect to the top-left corner of the image

Table 3 Datasets used in the experiments

Dataset	No. of images	Size (width × height)	View	Source	Scenario
MIT	888	64 × 128	Front, back	MIT	Outdoor scenes
APiS	3586	48 × 128	Mixed	KITTI, CBCL StreetScenes, INRIA, SVS	Street scenes (captured while driving), surveillance

We considered three different local patches, which we refer to as patch U, L and M, as shown in Fig. 2. By simply dividing the global patch into equal upper and lower halves, we obtained patch U and L, respectively. So, the size of the local patches is 227 × 114 and 224 × 112 for CaffeNet and VGG-19, respectively. Patch M is vertically the middle section of the global patch. It consists of the lower half of patch U and the upper half of patch L. The origin coordinates of the local patches can thus be derived, as shown in Table 2. Assuming the pedestrian is approximately centered in the image, the local patches would contain different regions of the body.

For CNN-1/2/3, we kept to the original size of the training images and derived the patch sizes as follows. For VGG-19, the ratio of the global patch height to image height is 224/256 or 0.875 (using the width would also give the same ratio). Applying this ratio, we thus obtained the size of the global patch as 42 × 112 (i.e. 0.875 of 48 × 128). The local patches size is then 42 × 56 (half the height of the global patch). The maximum amount of spatial jittering during training is 3 and 8 pixels in the horizontal and vertical directions, respectively. The origin coordinates of the global and local patches can thus be derived.

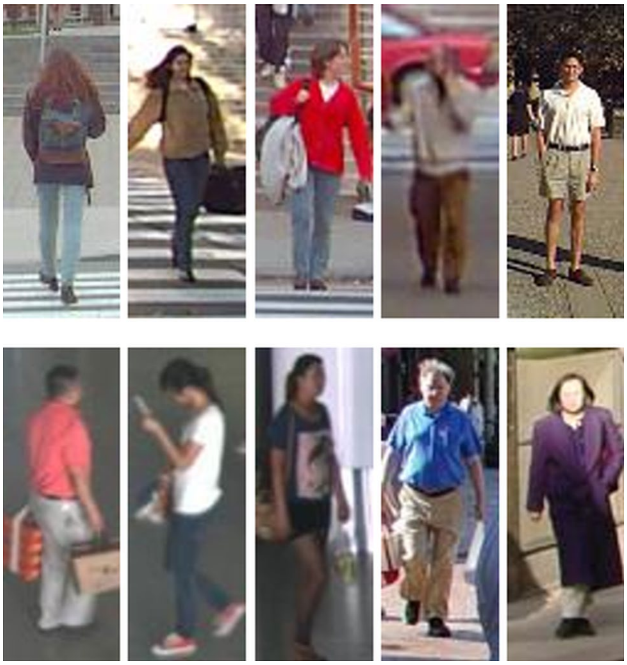

Fig. 3 Example of images from MIT dataset (top row) and APiS dataset (bottom row)

4 Experimental results and discussion

We conducted experiments to evaluate the proposed method using images from two publicly available datasets, with the details as summarized in Table 3. The datasets contain color images of pedestrians which have been cropped out and labeled with male or female gender by researchers.

The MIT pedestrian dataset [20] contains images of people captured in different seasons using several different digital cameras and video recorders. The images consists of only front and back views of people, as shown by the examples in Fig. 3 (top row). We used the 888 images with gender and view labeled by Cao et al. [2]. Each image is 64 × 128 pixels with the person centered in the image, but for standardization we further cropped it to 48 × 128, by removing eight pixels each from the left and right borders.

The Attributed Pedestrians in Surveillance (APiS) dataset was collected by Zhu et al. [28] for the purpose of evaluating pedestrian attribute classification. The images are actually a combination of various datasets which include street scenes and surveillance videos, with examples shown in Fig. 3 (bottom row). They located candidate regions using a pedestrian detector and then selected true positive images of sufficient size before finally resizing all the images to 48 × 128 pixels. For the male binary attribute, 3586 images have been labeled by them as male positive or negative, which we take to be male and female, respectively. The view of the pedestrians is a diverse mixture which is not constrained to front or back only. However, this label was not provided. So, for analysis purposes, we manually labeled the images as frontal or non-frontal views based on the yaw angle θ of the person's torso. We label an image as frontal view if the yaw angle is estimated to be in the range of $-90° < \theta < 90°$; otherwise, it is labeled as non-frontal view.

We combined both the datasets to obtain over 6000 images for supervised training of the network and 1000 images for testing. Specifically, we randomly selected 3470 images, of which 800 images were held out as the validation set, while the remaining 2670 images together with their mirror images, i.e. a total of 5340 images, were used for training. The breakdown for male and female numbers is shown in Table 4, with the proportion of males in each set approximately 69%. This percentage represents the accuracy if the classifier learnt to guess the majority class all the time, so a well-trained classifier should perform better than this.

We have no information of whether the datasets contain images of transgender or classes besides male and female. As such, our work does not consider these other classes.

4.1 Training the model

Off-the-shelf CNNs such as CaffeNet and VGG-19 can be fine-tuned to achieve good results in many image recognition problems. The usual practice is to pre-train these CNNs for object recognition in ILSVRC and then fine-tune (after a small modification on the final output layer to account for the change in the number of classes) the CNN for the target problem [3, 7]. We used the weights (including the biases) of the pre-trained CNN that is made available by the researchers to initialize each convolutional layer in the deep CNNs of our model before training for pedestrian gender classification. For the weights in the fully connected layers, the pre-trained weights could not be used due to size incompatibility, so they were initialized randomly from a Gaussian distribution of mean 0 and standard deviation 0.01, while the biases were set to 0. Following the recommended training practice, L2 weight regularization with the value 0.0005 and dropout with factor of 0.5 were also applied in the fully connected layers. Training by mini-batch stochastic gradient descent (SGD) was performed with momentum value of 0.9 and learning rate of 0.001 and 0.01 (for VGG-19 and CaffeNet, respectively) in the randomly initialized fully connected layers while using a smaller rate (by ten times) for the convolutional layers.

For CNN-1/2/3, we followed the method used in [18] to pre-train them for pedestrian gender classification. In particular, the first-layer convolutional weights were initialized using filters learnt by *k*-means clustering on a separate pedestrian dataset and trained for pedestrian classification. The weights were then used to initialize the convolutional layers of the CNNs in our model. The fully connected layers were initialized using Xavier method [8]. Dropout of 0.5 was used but without any weight regularization. A learning rate of 0.1 was used for the fully connected layers (ten times smaller for the convolutional layers) during mini-batch SGD training but no momentum was used.

The validation set was used for early stopping during training. Each training was repeated five times using different random seeds for the weights initializations to obtain an average percentage for the gender classification accuracy. When there was more than one network in the model, they were jointly trained to minimize the total loss. The model was implemented using Theano library [27] in Python.

Table 4 Breakdown of dataset

	Training set	Validation set	Test set
No. of males	1834	549	682
No. of females	836	251	318
Total	5340 (including mirror images)	800	1000

4.2 Results and discussion

In order to compare how effectively each patch is individually able to discriminate gender, only one patch was used as input to the model. That is, we used just a single network with the input being either the global patch or one of the local patches. The results, depicted in Table 5, show that patch U achieved the highest accuracy among the local patches, with patch L being the worst. This is consistent across all the different CNN architectures in the experiment. When compared to the global patch, patch U was only slightly outperformed.

Based on the results, we can conclude that the body parts in the middle and lower half regions are not as effective for discriminating the gender of a pedestrian as compared to the upper half of the body. Using only the upper body half is almost as effective as using the whole body, implying that most of the gender information that is learnt seems to be contained in the upper half of the body. This result supports the work by Li et al. [15], who effectively trained a classifier for the head–shoulder region. However, they did not

Table 5 Comparison between individual patches

Input	Average accuracy and standard deviation (%)				
	CNN-1	CNN-2	CNN-3	CaffeNet	VGG-19
Global	80.82 ± 1.08	80.48 ± 0.41	78.72 ± 0.34	82.76 ± 0.61	85.38 ± 0.52
Local U	80.8 ± 0.51	79.12 ± 0.51	79.58 ± 0.80	81.68 ± 0.37	84.86 ± 0.63
Local M	74.02 ± 0.44	73.88 ± 0.35	72 ± 0.46	75.18 ± 0.58	78.42 ± 0.75
Local L	70.62 ± 1.46	70.44 ± 0.57	68.24 ± 0.60	71.16 ± 0.60	73.58 ± 0.55

compare with the whole body, as the images of the dataset in their work contained only the head–shoulder region. Also, their dataset has very few images of a person from the back or side views, with almost full frontal faces in most images, which means there may have been a strong reliance on the face. In contrast, more than half of the test images in our dataset contains non-frontal views of the face.

Next, we examined the effect of combining the local and global patches. We used the parts-based CNN framework proposed, combining a different number of local CNNs with a global CNN. The choice of local patch was based on the results from the previous section, by selecting the patches with the best accuracy. So, if we used only one local CNN, the most discriminative patch U was used as input. If we used two local CNNs, patches U and M were used, and so on. The results are shown in Table 6.

By combining the local CNN of patch U with the global CNN, the accuracy is improved compared to using only an individual global CNN. However, by adding more local CNNs to the model, the accuracy does not improve but actually declines. The results implies that while the upper half body region seems to contain most of the gender information, there is information contained only in the whole body image that can contribute to the inference. This could perhaps be the overall shape or pose of the person, which would be beneficial, especially for cases when a pedestrian is viewed from the back.

For comparison purpose, we also tried the combination of local CNNs only to determine whether the global CNN can be excluded from the model. The accuracy was slightly worse in this case, hence verifying the importance of information from the global CNN.

Comparing between the various CNN architectures, VGG-19 obtains the highest accuracy. This is in line with its object recognition performance in ILSVRC which was better than CaffeNet. CNN-1, CNN-2 and CNN-3, being smaller networks, did not fare as well but can still provide decent accuracy above 80%.

Table 7 compares our best results against previous works. We also separated the results for frontal and non-frontal views.

The MIT dataset contains images of people from the front or back view only. Our classifiers achieve better accuracy for the combined MIT and APiS dataset that contains more variations in viewpoint and with a larger number of test images. This verifies the effectiveness of our model combining global and local information.

4.3 Feature visualization

To gain an understanding of what a trained CNN has learned, feature visualization methods have been proposed and advanced. We employed the guided backpropagation method by Springenberg et al. [25] which is implemented in the FeatureVis library by Grun et al. [10]. Basically, the method determines the contribution of the pixels in an input image to the classification decision, forming a heat map to visualize the most relevant features. We ran it through the

Table 6 Comparison between combinations of patches

Combination	Average accuracy and standard deviation (%)				
	CNN-1	CNN-2	CNN-3	CaffeNet	VGG-19
Global only	80.82 ± 1.08	80.48 ± 0.41	78.72 ± 0.34	82.76 ± 0.61	85.38 ± 0.52
Global + U	82.54 ± 0.27	80.86 ± 0.60	80.9 ± 0.33	83.84 ± 0.80	86.88 ± 0.41
Global + U + M	81.28 ± 0.60	80.64 ± 0.59	79.66 ± 1.05	82.56 ± 0.41	85.92 ± 0.26
Global + U + M + L	81.38 ± 0.56	80.38 ± 0.78	79.54 ± 0.49	82.58 ± 0.56	85.72 ± 0.61
U + M	80.94 ± 0.61	79.98 ± 0.52	80.06 ± 0.62	81.88 ± 0.58	85.48 ± 0.58
U + M +L	79.2 ± 1.02	79.04 ± 0.24	78.1 ± 1.05	81.26 ± 0.38	82.82 ± 0.44

Table 7 Comparison against previous works

Method	Average accuracy and standard deviation (%)			Dataset
	Mixed view	Frontal view	Non-frontal view	
Cao et al. [2]	75.0 ± 2.9	76.0 ± 1.2	74.6 ± 3.4	MIT
Collins et al. [4]		76.0 ± 8.1		MIT
Guo et al. [11]	80.6 ± 1.2	79.5 ± 2.6	84.0 ± 3.9	MIT
Geelen et al. [6]	80.9 ± 2.4	81.6 ± 1.6	82.7 ± 1.8	MIT
Ng et al. [18]	81.5 ± 2.2			MIT
Our model with CNN-1	82.54 ± 0.27	79.5 ± 0.55	85.0 ± 0.71	MIT+APiS
Our model with CaffeNet	83.84 ± 0.80	80.0 ± 0.71	87.0 ± 1.08	MIT+APiS
Our model with VGG-19	86.88 ± 0.41	84.4 ± 0.93	88.9 ± 0.9	MIT+APiS

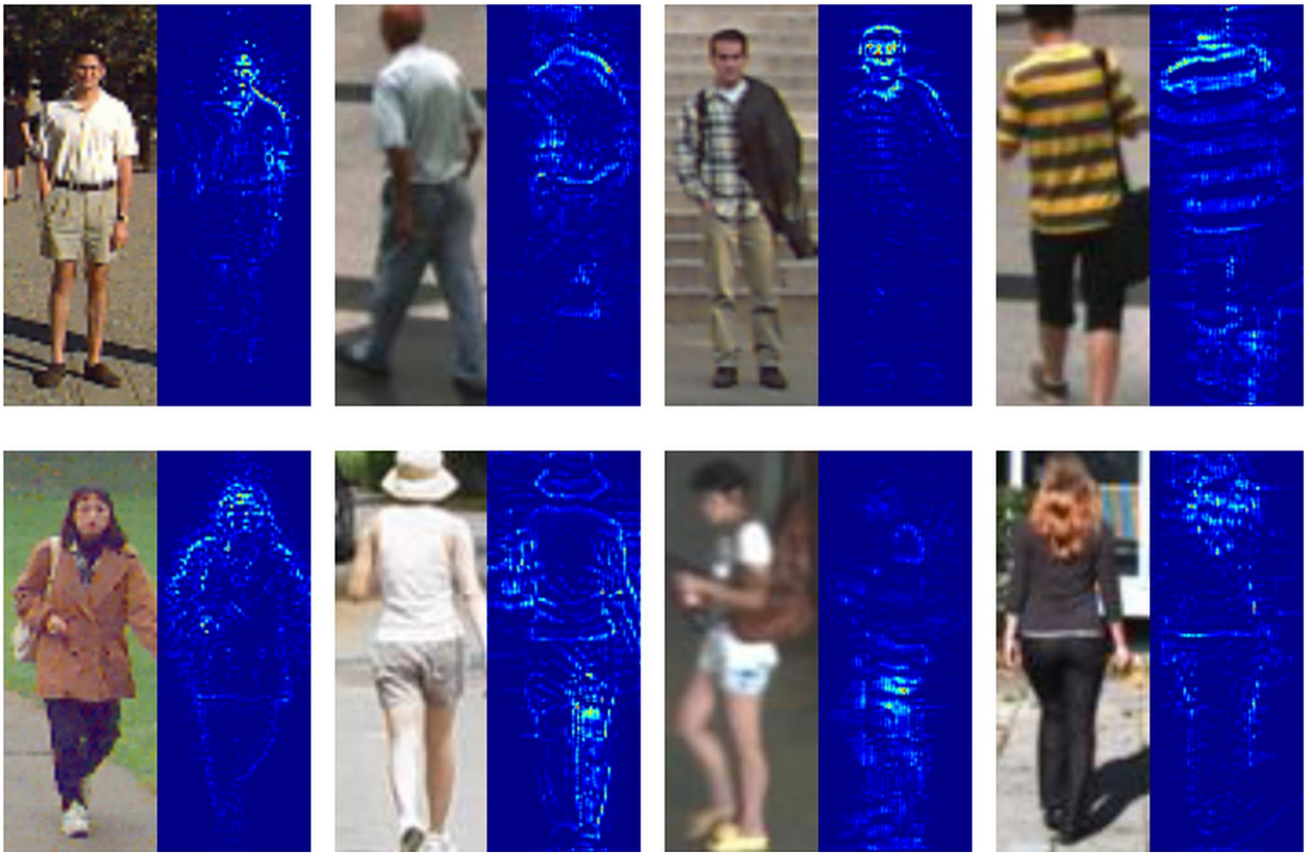

Fig. 4 Feature visualization of some male (top row) and female (bottom row) images (best viewed in color, available online)

VGG-19 network trained using global patch for pedestrian gender classification.

Figure 4 shows some examples of training images containing male and female pedestrians and the corresponding feature visualization on their right. The important pixels are seen mostly on the person or the outline rather than the background. In frontal views, the important pixels tend to be clustered around the facial regions, confirming that the face plays an important role in differentiating the gender of a person. When the face is not visible, the cue is then taken from the rest of the body, which could perhaps be the hairstyle, clothing or the body shape.

We next determined the average image of the visualizations. We summed up the pixel index values across all the heat maps and calculated the average value in order to obtain the average heat map. This was done for images in the training set, for all gender, females only and males only. Looking at the heat maps in Fig. 5, the brightest pixels tend to be concentrated in the upper region of the images. This concurs with our result that shows the upper half region is more discriminative than the other regions. In terms of the difference between the male and female average heat maps, the male is more concentrated in the head region while the female is more diffused throughout the body. This suggests that for females the body shape or clothing plays an important role.

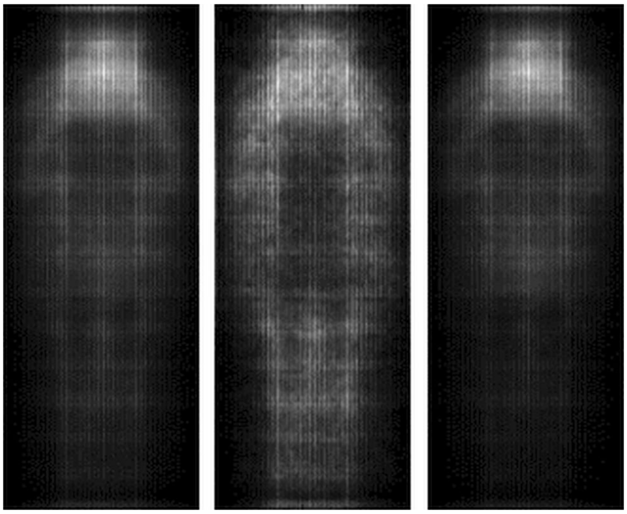

Fig. 5 Average image of the visualizations for all images (left), females (center) and males (right)

4.4 Error analysis

We performed an error analysis, in an attempt to find the possible reasons that led our trained model to make the wrong inference, by examining the wrongly classified images based on human observation and intuition on the difference between the male and female genders. We considered the best performing model combining global CNN and local CNN for patch U, with VGG-19 architecture as the network. Since the results were obtained from an average of five trials, the errors made may be different from one trial compared to another. We determined the images that were classified incorrectly in all the trials, i.e. all the models made the same mistake on these images. In total, there were 70 such images, consisting of 20 images labeled as male (incorrectly classified as female) and 50 images labeled as female (incorrectly classified as male).

Figure 6 shows examples of the wrongly classified male images, which we divide into several common categories as follows, with the total number of cases denoted in brackets:

a. wearing short sleeves and/or short pants (5)
b. long hairstyle or apparent long hair (4)
c. style of clothing (2)
d. body hidden under thick clothing (4)
e. no obvious reason (5)

Figure 7 shows examples of the wrongly classified female images, which we divide into several common categories as follows, with the total number of cases denoted in brackets:

a. short hairstyle or apparent short hair (26)
b. poor image quality such as blur, low light or poor contrast (7)
c. body hidden under thick clothing (6)
d. possibly label error (2)
e. no obvious reason (9)

It is difficult even for humans to distinguish the gender in some of the images, especially when the person is wearing thick clothes that hide the body shape. Some images have poor quality such as low resolution, lighting or contrast.

The classifier made many mistakes in the case of females with short hair, perhaps because it has learnt to use hair length (instead of hairstyle) as a strong cue, which overwhelms the weaker cues. Most of the females with short hair also wear long pants; hence, clothing could not act as a strong cue. Since there is a large variety in clothing and hairstyle because of fashion and culture, the classifier may

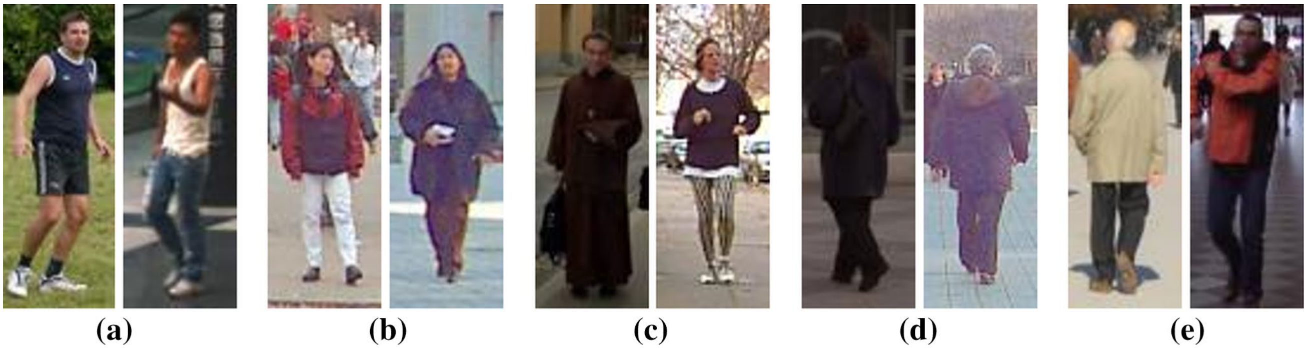

Fig. 6 Examples of males wrongly classified as females

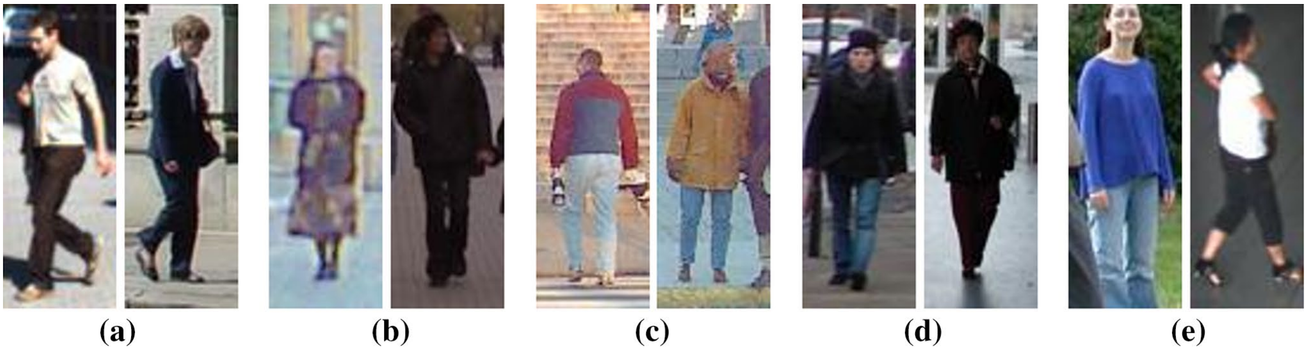

Fig. 7 Examples of females wrongly classified as males

Table 8 Effect of image resolution on classification accuracy

Image size %	Average accuracy and standard deviation (%)	
(width × height)	CaffeNet	VGG-19
100% (256 × 256)	83.84 ± 0.80	86.88 ± 0.41
90% (230× 230)	82.80 ± 0.64	86.00 ± 0.43
80% (204× 204)	82.54 ± 0.32	86.30 ± 0.32
70% (180× 180)	81.4 ± 1.44	85.80 ± 0.75
60% (154× 154)	81.52 ± 0.60	85.5 ± 0.49
50% (128× 128)	–	85.88 ± 0.34
40% (102× 102)	–	84.58 ± 0.31
30% (77× 77)	–	83.44 ± 0.21

not have learnt these adequately from the training images. In future work, a larger dataset should be used to mitigate this.

4.5 Effect of image resolution

We studied the effect of image resolution on the best models using the deep CNNs, CaffeNet and VGG-19. For these CNNs, the default image size is 256 × 256 pixels, the value to which we resized the images from our dataset. Here, we explored using various smaller image sizes to train the CNN. In particular, we resized the images to values ranging from 30 to 90% of the default size. For example, 90% corresponds to 230 × 230 pixels. The size and location of the patches were derived accordingly as described in Sect. 3.2. The model was then trained on these images and the classification accuracy obtained.

The results are shown in Table 8. Note that for CaffeNet, no result is given for image sizes at 50% or less, because the feature maps in the network shrink to zero at these small input sizes. Similarly, for VGG-19 the minimum size is at 30%.

For CaffeNet, the reduction of accuracy is approximately 2.3% at the smallest image size. For VGG-19, there is a reduction in accuracy of only 1% at 50% image size, and approximately 3.44% at the smallest image size. So, for 50% image size and above, VGG-19 is not as much affected by the image resolution when compared to CaffeNet. The deep CNNs can still perform near state-of-the-art levels with smaller training images.

5 Conclusion

The work presented here proposed a novel method for gender classification of pedestrians, leveraging on the feature learning power of convolutional neural networks. Our parts-based framework combines information from both the global and local level, without attempting to explicitly locate body parts. The advantage of our method is that only the gender label is required for the training images, without any other data annotation such as body parts, key points or other attributes, which would be costly to obtain for a large number of images. Our experiments show that the upper half body region is the most discriminative for gender when compared to the middle or lower half. This result is supported by feature visualization of the gender information learnt by a trained CNN, which provides significant understanding of the problem. This led to the best model, which combines a jointly trained global CNN and local CNN that takes input from the upper half region of the body. The information from the CNNs is combined by taking the average score. The classification accuracy of the best model on publicly available datasets which contain images with a large variety of viewpoints is higher than previous state-of-the-art methods.

Acknowledgements This work has been supported by the Ministry of Science, Technology and Innovation (MOSTI) Science Fund, Project No: 01-02-11-SF0199. We thank the anonymous reviewers for their comments and suggestions.

References

1. Antipov G, Berrani SA, Ruchaud N, Dugelay JL (2015) Learned vs. hand-crafted features for pedestrian gender recognition. In: Proceedings of the 23rd ACM international conference on multimedia. ACM, pp 1263–1266
2. Cao L, Dikmen M, Fu Y, Huang TS (2008) Gender recognition from body. In: Proceedings of the 16th ACM international conference on multimedia. ACM, pp 725–728
3. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014) Return of the devil in the details: delving deep into convolutional nets. arXiv:1405.3531
4. Collins M, Zhang J, Miller P, Wang H (2009) Full body image feature representations for gender profiling. In: 2009 IEEE 12th international conference on computer vision workshops (ICCV workshops). IEEE, pp 1235–1242
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on Computer vision and pattern recognition, 2005. CVPR 2005, vol 1. IEEE, pp 886–893
6. Geelen CD, Wijnhoven RG, Dubbelman G et al (2015) Gender classification in low-resolution surveillance video: in-depth comparison of random forests and svms. In: SPIE/IS&T electronic imaging, international society for optics and photonics, pp 94070M–94070M

7. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
8. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
9. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
10. Grün F, Rupprecht C, Navab N, Federico T (2016) A taxonomy and library for visualizing learned features in convolutional neural networks. In: ICML workshop on visualization for deep learning (ICML-W)
11. Guo G, Mu G, Fu Y (2009) Gender from body: A biologically-inspired approach with manifold learning. In: Asian conference on computer vision. Springer, pp 236–245
12. 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: Proceedings of the 22nd ACM international conference on multimedia. ACM, pp 675–678
13. Khan FS, van de Weijer J, Anwer RM, Felsberg M, Gatta C (2014) Semantic pyramids for gender and action recognition. IEEE Trans Image Process 23(8):3633–3645
14. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
15. Li M, Bao S, Dong W, Wang Y, Su Z (2013) Head-shoulder based gender recognition. In: 2013 20th IEEE international conference on image processing (ICIP). IEEE, pp 2753–2756
16. Ng CB, Tay YH, Goi BM (2013) Comparing image representations for training a convolutional neural network to classify gender. In: 2013 1st international conference on artificial intelligence, modelling and simulation (AIMS). IEEE, pp 29–33
17. Ng CB, Tay YH, Goi BM (2015) A review of facial gender recognition. Pattern Anal Appl 18(4):739–755
18. Ng CB, Tay YH, Goi BM (2017) Training strategy for convolutional neural networks in pedestrian gender classification. In: Second international workshop on pattern recognition, international society for optics and photonics, vol 10443, pp 104431A
19. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
20. Oren M, Papageorgiou C, Sinha P, Osuna E, Poggio T (1997) Pedestrian detection using wavelet templates. In: Proceedings, 1997 IEEE computer society conference on computer vision and pattern recognition, 1997. IEEE, pp 193–199
21. Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. Nat Neurosci 2(11):1019–1025
22. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
23. Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. Artifl Neural Netw ICANN 2010:92–101
24. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
25. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for simplicity: the all convolutional net. arXiv:1412.6806
26. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
27. Theano Development Team (2016) Theano: a python framework for fast computation of mathematical expressions. arXiv:1605.02688
28. Zhu J, Liao S, Lei Z, Yi D, Li S (2013) Pedestrian attribute classification in surveillance: database and evaluation. In: Proceedings of the IEEE international conference on computer vision workshops, pp 331–338