2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

A Comprehensive Study on Upper-Body Detection with Deep Neural Networks

Yamei Zhu
School of Software Engineering
Tongji University
Shanghai, China
Email: zhuyamei@tongji.edu.cn

Lin Zhang*
School of Software Engineering
Tongji University
Shanghai, China
Email: cslinzhang@tongji.edu.cn

Abstract—**The pedestrian detection task which aims to predict bounding-boxes of all the pedestrian instances in an image is of paramount importance for many real-world applications and has attracted much attention within the computer vision community. However, the researchers generally ignore the critical issue that due to the reasons of partial occlusion or being out of FOV, the definition for pedestrian is ill-posed in many cases and even humans will find it difficult to give accurate bounding-boxes. It is found that in many real applications, pedestrian detection can be substituted by upper-body detection, which is more robust and is much less affected by occlusion or being partially out of FOV. However, few studies have been conducted in this area. To fill this research gap to some extent, we make two contributions in this paper. Firstly, in order to facilitate the study of upper-body detection, a large-scale benchmark dataset is established. This dataset comprises 9585 images extracted from typical surveillance video clips and for each image, all the upper-body instances were carefully labeled. Secondly, the performances of four state-of-the-art object-detection frameworks were thoroughly evaluated in the context of upper-body detection, which can serve as a baseline for other researchers to develop even more sophisticated methods. To make the results fully reproducible, the collected dataset has been made publicly available at https://github.com/AmazingMei/upper-body-detection.**

I. Introduction

Pedestrian detection is meaningful in many fields, such as Advanced Driver Assistant System (ADAS) [1], Pedestrian Protection System (PPS), robotic and surveillance. There are extensive researches on pedestrian detection and many detectors turn out to have good results. The main paradigms for object detection, including Viola&Jones variants [2], HOG+SVM rigid templates [3], deformable part model (DPM) [4]and convolutional neural network (CNN) [5], are all good solutions for this task.

At the same time, pedestrians are one of the most challenging categories for object detection. Because of the various types and styles of clothing, their local and global appearance have a large variability. In addition, the global shape undergoes a large range of transformations caused by occlusion and movement. To solve these problems, some researchers focus on pedestrian detection in crowded scenes [6] and some focus on part-based approaches [7].

But all these researches have ignored the disadvantages of pedestrian detection. It is hard to give an exact definition to "pedestrian". In Figure 1, people in these images are various in many ways: taking bicycles or motorcycles, sitting or squatting, or even using handcart. Should they be defined as pedestrian instances? Because of occlusions, some people only have heads in sight in Figure 1. There are also many people partially out of the field of view (FOV) caused by the short distance to the camera. These people are usually on the down side of the images and only half of their bodies can be seen. Should we treat them as pedestrians as well?

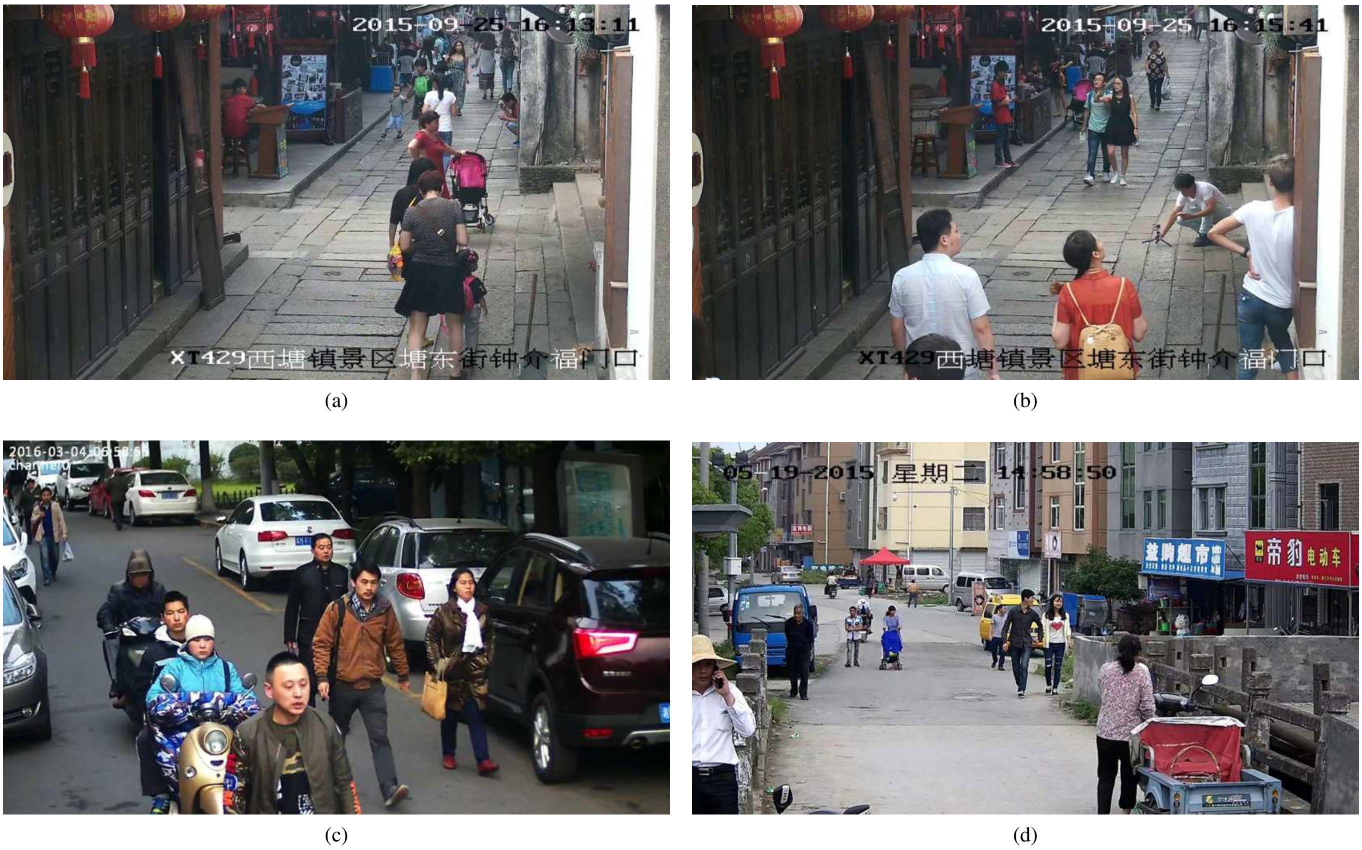
To detect every person in the image, most answers to the questions above are "yes". But this means the appearances of pedestrians become various and the performance of pedestrian detector will be weakened. To make the instance become consistent, we can just substitute upper-body for pedestrian. Upper-body regions have low flexibility and their shapes are basically stable. Different transports also have no effect on the appearance of upper-body. For the people who are partially out of FOV or occluded by other people, their upper-bodies are still in the field of view. This makes upper-body detection easier to have a better performance.

Actually, in many real-world applications, upper-body detection can completely replace pedestrian detection. The main purpose of pedestrian detection is to find and track people in the images or videos. In ADAS, pedestrian detection is used to find the position and distance to the people in sight. Upper-body detection is enough to achieve these goals. At the same time, compared with upper-body, the lower-body contains few information and is more flexible. Head detection is easier and popular, but also contains less information. It is hard to tell the human's action, pose or position through head. So using upper-body detection as a substitute of pedestrian detection is more reasonable.

Most of the upper-body datasets are acquired from TV shows for motion classification or action recognition, like the BBC TV Signing dataset [8], TV Human Interaction dataset [9] and VGG Upper Body Dataset [10]. The viewpoints of these datasets are similar and the backgrounds are simple. The numbers of people in the images are limited, usually 2 to 3 people, which means there are basically no occlusion in these

* Corresponding author.

978-1-5386-3788-3/18/$31.00 ©2018 IEEE


2015-09-25 16:13:11
XT429西塘镇景区塘东街钟介福门口
2015-09-25 16:15:41
XT429西塘镇景区塘东街钟介福门口
2016-03-04
05-19-2015 星期二 14:58:50
帝豹电动车

(a)

(b)

(c)

(d)

Fig. 1. Examples for ambiguous pedestrians. (a) Sitting people. (b) Squatting people. (c) People taking motorcycle. (d) People using handcart.

pictures. As a result, these datasets are not suitable to detect the upper-body instance as a substitute for pedestrian detection. For applications in ADAS or surveillance, the scenes need to be on the road, with low image resolution and high viewpoint. In surveillance videos, the crowd may cause occlusion. At the same time, these datasets are very small. BBC TV Signing dataset contains 300 video clips and each one of them is only 1-3 seconds. VGG Upper Body Dataset has 290 images. These datasets are too small to train deep learning models. Some researchers use INRIA Person Dataset [11] to train upper-body detectors. But as INRIA Person Dataset only contains annotated pedestrians, researchers have to refine the annotations by themselves, which means a lot of extra work. So the existing datasets are not enough for upper-body detection.

Many researchers have investigated upper-body detection and most of the approaches are the same as pedestrian detection. The most frequent features used in these works are the Histogram of Oriented Gradient (HOG) [3] features. At the same time, Dollár proposed an alternative representation of the channel feature called Aggregated Channel Features (ACF) [11] for pedestrian detection. ACF is quite appealing for its good performance in terms of detection results and computation time.

On the other side, recent years more and more deep learning models have been proposed to solve problems in different areas. Many DCNN-based methods turn out to have fantastic results for object detection, such as Faster R-CNN [12], SSD [13] and YOLOv2 [14]. The original detection methods using features and classifiers seem out-of-date and slow compared with them. But few of deep learning methods have been used in this area.

To fill this research gap, we make two contributions in this paper: (1) We present a new dataset for upper-body detection. Unlike other existing upper-body datasets acquired from TV shows, the images in our dataset are acquired from surveillance cameras. There are more people and more complicated background than the exit datasets. Our dataset is more suitable for applications in ADAS or surveillance. (2) We investigate the performance of four state-of-the-art object-detection methods in the context of upper-body detection. We adopt three DCNN-based deep learning methods, including Faster R-CNN [12], SSD [13] and YOLOv2 [14], for evaluation. In addition, we also investigate the performance of the ACF-based framework for upper-body detection on our collected dataset.

The rest of this paper is structured as follows: Section 2 introduces our new dataset. Section 3 gives a brief glance at architectures of different DCNN-based learning models. Section 4 shows the experimental results and Section 5 concludes the work.

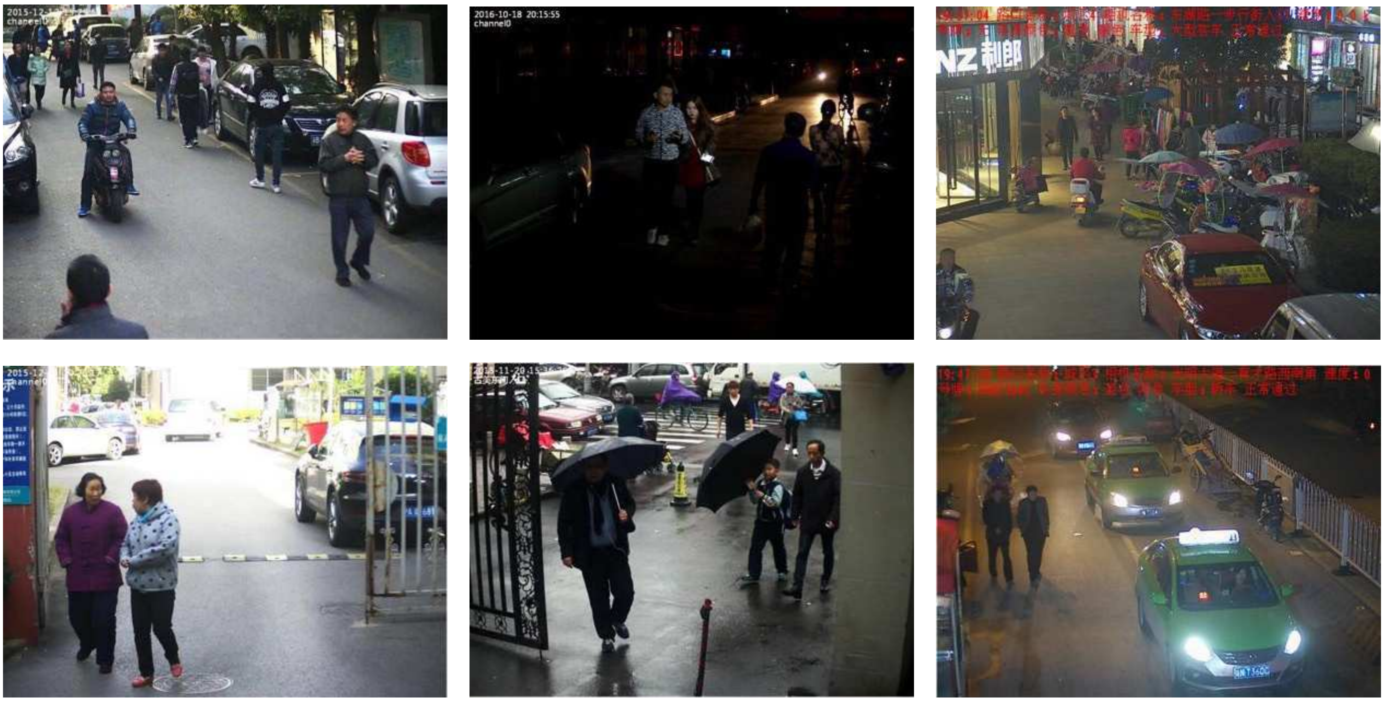


Fig. 2. Example images in our dataset with different weather and illumination.

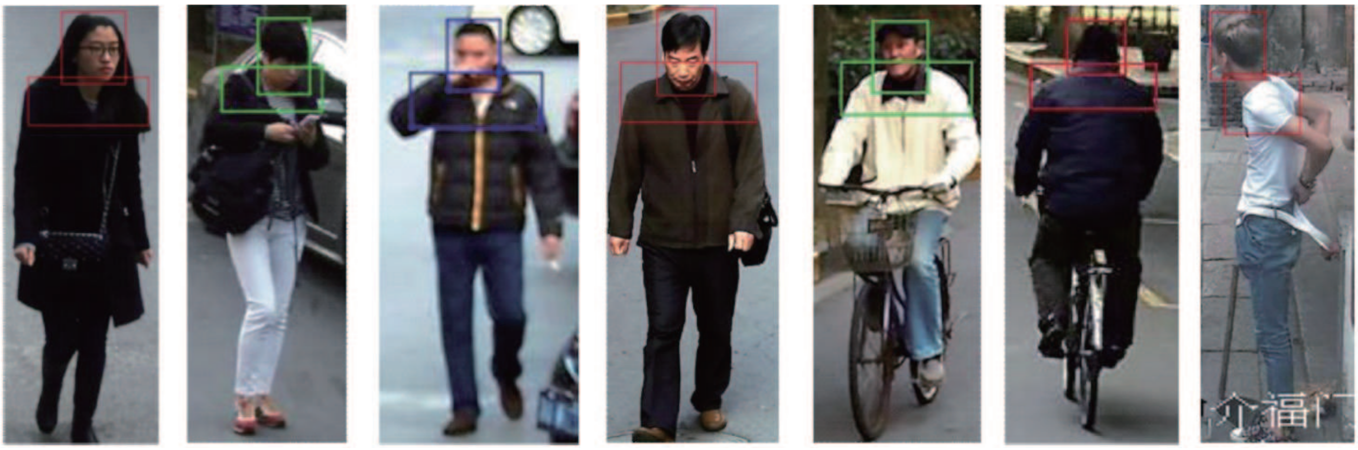


Fig. 3. The upper-body regions of different pedestrians. In our dataset, there are male and female, pedestrians and cyclists, taken from front, back and sides.

II. Database

Our dataset comprises 9585 images acquired from 26 different surveillance cameras. The surveillance videos are collected in various scenes, such as shopping street, housing estate and community gate. These places are very common in our daily life. The selected scenes have various weathers: sunny, rainy or cloudy. And the images are taken at different time, vary from late night without light to noon with bright sunlight. These pictures also vary in crowd density, motion direction and shooting distance. This makes our dataset have high diversity and is more challenging for detection. Figure 2 shows some examples in our dataset.

As shown in Figure 3, examples are acquired from the front, back and sides of people and they face to different directions. This makes it hard to be detect by traditional methods. There are also many images with crowd or acquired at night. The occlusion and illumination also increase the difficulty to tell the upper-body regions.

After we get the raw video data, we extracted static frames from the video and remove the frames without human or blurred. Then we use an application implemented by C++ to annotate the upper-body regions manually. This application can annotate different parts of human body such as head, shoulder, leg or foot. Besides upper-body detection, this application can be used in many other computer vision areas such as pedestrian detection or part-based human detection.

However, unlike head or pedestrian instance, upper-body is still an ambiguous definition. Some may think the breast is the bottom line of upper-body region while some think the hip is. In our dataset, we define the region between the top of head to the bottom of shoulders as upper-body.

To increase the accuracy of our annotations, we annotate the head and shoulder areas respectively and then merge the two regions together to get a new rectangle as the upper-body region. In this way, the annotation we use can be more accurate, because it is much easier and clearer for people to tell the regions of head and shoulders. After that, to make our dataset easy to be trained by deep learning methods, we convert the form of our annotations to the same as VOC2007 database [15]. The images and annotations have been publicly online at https://github.com/AmazingMei/upper-body-detection.

III. Deep CNN-based object detection frameworks

In recent years, deep learning has achieved many successful results in different fields. Compared with traditional machine learning methods, deep learning methods have a stronger learning ability and can make better use of the training data. Convolutional Neural Network (CNN) is an excellent model that can accomplish detection tasks efficiently. Many Deep CNN-based frameworks do a great job to solve object detection problems. Upper-body is an object detection task and can use these frameworks. In the following, we will give a brief introduction to several methods of them, Faster R-CNN, SSD and YOLOv2 in particular. In section 4, we will use these methods in our experiments.

A. Faster R-CNN

Region-based convolutional neural network (R-CNN) [16] is a popular approach for object detection based on CNN. While accurate, R-CNN is computationally expensive. But the performance has become better in the following incarnation: Fast R-CNN [17] . Although Fast R-CNN takes advantage of GPUs, the region proposal methods used in research are implemented on the CPU, costing a lot of time. The researchers observed that the convolutional feature maps used by region-based detectors could also be used for generation region proposals. So Faster R-CNN introduced novel Region Proposal Network (RPN) that shares convolutional layers with Fast R-CNN, which makes the cost for computing proposals small (e.g. 10ms per image).

Faster R-CNN achieves 73.2% mAP and 7 FPS on VOC2007test on Nvidia Titan X.

B. SSD

Single Shot MultiBox Detector (SSD) [13] used a set of default boxes with different aspect ratios and scales to represent the output space of the bounding boxes. When predicting, SSD gives scores for each class in each default box and adjusts the box to match the object shape better. By eliminating object proposal and following pixel or feature resampling stages, SSD encapsulates all computation in a single network, which makes it easy to train and integrate into other systems.

The elimination of object proposal step does not reduce the detection accuracy and make it much faster. For 300×300 input, SSD achieves 74.3% mAP on VOC2007test at 59 FPS on Nvidia Titan X and for 512×512 input, it achieves 76.9% mAP.

C. YOLOv2

You Only Look Once (YOLO) [18] reframes object detection as a single regression problem, from pixels to bounding boxes and associated class probabilities.

The pipeline of YOLO is quite simple. First resize the input picture and then run a single convolutional network on the image to predict both bounding boxes and class probabilities. Finally, YOLO sorts the results by the model's confidence score. Such a simple pipeline turns to make YOLO extremely fast, runs at 45 FPS on Nvidia Titan X. Besides, YOLO uses the whole image when making predictions, which means it encodes contextual information about classes and their appearances and make less background errors compared with other methods. However, YOLO makes more localization errors.

YOLOv2 [14] uses a few tricks to improve training and increase performance. Like SSD, YOLOv2 uses a fully convolutional model, but train on whole images instead of hard negatives. Like Faster R-CNN, it adjusts priors on bounding boxes instead of predicting height and weight, but predicts the x and y coordinates directly. At 67 FPS, YOLOv2 get 76.8% mAP on VOC2007test and at 40 FPS, YOLOv2 gets 78.6% mAP.

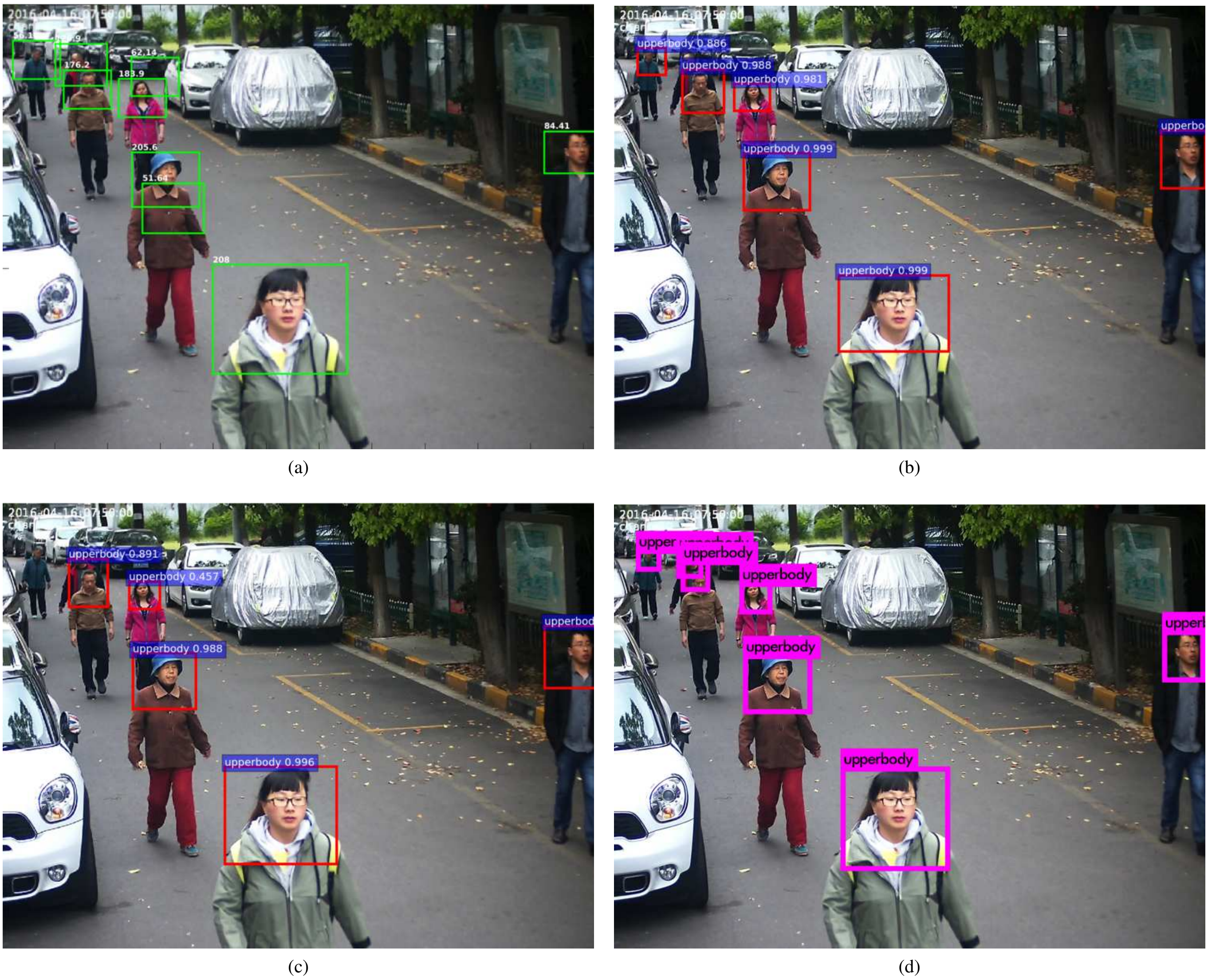
IV. Experiments

A. Experimental protocol

Experiments were conducted on a workstation with Ubuntu14.04 and Nvidia Tesla K40. Training set is composed of 6709 images randomly chosen from the dataset, and the rest 2876 images constitute testing set.

As for measurement, many researchers use precision/recall curve or average precision to measure the object detection performance. The precision/recall curve used in the PASCAL object detection challenges [15] shows the relationship between precision and recall rate, by plotting precision $p(r)$ as a function of recall r. Average precision (AP) is used to summarize the performance of precision/recall curve and is the average value of $p(r)$ over the interval from $r = 0$ to $r = 1$. In our experiments, to make the training results comparable and more intuitive, AP is applied to measure the performances of different detection methods and the threshhold of Intersection over Union (IoU) is changed to compare their localization accuracy.

Intersection over Union (IoU) is an evaluation metric used to measure the accuracy of an object detector. To compute the IoU, the ground-truth bounding box and the predicted bounding box are necessary. Then the area between two bounding boxes and the area encompassed by both two bounding boxes need to be compute. Dividing the area of overlap by the area of



(a)

(b)

(c)

(d)

Fig. 4. The detection result of different methods. (a) ACF Detector, (b) Faster R-CNN, (c) SSD, (d) YOLOv2.

union yields the final score. When the predict position is more accurate, the overlap area and union area are closer, and the IoU is closer to 1. A threshhold is used to calculate the AP of our detector. If the IoU of a predicted bounding box is bigger than the threshhold, it is accurate and average precision is calculated. The detection position accuracy can be seen through the change of average precisions under different threshholds. If the average precisions change a lot, the position accuracy of the detector is bad. In our experiments, the latest version of Dollars Computer Vision MATLAB Toolbox [19] is used to train aggregate channel features (ACF) [11] upper-body detector and compared with other DCNN-based detection methods. The negative samples are extracted from the background of each image randomly. The model's size is changed from 100×41 to 36×45, because of the shape of upper-body region.

Then Faster R-CNN is applied with network VGG16 [20] . The open-source Faster R-CNN has both Python version and MATLAB version. In our experiments, we used Python version. The sizes of input images were not generalized and negative samples were extracted from the background of positive images. Other network configurations and parameters were the same as the original paper.

We also applied Python version of SSD with pre-trained network VGG16. The input images are generalized to 300×300 pixels. The negative samples and other settings are the same as Faster R-CNN.

For YOLOv2, its Python version is used and YOLOv2 uses its own darknet instead of VGG16. At the beginning, it resizes the input images to 448×448. The other network configurations are the same as above.

B. Comparisons

Table 1 shows the upper-body detection average precision (AP) on the test dataset. Obviously, all the deep learning approaches have better results than ACF-based detection method. When IoU is 0.5, the AP of it is 0.817. SSD and Faster R-CNN have the approximate AP by 0.9056 and 0.9074, while YOLOv2 achieves 0.9710 when the IoU is 0.5. But when we

	ACF	Faster R-CNN	SSD	YOLOv2
IoU = 0.25	0.9076	0.9065	0.9081	0.9893
IoU = 0.5	0.8466	0.9056	0.9074	0.9712
IoU = 0.7	0.4576	0.797	0.9036	0.8656
IoU = 0.8	0.1306	0.5203	0.8049	0.5112

TABLE I
AVERAGE PRECISION (AP) OF DIFFERENT METHODS WITH DIFFERENT IOU

	ACF	Faster R-CNN	SSD	YOLOv2
Detection time	0.642	0.244	0.092	0.056

TABLE II
AVERAGE DETECTION TIME

increase the IoU to 0.7 or even 0.8, the average precisions of YOLOv2 and Faster R-CNN drop rapidly to 0.5, while 0.8 for SSD, which means the positions these two methods give are not as exact as SSD.

Figure 4 shows the different detection results for the same image by the four methods mentioned above when the IoU is 0.5. As we can see in Figure 4(a), there are many false positive samples in the result of ACF detection. For Faster R-CNN in Figure 4(b), the results and positions are good and the detected upper-body regions are all with high scores. In Figure 4(c), the result of SSD detection fails to find some small upper-body regions, like the man in the top left corner. Figure 4(d) shows the YOLOv2 detection result. The locations of resultant bounding boxes are not accurate compared with other methods. Like the man in top left corner, some people are only cropped with head.

As the IoU changes, the performances of the four methods change a lot. As we can see, SSD detection result seems stable and changes only a little from 0.8 to 0.9. At the same time, the results of YOLOv2, Faster R-CNN and ACF detection change a lot. This means detection of SSD has a better location accuracy.

Table 2 shows the average detect time for the test set. Of course, all the deep learning methods are better than ACF. Among them, YOLOv2 turns to be the fastest and is about 4 times better than Faster R-CNN. YOLOv2 and SSD both skip the proposal step and predict bounding boxes directly. This gives them a high speed in return. SSD and Faster R-CNN both use a deep and complicated network, VGG-16. At the same time, YOLOv2 uses a custom network based on the Googlenet [21] architecture, which is much lighter and faster than VGG-16.

In our experiments, we can find that YOLOv2 and SSD are fast enough to do real-time detections. SSD has a weakness on detecting small objects but the positions of its bounding boxes are very accurate. On the other hand, YOLOv2 is good at detections when its location accuracy is not required.

V. CONCLUSION

In this paper, we thoroughly investigated the problem of vision-based upper-body detection. Specifically, we made two contributions. Firstly, we established a new dataset for human upper-body detection. Unlike other upper-body datasets acquired from TV shows, the images in our dataset are captured from surveillance cameras and can be used in many applications such as public security system or ADAS. Secondly, the performances of four representative object detectores were thoroughly evaluated in the context of upper-body detection. Among them, there are DCNN-based while the other is ACF-based. The evaluation results can serve as a baseline when other researchers develop even more advanced approaches in this area. In the future, we will expand our dataset and collect more images for surveillance system.

VI. ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of China under Grant No. 61672380 and in part by the Fundamental Research Funds for the Central Universities under Grant No. 2100219068.

REFERENCES

[1] D. Geronimo, A.M. Lopez, A.D. Sappa, T. Graf: Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans. PAMI, vol. 32, no. 7, pp. 1239–1258 (2010)

[2] P. Viola, M. Jones: Rapid object detection using a boosted dascade of simple features. In CVPR, pp. 511–518 (2003)

[3] N. Dalal, B. Triggs: Histograms of oriented gradients for human detection. In CVPR, pp. 886–893 (2005)

[4] P. Felzenszwalb, D. Mcallester, D. Ramanan: A discriminatively trained, multiscale, deformable part model. In CVPR, pp. 1–8 (2008)

[5] A. Krizhevsky, I. Sutskever, G.E. Hinton: Imagenet classification with deep convolutional ceural networks. In NIPS, pp. 1097–1105 (2012)

[6] B. Leibe, E. Seemann, B. Schiele: Pedestrian detection in crowded scenes. In CVPR, pp. 878–885 (2005)

[7] A. Prioletti, A. Møgelmose, P. Grisleri, M.M. Trivedi, A. Broggi, T.B. Moeslund: Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation. IEEE Trans. ITS, vol. 14, no. 3, pp. 1346–1359 (2013)

[8] J. Charles, T. Pfister, M. Everingham, A. Zisserman: Automatic and efficient human pose estimation for sign language videos. Int'l J. Comp. Vis, vol. 110, no. 1, pp. 70–90 (2014)

[9] A. Patron-Perez, M. Marszalek, I. Reid, A. Zisserman: Structured learning of human interactions in TV shows. IEEE Trans. PAMI, vol. 34, no. 12, pp. 2441–2453 (2012)

[10] V. Ferrari, M. Marin-Jimenez, A. Zisserman: Progressive search space reduction for human pose estimation. In CVPR, pp. 1–8 (2008)

[11] P. Dollár, R. Appel, S. Belongie, P. Perona: Fast feature pyramids for object detection. IEEE Trans. PAMI, vol. 36, no. 8, pp. 1532–1545 (2014)

[12] S. Ren, K. He, R. Girshick, J. Sun: Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, pp. 91–99 (2015)

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg: SSD: Single shot multibox detector. In ECCV, pp. 21–37 (2016)

[14] J. Redmon, A. Farhadi: YOLO9000: Better, Faster, Stronger. arXiv:1612.08242 (2016)

[15] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman: The pascal visual object classes (voc) challenge. Int'l J. Comp. Vis, vol. 88, no. 2, pp. 303–338 (2010)

[16] R. Girshick, J. Donahue, T. Darrell, J. Malik: Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, pp. 580–587 (2014)

[17] R. Girshick: Fast r-cnn. In ICCV 2015, pp. 1440–1448 (2015)

[18] J. Redmon, S. Divvala, R. Girshick, A. Farhadi: You only look once: Unified, real-time object detection. In CVPR, pp. 779–788 (2016)

[19] Piotr's Computer Vision Matlab Toolbox (PMT), https://github.com/pdollar/toolbox

[20] K. Simonyan, A. Zisserman: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, A. Rabinovich: Going deeper with convolutions. In CVPR, pp. 1–9 (2015)