PART-BODY DETECTION FRAMEWORK FOR PEOPLE DETECTION USING SLICED HOG DESCRIPTORS

Ahmad Sani, Mohd Daud Kasmuni, Mahardhika Candra Prasetyahadi, Mohd Shafry Mohd Rahim and Mohd Shahrizal Sunar

UTM-IRDA Digital Media Centre
Faculty of Computing
Universiti Teknologi Malaysia,
81310 Skudai, Johor-Malaysia

ABSTRACT. *We investigate the possibility for using portions of Histograms of Oriented Gradients (HOG) descriptors in a part- based people detection framework. Instead of extracting descriptors from isolated or pre-cropped human parts, we slice the extracted HOG descriptor from whole windows into four, one slice per one human part. Support Vector Machines (SVMs) are used for classifying the slices and the outcome detections are handled by a finite-state machine where three detected parts means that one assumed person is in the window being scanned. Experiments were conducted for our detection framework and another conventional one that uses whole HOG descriptors using images from the INRIA Person Dataset, in which our framework achieved better; detecting 46/50 of occluded people comparing to 36/50 for the conventional framework. Moreover, we achieved less false positive detections of 80 windows comparing to 289 for the conventional framework.*

KEYWORDS. People detection; object detection; histograms of oriented gradients; part-based detection framework

INTRODUCTION

The research on computer vision has been grown well during the last decade. People detection received good attention for its promising applications where computers can see and decide. Many implementations in road safety and surveillance run algorithms that have been improved over and over. And with the advance in computer hardware, these implementations are becoming capable of running highly computing algorithms. Modern solutions for people detection include frameworks composed of two running algorithms; (a) feature extraction algorithms; and (b) machine learning. Moreover, these frameworks densely scan images for any possibility of people by sliding a window from the top-left to the bottom-right where each window's patch of the image has its features extracted and then classified. Many feature extraction algorithms were proposed and used for people/object detection, examples of popular features include Haar wavelets in (Oren *et al.,* 1997; Viola & Jones, 2001), scale-invariant feature transform (SIFT) in (Mikolajczyk & Schmid, 2005), and the histogram of oriented gradients (HOG) in (Dalal & Triggs, 2005). However, many recent works on object/people detection preferred (HOG) descriptors for its robustness to many issues in object detection, such as illumination, cluttered background, variance in the shape, etc.

In order to increase its power, researches have been using HOG in different models or with other feature extracting algorithms, aiming for tackling advanced issues like occlusion. The deformable part model of (Felzenszwalb *et al.,* 2008) was basically introduced for recognising the different appearances that an object could take. It was improved to handle occlusion of different conditions, such as (Tang *et al.,* 2013; Azizpour & Laptev*,* 2012). Other approaches searches for cluttered areas in images and extracting hybrid features, such as Wang *et al.,* (2009) and Marin *et al.,* (2014). While these works have presented good results by improving HOG descriptors to adapt their detection systems, more research is needed for handling occlusion and different approaches should be presented. And moreover, we believe the increase of complexity for using more algorithms in their detection systems. We propose a much simpler system that utilises original HOG descriptors in different forms by slicing the blocks that belongs to each part we define in our framework. This approach requires only extracting HOG descriptors, slicing them, and then decide whether these slices might belong to a person in the input image. We have detailed these processes in the following sections.

RELATED WORKS

Detecting people using part-based approach has been in research as an alternative way to the conventional whole-object approach and mostly for countering occlusion. An early work like Mohan *et al.,* (2001) used Haar wavelets and SVM for detecting four parts (head, left/right arms, and legs). A much flexible detector in Mikolajczyk *et al.,* (2004) searches for a number of parts then use local context to join them. The deformable part model in (Felzenszwalb *et al.*, 2008) is a recent work which joins different parts regardless of their object's different pose, and later was the basis for Azizpour & Laptev*,* (2012) and Tang *et al.,* (2013). Slicing or cutting from scanning window was seen in Linh *et al.,* (2011) yet they extract HOG descriptors directly from human parts.

METHODOLOGY

There are two detection framework that were taken into study; (i) the conventional whole-body detection framework that utilises the whole window's HOG descriptor and (ii) our part-body detection framework that utilises slices taken from the whole window's HOG descriptor. Both detectors follow similar procedures for training and testing and use a subset of the INRIA Person Dataset which is composed of two sets, i.e. train and test sets and each has positive and negative images. We opted not to use the whole dataset since they contain images that cannot fit well with our approach. Training detection frameworks uses the method of bootstrapping where we train the framework with initial round of positive images and negative ones, then we search the negative images for false positive windows (hard samples) that would be later included in the second and final round of training.

A. Extracting HOG Descriptors

We used the same algorithm presented by Dalal & Triggs, (2005) for extracting a window's descriptor. The process within include: computing gradients, construct local histograms at every cell (a patch of pixels in the window) of oriented gradients, apply overlapping normalisation for each block of four cells, and finally collect overlapping normalised blocks.

B. Slicing HOG Descriptors

This process is used by our part-based detection framework where an HOG descriptor is sliced into four parts; i.e. the human head, left arm, right arm, and legs. We preferred to have both legs together in one slice while separating the arms into two slices although there could be other options for how slicing is made yet we limit ourselves to discovering different sizes for each slice of the aforementioned human body parts. Slicing an HOG descriptor is basically collecting a group blocks from the overlapped normalised blocks which resulted from §III.A. Knowing that a window's 1D descriptor was collected from overlapping blocks (horizontal and vertical, respectively) and each block is made of 36 elements (9 bins for each cell's histogram), we can put the 1D descriptor into a 2D representation we call a 'mapping table' (Fig. 1) at which every cell is the starting index (zero-based) for every block of the descriptor. The mapping table is constructed using (1) and the algorithm in Fig. 2. Based on the above, each part of the four is fixed at one known location in the window, thus training is restricted to images that has these parts located properly. In other words, we cannot train using images of displaced parts such as side views where one arm is shown only, images with arms raised in the air, and so forth. While this could be a backward in our approach, as other works pre-crop and separate their parts for training, we based our approach on the assumption that slicing from a whole window's descriptor retains the benefit of applying normalization on overlapping block, in which neighbour blocks contribute positively on each other even those that lay out of the slice.'

0	36	72	108	144	180	216
252	288	324	360	396	432	468
504	540	576	612	648	684	720
756	792	828	864	900	936	972
1008	1044	1080	1116	1152	1188	1224
1260	1296	1332	1368	1404	1440	1476
1512	1548	1584	1620	1656	1692	1728
1764	1800	1836	1872	1908	1944	1980
2016	2052	2088	2124	2160	2196	2232
2268	2304	2340	2376	2412	2448	2484
2520	2556	2592	2628	2664	2700	2736
2772	2808	2844	2880	2916	2952	2988
3024	3060	3096	3132	3168	3204	3240
3276	3312	3348	3384	3420	3456	3492
3528	3564	3600	3636	3672	3708	3744

Figure 1: The descriptor's mapping table

$$I_{(x,y)} = \begin{cases} 0, & x = 1 \text{ and } y = 1 \\ (7 \times y - 1 + B) + (x - 1 \times B), & 1 < x < 8,\ 1 < y < 16 \end{cases} \quad (1)$$

Input: 64x128 window's descriptor, part's start index, part's width, part's length.
Output: The part's descriptor vector.

1. Allocate a 2-D vector (as wide and high as the part's dimensions according to the mapping table) to store the part's descriptor.
2. Point to the group's first block of the window's descriptor with the help of the mapping table end (1)
3. For every row of blocks between the part's start index and the part's height, do the following:
   a. Allocate a 1-D vector (as wide as the part's width) to hold the current group of blocks.
   b. For every element in the window's descriptor between this group's first block and the part's width, copy into the 1-D vector
   c. Push the 1-D vector's contents into 2-D vector
   d. Point to the next group's first block of the window's descriptor by increasing by 1 in (1)
4. After pushing all part's block into the 2-D vector, the latter is reshaped into a 1-D vector for SVM classification.

Figure 2: the algorithm for slicing HOG descriptor

Figure 3: Invalid and valid images; images in group (a) have human parts located out of their slices while images in group (b) can be used for training

C. Classifier

We use linear support vector machines (SVMs) with as the learning machines for both frameworks as follows: one SVM for the whole-body framework and four for the body-part framework (one for each slice)

D. Training

The classifier(s) of both frameoksis trained with the accumulated training vector of extracted HOG descriptors from all windowsin the train set. This practice is followed exactly by the whole-body framework's SVM. But for the part-body framework, each HOG descriptor from each window is sliced and distributed to four accumulating training vectors of sliced HOG descriptors to train the framework's four SVMs.

A. Testing

Testing detection frameworks follows similar procedures of training detection frameworks. Yet, rather than accumulating training vectors, the classifiers in both frameworks examine and

$$\alpha + \beta = \varkappa \qquad (1) \qquad (1)$$

and return the answer whether a person (or a part of the body) exists or not. Testing is straightforward in the whole-body detector, but more processes are required for our part-body framework, i.e. slicing the descriptor and determine whether a person exists if classifications were positives for three slices. However, we had first to test what size is better for a slice by testing each slice with two sizes exclusively on the positive test set, where the size that achieved more true positive detections is used in the part-body detection framework.

B. Handling Detected Parts

We used a simple implementation of finite-state machine (FSM) for handling detected parts in the part-body framework where a person is detected when three human parts are detected. The FSM (Fig. 4) is loaded with each output from each slice's SVM and in the following order: the head, left arm, right arm, and legs.

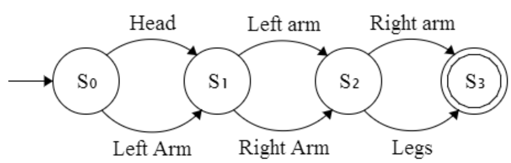

Figure 4: The finite-state machine of the part-body detection framework

C. Scanning and Grouping Multi-Detections

We basically work on isolated windows of individuals, yet, on the processes of generating hard samples and testing on the negative set, a window is slid scanning negative images from top-left to bottom-right searching for any person. The sliding window computes the HOG descriptor and classify at the current position then the window shifts by 8 pixels right or down. Since the classifier(s) may produce multi-detections for one assumed person, we group detections that are close to each other and eliminate 'orphan' detections that do not have one- minimum detection nearby.

EXPERIMENTS

We performed our experiments on an Intel Core i-5 processor at 2.40 GHz Windows PC with 6 GB of RAM and using Microsoft Visual C++ with OpenCV libraries. Our sets of images were initial 1,314 positive cropped windows and 12,180 random windows from negative full images for training; and 50 positive cropped windows and 453 whole-size negative images for testing which will be our evaluation for the two detection frameworks. We implement the conventional whole-body detection framework first in order to prepare the hard samples that will be used for training the second and final round for both frameworks.

A. Whole-Body Detection Framework

We trained this framework using OpenCV's SVMwiththe initial train set and then we scanned negative whole-size images (with grouped multi-detections using an OpenCV built- in function) for hard samples. This added 371 hard samples windows to the initial negative set for the final round of training. Testing this framework is straightforward; for the positive test set, each cropped window is loaded and its HOG descriptor is computed and then classified; and for negative test set, each whole-size image is scanned as explained in §III.G. The testing gave good results (See Table II), detecting 36 occluded persons and achieving a number of 289 false positive detections.

B. Part-Body Detection Framework

We first choose the best size for each slice by testing two sizes on the positive test set. The proposed sizes were based on observing how these human parts are located in the images of the INRIA Person Dataset, in which the head's first upper pixels are located 16 pixels down from the window's top border; the arms are at 24-32 pixels from top; and the legs are at 56-64 pixels from top.See fig. 5 for the proposed sizes and their locations on the mapping table and table I for their performance. We conclude the training for this framework by training each SVM designated for each slice using the same train set including the hard samples that was used in the previous framework. Testing this framework includes the processes of extracting HOG descriptors, slice them, classify them, and then pass the classifications to the finite-state machine to determine if the window has a person. Testing this framework performs this chain of processes once per window for the positive test set, but multiple times using the scanning and grouping method in § III.G for the negative test set. The result for this framework came better than the previous one; the framework detected more people (46 of 50) and avoided more false positives (80 false positive windows), see table II.

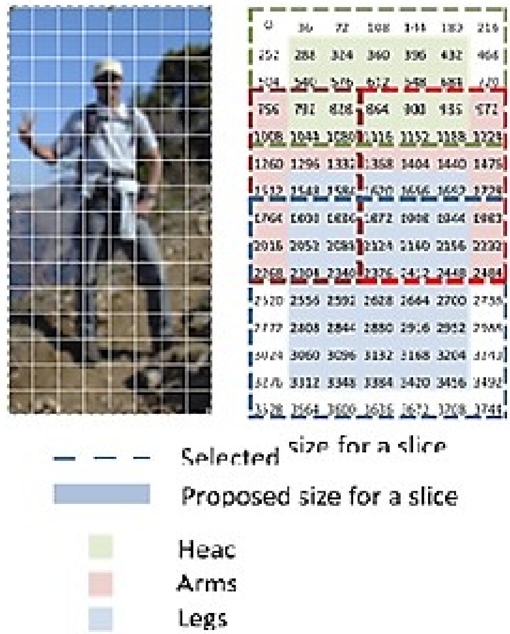

Figure 5: Illustrating the different sizes for each slice on the mapping table

Table I: Detection Frameworks Performance on Test Sets

Slice	Horizontal Blocks	Vertical Blocks	Performance on the Positive Test Set
Head	7	5	36*
	5	4	21
Left Arm	4	7	41
	3	7	46*
Right Arm	4	7	43*
	3	7	40
Legs	7	8	47*
	5	8	40

*. Best performance

C. Disscussion

The results (table II and fig. 6) show that our part-body framework is capable to detect people under occlusion better than the whole-body framework. However, we unexpectedly saw some detection cases had their hidden arms discovered (fig. 7), and thus the finite-state machine returned the answer of people's existence. We can only return this to the training that included few images with arms overlapping over other people.

On the other hand, the whole-body framework still retains some advantage with the ability to estimate hidden arms since it was able to detectmore than half of the positive train set. Albeit this becomes a disadvantage since the estimation could return false positive windows directly, while the part-body framework perform multiple checks before declaring any detection.

Table II: Detection frameworks performance on test sets

Detection Framework	True Positive (Positive Test Set)[a]	False Positive (Negative Test Set)[b]
Whole-Body	36	289
Part-Body	46	80

a. Total windows count is 50

b. Total windows scanned from 453 whole-size images is 865595

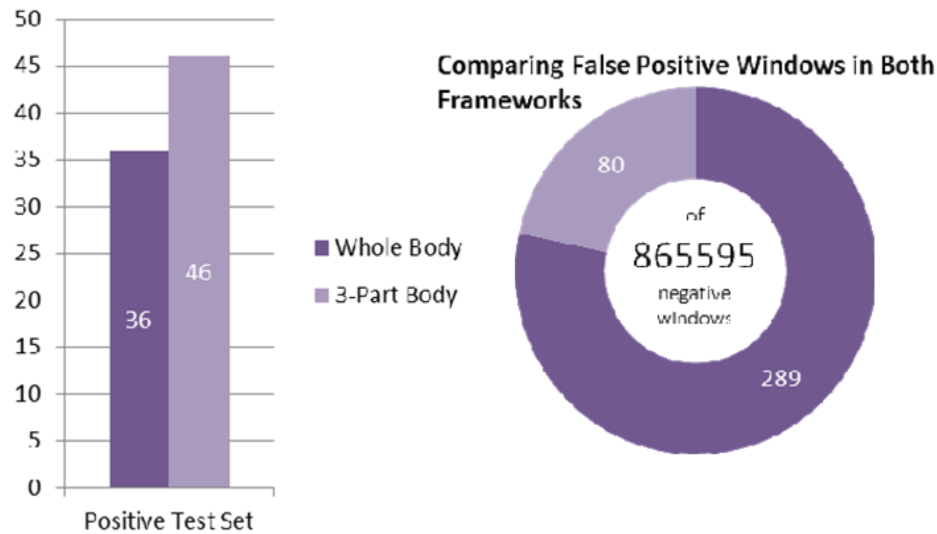

Figure 6: Comparing performance for both frmaeworks on the positive test set (right) and the negative test set (left)

CONCLUSION

We have introduced a part-object detection framework that is powered by a new utilisation of HOG descriptors by slicing them rather than extracting descriptors from pre-cropped parts, and a finite-state machine for handling detected parts.

Figure 7: Sample of results; the output from the whole-body detection framework is on the left on each pair while the part-body framework is on the right.

ACKNOWLEDGEMENTS

This research is supported by UTM-IRDA Digital Media Centre, Faculty of Computing, Universiti Teknologi Malaysia (UTM).

REFERENCES

Azizpour, H and Laptev, I, 2012 "Object Detection Using Strongly- Supervised Deformable Part Models," in Computer Vision. vol. 7572, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., ed Berlin: Springer Berlin Heidelberg, pp. 836-849.

Dalal, N. and Triggs. B, 2005 "Histograms of oriented gradients for human detection," in Conference on Computer Vision and Pattern Recognition San Diego, CA, pp. 886-893.

Felzenszwalb, F., McAllester, D., and Ramanan, D., 2008 "A discriminatively trained, multiscale, deformable part model," in Computer Vision and Pattern Recognition, Anchorage, AKpp. 1-8.

Linh, D., Buu, B., Vo, P. D., Tran, T. N. and Le, B. H., "Improved HOG Descriptors," in 3rd International Conference on Knowledge and Systems Engineering, Hanoi, 2011, pp. 186-189.

Marin, J., Vazquez, D., Lopez, A. M., Amores, J and Kuncheva, L. I., 2014 "Occlusion Handling via Random Subspace Classifiers for Human Detection," Transactions on Cybernetics, vol. 44, pp. 342-354.

Mikolajczyk, K., Schmid, C., and Zisserman, A., 2004 "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," in Computer Vision - ECCV 2004. vol. 3021, T. Pajdla and J. Matas, Eds., ed Berlin: Springer, pp. 69-82.

Mikolajczyk, K., and Schmid, C., 2005 "A performance evaluation of local descriptors," Pattern Analysis and Machine Intelligence, vol. 27, pp. 1615-1630.

Mohan, A., Papageorgiou, C., and Poggio, T., 2001 "Example-based object detection in images by components," Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 349-361.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T., 1997 "Pedestrian detection using wavelet templates," in Conference on Computer Vision and Pattern Recognition, San Juan, pp. 193-199.

Tang, S., Andriluka, M., and Schiele, B., 2013 "Detection and Tracking of Occluded People," International Journal of Computer Vision, vol. 11263, pp. 1-12.

Viola, P. and Jones, m., 2001 "Rapid object detection using a boosted cascade of simple features," in Conference on Computer Vision and Pattern Recognition, Kauai, pp. 511-518.

Wang, X., Han, T. X., and Yan, S., 2009 "An HOG-LBP human detector with partial occlusion handling," in 12th International Conference on Computer Vision, Kyoto, pp. 32-39.