Prototype of Real-time Pedestrian Crossing Detection on Moving Automobile

Teera Siriteerakul and Sirikul Siriteerakul
Faculty of Science, King Mongkut's Institute of Technology Ladkrabang

1. Summary

In Thailand, there are approximately 75,000 accidents per year. These accident resulted in 13,000 casualties, 900,000 injuries, and cost around 100 billion BAHT. Within these numbers, 9.3% of casualties and 5.6% of injuries involves pedestrian. Thus, the prototype of real-time pedestrian crossing detection on moving automobile was develop in order to help reducing both casualties and injuries accident involving pedestrians. The prototype is an onboard device which can detect any pedestrians in the front and alert the driver. The pedestrian detection system adopted Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM) techniques with some modification to run in real-time. We manage to achieve the recall rate of 80% at 10^{-3} false positive per windows at an acceptable speed.

2. Aim of Research

This research aims to develop a working prototype of real-time pedestrian detection which can be attached onboard a moving vehicle. The system must be able to alert the driver when pedestrians are detected. We aims to make the system performed in real-time with good recall rate and acceptable false positive rate. Optimistically, the system should help reduce the number of injuries and casualties due to pedestrian related accident.

3. Method of Research & Progression

3.1 Hardware Component

There are two parts of this research: hardware and software. The design principle of the hardware part consists of low power system, acceptable cost, and easy to install. The system must be powerful enough to perform the detection in real-time. In the same time, the software part must be able to utilize the hardware to perform. Also, the software must follows the design principle: it must be easy to install.

<u>Key Hardware Components</u>

1. Computing unit: Raspberry PI 2 Model B

The choice of computing unit is Raspberry PI 2 model B (see the assembled unit in Fig. 1 and bare Raspberry PI in Fig. 2). The unit run on 900MHz quad-core ARM Cortex-A7 CPU with 1GB of RAM with 40 GPIO (general purpose input/output) which can be connect to external circuit or device such as speaker. Raspberry PI provides its own Linux based operating system which can be installed into a Micro SD card. The unit is powered by cigarette lighter power adapter which can power up to 2 amp at 5 volt (up to 10 watt).

The selected Raspberry PI unit can be regarded as a simple computer board. Thus, we can control it by a keyboard via USB port and view its output in a typical computer monitor via its HDMI port. However both standard input and output for computing unit are unnecessary in our hardware module.

Note that the Raspberry PI also comes with VideoCore IV 3D graphics core and also a port of its on Raspi-Camera. However, we have not yet utilized its GPU unit nor its camera due to time constrain and the limitation of its camera.

Fig. 1 Assembled computing unit

Fig. 2 Bare Raspberry PI 2 model B

Fig. 3: Lighter socket power adapter

2. Camera unit: Logitech C930

Rather than using its own Raspi-Camera, we decided to use Logitech C930 webcam camera due to various reasons. Firstly, the Logitech C930 comes with long USB cable with allowed us to place the computing unit anywhere independent to the camera unit. Secondly, C930 comes with wider field of view (90 degrees) and works better in low light condition. Next, the C930 camera's PTZ (pan, tilt, and zoom) can be control programmically. Lastly, the camera housing allowed it to be placed easily on the car dashboard or windshield.

Fig. 4: Logitech C930 camera unit

Fig. 5: Camera unit attached on windshield

3.2 Software Component

The system software unit developed using C++ with OpenCV. The system utilized Histogram of Oriented Gradient (HOG) for feature extraction and Support Vector Machine (SVM) for classification between pedestrian/none-pedestrian images.

To speed up our system, we limited search area in for the sliding windows and also fixed scale-resolution. The detail of these optimization can be found in the results section.

Key Theories and Techniques

1. Histogram of Oriented Gradient – HOG

For features, we adopted Histogram of Oriented Gradient (HOG) which was designed for detecting pedestrians (upright full body image of human) (Dalal and Triggs, 2005). The HOG of an image is computed by consider the discretized gradient direction of each pixel. Then, a histogram vector is constructed by adding up gradient magnitude using the discretized directions as histogram bins. Then, to lessen the effect of localization (for example, the detected human is not in the center of the detection window), these features are collected as histogram vectors. To do so, each image is divided into blocks (for example, 5x5 blocks). Then in each blocks, a histogram is corrected according to the HOG scheme.

2. Support Vector Machine – SVM

The Support Vector Machine (SVM) (Cortes and Vapnik, 1995) is a classification tool where features vectors are separated into two classes using a hyper-plane. This hyper-plane is founded by searching for the optimal hyper-plane which best separate the training input vectors which are on the boundary and closest to the other clas $\omega^T x + \omega_0$ ne, can be found by minimize the objective function:

$$\min_{\omega,\omega_0,\xi_i} \frac{||\omega||^2}{2} + C \sum_i \xi_i \qquad \text{-- (1)}$$

subject to $y_i(\omega^T x_i + \omega_0) \geqslant 1 - \xi_i$ and $\xi_i \geqslant 0$ where C is the parameter to be set in training.

3.3 Data Collection

Data collection is done by recording image sequences using a separate program (also developed for this research) running on the computing unit. The video record is of size 640x480. The data collected is more than 40 hours of video from location nearby Ladkrabang, Bangkok. The data is then manually labeled manually for pedestrian location in image for experiment.

4. Results of Research

The resulting prototype can be attached onboard a moving vehicle and alert the driver upon pedestrian detection. The system runs approximately 5-6 frame-per-second. The system achieved approximately 80% recall rate with 10^{-3} false-positive-per-window.

There are two main optimization techniques we applied in order to improve the frame rate: space and scale reduction. For space reduction, we limit sliding window search space into roughly 30% of the 640x480 pixel image (illustrated in figure 6). For scale reduction, we only perform search only in one scale (contrasting the original Dalal and Triggs technique which resize the image for multi-scale search).

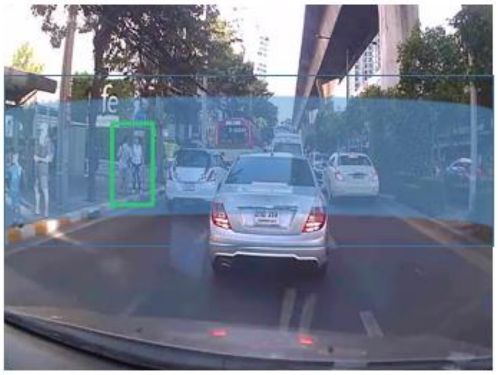

Fig. 6 Example of search space (highlight area), true positive (green rectangle), and missed (two person on the left).

5. Future Areas to Take Note of, and Going Forward

One way to improve the system speed is trying to utilize GPU available on the Raspberry PI unit. Also, we can consider another hardware such as nVidia Jetson TK1 which compose of 192 GPU CUDA cores and a CPU with two times faster in speed comparing to Raspberry PI 2 Model B. However, cost of the nVidia Jetson TK1 is approximately four times the cost of the Raspberry PI unit.

Another way to improve speed is to apply different technique such as HAAR-like feature with cascade classifiers, similar to face detection techniques purposed by Viola and Jones (2001). However, Enzweiler and Gavrila (2009) reported that even though HAAR-like techniques perform well in lower resolution, they perform inferior to HOG-based techniques when enough resolution available.

6. Means of Official Announcement of Research Results

There are several academic journals we consider for publication. First, we have KMITL Science Journal. However, we can consider journal with more impact such as those in Scopus or ISI-based. The choice of academic journal will become clear when we complete our academic paper.

Reference

- Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511). IEEE.
- Enzweiler, M., & Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(12), 2179-2195.