An Efficient Walking Safety Service for Distracted Mobile Users

Abstract—**There is a growing number of incidents related to using cell phones while walking on the street. To address this issue, this paper proposes a new system based on tactile paving detection on the sidewalk to alert distracted mobile users to avoid traffic hazard. Our system (namely `Inspector`) is deployed as an application on an off-the-shelf normal mobile phone equipped with back camera. `Inspector` plays as a third eye to alert users when they step out of the safe zones where no tactile paving is detected. In order to obtain reliable yet effective decision results, we exploit a lightweight image processing method, simple classifiers along with a smart sampling strategy. The main idea is that the application capture more images of the surrounding environments when we doubt that the result from one image is not sufficient. The sampling interval is adjusted dynamically so that we can save the energy and thus extend the working time of the system. Real-scenario tests show that `Inspector` can detect whether a mobile user is walking along a blind sidewalk with an accuracy of 92.72%, 98.78%, and 99.44% when the detection algorithm sample 2 times, 3 times, and 4 times continuous detection respectively. The reaction time, which is measured by the difference between the alert time and the time when a user steps out of a safety zone, is 0.52 seconds early with a sampling interval of 2 seconds.**

I. Introduction

There is a growing concern on the danger of distracted walking involving cellphones. Researchers in Ohio State University discovered that more than 1,500 pedestrians were treated in emergency rooms in 2010 for injuries related to using a cell phone while walking [1]. Pedestrians fatalities when using the mobile phones have increased from less than 1% in 2004 to 3.6% in 2010 [2]. Recently, a study [3] published by New York's Stony Brook University found that those, who are texting while walking, were 60 percent more likely to veer off line than non-texters. Unfortunately, there is no valid way to low down the distracted walking risk regardless of the efforts of Arkansas, Illinois, and New York to ban the behavior of using a mobile device while walking ([4] [5]).

To address this problem, research community has devoted a lot of efforts using technologies such as wearable devices, mobile sensing, RFID embedding, to name a few. These technologies are able to tract user's behaviors and positions to avoid dangerous situations. Wearable sensing on shoes is introduced on terrain gradient profiling [6] to protect users from walking onto the road at cross-points. The result is collected by mobile sensors through a wireless bluetooth connection. Unfortunately, detecting transitions (inclination of the ground or ramp) can not avoid people incidentally walk to curbs and falling there. Also, users may refuse to put sensors and battery on their shoes. In other attempts, safety related applications (on smartphones) are developed to reduce the distractions caused by these smartphones. Using human activity analysis, the movement pattern of mobile phone users can be computed by sampling from embedded sensors [7]. An alert will be shown on the screen when people are using mobile phones and walking. Walksafe project [8] uses cameras on pedestrian's smartphones to detect oncoming vehicles. However, previous image-based approaches have many drawbacks. First, since the phone cameras usually face to the ground when people are walking while reading on the mobile, special devices (like camera on hat) are usually required to monitor the front scene. Moreover, they need to track multiple objectives in a complicated background, which significant improvements on accuracy and simplifying the computation may required to make these approaches feasible to be deployed on mobiles. Besides, continuous camera operations can cause high energy consumption and thus users might be reluctant to install such applications.

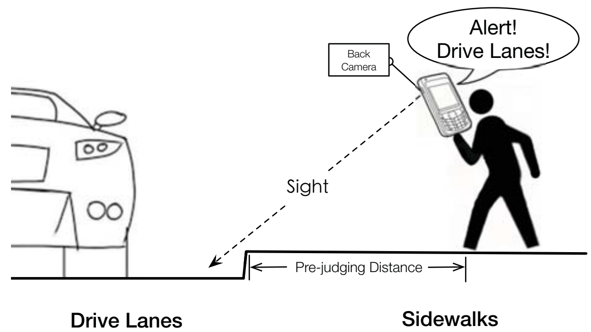


Fig. 1: Pedestrian safety service on mobile for distracted walking

In this paper, we propose an energy efficient vision-based approach to reduce the risk caused by distracted walking. The main idea is motivated by the tactile ground surface indicator on footpaths in urban environments for visually impaired pedestrians, *the tactile paving*. We leverage this widely deployed design feature in our society and develop a vision-based system to detect the distinctive surface pattern of truncated domes, cones or bars. While a pedestrian is

978-1-5090-2833-7/16 $31.00 © 2016 IEEE
DOI 10.1109/MASS.2016.55

texting or reading on her mobile phone, its back camera is able to serve as a third eye. As cellphones are often held in a inclined angle, the back camera will always capture the variation on the road surfaces. Using a limited number of captured images, the system can detect whether the user is in the safety area along the tactile ground surface indicators with high accuracy but less energy consumption. Since we focus only on single object detection, the problem is much simpler than multi-object detection approach [8]. Because the tactile paving often has distinctive features, we can reduce the rate of wrongly detecting drive lanes as safety zones. Moreover, since all the process is realized on mobile, it does not introduce any communication cost, and thus, this simple design much easier to be accepted by users. One may consider that the vision-based approach is not suitable for crowded areas or using at night. However, in such scenarios people usually avoid reading or texting on mobiles actively due to the inconvenience. Since a majority of accidents happen during road crossing at intersection or mid-block locations [6], the proposed approach aims to keep pedestrians in relatively safe areas on a sidewalk, and alert the distracted pedestrians when approaching to the road or crossing the street.

We build a pedestrian safety service on mobile phones based on the detection of directional blocks of tactile paver on sidewalks, as illustrated in Fig. 1. There are two challenges in building such systems. First, the system must be fast and lightweight to reduce the energy and computation requirement. And second, it must be reliable in detecting safety zones, i.e. the tactile paver on the sidewalks. We target the first challenge by making use of fast image processing technique [9] in combination with simple yet effective classification methods, i.e. Normal Bayes classification and K-nearest neighbors (KNN) [10], [11]. To achive reliability, we propose an adaptive sampling method to make decision based on continuous captured images. It is observable that using a newly captured image combined with previous images around that time will provide more robust decisions. It has been proven that ensembling detection results is able to obtain very good performance even with very basic classifiers [12]. This design is particularly useful when walking on long-distance blind-sidewalks.

In summary, the contributions of this work list as follow:

- Proposing the use of tactile paver detection on sidewalk to build a pedestrian safety service on mobiles. Since we focus on one object (the tactile paver) instead of multi-object detection, the requirement is simpler than previous vision-based approach [8], and thus more suitable to be implemented in mobile devices.
- Designing adaptive sampling algorithm which automatically adjusts the sampling interval to improve the accuracy of safety detection and reduce the energy consumption. The algorithm is simple but effective for the long distance sidewalk by leveraging the results of previous detections.
- Prototyping the design on off-the-shelf mobiles which demonstrates the feasibility of our vision based pedestrian safety service in real environment.
- Extensive experiments at different cities have shown that our system is capable of alert detracted walkers with an average accuracy of 92.72%, 98.78% when the algorithm demands 2 times, 3 times continuous detections respectively. The reaction time, which is measured by the alert time minus the time when a user steps out of a safety zone is around 0.52 seconds early with a sampling interval of 2 seconds.

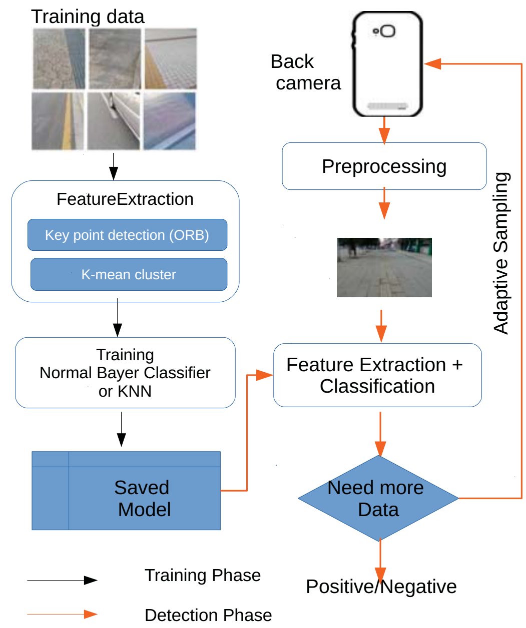


Fig. 2: System Architecture

The rest of this paper is organized as follows. Section II describes the system design and corresponding algorithms for safety monitoring. Section III shows experimental results. Section IV provides an overview of related works. A discussion about the extension of our works is depicted in Section V. The paper is concluded in section VI.

II. System Design

A. System Overview

Our primary idea is to distinguish the safe sidewalk and dangerous road for pedestrian through the directional blocks of tactile paver recolonization based on the image captured by back camera of mobiles. The architecture of our system is illustrated in Fig. 2. First of all, we collect the sampling images with and without the detectable warning surfaces in urban and suburban area. Secondly, through training the sampling images and careful tuning of parameters, a classifier model is generated. Thirdly, we propose an algorithm to analyze the result achieved from the sampling data and determine the sampling interval to achieve high accuracy but low energy cost. The key components of our system are *building classifier model* and *adaptive sampling algorithm.*

Building Classifier Models Given a set of training examples, which are images labeled manually as positive and negative

Fig. 3: Some samples of positive images with tactile paving.

Fig. 4: Some samples of negative images with drive lanes.

classes, the objective of this step is to build a classification model to automatically detect tactile paving in future scenarios. The training phase is shown on the left side of Fig. 2, which consists of two sub-components: 1) the feature extraction based on key point detection and clustering; and 2) classification. In this study, we made use of simple key point detection [9], and K-mean clustering [11] for the feature extraction sub-component, and Normal Bayes Classification [10] and KNN [11] for classification.

Adaptive sampling algorithm When a pedestrian walks along the sidewalks, the photos captured by the back camera are supposed to contain the tactile ground surface. Due to some interferences such as poor light conditions, obstacles on the sidewalk, we may get conflicting results from two continuous photos. This case also happens when the pedestrian walks out of the safety area of sidewalk or at the edge of the sidewalk. To resolve the conflict, the system needs to take into account the decision result from a third photo. We follow an adaptive sampling algorithm to sample and combine decision results from a series of photos. The sampling interval is adjusted dynamically so that we can save the energy and thus extend the working time of the system.

In the following sections, we introduce the details of our system design.

B. Samples Collection Method

In reality, the captured images contain not only the tactile paving but also many other objects like trees, pedestrians, telegraph pole, etc. As a result, the images in training dataset should fit the practical situation. We collect data by emulating a pedestrian who were texting or surfing on the Internet when walking on the sidewalk. The raw data are collected from both downtown area and suburban area in Nanjing and Chengdu, China. And it is discovered that the back camera is able to capture a clear photo with a normal walking speed. The raw data are classified into positive and negative sets according to whether there is a tactile ground surface indicator there. Fig. 3 and Fig. 4 show the samples of positive and negative images.

C. Building Classifier Models

We cast the problem of detecting safe walking zone based on tactile paving into a binary classification problem. In other words, we would like to categorize a captured image into 2 classes, which are safe (positive) or not safe (negative). The training phase to build classifier models is shown on the left side of Fig. 2. This section will briefly introduce the phase of feature extraction and classification.

For feature extraction, we first use ORB (Oriented Fast and Rotated BRIEF) [9] to detect key points from every image. ORB has similar performance with SIFT [13], the famous key point description for image matching. However, ORB is faster and more suitable for low-energy devices such as mobile phones. Using ORG, we obtain a set of key-point description vectors in R^{32}, the 32-dimensional space, for each image. An example of keypoints is shown in Fig. 5. We then use "K-means" clustering algorithm [11] to obtain a set of T prototypes (visual words) from a sample of keypoint descriptors. This prototypes capture common "patterns" of blind sidewalks in the training dataset. We map each key point in one image into the nearest prototype, thus obtain a set of prototypes for each image. This representation is also referred as bags of "visual words", which borrows the concept of bags of words in text analysis. The histogram of the prototypes of an image forms a feature vector of size T for that image. The number of prototypes (features) T is one important parameter to tune during training process.

As for classification, we consider two simple classifiers, the Normal Bayes classifier [10] and KNN [11]. Normal Bayes classification model assumes that feature vectors from each class are normally distributed, and the whole data distribution is a mixture of Gaussian distributions, one from the positive class, and one from the negative class. In contrast, KNN classify an object by majority voting of its nearest neighbors in the feature space, i.e., the T dimensional space in our case. The main reason for selecting these methods is because we would like to have lightweight classifiers to suit the computation power of mobile phones. It has been shown that even with simple classifiers, a smart decision strategy can result in a robust and efficient system [12]. In our experiments, we will compare these two classification methods in order to pick the most suitable one for our system.

In summary, there are several parameters that may influence the performance and efficiency of our system, i.e., the size of the input images, the number of features (T), and the classification methods.

- Image size. The sampling images captured from different devices are various, as well as their size, so that we need to resize them to a standard size for building the model. Another reason is that if an image is captured from smartphone camera in a common way, the image size will be so large that there are too much information to process in the smartphone. Accordingly, we try different size standards to resize the picture, and after that, we train and test the processed images to get results.

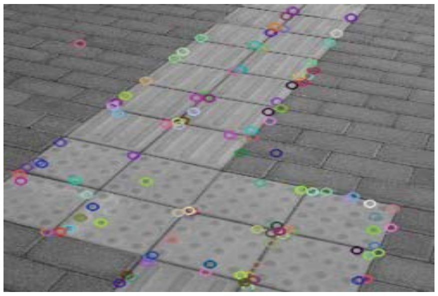


Fig. 5: Key points are highlighted on one sample image

- Number of Features. The number of features (or prototypes) T is essential to the performance of our system. A large number of features may result in better performance, but also consume more energy. The objective is to balance between the performance and the energy cost.
- Classifier. We compare two simple classifiers Bayes classifier and KNN to select a suitable method to be deployed. For experiments, we use the implementation of these two methods in OpenCV [14].

When all tests in studying parameter influence are finished, we select a satisfying model with specific parameters for deployment to mobile phones.

D. Adaptive Sampling

Considering a mobile user who is texting when walking on the sidewalk, during the time she is moving to the vehicle lanes, the cellphones back-camera will capture a serie of photos with tactile ground surface indicator disappearing in the frames. That is because the sidewalks and the driveways usually have a distance and pedestrians can not get into another immediately without traces. We can recognize the danger by detecting the exception in the pattern that a negative image follows a series of negative images or a positive image follows a series of positive ones.

Another issue to be addressed is the false positive caused by environment change or wrong detection. Through a period of mobile phone testing on multiple sidewalks in two cities, it is discovered that most of the false predictions are scattered, which means a false tactile paving recognition can be corrected by continuous sampling. Based on this observation, the application will adjust the sampling frequency dynamically when seeing an exception.

In general, a fixed sampling interval is applied when capturing the images through back-camera of mobiles. Considering the average human walking speed is about 5.0 kilometers per hour (km/h), we select the interval of 2 seconds and 5 seconds in urban and sub-urban area respectively, which represents the distance of 2.7 m and 6.5 m accordingly. The judgment of "positive" (resp. "negative") is given by the application when n continuous positive (resp. negative) detection of captured images appears. By so doing, we can significantly improve the accuracy. On the other hand, *this operation reduces the accuracy requirement of the detection of a single image, which allows to deploy a simple but fast and low energy cost algorithm on our mobiles.*

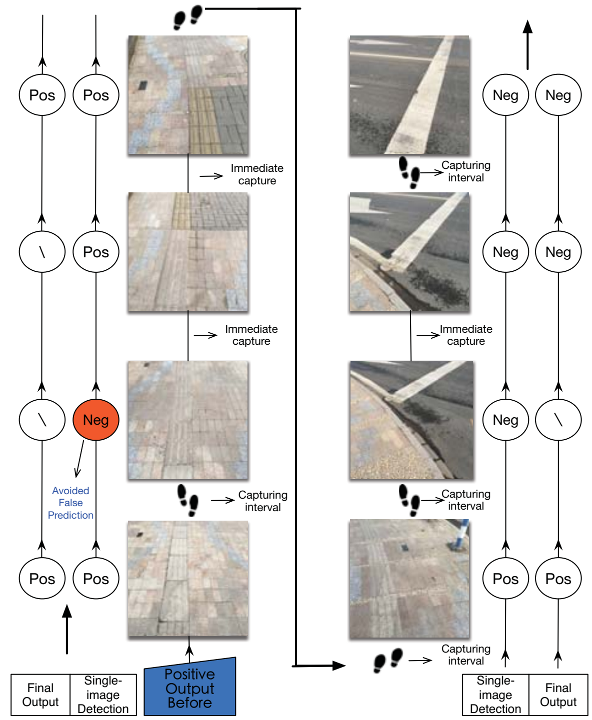


Fig. 6: An example of adaptive sampling (sample 2 times) to detect the directional blocks of tactile paver at sidewalk

Once an exception happens, n times of continuous sampling will be performed directly to evaluate the correctness of the detection and make an adjustment afterward. We use an example to explain this procedure. The corresponding algorithm is shown in Algorithm 1. As illustrated in Fig. 6, a pedestrian is walking on the sidewalk. The application captures the photo and gives a positive (see the tactile paving on the image) judgment. After an interval of time, one image is captured but a wrong recognition is shown. This is a false negative result. Instead of giving a warning to the pedestrian, the application will take continuous two photos, i.e., $n = 2$, and see the recognition results. Since both two photos indicate the positive result, the application will consider the situation is safe. After another two intervals of time, the pedestrian comes to the cross-point. And the image recognition result becomes negative again. The application will take another two continuous photos, both of which show the negative results. Therefore, the application will send an alert to the pedestrian.

III. Evaluation

Our evaluation includes two cases: 1) the training and evaluation on computers; and 2) the real-scenario evaluation on mobile phones. The first evaluation, which is performed on computers, is conducted with different parameters, i.e. image sizes, the number of features, and the types of classification

Algorithm 1: Adaptive Sampling Algorithm

Input: P_{now} - Recognition result on new photo
P_{prv} - Recognition result on previous photo
R_{prv} - Previous Result given by application
n - Requirement of continuous detecting times
T_{det} - Number of detections
Output: R_{now} - Result given by application now

if $P_{now} == R_{prv}$ **then**
  $R_{now} \leftarrow P_{now}$
else
  while true **do**
    if $P_{now} == P_{prv}$ **then**
      $T_{det} \leftarrow T_{det} + 1$
    else
      $T_{det} \leftarrow 0$
    end if
    if $T_{det} >= T_{req}$ **then**
      return $R_{now} \leftarrow P_{now}$
    end if
  end while
end if

methods on a training dataset. The main purpose of this step is to select and build a suitable model for the second step (the real-scenario on mobile phones).

A. Training and Testing on Computers

To develop and evaluate our approach, we have collected over 20 hours of transportation data from 4 individuals. We implement the mobile phone applications with the Android platform. From the recorded data, we sample 203 images (102 positive, and 101 negative images) for building models, which seems not enough in quantity but actually already makes the model perform well in real-scenario.

Given the training dataset, for each of 51 values T (from 100 to 600), we perform 3-time 10-fold cross validation. Here, 10-fold cross validation means we divide the training dataset into 10-folds, and in turn, take one fold for testing and the rest for training. By doing that way, every image in the training dataset is tested once. In order to randomize the partition of 10-folds, we randomly shuffle the training dataset 3 times, thus obtain 3-time 10-fold cross validation. As a result, for each value of T, we obtain an average result of 30 tests (3*10 folds). All the experiments are conducted in Apple MacBook Pro with a 2.7 GHz Intel Core i5 in the software Eclipse.

1) Influence of the Number of Features and Classification Methods: By fixing the image size of 250×250, we study the impact of the number of features T on the performance. Changing T from 100 to 600, we obtain 51 values with a step 10 from 100 to 600, Fig. 7a demonstrates the class-dependent accuracies with Bayes Classifier. It is observable that a larger number of features leads to a higher accuracy for positive class, and a slightly lower accuracy for negative class. The class-independent accuracy, however, tends to increase with the larger number of features in general (from 0.6 to 0.8). The number of features reach acceptable performance with $T = 200$ features. Fig. 7b shows the influence of T on KNN Classifier. As for KNN Classifier, it performs worse when increasing the number of features, reflecting in the decrease of the positive rate. This might be because when more features are used, more noises are introduced instead of helpful features, KNN is more sensitive to noises than Bayes classifier in this specific problem.

The cost of detecting time also varies when T changes, which is showed in Fig. 7c. In Bayes Classifier and KNN Classifier, the detecting time both get a slight growth and the Bayes Classifier costs a bit more than KNN.

We also evaluates the time for loading the save classifier models with KNN and Bayes, which determines the start-up times of `Inspector`. The analysis is shown in Fig. 7d, and the obtained curve pretty fits a quadratic function. In other words, the loading time grows rapidly as feature points quantity increases. Fortunately, we only need to load the model once for `Inspector` to operate.

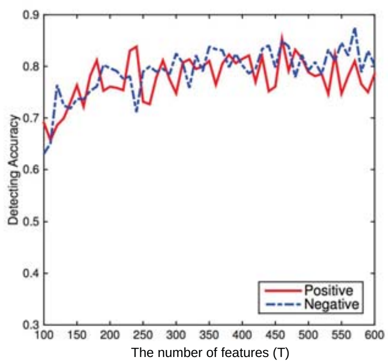
In short, the above experiments suggest that the number of features T of 350 and Bayes Classifier are suitable for our systems.

2) Influence of Image Size: Using the Bayes Classifier and $T = 350$, we measure the detecting accuracy and time for images sizes vary from 100×100 to 400×400. The results are illustrated in Fig. 7e and Fig. 7f. We find out that the detecting time increases rapidly as images get larger, which is easy to understand. For detecting accuracy rate, it shows a great increase from 100×100 to 300×300 and a slight change after 300×300. Therefore, we use the size of 300×300 as the standard size to captured photos.

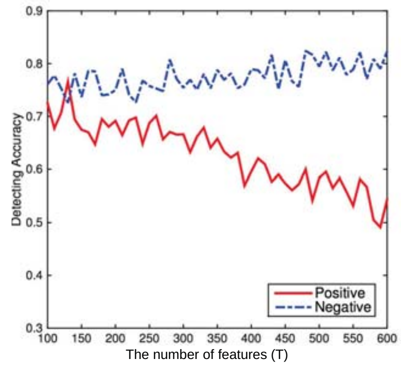
The best accuracy result that we obtained on the training dataset with Bayes classifier, $T = 350$, and image sizes of 300×300 is 79.28%. Although this is not so high, it is enough to be used for single-image detection. In reality, we will make decision by combining results from 2 or 3 continuous captured images, and thus the performance is expected to be higher (see Section II-D).

B. Real-scenario Testing on Mobile Phones

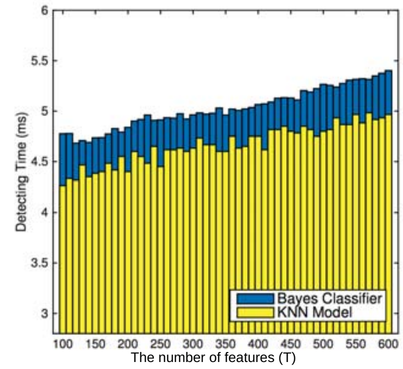
The following section sets out our work on measuring `Inspector`'s performance with a series of real-scenario mobile tests. Compared with the scenarios where we capture the training pictures, the testing scenarios have the same lighting condition but different location condition. In details, we implement `Inspector` from 9:00 a.m to 5:00 p.m when the sunlight is bright enough for cellphone to capture a clear picture, and we choose three streets, paved three kinds of tactile sidewalk strips, as mobile phone testing scenarios. In these tests, `Inspector` is running on the cellphone Samsung SCH-i579 Galaxy Ace Duos Specs that released in 2012 and equipped with a 1350mAh battery and a 800MHz single core CPU. The version of the Android system is 2.3.6. In fact, the testing phone we choose has modest computational strength in comparison with the popular smartphone in markets. Nev-



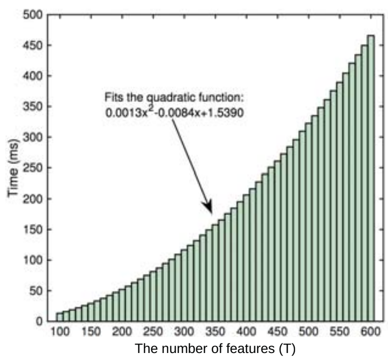
(a) Bayes Classifier



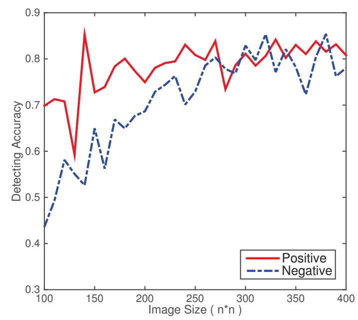
(b) KNN Classifier



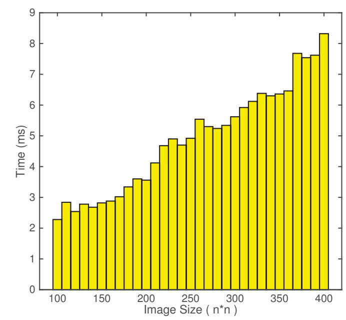
(c) Detection time w.r.t T



(d) Loading time w.r.t T



(e) Accuracy w.r.t image sizes



(f) Detecting time w.r.t image sizes

Fig. 7: Experiment results on computer with different parameters. (a).The detecting accuracy with Bayes Classifier and various feature points amounts. (b).The detecting accuracy with KNN Classifier and various feature points amounts. (c).The detecting time with two classifiers and various feature points amounts. (d).The loading model time with Bayes Classifier and various feature points amounts. (e).The detecting accuracy with Bayes Classifier and various image size set. (f).The detecting time with Bayes Classifier and various image size set.

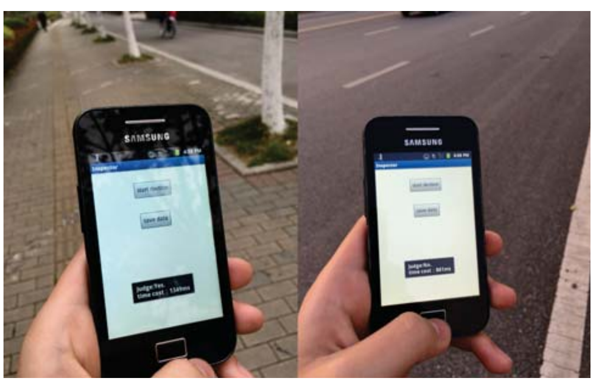
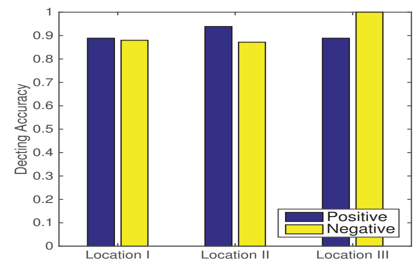


Fig. 8: Testing results in positive and negative scenarios

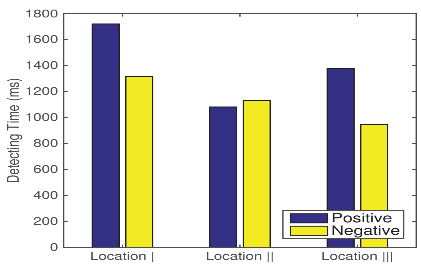
ertheless, we can obtain good performance with fundamental back camera, CPU and the application `Inspector`.

1) Detection Accuracy and Detection Time: In order to consider environmental discrepancies, three places in Nanjing City are involved into our experiments and the actual testing scenes is showed in Fig. 8.

The method of measuring `Inspector`'s performance in real-scenario is to emulate the potential behaviors of distracted walkers, particularly, including two kinds of routes: 1) the safe routes where users walk on the sidewalks all the time and 2) the unsafe routes where users step into the drive lanes from different positions of a sidewalk. The results of single-image detection accuracy and detection time are shown in Fig. 9. We can find that the accuracy is about 91% which is higher than that on the training dataset. The main reason is that training the dataset includes all kinds of tactile paving (e.g. various color, various dimensions and even a corner of tactile paving), and thus a lot of images included are difficult to distinguish. In contrast, in our real-scenario evaluation, pedestrians usually walk on the safety area (e.g. the middle part of the sidewalks) where `Inspector` can easily get a correct judgment. When we combine decision results using our adaptive sampling, the performance improves significantly, which are shown in TABLE. I. Combing two continuous detections gives the performance of 95.08% in comparison with 91.14% in the case of single-image detection. The trade-off we have to make for better performance is the higher



(a) Detection accuracy (sample one time)



(b) Detection time (sample one time)

Fig. 9: Mobile phone testing results in three different scenarios

Required Continuous Detection Times	1	2	3
Accuracy (%)	91.14	98.09	98.96
Average Detecting Times in Real	1	1.25	1.40
Periods Between Two Output (ms)	1262	1577	1767

TABLE I: Results of continuous detection on mobiles

computational cost. It should be mentioned that there are about two orders of magnitude difference between computer testing results (2.7 GHz CPU and cost 5 ms) and mobile phone testing results (800 MHz CPU and cost 1262 ms). Besides the factors affecting processor speed such as CPU clock speed, the amount of RAM, it worths noticing that the processing time on the computer doe snot include the time to take photos, and preprocess photos.

2) Reaction Time: In this experiment, we perform testing near the edge between sidewalks and drive lanes, which is the boundary between safe zones and danger zones for pedestrians. The reaction time is the difference between the time, when `Inspector` gives a warning, minus the time, when the user approaches the edge. A positive value means that the alert is given after the user approaches the edge, or it is a late warning. In contrast, the negative value reflects the system ability to give an early warning. We expect this number to be negative (an early alert). By setting the capturing intervals of five seconds and two seconds, the results of average reaction time is 1.6 seconds. When we cut down the capturing interval to two seconds, the results is -0.54 seconds. This result shows that `Inspector` is able to give a timely alert with 2 second sampling intervals.

3) Energy and Computational Cost: An essential part of our experiment is the evaluation of system running costs,

	RAM(MB)	CPU	Exhausting Time
Running Service	13.21	17.22%	4h10min
Without Running	5.84	0.003%	8h30min

TABLE II: Results of application cost.

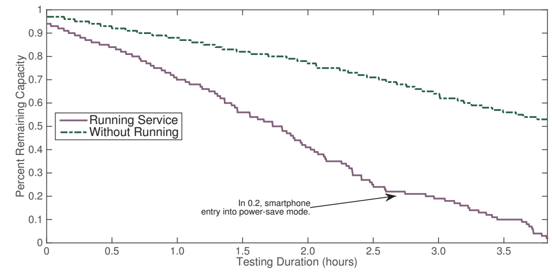


Fig. 10: Tendency of cellphone dump energy with time.

which consists of CPU/RAM occupation and the energy cost. Generally, people consider that taking photos or recording videos via camera costs a lot of power [15]. However, two points make `Inspector` energy efficient. First, we find out that the primary consumption of photographing spends on camera previewing action which does not required in `Inspector` [16]. Second, `Inspector` is lightweight and does not use any heave energy consuming component of smartphone such as LCD, or GPS sensors [17]. `Inspector` along with back camera will automatically run as background process while the LCD screen is displaying another app's interface. We measure the dump energy of the cellphone battery every ten seconds for about four hours. The dump energy tendency is shown in Fig. 10 and average occupation of CPU&RAM is shown in TABLE II. In this experiment, the capturing interval is five seconds, and the power monitor is Emmagee 2.0. It can be concluded from the TABLE II that the actual energy cost of `Inspector` per hour is 12.24% with the 1350mAh 3.7V battery, and the power consume is 611 mW. When doing experiments in real scenario, it usually takes four hours to drain our mobile's battery.

IV. RELATED WORKS

With a growing number of accidents caused by people immersed in smartphones, the problem of distracted walking has attracted a lot of attentions from research community [18] [19] [20]. In recent years, various kinds of mobile applications have been presented. To eliminate the risk when distracted pedestrians cross a street, Tianyu Wang *et al.*[8] implement an application to detect vehicles approaching the user and alert the user of a potentially dangerous situation. However, considering the back camera usually faces to the ground while people are texting or watching, it is hard to catch the image of the vehicles. Another example based on camera and image processing technique is proposed in [21] which provides a method to detect obstacles in front of the users. But how to improve the detection accuracy with regard to the complicated circumstances remains a challenge.

In [7], Zhou *et al.* use accelerometer and gyroscope to judge the state of users. But the broad coverage of the program may fret its users since it does not considered whether users are in danger or not. In [22], [23], ultrasonic is used to avoid pedestrians from abrupt fall or monitor approaching cars. But pedestrians on the sidewalk may receive a lot of false alerts from the cars on the road which is close to the sidewalk. The author of [24] depends on GPS localization to alert pedestrians while they are considered to be in danger area. However, the low accuracy of GPS in urban area is still a problem.

According to [25], Datta *et al.* estimate users' state by referring to data collected from a large number of pedestrians' behavior and alert them while they are considered having the intention of crossing roads illegally. However, further works are required when considering the differences of habits among people. In [6], Jain *et al.* judge the state of users via shoe sensing and warn them while they are considered to have come to the edge of sidewalks, but the inconvenience of shoe sensing has greatly limited the usage of the program.

V. Discussion

Our adaptive sampling method can be applied to detect a lot of obstacles such as lane markings, telegraph poles and street lamps. The reasons are as follows: (i) Our detecting method is based on the OpenCV vision library which is an open and common approach for all kinds of images targets; and (ii) Our decision algorithm, based on the continuity and predictability of detecting targets, does not require a high detecting accuracy, since it is able to improve the accuracy with acceptable cost. In fact, we have applied the same system to detect the zebra crossings. The single detecting accuracy is 70.75% and it grows above 90% when using a 3-times continuous detections.

VI. Conclusion

The paper introduces a system for detecting tactile pavement elements on sidewalks as indication of being in a "safe walking zones" for smartphone users who are using their phones while walking. The system periodically collects images from the smartphone camera, performs feature extraction and then a binary classification to decide if the user is in a safe or not safe zone. The classification accuracy is boosted by repeating the decision process several times upon a change in the detected state. Extensive experiments have shown that the system can achieve the accuracy of 92.72%, 98.78% when the algorithm demands 2 times, 3 times continuous detections respectively. The reaction time is 0.54 seconds early with a capturing interval of 2 seconds.

Acknowledgment

This work was partially supported by NSFC Grants (No. 61370028, No.91218302, No.61321491); Key Project of Jiangsu Research Program Grant (No. BE2013116); the EU FP7 IRSES MobileCloud Project (Grant No. 612212), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Distracted walking: Injuries soar for pedestrians on phones. http://researchnews.osu.edu/archive/distractwalk.htm, 2013.

[2] Mark Huffman. It's getting more dangerous to travel on foot. https://www.consumeraffairs.com/news/its-getting-more-dangerous-to-travel-on-foot-081015.html, 2015.

[3] Walking and texting at the same time? stony brook study says think again. http://commcgi.cc.stonybrook.edu/am2/publish/Medical_Center_Health_Care_4/Walking_and_Texting_at_the_Same_Time_Stony_Brook_Study_Says_Think_Again.shtml, 2012.

[4] Distracted walking a major pedestrian safety concern. http://safesoundfamily.com/blog/distracted-walking-a-major-pedestrian-safety-concern/, 2016.

[5] Distracted walking laws, legislation. http://handsfreeinfo.com/distracted-walking-pedestrians-law/, 2016.

[6] Shubham Jain, Carlo Borgiattino, Yanzhi Ren, Marco Gruteser, Yingying Chen, and Carla Fabiana Chiasserini. Lookup: Enabling pedestrian safety services via shoe sensing. In *MobiSys*. ACM, 2015.

[7] Zhengjuan Zhou. Headsup: keeping pedestrian phone addicts from dangers using mobile phone sensors. *International Journal of Distributed Sensor Networks*, 2015:5, 2015.

[8] Tianyu Wang, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads. In *HotMobile '12, San Diego, 2012*. ACM, 2012.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, Nov 2011.

[10] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[11] Christopher M Bishop. Pattern recognition. *Machine Learning*, 2006.

[12] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[13] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[14] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[15] Xiang Chen, Yiran Chen, Zhan Ma, and Felix C. A. Fernandes. How is energy consumed in smartphone display applications? In *HotMobile*. ACM, 2013.

[16] Yuanyuan Fan, Lei Xie, Yafeng Yin, and Sanglu Lu. A context aware energy-saving scheme for smart camera phones based on activity sensing. In *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*, pages 64–72. IEEE, 2015.

[17] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *USENIX ATC*, 2010.

[18] Jack Nasar, Peter Hecht, and Richard Wener. Mobile telephones, distracted attention, and pedestrian safety. *Accident Analysis & Prevention*, 40(1):69–75, 2008.

[19] Despina Stavrinos, Katherine W Byington, and David C Schwebel. Distracted walking: Cell phones increase injury risk for college pedestrians. *Journal of safety research*, 42(2):101–107, 2011.

[20] David C Schwebel, Despina Stavrinos, Katherine W Byington, Tiffany Davis, Elizabeth E ONeal, and Desiree De Jong. Distraction and pedestrian safety: how talking on the phone, texting, and listening to music impact crossing the street. *Accident Analysis & Prevention*, 45:266–271, 2012.

[21] Klaus-Tycho Förster, Alex Gross, Nino Hail, Jara Uitto, and Roger Wattenhofer. Spareeye: enhancing the safety of inattentionally blind smartphone users. In Arkady B. Zaslavsky, Seng W. Loke, Lars Kulik, and Evaggelia Pitoura, editors, *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia, Melbourne, VIC, Australia, November 25-28, 2014*, pages 68–72. ACM, 2014.

[22] Jiaqi Wen, Jiannong Cao, and Xuefeng Liu. We help you watch your steps: Unobtrusive alertness system for pedestrian mobile phone users. In *PerCom*, pages 105–113. IEEE, 2015.

[23] O.L. Technology. One llama. http://www.onellama.com/, 2013.

[24] Shubham Jain, Carlo Borgiattino, Yanzhi Ren, Marco Gruteser, and Yingying Chen. On the limits of positioning-based pedestrian risk awareness. In *Workshop on Mobile Augmented Reality and Robotic Technology-Based Systems, MARS*, 2014.

[25] Trisha Datta, Shubham Jain, and Marco Gruteser. Towards city-scale smartphone sensing of potentially unsafe pedestrian movements. In *MASS*. IEEE, 2014.