Paper ID #

All about the road: detecting road type, road damage, and road conditions

Nikitha S. Poddatur[1], Christoph Mertz[2*]
1. Carnegie Mellon University, USA; current address: Mumbai India
2. Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA 15213, +1-4122687955, cmertz@andrew.cmu.edu

Abstract
In this paper, we describe a cost-effective system that can continuously monitor the status of the road infrastructure. It can measure the amount of cracking in the pavement, determine the type of road, and can measure the amount of snow or slush on the road. The image and sensor data is collected by smart-phones mounted on vehicles that travel the roads on a regular basis. The images are analysed by a machine vision algorithm that uses bag-of-words and other visual features within superpixels and a SVM as the machine learning classifier. We evaluate the accuracy of the system and find better than 95%/95% precision/recall.

KEYWORDS:
Infrastructure monitoring, computer vision

Introduction
In order to properly manage transportation infrastructure assets it is very important to have an up-to-date inventory and status. Transportation departments are tasked to inspect the road surface regularly so that they can recondition them if required. Surveying the infrastructure includes looking for cracks and potholes and identifying other problem areas. One of the ways of collecting the data required to do this is by using specialized vehicles called Mobile Data Recorder (MDR) Figure 1 (left). These vans installed with specialized cameras, laser sensors, accelerometers and computer control systems [1]. This equipment is expensive, up to 500,000 dollars, and it involves a dedicated driver and an operator. This system is highly accurate in recording information required for road assessment like cracks but because of the high cost the data is usually only collected every two years for highways, three to five years for city roads. For most smaller townships, it is not affordable.

Figure 1: Left: Mobile Data Recorder (MDR) used by the Pennsylvania department of transportation for automatic pavement distress condition surveying. Right: High resolution images of the road taken by specialized downward facing cameras.

Another way road distress inspection is done is by the personnel who patrols the streets and visually inspects the road condition. This method is tedious, subjective, and error prone. Sometimes problems are also recorded through complaints filed by citizens. These complaints are usually only of severe damage. In our previous work [6] we proposed a solution that makes road assessment and surveying inexpensive and easier. We use computer vision algorithms on image data collected using smart phones. These smart phones are mounted on service vehicles like garbage trucks, fleet vehicles or other cars that travel the roads on a regular basis. Hence, we don't require dedicated drivers and operators, further reducing the cost significantly. Since these vehicles drive across the city almost every day, there is a large amount of image data that gets collected. Using this data, the road assessment can be done on a continuous basis as opposed to every two or more years, which is the current practice. Our method is machine vision based and requires little human intervention.
We use a supervised learning method to train a machine learning classifier. Our initial system [8] could detect cracks. Now we have extended it to also detect other parts of the road: road type like cobblestone or road conditions like snow or slush. The hardware requirement to run the analysis is modest, we can train, test and run the classifier on regular laptops. In the following sections, we begin by giving some background information. Then we describe our data pipeline, from data collection to analysis method. Next we discuss how the system was applied to various use cases. Finally, we discuss the storage and CPU requirements and evaluate the accuracy of the system.

Overview of crack detection with images

There have been several methods used in the past for road crack detection. Image based solutions to crack detection are not new. Most of the image based solutions use a classifier trained using either supervised or unsupervised [7] learning methods. Supervised learning algorithms need accurate training examples to train their classifier, which involves manual labelling of data. [3] is one of the few image based solutions that does not involve training a classifier but uses minimal paths between pixels in a neighbourhood to detect cracks. One key limitation to these methods is that they cannot be generalized to be used in identifying potholes and other data that might be useful in road assessment. Although these methods are accurate in detecting cracks, another high-cost task these methods encounter is data collection. They require specialized downward facing durable cameras, that are weather-resistant and damage proof, increasing the costs. These cameras capture high resolution images taken parallel to the surface of the road so that they do not capture any other areas apart from the road Figure 1 (right). A review of image based crack detection can be found in [9]. We follow a supervised learning approach as well, the details of which will be discussed in the following sections.

Figure 2: Data collection setup: A smart phone is mounted on the windshield with a setup similar to a GPS mount. The collection app is running on the phone.

Method

The algorithm and setup have been discussed in detail in [8] and [6], here we will give a summary. In the data collection step, we mount a smart-phone on the windshield of service vehicles in a setup similar to mounting a navigator (Figure 2). On clicking the start and stop button, an in-built app starts recording the GPS tagged images. The data is automatically downloaded as soon as the smart-phone connects with WiFi.

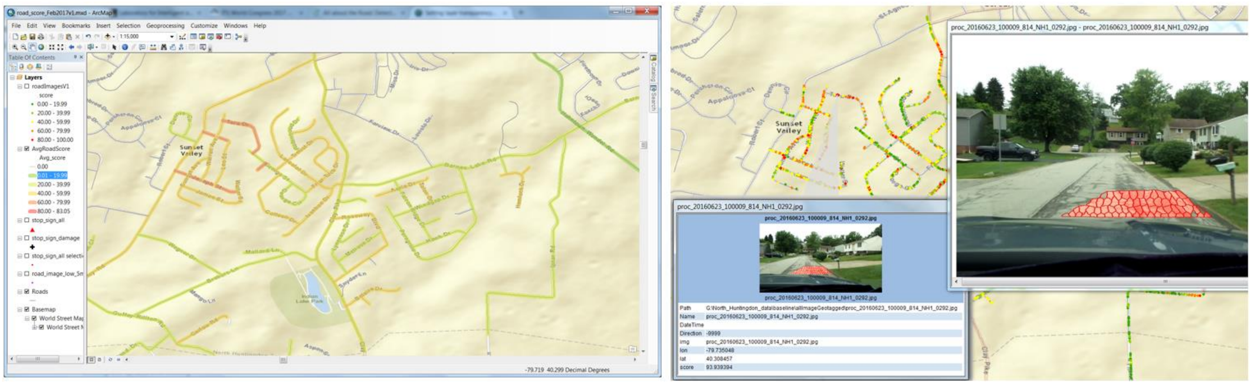

Figure 3: Example display of data on GIS map. One the left are the road segments colour coded according to the average distress score. On the right are the scores for each image mapped as a colour coded point. Clicking on the point displays the image itself and its attributes.

The data can be displayed on a Google Earth or a GIS map. Figure 3 shows the GPS location of the images as points, clicking on the points displays the corresponding image and other information (Figure 3 right). It is also possible to display the data for user-selected time windows. These images have a lot of background besides the road surface, like shown in the street view Figure 4. Since we are only interested in the road surface pixels and all the images are taken from the same camera orientation, we create generic masks for images. This hides the background and only takes the road surface information. We run our algorithm only on the masked region of the image, speeding up the processing time. In Figure 4 the mask is indicated in grey.

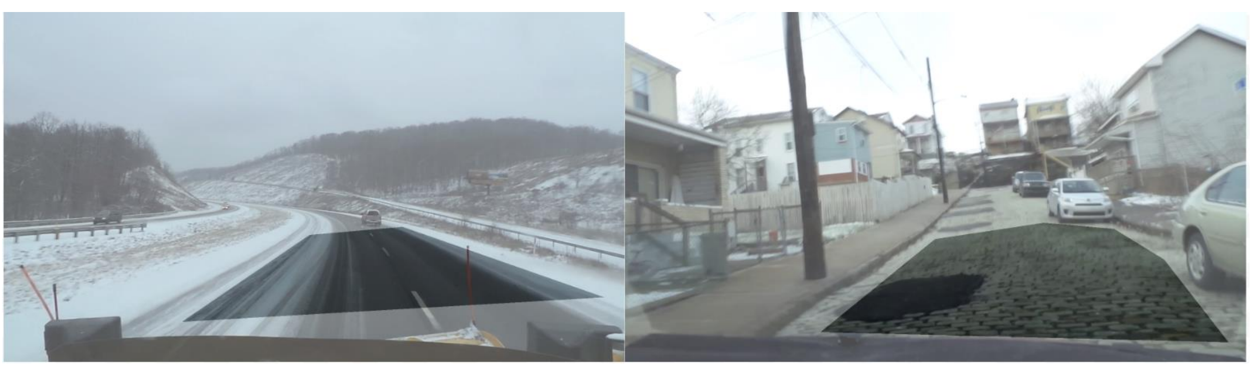

Figure 4: Masks placed on example images. The left image shows an example of snow covered road and the right image shows an example of cobblestone road area.

Classification Algorithm

Cracks cannot be differentiated based on colour because they are only a shade darker than the road surface. This logic can also be extended to cobblestone road areas. They are heavy in texture and hence we use texture based features. We run texture classification filters [5]. Those are edge and bar filters - first and second order derivatives of Gaussian in six equally spaced orientations. Since cracks are rotationally invariant, we order the filter responses for different orientations by making the highest absolute valued response as the base direction. We also use rotationally symmetric LoG (Laplacian of Gaussian) filters for three different scales. Based on pixel-wise filter response vector, all the pixels are partitioned into K clusters to generate texton maps of images (Figure 5).

Figure 5: Texton maps generated of the example images shown in Figure 4. Notice that the textons are generated only on the masked area

The texton maps generated in the previous step are now used in an over-segmentation step. We group pixels into larger segments called superpixels, based on Euclidean, colour, and texture distance. The SLIC (Simple Linear Iterative Clustering) superpixel algorithm [2] uses colour and proximity to compute the superpixels. But since cracks are texture rich, we compute texture based distance and give it a high weight (Figure 6). Next we calculate features of each superpixel. In contrast to pixel-based features the features of superpixels carry more semantic and perceptual information. Colour, hue, saturation, and texture based histograms are used to compute this feature vector.

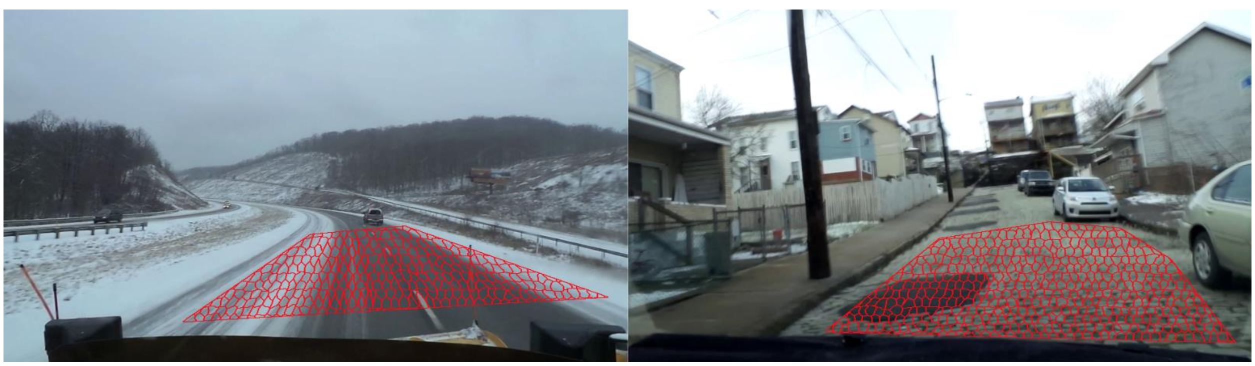

Figure 6: Superpixel generated of the images in Figure 5.

These features vectors are then used to train the SVM classifier [4]. Using the SVM model, the superpixels are predicted as cracked or non-cracked with a certain probability (Figure 7). Since the dimension of the superpixel feature vector is high (depends on k in k-clustering used in texton map generation), we use RBF kernel to train the classifier by setting its parameters using cross-validation.
In contrast to our earlier system [8] we no longer use multiple-instance learning and no longer define the cracked regions as consensus of several labeller. Instead we take great care during labelling that no region of cracks contains sub-regions without cracks. This is quite tedious, but it improves the results considerably.

Figure 7: Sample results of testing phase. Coloured ones denoting the predictions made by the SVM classifier. Classification result for snow covered road (left-top). Areas where the road is clear is shown to be negative. Results for cobblestone-vs-asphalt road area (right-top), cracked road (left-bottom) and good road (right-bottom).

Experiments

We trained and tested the algorithms to distinguish road damage (cracked vs. non-cracked). Figure 7 (bottom) shows examples of a road in poor condition and a newly paved road. We collected data in a township, calculated a damage score (= cracked vs non-cracked area) for each image, and then averaged this score over a road segment. The resulting road damage map is shown in Figure 3.
This method has also been successfully extended to find cobblestone road areas and distinguish between snow/slush and clear road. With this information, one can easily get surveying maps of the city showing cracks, potholes, and cobblestone road areas or determine winter road conditions. We conducted experiments to investigate the trade-offs between number of textons (k), number of superpixels, accuracy and CPU time, and to create precision-recall curves. We tested our system with k = 10 to 150 and for a varying number of superpixel sizes (500 to 3000 superpixels per full image). With increased k and number of superpixels, the run time of the algorithm became higher (Figure 9 right). Our algorithm gave more accurate results for higher valued k (Figure 9 left) irrespective of the superpixel size. Our precision-recall curves are shown for k values of 10 to 150 (Figure 8). In our earlier system [8] we reached only precision/recall of 35%/70%, which was mainly limited by the quality of the labelling of 50%/70%. Now we get much better results. For k = 150 one gets the best results, exceeding 95%/95% precision/recall for all three classifiers. The best performance is for snow/slush vs. clear road with 99%/97% precision/recall.

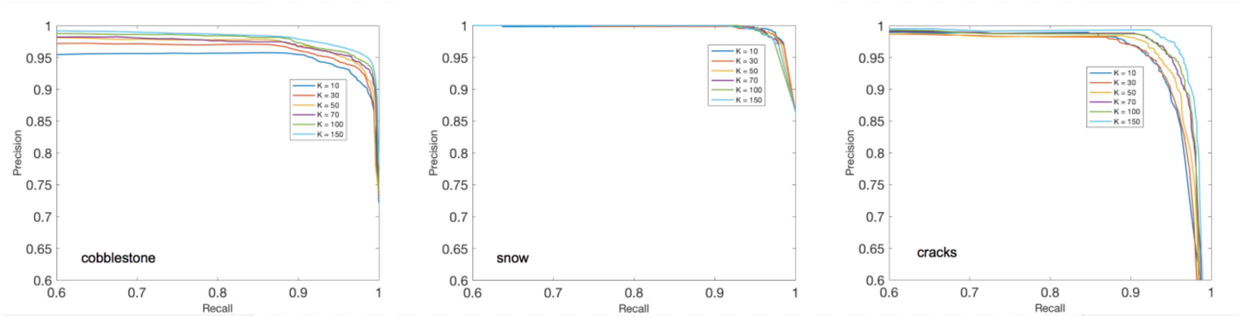

Figure 8: Precision-Recall curves for cobblestone vs paved road(left), snow/slush covered vs clear road(middle) and cracked vs. undamaged road (right).

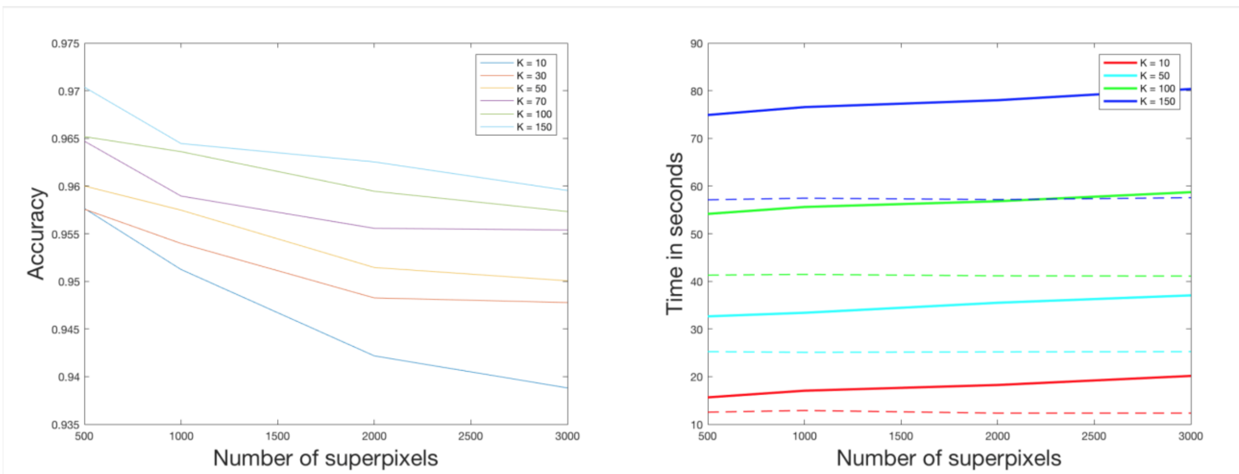

Figure 9: Left: Accuracy of our algorithm in classifying cracks. Right: Shows the time taken to do the full analysis per image (solid line) and the time it takes to calculate the texton map alone. The times are shown for different number of superpixels per image and for different k.

We want to run this algorithm on large sets of images. Hence time taken for each image becomes crucial. The superpixel size has very little influence on the run-time whereas the number of clusters (k) has a significant impact. Our performance bottleneck was the time taken to compute the texton map, more specifically, finding the nearest cluster each pixel belonged to. We implemented parallel processing techniques to compute the texton map. This can further be improved by implementing k-d trees to find nearest neighbours. There is only a marginal time difference between computing the texton map and generating superpixels. We also present the time it takes to compute texton maps and superpixel feature descriptor for different k centres and superpixels (Figure 9 right). It can be seen from Figure 9 (right) that the most time is taken to compute texton maps. Higher k values gave better accuracy with the trade off of larger processing time. This way we can adjust the k value depending on our requirement. We used an intel i7 quad core with 8 threads to test our system.

Conclusion

We were able to significantly improve the performance of our system from an earlier version with changes in code, method, and labeling strategy. The system can classify textures from damaged road, road types, or road conditions with high reliability, precision/recall is greater than 95%/95%. It is straightforward to train the classifier for other textures, like vegetation. The full system of data collection, data analysis, and display of results is used for infrastructure monitoring.

References

[1] www.penndot.gov/projectandprograms/researchandtesting/roadwaymanagementandtesting/pages/default.aspx.
[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11):2274–2282, 2012.
[3] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. IEEE Transactions on Intelligent Transportation Systems, 17(10):2718–2729, 2016.
[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27, 2011.
[5] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. International journal of computer vision, 43(1):29–44, 2001.
[6] Christoph Mertz, Srivatsan Varadharajan, Sobhagya Jose, Karan Sharma, Lars Wander, and Jina Wang. City-wide road distress monitoring with smartphones. In 21st World congress on intelligent transport systems: reinventing transportation in our connected world, ITSWC, 2014.
[7] Henrique Oliveira and Paulo Lobato Correia. Automatic road crack detection and characterization. IEEE Transactions on Intelligent Transportation Systems, 14(1):155–168, 2013.
[8] Srivatsan Varadharajan, Sobhagya Jose, Karan Sharma, Lars Wander, and Christoph Mertz. Vision for road inspection. In Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, pages 115–122. IEEE, 2014.
[9] H Zakeri, Fereidoon Moghadas Nejad, and Ahmad Fahimifar. Image based techniques for crack detection, classification and quantification in asphalt pavement: A review. Archives of Computational Methods in Engineering, pages 1–43, 2016.